

Cisco Edition OpenStack

User Guide

Contents

Cisco Edition of OpenStack	1
User Guide	1
Contents	2
Figures	4
1.0 Introduction	6
1.1 Preface	6
1.2 Audience	7
1.3 Document Scope	7
2.0 Web-Based OpenStack Tenant Operations	8
2.1 Project Overview	9
2.2 Image Overview	9
2.3 Creation of Key-Pair	10
2.4 Creation of Security Groups	11
2.5 Instance Management	14
2.6 Floating IP Management	18
2.7 Accessing Instance using Floating IP	20
2.8 Instance Snapshots	21
2.9 Nova Volumes	24
2.10 Swift Containers	27
3.0 CLI-Based OpenStack Tenant Operations	30
3.1 OpenStack Python-Based Command Line Interface Clients	30
3.2 User Credentials	31
3.3 Tenant Operations	34
3.3.1 Project Overview	34
3.3.2 Image Overview	35
3.3.3 Creation of Key-Pair	35
3.3.4 Creation of Security Groups	36
3.3.5 Instance Management	38
3.3.6 Floating IP Management	40
3.3.7 Accessing Instance using Floating IP	41
3.3.8 Instance Snapshots	42
3.3.9 Nova Volumes	43
3.3.10 Swift Containers	45
3.4 Image Management	46
3.4.1 Adding Images	46
3.4.2 Adding AMI (Amazon Machine Image) Images	46
3.4.3 Adding QCOW (Qemu Copy on Write) Images	48

3.4.4 Listing Images	49
3.4.5 Deleting Images	50
4.0 Appendix	52
4.1 Glossary	52
4.2 Caveats	53
4.3 References	53

Figures

Figure 1: OpenStack Dashboard Login	8
Figure 2: Project-based Overview of Usage Summary	9
Figure 3: Glance Image Details	10
Figure 4: Access and Security	11
Figure 5: Creating a new keypair	11
Figure 6: Create Security Groups	12
Figure 7: Add Rules to Security Group	12
Figure 8: Edit Security Group	13
Figure 9: Security Group Rules	14
Figure 10: List of Instances and Volumes	15
Figure 11: Images Available	15
Figure 12: New Instances Required Parameters	16
Figure 13: Status of the new Instance	17
Figure 14: Console View per Instance	17
Figure 15: Floating IP list in Dashboard	18
Figure 16: Floating IP Allocation by Project	18
Figure 17: Confirmation message after assigning Floating IPs	19
Figure 18: Associate Floating IP to Instance	19
Figure 19: Confirmation message after associating Floating IP	19
Figure 20: Verify Instance to Floating IP Association	20
Figure 21: Dis-associate Floating IP from Instance	20
Figure 22: Instance Snapshot	22
Figure 23: Snapshot name assignment	22
Figure 24: Instance Snapshot	23
Figure 25: Launch Instance from Snapshot	23
Figure 26: New Instance from VM1 Snapshot	24
Figure 27: Create Volume	24
Figure 28: Fields required for creating a volume	25
Figure 29: Attaching a volume to an instance	26
Figure 30: Volume Attached Overview	27
Figure 31: Containers section in Project Panel	28
Figure 32: New Container Name Assignment	28
Figure 33: List of Containers	29
Figure 34: Uploading Objects to Containers	29
Figure 35: Download OpenStack User Credentials	31
Figure 36: Download OpenStack RC File	32
Figure 37: Source Openrc File	33
Figure 38: EC2 Credentials	34
Figure 39: Tenant Instance List	35
Figure 40: Nova Image list for project "mycompany"	35
Figure 41: Create Key-Pair	36
Figure 42: View Existing Security Groups and list rules	37
Figure 43: Create New Security Group and add Rules	38

Figure 44: Flavor list, image-list and secgroup-list	39
Figure 45: Adding an instance using nova CLI	39
Figure 46: List of existing Instance within a project	40
Figure 47: Create Floating-IP	40
Figure 48: Associate Floating IP to an existing Instance	41
Figure 49: Accessing Instance using Floating IP	41
Figure 50: SSH to new VM using floating-ip	42
Figure 51: Creating a Snapshot of an Instance	43
Figure 52: Nova Volume List	43
Figure 53: Nova Volume Attached to Instance	44
Figure 54: Create a new Nova Volume	45
Figure 55: Attach New Nova Volume to Instance	45
Figure 56: AMI Image Download	47
Figure 57: Verify AMI Kernel and RAM Image upload	48
Figure 58: Verify AMI Image Upload	48
Figure 59: QCOW Image download	49
Figure 60: QCOW2 Image upload	49
Figure 61: Glance Image List	50
Figure 62: Nova Image List	50
Figure 63: Deleting Images Using Dashboard	51
Figure 64: Delete an image using Glance CLI	51
Figure 65: Delete an image using nova CLI	51

1.0 Introduction

1.1 Preface

This document is an OpenStack User Guide intended as a reference for both OpenStack Tenant User and Tenant Admin.

OpenStack is a global collaboration of developers and cloud computing technologists producing the ubiquitous open source cloud computing platform for public and private clouds. The project aims to deliver solutions for all types of clouds by being simple to implement, massively scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution.¹

OpenStack provides a tool to orchestrate a cloud, including running instances, managing networks, storing files, and controlling access to the cloud through users and projects. It provides the software that can control an Infrastructure as a Service (IaaS) cloud computing platform. It is similar in scope to Amazon EC2 Cloud Servers and Amazon S3. Major components include OpenStack Compute (also known as Nova), OpenStack Storage (also known as Swift), OpenStack Image Service (also known as Glance), OpenStack Dashboard (also known as Horizon), and OpenStack Identity Services (also known as Keystone). OpenStack Compute does not include any virtualization software; rather it defines drivers that interact with underlying virtualization mechanisms that run on your host operating system, and exposes functionality over a REST APIs.²

All components in OpenStack offer REST APIs that may be invoked by user-friendly web-based user interfaces, Command Line Interfaces (CLIs) or customized interfaces. In this guide we will explain how to use the web-based UI (Horizon) and python-based CLI Clients.

OpenStack can be used by many different projects (tenants) sharing resources in the same system. Earlier versions of OpenStack used the term "project" instead of "tenant". Because of this legacy terminology, the word projects and tenants are used interchangeably this document.

This User guide assumes the successful deployment of OpenStack based on the installation process indicated in the OpenStack Installation Guide [1].

¹ Description from <http://OpenStack.org/>, 06/26/2012.

² Adapted from <http://nova.OpenStack.org/nova.concepts.html>, 06/26/2012.

1.2 Audience

This document is intended for Cisco Advanced Services Data Center Practice Team and Customer Network Architecture Team.

1.3 Document Scope

The scope of this document will include all the steps necessary for a tenant user or tenant admin to administer all functions within a project such as creating key-pair, managing instances, security groups, etc. The document is divided into two major sections. First, Web-based OpenStack Tenant Operations and the second is a CLI-based OpenStack Tenant Operations.

