
openstack.compute Documentation

Release 2.0a1

Jacob Kaplan-Moss

Sep 27, 2017

Contents

1	The <code>openstack-compute</code> shell utility	3
2	The <code>openstack . compute</code> Python API	5
2.1	Usage	5
3	Release notes	9
3.1	2.0 (TBD)	9
3.2	1.2 (August 15, 2010)	9
3.3	1.1 (May 6, 2010)	9
4	Contributing	11
5	Indices and tables	13
	Python Module Index	15

This is a client for the OpenStack Compute API used by Rackspace Cloud and others. There's *a Python API* (the `openstack.compute` module), and a *command-line script* (installed as `openstack-compute`). Each implements the entire OpenStack Compute API (as well as a few Rackspace-only addons).

To try this out, you'll need a [Rackspace Cloud](#) account — or your own install of OpenStack Compute (also known as Nova). If you're using Rackspace you'll need to make sure to sign up for both Cloud Servers *and* Cloud Files – Rackspace won't let you get an API key unless you've got a Cloud Files account, too. Once you've got an account, you'll find your API key in the management console under “Your Account”.

See also:

You may want to read [Rackspace's API guide](#) (PDF) – the first bit, at least – to get an idea of the concepts. Rackspace/OpenStack is doing the cloud hosting thing a bit differently from Amazon, and if you get the concepts this library should make more sense.

Contents:

The `openstack-compute` shell utility

The `openstack-compute` shell utility interacts with OpenStack Compute servers from the command line. It supports the entirety of the OpenStack Compute API (plus a few Rackspace-specific additions), including some commands not available from the Rackspace web console.

To try this out, you'll need a [Rackspace Cloud](#) account — or your own install of OpenStack Compute (also known as Nova). If you're using Rackspace you'll need to make sure to sign up for both Cloud Servers *and* Cloud Files – Rackspace won't let you get an API key unless you've got a Cloud Files account, too. Once you've got an account, you'll find your API key in the management console under “Your Account”.

You'll need to provide `openstack-compute` with your Rackspace username and API key. You can do this with the `--username` and `--apikey` options, but it's easier to just set them as environment variables by setting two environment variables:

OPENSTACK_COMPUTE_USERNAME

Your Rackspace Cloud username.

OPENSTACK_COMPUTE_API_KEY

Your API key.

For example, in Bash you'd use:

```
export OPENSTACK_COMPUTE_USERNAME=yourname
export OPENSTACK_COMPUTE_API_KEY=yadayadayada
```

From there, all shell commands take the form:

```
openstack-compute <command> [arguments...]
```

Run `openstack-compute help` to get a full list of all possible commands, and run `openstack-compute help <command>` to get detailed help for that command.

The `openstack.compute` Python API

Usage

First create an instance of `Compute` with your credentials:

```
>>> from openstack.compute import Compute
>>> compute = Compute(username=USERNAME, apikey=API_KEY)
```

Then call methods on the `Compute` object:

class `openstack.compute.Compute`

backup_schedules

A `BackupScheduleManager` – manage automatic backup images.

flavors

A `FlavorManager` – query available “flavors” (hardware configurations).

images

An `ImageManager` – query and create server disk images.

ipgroups

A `IPGroupManager` – manage shared public IP addresses.

servers

A `ServerManager` – start, stop, and manage virtual machines.

For example:

```
>>> compute.servers.list()
[<Server: buildslave-ubuntu-9.10>]

>>> compute.flavors.list()
[<Flavor: 256 server>,
 <Flavor: 512 server>,
 <Flavor: 1GB server>]
```

```
<Flavor: 2GB server>,
<Flavor: 4GB server>,
<Flavor: 8GB server>,
<Flavor: 15.5GB server>]

>>> compute.images.list()
[<Image: Windows Server 2008 R2 x64 - MSSQL2K8R2>, ...]

>>> fl = compute.flavors.find(ram=512)
>>> im = compute.images.find(name='Ubuntu 10.10 (maverick)')
>>> compute.servers.create("my-server", image=im, flavor=fl)
<Server: my-server>
```

For more information, see the reference:

API Reference

Backup schedules

Rackspace allows scheduling of weekly and/or daily backups for virtual servers. You can access these backup schedules either off the API object as `CloudServers.backup_schedules`, or directly off a particular `Server` instance as `Server.backup_schedule`.

Classes

Constants

Constants for selecting weekly backup days:

```
openstack.compute.BACKUP_WEEKLY_DISABLED
openstack.compute.BACKUP_WEEKLY_SUNDAY
openstack.compute.BACKUP_WEEKLY_MONDAY
openstack.compute.BACKUP_WEEKLY_TUESDAY
openstack.compute.BACKUP_WEEKLY_WEDNESDA
openstack.compute.BACKUP_WEEKLY_THURSDAY
openstack.compute.BACKUP_WEEKLY_FRIDAY
openstack.compute.BACKUP_WEEKLY_SATURDAY
```

Constants for selecting hourly backup windows:

```
openstack.compute.BACKUP_DAILY_DISABLED
openstack.compute.BACKUP_DAILY_H_0000_0200
openstack.compute.BACKUP_DAILY_H_0200_0400
openstack.compute.BACKUP_DAILY_H_0400_0600
openstack.compute.BACKUP_DAILY_H_0600_0800
openstack.compute.BACKUP_DAILY_H_0800_1000
openstack.compute.BACKUP_DAILY_H_1000_1200
```

```
openstack.compute.BACKUP_DAILY_H_1200_1400
openstack.compute.BACKUP_DAILY_H_1400_1600
openstack.compute.BACKUP_DAILY_H_1600_1800
openstack.compute.BACKUP_DAILY_H_1800_2000
openstack.compute.BACKUP_DAILY_H_2000_2200
openstack.compute.BACKUP_DAILY_H_2200_0000
```

Exceptions

Exceptions

Exceptions that the API might throw:

Flavors

From Rackspace's API documentation:

A flavor is an available hardware configuration for a server. Each flavor has a unique combination of disk space, memory capacity and priority for CPU time.