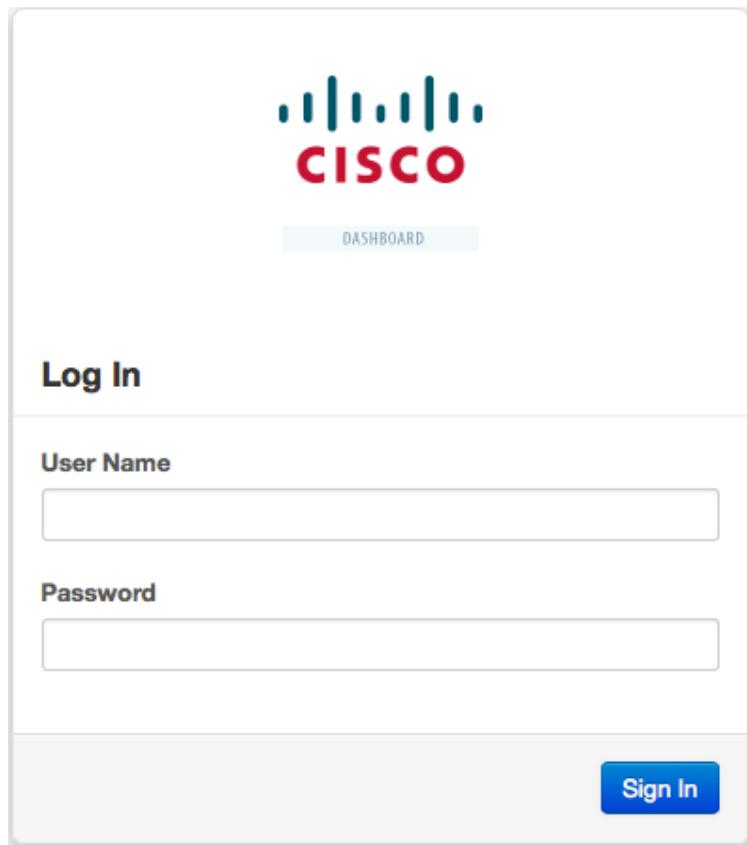
This document does not include anything that relates to OpenStack Administrative functions, such as Creating projects within OpenStack, creating new users, managing project quotas, etc. These items will be covered in a separate Cisco OpenStack Admin Guide document.

2.0 Web-Based OpenStack Tenant Operations

OpenStack offers a user-friendly graphical interface called the OpenStack Dashboard (also known by its codename: Horizon). The dashboard offers two different views: the Admin System Panel and the Project Panel. The Admin System Panel is exclusively for cloud administration activities such as creating projects and users or registering images. Tenants use the Project Panel to control their projects and perform functions such as managing compute servers or creating new instances and volumes. In this guide we will focus on the Project Panel.

The OpenStack Dashboard is a web-based interface compatible with Firefox 13 (recommended), IE 7 and Chrome 19. To access the Dashboard, open your browser and type the IP address or hostname of the Horizon host server. Figure 1 illustrates the OpenStack Dashboard Log In screen.

Figure 1: OpenStack Dashboard Login



2.1 Project Overview

Once project users have logged into the OpenStack Dashboard using the username and credentials provided by the OpenStack Administrator, they will be directed to the Overview section where they will find a project-based usage summary of the cloud infrastructure. The summary includes information such as the number of instances running, memory utilization, and disk space. Figure 2 shows the summary for the “mycompany” project (tenant). Since this is a new project, we have yet to create instances.

Figure 2: Project-based Overview of Usage Summary

Instance Name	vCPUs	Disk	RAM	Uptime
Displaying 0 items				

2.2 Image Overview

An image is a file containing information about a virtual disk that completely replicates all information about a working computer at a point in time including operating system information and file system information. OpenStack integrates Glance as the image service. Glance is in charge of fetching all available images on to the host machine. The OpenStack compute service then boots the images it finds on the host machine.

The OpenStack Dashboard has limited integration with the full set of Glance APIs [2]. Basically, it just provides an overview of the already uploaded images. There will be images owned by the Administrator that cannot be deleted or modified by users.

On the Project Panel select the section “Images & Snapshots”. The complete list of available images will be displayed. Then, just select the name of the desired image

and the information will be displayed. Figure 3 shows detailed information from the Ubuntu image.

Figure 3: Glance Image Details

The screenshot shows the Cisco OpenStack tenant interface. On the left, there's a sidebar with a Cisco logo and navigation links: DASHBOARD, Project (selected), PROJECT mycompany, Manage Compute (Overview, Instances & Volumes, Images & Snapshots, Access & Security), Object Store (Containers), and Custom Properties. The main area is titled 'Image Detail' and shows the 'Image Overview' tab selected. It displays the following details for the 'precise-amd64' image:

- Name:** precise-amd64
- ID:** 651d048a-b1a7-4af7-bbe4-a11ae7b8f420
- Status:** Active
- Public:** True
- Checksum:** 28c510b1e49a65d638cfb469754a93d3
- Created:** 2012-08-24T18:05:33
- Updated:** Never updated
- Specs:**
 - Size:** 222.4 MB
 - Container Format:** OVF
 - Disk Format:** QCOW2
- Custom Properties:** (Empty)

2.3 Creation of Key-Pair

Now that we've seen a summary of a project, let's consider how users will interact with the project. Public/private key pairs control programmatic access to the tenant-facing API's in an OpenStack cloud. Key pairs are created for each user within a project. To create a keypair, click on "Access & Security" in

the navigation menu on the left (Figure 4). Scroll to the “keypairs” section of the page and click on the “Create Keypair” button to generate a new key pair. You’ll need to supply a name for each keypair – the name can be whatever you want, but in practice the name often reflects whom the keypair belongs to. A file with the key will be automatically downloaded to the computer used to interact with the OpenStack Dashboard.

Figure 4: Access and Security

Keypair Name	Fingerprint	Actions
mykey	8a:46:50:c4:3d:e0:f8:ad:a4:0f:67:42:a2:b0:e6:df	Delete Keypair

The file will have the extension .pem. Save this key to your laptop, as it will be needed to access the instances that will be created under this project.

OpenStack Dashboard will show the key created with the only available action of deleting it. Figure 5 shows the created keypair named “mykey”.

Figure 5: Creating a new keypair

Keypair Name	Fingerprint	Actions
mykey	8a:46:50:c4:3d:e0:f8:ad:a4:0f:67:42:a2:b0:e6:df	Delete Keypair

Section “Accessing Instances by Terminal Console” covers how to use keypair files to securely gain access to instances without the need for a password.

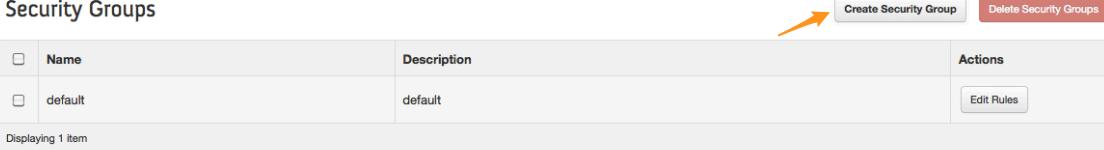
2.4 Creation of Security Groups

OpenStack provides ingress filtering for the instances based on the concept of security groups. A Security Group is a named set of rules that get applied to the incoming packets for the instances. You can specify a security group while

launching an instance. Each security group can have multiple rules associated with it. Each rule specifies the source IP/network, protocol type, destination ports etc. Any packet matching these parameters specified in a rule is allowed in. The rest of the packets are blocked.

To create a new Security Group, click on “Access & Security” in the navigation menu on the left. Scroll to the “Security Groups” section of the page and click on the “Create Security Group” button to generate a new Security Group.

Figure 6: Create Security Groups



The screenshot shows a table titled "Security Groups". It has three columns: "Name", "Description", and "Actions". There is one row displayed, representing a security group named "default" with a description of "default". In the "Actions" column, there is a "Edit Rules" button. At the top right of the table, there are two buttons: "Create Security Group" (highlighted with an orange arrow) and "Delete Security Groups".

	Name	Description	Actions
<input type="checkbox"/>	default	default	<button>Edit Rules</button>

Displaying 1 item

To add rules to the newly created security group, click on “Edit Rules” button.

Figure 7: Add Rules to Security Group



The screenshot shows a table titled "Security Groups". It has three columns: "Name", "Description", and "Actions". There are two rows displayed, representing security groups "default" and "security_group1". The "description" for "default" is "default" and for "security_group1" is "security group1". In the "Actions" column for "security_group1", there is a "Edit Rules" button with a dropdown arrow, highlighted with an orange arrow. At the top right of the table, there are two buttons: "Create Security Group" and "Delete Security Groups".

	Name	Description	Actions
<input type="checkbox"/>	default	default	<button>Edit Rules</button>
<input type="checkbox"/>	security_group1	security group1	<button>Edit Rules</button> ▾

Displaying 2 items

Add a rule to allow SSH traffic.

Figure 8: Edit Security Group

Edit Security Group Rules

Security Group Rules

	IP Protocol	From Port	To Port	Source	Actions
No items to display.					

Displaying 0 items

Add Rule

IP Protocol: TCP

From Port: 22

To Port: 22

Source Group: CIDR

CIDR: 0.0.0.0/0

Buttons: Cancel, Add Rule

Also, repeat previous step to add another rule to allow ICMP packets.

Figure 9: Security Group Rules

Edit Security Group Rules

Security Group Rules

	IP Protocol	From Port	To Port	Source	Actions
<input type="checkbox"/>	TCP	22	22	0.0.0.0/0 (CIDR)	Delete Rule
<input type="checkbox"/>	ICMP	-1	-1	0.0.0.0/0 (CIDR)	Delete Rule

Displaying 2 items

Add Rule

IP Protocol	From Port	To Port	Source Group	CIDR
TCP			CIDR	0.0.0.0/0

Add Rule

2.5 Instance Management

An instance is a running virtual machine within the OpenStack cloud. An instance has a life cycle that is controlled by OpenStack Compute. Compute creates the instances and it is responsible for building a disk image, launching it, reporting the state, attaching persistent storage, and terminating it.

We are ready to start new instances. On the Project Panel select “Instances & Volumes” and then simply click over the “Launch Instance” bottom at the top of this screen (Figure 10).

Figure 10: List of Instances and Volumes

Instance Name	IP Address	Size	Status	Task	Power State	Actions
No items to display.						

Name	Description	Size	Status	Attachments	Actions
No items to display.					

Select the desired VM image for the new instance and click over “Launch”. Figure 11 shows the ubuntu image “precise-amd64” ready to be launched.

Figure 11: Images Available

Image Name	Type	Status	Public	Container Format	Actions
<input checked="" type="checkbox"/> precise-amd64	Image	Active	Yes	OVF	Launch
<input type="checkbox"/> cirros image	Image	Active	Yes	BARE	Launch

Next, we need to assign a name to the instance and select the parameters such as flavor (size of the VM based on CPU, HD and Memory), a keypair and a security group.

We can also assign also a brief description of the instance. If we want to create several instances with the same characteristics, we can also set the number of instances we wish to create. Ensure your image does not exceed your user quota (i.e. disk space); otherwise the instance will fail to spawn.

Figure 12 shows an example in which we’ve created a single instance called “vm1” using the “m1.small” flavor. Available keypair files may be selected in this section. OpenStack

will inject the signed RSA private key into the instance after completing the instantiation process (image creating, booting and network assignment).

Figure 12: New Instances Required Parameters

Launch Instances

Server Name
VM1

User Data
VM1 first instance for project mycompany

Description:
Specify the details for launching an instance. The chart below shows the resources used by this project in relation to the project's quotas.

Project Quotas	
Instance Count (0)	10 Available
VCPU (0)	20 Available
Disk (0 GB)	1000 GB Available
Memory (0 MB)	51200 MB Available

Flavor
m1.small (1VCPU / 10GB Disk / 2048MB Ram)

Keypair
myKey

Instance Count
1

Security Groups
 default
 security_group1

Cancel Launch Instance

Once you've finished filling out the form, click the "Launch Instance" button to boot the instance. The instances section (Figure 13) will show the status of the newly created instance:

Figure 13: Status of the new Instance

The screenshot shows the 'Instances & Volumes' page. On the left sidebar, under the 'Project' section, 'mycompany' is selected. The main area displays a table of instances. A green success message at the top states 'Success: Instance "vm1" launched.' An orange circle highlights this message. Another orange circle highlights the 'Running' status of the 'vm1' row in the table.

Instance Name	IP Address	Size	Status	Task	Power State	Actions
vm1	10.4.0.6	512MB RAM 1 VCPU 0 Disk	Active	None	Running	Edit Instance

The Action section allows Tenant Admins to see the Console Log of the recently created instance, by simply selecting the “View Log” option.

In the same section, the Project Admin can launch the VNC Console view (Figure 14). This view allows the Project Administrator to interact with the instance as it boots up. Note that you cannot use VNC Console from a Chrome browser. You need both Flash installed and a Firefox browser.

For VM created using Ubuntu image, you need to first create a user via ssh to the VM and then use that user to login to this console.

Figure 14: Console View per Instance

The screenshot shows the 'Instance Detail: vm1' page. The left sidebar is identical to Figure 13. The main area has tabs for 'Overview', 'Log', and 'VNC'. The 'VNC' tab is selected, highlighted with a blue border. Below the tabs, a section titled 'Instance VNC Console' contains a message: 'If VNC console is not responding to keyboard input: click the grey status bar below.' A status bar is visible above the terminal window. The terminal window shows a connection message: 'Connected (unencrypted) to: QEMU (instance-00000009)' and a login prompt: 'Ubuntu 12.04.1 LTS vm1 tty1' and 'vm1 login: -'.

2.6 Floating IP Management

OpenStack automatically assigns one private IP address to each spawned instance. Users may assign a public IP that is commonly known as “Floating IP” because it is added to a running instance. Floating IP's let users connect to instances from a public network without having to first connect to the private IP.

Assigning a floating IP is done from the Access & Security section of the Project Panel. The first step is to allocate an IP for the project (tenant). Select Allocate IP To Project, as shown in the following figure:

Figure 15: Floating IP list in Dashboard

The screenshot shows the 'Access & Security' dashboard with the 'Floating IPs' section. The table has the following structure:

IP Address	Instance	Floating IP Pool	Actions
No items to display.			
Displaying 0 items			

A blue button at the top right of the table area is circled in orange and labeled 'Allocate IP To Project'.

The OpenStack administrator will already have IP pools available (Figure 16). A user needs to select an available pool from the Pool drop-down list and click “Allocate IP”.

Figure 16: Floating IP Allocation by Project

The dialog box has the following fields and buttons:

- Pool:** nova
- Description:** Allocate a floating IP from a given floating ip pool.
- Project Quotas:**
 - Floating IP (0)
 - 10 Available
- Buttons:** Cancel, Allocate IP

You should get a conformation message from the system about the allocation process. The next screen shows how to associate the floating IP to the instance.

Figure 17: Confirmation message after assigning Floating IPs

The screenshot shows the 'Access & Security' section of the OpenStack tenant interface. At the top, there's a green success message: 'Success: Successfully allocated Floating IP "172.29.75.11" to project "mycompany"'. Below this is a table titled 'Floating IPs' with one item listed:

<input type="checkbox"/>	IP Address	Instance	Floating IP Pool	Actions
<input type="checkbox"/>	172.29.75.11	-	nova	Associate IP

A red circle highlights the 'Associate IP' button in the Actions column.

Figure 18: Associate Floating IP to Instance

The screenshot shows the 'Associate Floating IP' dialog box. It has two main sections: 'Floating IP' containing the value '172.29.75.11' with a red arrow pointing to it, and 'Description' with the text 'Associate a floating ip with an instance.' Below these are dropdown menus for 'Instance' (containing 'vm1') and 'Pool' (containing 'nova'). At the bottom are 'Cancel' and 'Associate IP' buttons, with the 'Associate IP' button highlighted by a red circle.

Figure 19: Confirmation message after associating Floating IP

The screenshot shows the 'Access & Security' section again. A green success message at the top states: 'Success: Successfully associated Floating IP 172.29.75.11 with Instance: df9ad811-046a-43fc-8601-9ab366742889'. Below this is the same floating IP table as in Figure 17, now showing the association:

<input type="checkbox"/>	IP Address	Instance	Floating IP Pool	Actions
<input type="checkbox"/>	172.29.75.11	df9ad811-046a-43fc-8601-9ab366742889	nova	Disassociate IP

Figure below shows instance VM1 now has the floating IP assigned to it along with the private IP.

Figure 20: Verify Instance to Floating IP Association

The screenshot shows the 'Instances & Volumes' dashboard. In the 'Instances' section, there is a table with one item. The first column is 'Instance Name' (vm1), the second is 'IP Address' (10.4.0.6, 172.29.75.11), the third is 'Size' (512MB RAM | 1 VCPU | 0 Disk), the fourth is 'Status' (Active), the fifth is 'Task' (None), the sixth is 'Power State' (Running), and the seventh is 'Actions' (Edit Instance). The 'IP Address' column is highlighted with an orange border. In the 'Volumes' section, there is a table with one item, but it shows 'No items to display.' Below the tables, it says 'Displaying 1 item' and 'Displaying 0 items' respectively.