Classes

Images

An "image" is a snapshot from which you can create new server instances.

From Rackspace's own API documentation:

An image is a collection of files used to create or rebuild a server. Rackspace provides a number of pre-built OS images by default. You may also create custom images from cloud servers you have launched. These custom images are useful for backup purposes or for producing "gold" server images if you plan to deploy a particular server configuration frequently.

Classes

Shared IP addresses

From the Rackspace API guide:

Public IP addresses can be shared across multiple servers for use in various high availability scenarios. When an IP address is shared to another server, the cloud network restrictions are modified to allow each server to listen to and respond on that IP address (you may optionally specify that the target server network configuration be modified). Shared IP addresses can be used with many standard heartbeat facilities (e.g. `keepalived`) that monitor for failure and manage IP failover.

A shared IP group is a collection of servers that can share IPs with other members of the group. Any server in a group can share one or more public IPs with any other server in the group. With the exception of the first server in a shared IP group, servers must be launched into shared IP groups. A server may only be a member of one shared IP group.

See also:

Use `Server.share_ip()` and `Server.unshare_ip` to share and unshare IPs in a group.

Classes

Servers

A virtual machine instance.

Classes

Constants

Reboot types:

`openstack.compute.REBOOT_SOFT`

`openstack.compute.REBOOT_HARD`

2.0 (TBD)

- **Major renaming:** the library is now called `openstack.compute` to reflect that Rackspace Cloud is just one instance of the open source project. This ripples to a lot of places:
 - The library is now called `openstack.compute` instead of `cloudservers`, and the main API entry point is now `openstack.compute.Compute` instead of `cloudservers.CloudServers`.
 - The shell program is now `openstack-compute` instead of `cloudservers`. Yes, the name's a lot longer. Use `alias`.
 - The `env` variables are now `OPENSTACK_COMPUTE_USERNAME` and `OPENSTACK_COMPUTE_API_KEY`.

1.2 (August 15, 2010)

- Support for Python 2.4 - 2.7.
- Improved output of `cloudservers ipgroup-list`.
- Made `cloudservers boot --ipgroup <name> work` (as well as `--ipgroup <id>`).

1.1 (May 6, 2010)

- Added a `--files` option to `cloudservers boot` supporting the upload of (up to five) files at boot time.
- Added a `--key` option to `cloudservers boot` to key the server with an SSH public key at boot time. This is just a shortcut for `--files`, but it's a useful shortcut.
- Changed the default server image to Ubuntu 10.04 LTS.

CHAPTER 4

Contributing

Development takes place on [GitHub](#); please file bugs/pull requests there.

Run tests with `python setup.py test`.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

O

`openstack.compute`, 5

B

- BACKUP_DAILY_DISABLED (in module openstack.compute), 6
- BACKUP_DAILY_H_0000_0200 (in module openstack.compute), 6
- BACKUP_DAILY_H_0200_0400 (in module openstack.compute), 6
- BACKUP_DAILY_H_0400_0600 (in module openstack.compute), 6
- BACKUP_DAILY_H_0600_0800 (in module openstack.compute), 6
- BACKUP_DAILY_H_0800_1000 (in module openstack.compute), 6
- BACKUP_DAILY_H_1000_1200 (in module openstack.compute), 6
- BACKUP_DAILY_H_1200_1400 (in module openstack.compute), 6
- BACKUP_DAILY_H_1400_1600 (in module openstack.compute), 7
- BACKUP_DAILY_H_1600_1800 (in module openstack.compute), 7
- BACKUP_DAILY_H_1800_2000 (in module openstack.compute), 7
- BACKUP_DAILY_H_2000_2200 (in module openstack.compute), 7
- BACKUP_DAILY_H_2200_0000 (in module openstack.compute), 7
- backup_schedules (openstack.compute.Compute attribute), 5
- BACKUP_WEEKLY_DISABLED (in module openstack.compute), 6
- BACKUP_WEEKLY_FRIDAY (in module openstack.compute), 6
- BACKUP_WEEKLY_MONDAY (in module openstack.compute), 6
- BACKUP_WEEKLY_SATURDAY (in module openstack.compute), 6
- BACKUP_WEEKLY_SUNDAY (in module openstack.compute), 6
- BACKUP_WEEKLY_THURSDAY (in module openstack.compute), 6
- BACKUP_WEEKLY_TUESDAY (in module openstack.compute), 6
- BACKUP_WEEKLY_WEDNESDA (in module openstack.compute), 6

C

Compute (class in openstack.compute), 5

E

environment variable

- OPENSTACK_COMPUTE_API_KEY, 3
- OPENSTACK_COMPUTE_USERNAME, 3

F

flavors (openstack.compute.Compute attribute), 5

I

images (openstack.compute.Compute attribute), 5
 ipgroups (openstack.compute.Compute attribute), 5

O

openstack.compute (module), 5

R

REBOOT_HARD (in module openstack.compute), 8
 REBOOT_SOFT (in module openstack.compute), 8

S

servers (openstack.compute.Compute attribute), 5