Instance Name	IP Address	Size	Status	Task	Power State	Actions
vm1	10.4.0.6 172.29.75.11	512MB RAM 1 VCPU 0 Disk	Active	None	Running	Edit Instance

The process of releasing floating IP address is known as Disassociate IP. This is done by clicking on the “Disassociate IP” button next to the respective instance.

Figure 21: Dis-associate Floating IP from Instance

The screenshot shows the 'Access & Security' dashboard. In the 'Floating IPs' section, there is a table with one item. The first column is 'IP Address' (172.29.75.11), the second is 'Instance' (df9ad811-046a-43fc-8601-9ab366742889), the third is 'Floating IP Pool' (nova), and the fourth is 'Actions' (Disassociate IP). The 'Disassociate IP' button is highlighted with an orange circle.

IP Address	Instance	Floating IP Pool	Actions
172.29.75.11	df9ad811-046a-43fc-8601-9ab366742889	nova	Disassociate IP

Once a floating IP is disassociated from an instance, it can now be re-assigned to another instance as outlined in the previous steps.

2.7 Accessing Instance using Floating IP

An instance can be accessed using the floating IP assigned to the Virtual Machine. First, we must locate the key-pair downloaded to your laptop as documented in the “Creation of Key-Pair” section. Open a terminal on your laptop and locate the private key.

```
% pwd
/Users/username/Downloads
% ls -lrt | grep mykey
-rw-r--r-- 1 username staff 887 Sep 4 12:05 mykey.pem
```

Change private key permission to allow only read/write permission to the user.
% chmod 600 mykey.pem

You can now ssh to the VM1 instance.

```
% ssh ubuntu@172.29.75.11 -i /<path>/mykey.pem
```

Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-virtual x86_64)

....

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo_root" for details.

```
ubuntu@vm1:~$
```

2.8 Instance Snapshots

A virtual machine (instance) is physically represented by one or more files that contain the operating system configuration and state. This is managed differently between various hypervisors but all of them are capable of saving the state of an instance at certain point. This is known as taking a snapshot. Snapshots are very useful in rolling back a system configuration to a state where everything was working as desired but they are also a great alternative to clone instances for creating customized versions of images.

The system administrator will have uploaded some default images for use. Those images may not have all the software and configurations required to start building applications. Customizing a default image is vital to speed-up productivity and to provide a set of user-specific images with applications already configured. Snapshots are created from user instances, from the "Instances & Volumes" section select the instance to be snapshotted and select the option Snapshot from the "Edit Instance" menu as next Figure 22 shows.

Figure 22: Instance Snapshot

The screenshot shows the Cisco OpenStack dashboard with the 'Instances & Volumes' page selected. On the left, there's a sidebar with 'PROJECT mycompany' selected. The main area shows a table of instances with one row for 'VM1'. A context menu is open over 'VM1', and an orange arrow points to the 'Snapshot' option in the list.

Instance Name	IP Address	Size	Status	Task	Power State	Actions
VM1	10.4.0.6 172.29.75.11	2GB RAM 1 VCPU 10.0GB Disk	Active	None	Running	Edit Instance ▾ VNC Console View Log Snapshot Pause Instance Suspend Instance Reboot Instance Terminate Instance

After assigning a name for the snapshot, simply click on Create Snapshot and the system will create it automatically.

Figure 23: Snapshot name assignment

The screenshot shows the 'Create Snapshot' dialog box. It has fields for 'Instance ID' (703fe6da-5e46-4cd1-ba6f-b81073f8a612) and 'Snapshot Name' (VM1_snapshot). There's also a 'Description' section with a note about snapshots preserving disk state. At the bottom are 'Cancel' and 'Create Snapshot' buttons, with an orange arrow pointing to the 'Create Snapshot' button.

The newly created snapshot will appear on the Images and Snapshots section of the Project Panel (Figure 24).

Figure 24: Instance Snapshot

<input type="checkbox"/>	Image Name	Type	Status	Public	Container Format	Actions
<input type="checkbox"/>	precise-amd64	Image	Active	Yes	OVF	<button>Launch</button>
<input type="checkbox"/>	cirros image	Image	Active	Yes	BARE	<button>Launch</button>

Displaying 2 items

<input type="checkbox"/>	Image Name	Type	Status	Public	Container Format	Actions
<input type="checkbox"/>	VM1_snapshot	Snapshot	Queued	No	-	<button>Edit</button> <button>⋮</button>

Displaying 1 item

This is the place to launch a new Instance from that snapshot. In the Action column for the snapshot, there will be the launch button (figure 25). When clicked, OpenStack will request the same information that is requested when a new instance is created from a default image. This instance will be shown on the Instances and Volumes section.

Figure 25: Launch Instance from Snapshot

<input type="checkbox"/>	Image Name	Type	Status	Public	Container Format	Actions
<input type="checkbox"/>	VM1_snapshot	Snapshot	Active	No	OVF	<button>Launch</button> <button>⋮</button>

Displaying 1 item

Volume Snapshots

Figure 26, shows the vm-from-snapshot that we just created.

Figure 26: New Instance from VM1 Snapshot

The screenshot shows the 'Instances & Volumes' section of the OpenStack tenant interface. At the top, there's a green success message: 'Success: Instance "VM2_from_VM1_snap" launched.' On the right, it says 'Logged in as: mycompany' with 'Settings' and 'Sign Out' links. Below the message, the 'Instances' section displays two items in a table:

<input type="checkbox"/>	Instance Name	IP Address	Size	Status	Task	Power State	Actions
<input type="checkbox"/>	VM2_from_VM1_snap	10.4.0.9	512MB RAM 1 VCPU 0 Disk	Active	None	Running	<button>Edit Instance</button>
<input type="checkbox"/>	VM1	10.4.0.6 172.29.75.11	2GB RAM 1 VCPU 10.0GB Disk	Active	None	Running	<button>Edit Instance</button>

Below the table, it says 'Displaying 2 items'. To the right of the table are 'Launch Instance' and 'Terminate Instances' buttons. The 'Volumes' section below has a 'Create Volume' button.

2.9 Nova Volumes

OpenStack (optionally) provides block-level persistent storage to instances through Volumes. This is an iSCSI solution that uses Logical Volume Manager (LVM) for Linux. Users may create volumes based on their respective quota limits. Volumes are managed from the Instances & Volumes section of the Project Panel. We create a volume by clicking over the “Create Volume” bottom as shown on Figure 27.

Figure 27: Create Volume

The screenshot shows the 'Volumes' section of the OpenStack tenant interface. It has a table with the following columns: , Name, Description, Size, Status, Attachments, and Actions. The table is currently empty, displaying 'No items to display.' and 'Displaying 0 items'. An orange arrow points to the 'Create Volume' button located at the top right of the table area.

Users need to assign a reference name for the volume, a short description, and the size in GB. Then simply select “Create Volume” and the new volume will appear in the Instances & Volumes screen.

Figure 28: Fields required for creating a volume

Create Volume

Volume Name
vol-mycompany

Description:
Volumes are block devices that can be attached to instances.

Description
volume1 for mycompany

Size (GB)
1

Cancel Create Volume

The screenshot shows a 'Create Volume' dialog box. At the top left is the title 'Create Volume'. Below it is a 'Volume Name' input field containing 'vol-mycompany', which is highlighted with an orange border. To the right of the input field is a 'Description:' label followed by a text area containing 'Volumes are block devices that can be attached to instances.' Below the input field is a 'Description' label with a text area containing 'volume1 for mycompany'. Underneath these fields is a 'Size (GB)' label with an input field containing '1', also highlighted with an orange border. At the bottom right of the dialog are two buttons: 'Cancel' and a blue 'Create Volume' button, with an orange arrow pointing towards the 'Create Volume' button.

Once the volume has been created, it needs to be attached to an instance. Selecting “Edit Attachments” for the respective volume will open a new window where an instance needs to be selected. This is the attachment process shown in the Figure below.

Figure 29: Attaching a volume to an instance

Manage Volume Attachments

Attachments

	Instance	Device	Actions
No items to display.			

Displaying 0 items

Attach To Instance

Attach to Instance

VM1 (703fe6da-5e46-4cd1-ba6f-b81073f8a61)

Device Name

/dev/vdc

Cancel **Attach Volume**



The next screenshot shows the association between the volume created (vol-mycompany) and the instance selected.

Figure 30: Volume Attached Overview

The screenshot shows a web-based interface for managing volumes. At the top, a header bar contains the title "Volume Detail". Below the header, there is a navigation menu with a single item "Overview" selected. The main content area is titled "Volume Overview: vol-mycompany". Under the "Info" section, details are listed: Name (vol-mycompany), ID (2), Description (volume1 for mycompany), and Status (In-use). The "Specs" section includes Size (1 GB) and Created (09/06/12 at 00:12:56). The "Attachments" section lists "Attached To" with the value "Instance VM1 (703fe6da-5e46-4cd1-ba6f-b81073f8a612) on /dev/vdc", which is highlighted with an orange border.

2.10 Swift Containers

A container is part of the optional OpenStack Object Storage System. A container is basically a reserved storage location where users can allocate data. Swift containers offer high capacity and scalability, strong data consistency, REST APIs. Keep in mind that containers cannot be nested. The data allocating process in OpenStack is done in two steps. First, users need to create a container and second, upload all documents to be stored. Containers are created from the Project Panel: there is section called “Containers” just below “Object Store” shown in Figure 31.

Figure 31: Containers section in Project Panel

The screenshot shows the Cisco OpenStack Project Panel. On the left, there's a sidebar with a 'mycompany' project selected under 'Object Store'. The main area is titled 'Containers' and shows a table with one row: 'No items to display.' Below the table, it says 'Displaying 0 items'. At the top right of the table, there's a 'Create Container' button with a red arrow pointing to it.

At the right side there two available actions: Create and Delete Containers. After selecting “Create Container”, the user will be asked to assign a name for the container and confirmation (Figure 32) After the container is created, it will be listed on the containers section.

Figure 32: New Container Name Assignment

The screenshot shows a 'Create Container' dialog box. It has a 'Container Name' field containing 'backup-mycompany'. To the right, there's a 'Description:' section with the following text:
A container is a storage compartment for your data and provides a way for you to organize your data. You can think of a container as a folder in Windows ® or a directory in UNIX ®. The primary difference between a container and these other file system concepts is that containers cannot be nested. You can, however, create an unlimited number of containers within your account. Data must be stored in a container so you must have at least one container defined in your account prior to uploading data.
At the bottom, there are 'Cancel' and 'Create Container' buttons.

The second step is to upload objects (which is in fact the data from the users). In the Actions column from the containers list, the user may select Upload Objects. Figure 33 illustrates the location of this action.

Figure 33: List of Containers

Containers

Containers

Container Name	Objects	Size	Actions
backup-mycompany	0	0 bytes	List Objects ▾ Upload Object Delete Container

Displaying 1 item

The user will be requested to assign a name to the object and upload the file as the following figure shows.

Figure 34: Uploading Objects to Containers

Upload Object To Container: backup-mycompany

Object Name: Cisco OpenStack User Guide

Description: An object is the basic storage entity and any optional metadata that represents the files you store in the OpenStack Object Storage system. When you upload data to OpenStack Object Storage, the data is stored as-is (no compression or encryption) and consists of a location (container), the object's name, and any metadata consisting of key/value pairs.

File: /Users/arneldionisio/DCN_S [Browse...](#)

[Cancel](#) **Upload Object**

Users may access the list of object from each container by selecting the container

Users may access the list of object from each container by selecting the container

3.0 CLI-Based OpenStack Tenant Operations

An alternative for invoking the OpenStack APIs is to use the python-based command line interface clients. These clients are available as individual packages that need to be installed in the Tenant user system. This section will include all equivalent CLI's needed to repeat the steps outlined in Section 2 (Web-Based OpenStack Tenant Operations) as well as CLI commands that are required for performing certain operations that are not included in the Dashboard.

3.1 OpenStack Python-Based Command Line Interface Clients

Pre-Requisite: All CLI-based commands will need to be executed from a Tenant system. For the purpose of this user guide, we used a Linux system with Ubuntu 12.04 LTS release.

In this Tenant Linux System, install the following OpenStack Service clients.

Installing Nova Client:

```
$ sudo apt-get install python-novaclient
```

Installing Glance Client:

```
$ sudo apt-get install glance-client
```

Installing Keystone Client:

```
$ sudo apt-get install python-keystoneclient
```

Installing Swift Client:

```
$ sudo add-apt-repository ppa:swift-core/release  
$ sudo apt-get update  
$ sudo easy_install pip  
$ sudo pip install python-swiftclient
```

Clients include a help section that lists the available sub-commands and optional arguments. To verify the installation is successful, try the client help commands below.

```
$ nova help  
$ glance -help
```

```
$ keystone help
$ swift help
```

3.2 User Credentials

Prior to using any of the OpenStack clients, users should set up few essential environment variables that are used by these clients to construct the proper API calls. These variables are:

- OS_AUTH_URL
- OS_TENANT_ID
- OS_TENANT_NAME
- OS_USERNAME
- OS_PASSWORD

For descriptions of these variables, please refer to the [Unified CLI wiki page](#).

These variables information can be retrieved from the Dashboard for a given tenant user. On the top right corner, select the “Settings” link. Then, select the section named “OpenStack Credentials” on the left panel as shown in Figure 35 and click over the Download RC File bottom.

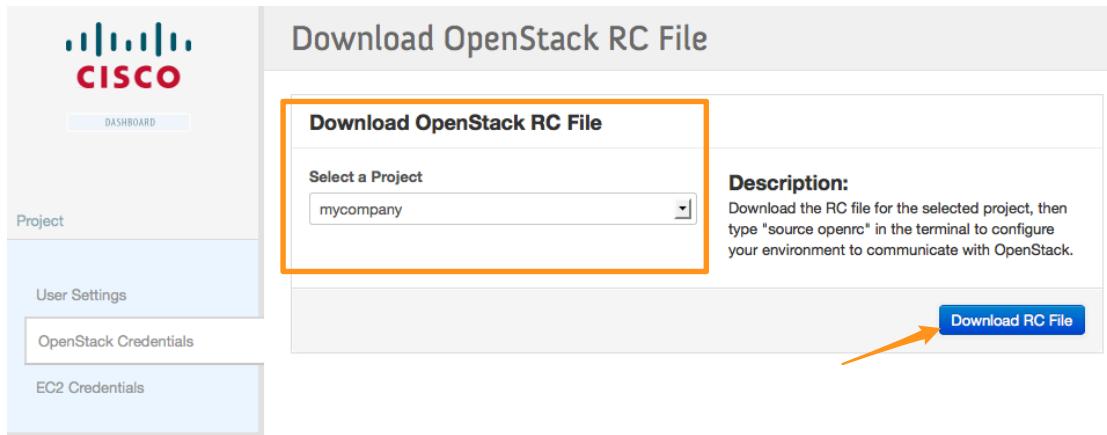
Figure 35: Download OpenStack User Credentials

The screenshot shows the Cisco OpenStack Dashboard. On the left, there's a sidebar with 'PROJECT mycompany' selected. The main area has a title 'Overview'. Below it, a section titled 'Select a month to query its usage:' shows dropdown menus for 'September' and '2012', with a 'Submit' button. A message at the bottom says 'Active Instances: 2 Active Memory: 2GB This Month's VCPU-Hours: 30.80 This Month's GB-Hours: 585.71'. To the right, there's a 'Usage Summary' table with two rows:

Instance Name	VCPUs	Disk	RAM	Uptime
VM1	1	30	2GB	19 hours, 31 minutes
VM2_from_VM1_snap	1	-	512MB	1 hour, 10 minutes

At the bottom right of the summary table is a 'Download CSV Summ' button. In the top right corner of the main dashboard area, there are three links: 'Settings' (with an orange arrow pointing to it), 'Sig', and 'Sig'.

Figure 36: Download OpenStack RC File



The file downloaded (openrc) will contain private information about the user account, it is important to not share it and keep in a private secure location.

```
#!/bin/bash

# With the addition of Keystone, to use an openstack cloud you should
# authenticate against keystone, which returns a **Token** and **Service
# Catalog**. The catalog contains the endpoint for all services the
# user/tenant has access to - including nova, glance, keystone, swift.
#
# *NOTE*: Using the 2.0 *auth api* does not mean that compute api is 2.0. We
# will use the 1.1 *compute api*
export OS_AUTH_URL=http://172.29.74.193:5000/v2.0

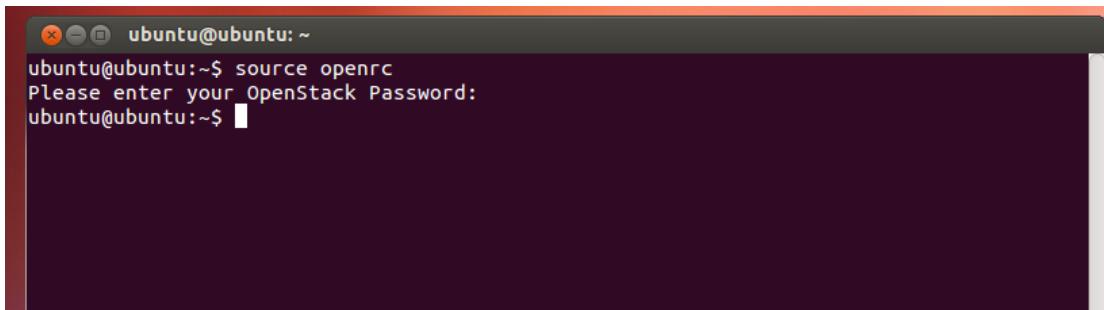
# With the addition of Keystone we have standardized on the term **tenant**
# as the entity that owns the resources.
export OS_TENANT_ID=6d35d471937e4862997372b1a1d3f7ff
export OS_TENANT_NAME=mycompany

# In addition to the owning entity (tenant), openstack stores the entity
# performing the action as the **user**.
export OS_USERNAME=mycompany

# With Keystone you pass the keystone password.
echo "Please enter your OpenStack Password: "
read -s OS_PASSWORD_INPUT
export OS_PASSWORD=$OS_PASSWORD_INPUT
```

Open a console terminal in your Tenant linux system and type "source openrc" to configure your environment to communicate with OpenStack. Note that you will need to do this in each shell from which you intend to use CLI clients. Note the openrc file will ask you to input the Tenant user password.

Figure 37: Source Openrc File



A screenshot of a terminal window titled "ubuntu@ubuntu: ~". The window contains the following text:
ubuntu@ubuntu:~\$ source openrc
Please enter your OpenStack Password:
ubuntu@ubuntu:~\$ █

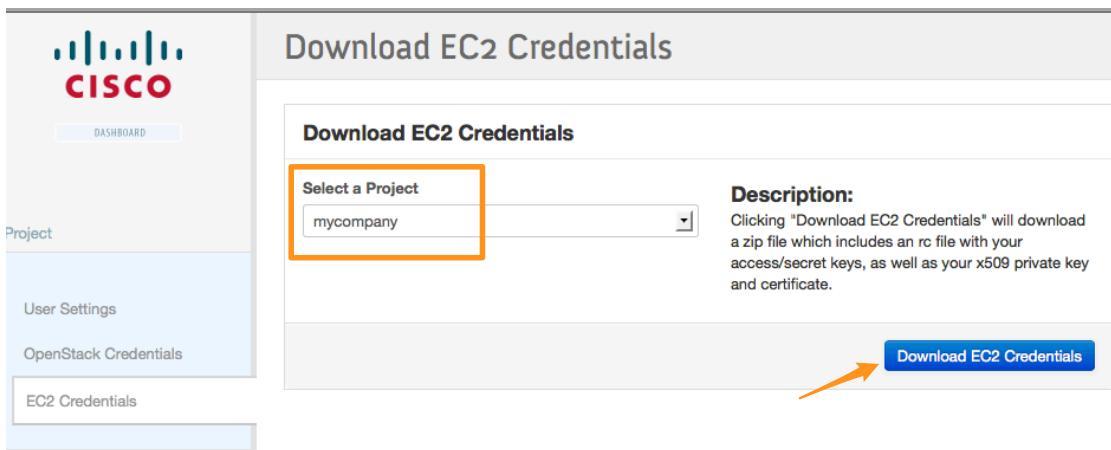
OpenStack offers a high level of compatibility for the EC2 APIs. The complete API compatibility matrix is included on section 4. When users use euca2ools as a client to construct EC2 APIs calls, different variables need to be set.

These are:

EC2_ACCESS_KEY
EC2_SECRET_KEY
EC2_URL
EC2_USER_ID
EC2_PRIVATE_KEY
EC2_CERT
EUCALYPTUS_CERT

For descriptions of these variables, please refer to the Eucalyptus documentation. The process to set up the EC2 credentials is very similar to the one for OpenStack credentials, but the settings will be provided in a zip file which includes the EC2 cert.pem keys as well as an rc file. This zip file can be downloaded from the same “settings” page.

Figure 38: EC2 Credentials



Sourcing User Credentials EC2 File:

```
# source /PATH_TO_FILE/ec2rc.sh
```

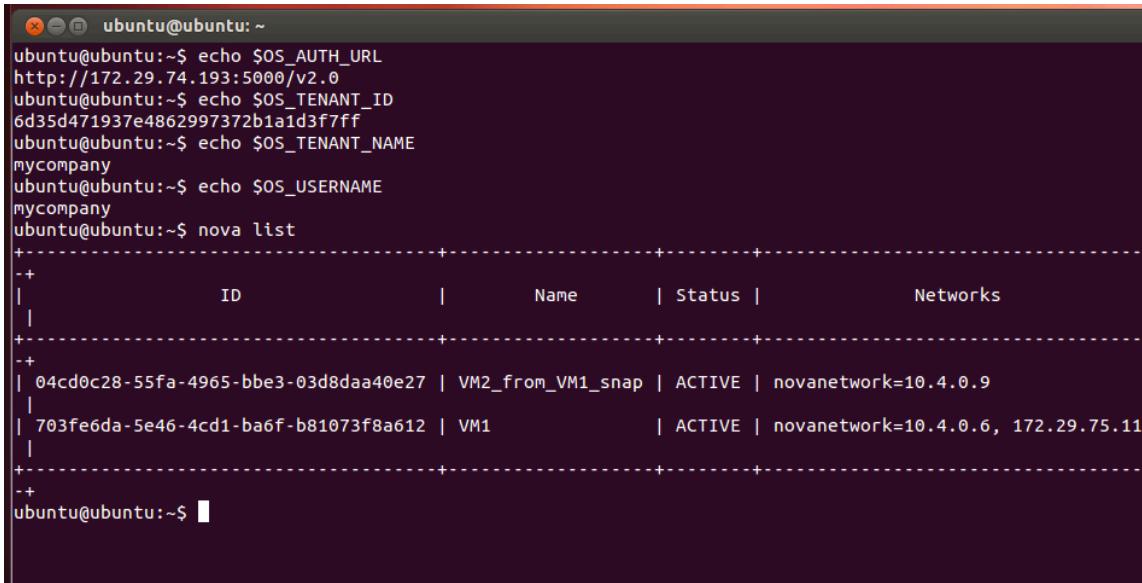
3.3 Tenant Operations

3.3.1 Project Overview

- # To list all existing instances in a project use the following nova command:
nova list
- # To list all command options for “nova list”
nova help list
- # To list an instance matching a particular name
nova list -name=<instance_name>

Figure 39; list all the existing instances in project “mycompany”

Figure 39: Tenant Instance List



```
ubuntu@ubuntu:~$ echo $OS_AUTH_URL
http://172.29.74.193:5000/v2.0
ubuntu@ubuntu:~$ echo $OS_TENANT_ID
6d35d471937e4862997372b1a1d3f7ff
ubuntu@ubuntu:~$ echo $OS_TENANT_NAME
mycompany
ubuntu@ubuntu:~$ echo $OS_USERNAME
mycompany
ubuntu@ubuntu:~$ nova list
+-----+-----+-----+
| ID      | Name    | Status |
+-----+-----+-----+
| 04cd0c28-55fa-4965-bbe3-03d8daa40e27 | VM2_from_VM1_snap | ACTIVE |
| 703fe6da-5e46-4cd1-ba6f-b81073f8a612 | VM1               | ACTIVE |
+-----+-----+-----+
ubuntu@ubuntu:~$
```

3.3.2 Image Overview

To list all existing nova image related CLI's:

nova help | grep image

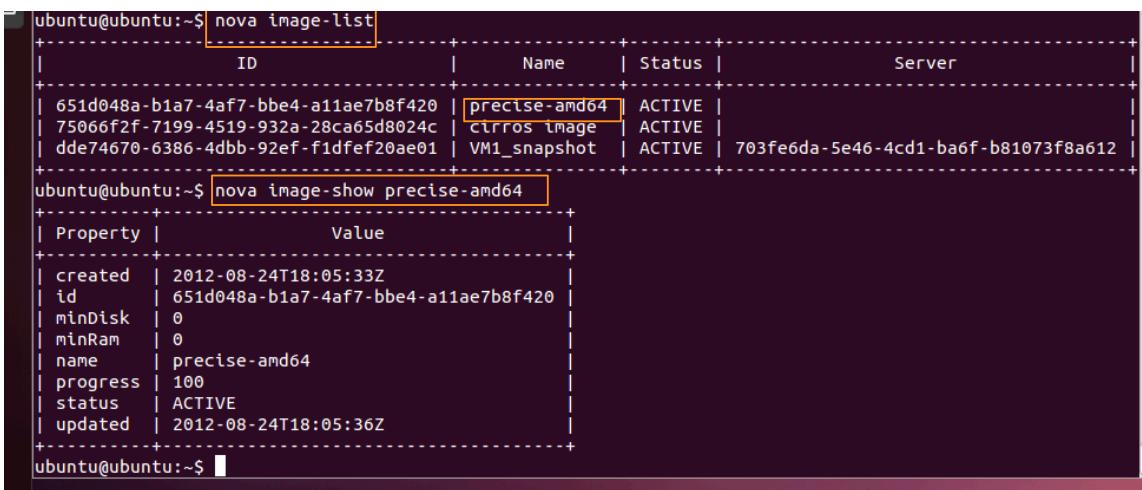
To list all existing image(s) in a project

nova image-list

To view details of a particular image

nova image-show <image_name>

Figure 40: Nova Image list for project "mycompany"



```
ubuntu@ubuntu:~$ nova image-list
+-----+-----+-----+
| ID      | Name    | Status |
+-----+-----+-----+
| 651d048a-b1a7-4af7-bbe4-a11ae7b8f420 | precise-amd64   | ACTIVE |
| 75066f2f-7199-4519-932a-28ca65d8024c | cirros_image    | ACTIVE |
| dde74670-6386-4dbb-92ef-f1dfef20ae01 | VM1_snapshot    | ACTIVE |
+-----+-----+-----+
ubuntu@ubuntu:~$ nova image-show precise-amd64
+-----+
| Property | Value
+-----+
| created | 2012-08-24T18:05:33Z
| id      | 651d048a-b1a7-4af7-bbe4-a11ae7b8f420
| minDisk | 0
| minRam  | 0
| name    | precise-amd64
| progress| 100
| status   | ACTIVE
| updated | 2012-08-24T18:05:36Z
+-----+
ubuntu@ubuntu:~$
```

3.3.3 Creation of Key-Pair

To list all existing nova keypair related CLI's:

nova help | grep keypair

```
# To list all existing keypair (s) in a project
nova keypair-list
# To create a new keypair to be used for new instances
nova keypair-add --pub_key <id_rsa.pub> <key_name>
```

Figure 41: Create Key-Pair

The screenshot shows a terminal session on an Ubuntu system. It starts with listing existing keypairs, then generating a new RSA key pair using ssh-keygen, and finally adding the public key to a Nova keypair.

```
ubuntu@ubuntu:~$ nova keypair-list
+-----+-----+
| Name | Fingerprint |
+-----+-----+
| myKey | 8a:46:50:c4:3d:e0:f8:ad:a4:0f:67:42:a2:b0:e6:df |
+-----+-----+
ubuntu@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Created directory '/home/ubuntu/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
2b:fd:bb:ab:69:fb:de:b1:7c:be:31:2d:39:2d:67:b1 ubuntu@ubuntu
The key's randomart image is:
+---[ RSA 2048]---+
| |
| |
| |
| |
| S . |
| .. +o|
. o . BE=|
..o o o.0 |
.=B++oo. |
+-----+
ubuntu@ubuntu:~$ cd .ssh
ubuntu@ubuntu:~/ssh$ nova keypair-add --pub_key id_rsa.pub mykey2
ubuntu@ubuntu:~/ssh$ nova keypair-list
+-----+-----+
| Name | Fingerprint |
+-----+-----+
| myKey | 8a:46:50:c4:3d:e0:f8:ad:a4:0f:67:42:a2:b0:e6:df |
| mykey2 | 2b:fd:bb:ab:69:fb:de:b1:7c:be:31:2d:39:2d:67:b1 |
```

3.3.4 Creation of Security Groups

```
# To list all existing nova security group related CLI's:
nova help | grep secgroup
# To list all existing security group (s) in a project
nova secgroup-list
```

```
# To list all existing security group rules(s) in a specific security-group  
nova secgroup-list-rules <secgroup_name>  
# To create a new security-group to be used for new instances  
nova secgroup-create <secgroup_name> <description>  
# To add a new security-group-rule to an existing security group  
nova secgroup-add-rule <ip_protocol> <from_port> <to_port> <cidr>
```

Figure 42: View Existing Security Groups and list rules

```
ubuntu@ubuntu:~/.ssh$ nova secgroup-list  
+-----+  
| Name | Description |  
+-----+  
| default | default |  
| security_group1 | security group1 |  
+-----+  
ubuntu@ubuntu:~/.ssh$ nova secgroup-list-rules  
usage: nova secgroup-list-rules <secgroup>  
error: too few arguments  
Try `nova help secgroup-list-rules` for more information.  
ubuntu@ubuntu:~/.ssh$ nova secgroup-list-rules security_group1  
+-----+-----+-----+-----+-----+  
| IP Protocol | From Port | To Port | IP Range | Source Group |  
+-----+-----+-----+-----+-----+  
| icmp | -1 | -1 | 0.0.0.0/0 | |  
| tcp | 22 | 22 | 0.0.0.0/0 | |  
+-----+-----+-----+-----+-----+  
ubuntu@ubuntu:~/.ssh$
```

Figure 43: Create New Security Group and add Rules

```

ubuntu@ubuntu:~/.ssh$ nova secgroup-create security_group2 "security group2"
+-----+
| Name      | Description   |
+-----+
| security_group2 | security group2 |
+-----+
ubuntu@ubuntu:~/.ssh$ nova secgroup-add-rule security_group2 tcp 22 22 0.0.0.0/0
+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+
| tcp          | 22        | 22      | 0.0.0.0/0 |           |
+-----+-----+-----+-----+
ubuntu@ubuntu:~/.ssh$ nova secgroup-add-rule security_group2 icmp -1 -1 0.0.0.0/0
+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+
| icmp         | -1        | -1      | 0.0.0.0/0 |           |
+-----+-----+-----+-----+
ubuntu@ubuntu:~/.ssh$ nova secgroup-list-rules security_group2
+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+
| icmp         | -1        | -1      | 0.0.0.0/0 |           |
| tcp          | 22        | 22      | 0.0.0.0/0 |           |
+-----+-----+-----+-----+
ubuntu@ubuntu:~/.ssh$ nova secgroup-list
+-----+
| Name      | Description   |
+-----+
| default   | default       |
| security_group1 | security group1 |
| security_group2 | security group2 |
+-----+
ubuntu@ubuntu:~/.ssh$ █

```

3.3.5 Instance Management

- # To list all existing nova instance flavors
nova flavor-list
- # To list all existing image (s) in a project
nova image-list
- # To list all existing keypair (s) in a project
nova keypair-list
- # To list all existing security group(s) in a specific project
nova secgroup-list
- # To create a new instance within a project
nova boot -flavor <flavor_id> --image <image_id> --keyname <mykey_name> --security_groups <secgroup_name> <Instance_name>
- # To show details of an instance within a project
nova show <instance_name> or <instance_id>

Figure 44: Flavor list, image-list and secgroup-list

```
ubuntu@ubuntu:~$ nova flavor-list
+---+---+---+---+---+---+---+
| ID | Name | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor |
+---+---+---+---+---+---+---+
| 1 | m1.tiny | 512 | 0 | 0 | 1 | 1.0 |
| 2 | m1.small | 2048 | 10 | 20 | 1 | 1.0 |
| 3 | m1.medium | 4096 | 10 | 40 | 2 | 1.0 |
| 4 | m1.large | 8192 | 10 | 80 | 4 | 1.0 |
| 5 | m1.xlarge | 16384 | 10 | 160 | 8 | 1.0 |
+---+---+---+---+---+---+---+
ubuntu@ubuntu:~$ nova image-list
+-----+-----+-----+-----+
| ID | Name | Status | Server |
+-----+-----+-----+-----+
| 651d048a-b1a7-4af7-bbe4-a11ae7b8f420 | precise-amd64 | ACTIVE | |
| 75066f2f-7199-4519-932a-28ca65d8024c | cirros image | ACTIVE |
| dde74670-6386-4dbb-92ef-f1dfef20ae01 | VM1_snapshot | ACTIVE | 703fe6da-5e46-4cd1-ba6f-b81073f8a612 |
+-----+-----+-----+-----+
ubuntu@ubuntu:~$ nova keypair-list
+-----+-----+
| Name | Fingerprint |
+-----+-----+
| myKey | 8a:46:50:c4:3d:e0:f8:ad:a4:0f:67:42:a2:b0:e6:df |
| mykey2 | 2b:fd:bb:ab:69:fb:de:b1:7c:be:31:2d:39:2d:67:b1 |
+-----+-----+
ubuntu@ubuntu:~$ nova secgroup-list
+-----+-----+
| Name | Description |
+-----+-----+
| default | default |
| security_group1 | security group1 |
| security_group2 | security group2 |
+-----+-----+
```

Figure 45: Adding an instance using nova CLI

```
+-----+
ubuntu@ubuntu:~$ nova boot --flavor 1 --image 651d048a-b1a7-4af7-bbe4-a11ae7b8f420 --key_name mykey2 --security_groups security_group1 VM3
+-----+
| Property | Value |
+-----+
| OS-DCF:diskConfig | MANUAL
| OS-EXT-STS:power_state | 0
| OS-EXT-STS:task_state | scheduling
| OS-EXT-STS:vm_state | building
| accessIPv4 |
| accessIPv6 |
| adminPass | 7B2XK2bDPDUdu
| config_drive |
| created | 2012-09-06T18:04:58Z
| flavor | m1.tiny
| hostId |
| id | f7c5cae2-ae43-4714-96b7-87970ee0f7da
| image | precise-amd64
| key_name | mykey2
| metadata | {}
| name | VM3
| progress | 0
| status | BUILD
| tenant_id | 6d35d471937e4862997372b1a1d3f7ff
| updated | 2012-09-06T18:04:58Z
| user_id | 204b7ad6fe8a4e0fa4c81cab42b98d26
+-----+
ubuntu@ubuntu:~$ nova list
```

Figure 46: List of existing Instance within a project

ID	Name	Status	Networks
04cd0c28-55fa-4965-bbe3-03d8daa40e27	VM2_from_VM1_snap	ACTIVE	novanetwork=10.4.0.9
703fe6da-5e46-4cd1-ba6f-b81073f8a612	VM1	ACTIVE	novanetwork=10.4.0.6, 172.29.75.11
f7c5cae2-ae43-4714-96b7-87970ee0f7da	VM3	ACTIVE	novanetwork=10.4.0.8

3.3.6 Floating IP Management

```
# To list all existing nova floating-ip commands
nova help | grep floating-ip
# To list all existing ip pool list (s) in a project
nova floating-ip-pool-list
# To list all existing ip pool in a project
nova floating-ip-list
# To create a new floating-ip
nova floating-ip-create
# To associate a new floating-ip to an instance
nova add-floating-ip <instance_name> <floating_ip_address>
```

Figure 47: Create Floating-IP

nova floating-ip-pool-list			
name			
nova			
nova floating-ip-list			
Ip	Instance Id	Fixed Ip	Pool
172.29.75.11	703fe6da-5e46-4cd1-ba6f-b81073f8a612	10.4.0.6	nova
nova floating-ip-create			
Ip	Instance Id	Fixed Ip	Pool
172.29.75.12	None	None	nova
nova floating-ip-list			
Ip	Instance Id	Fixed Ip	Pool
172.29.75.11	703fe6da-5e46-4cd1-ba6f-b81073f8a612	10.4.0.6	nova
172.29.75.12	None	None	nova

Figure 48: Associate Floating IP to an existing Instance

```
+ ubuntu@ubuntu:~$ nova add-floating-ip VM3 172.29.75.12
ubuntu@ubuntu:~$ nova floating-ip-list
+-----+-----+-----+
|   Ip    |     Instance Id      | Fixed Ip | Pool |
+-----+-----+-----+
| 172.29.75.11 | 703fe6da-5e46-4cd1-ba6f-b81073f8a612 | 10.4.0.6 | nova |
| 172.29.75.12 | f7c5cae2-ae43-4714-96b7-87970ee0f7da | 10.4.0.8 | nova |
+-----+-----+-----+
ubuntu@ubuntu:~$
```

#To remove floating IP's, first dis-associate floating ip from the server
nova remove-floating-ip <server> <address>

#Then de-allocate the floating ip and return back to the pool
nova floating-ip-delete <address>

3.3.7 Accessing Instance using Floating IP

Verify the floating IP is associated with the correct Instance and the VM is currently active using “*nova show <instance_name>*”

Figure 49: Accessing Instance using Floating IP

```
ubuntu@ubuntu:~$ nova show VM3
+-----+-----+
|       Property      |             Value
+-----+-----+
| OS-DCF:diskConfig | MANUAL
| OS-EXT-STS:power_state | 1
| OS-EXT-STS:task_state | None
| OS-EXT-STS:vm_state | active
| accessIPv4
| accessIPv6
| config_drive
| created | 2012-09-06T18:04:58Z
| flavor | m1.tiny
| hostId | b289f313ac0aa18e8df6a0bb5a7ebfa25160913f685c14e38a204a88
| id | f7c5cae2-ae43-4714-96b7-87970ee0f7da
| image | precise-amd64
| key_name | mykey2
| metadata | {}
| name | VM3
| novanetwork network | 10.4.0.8, 172.29.75.12
| progress | 0
| status | ACTIVE
| tenant_id | 6d35d471937e4862997372b1a1d3f7ff
| updated | 2012-09-06T18:05:14Z
| user_id | 204b7ad6fe8a4e0fa4c81cab42b98d26
+-----+-----+
```

Figure 50: SSH to new VM using floating-ip

```

ubuntu@ubuntu:~$ cd .ssh
ubuntu@ubuntu:~/ssh$ ls
id_rsa  id_rsa.pub
ubuntu@ubuntu:~/ssh$ ssh ubuntu@172.29.75.12 -i id_rsa.pub
The authenticity of host '172.29.75.12 (172.29.75.12)' can't be established.
ECDSA key fingerprint is 0b:d9:98:9b:71:79:05:9e:d9:69:5d:3b:79:a2:65:44.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.29.75.12' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-virtual x86_64)

 * Documentation:  https://help.ubuntu.com/

 System information as of Thu Sep  6 18:41:16 UTC 2012

 System load:  0.0          Processes:      59
 Usage of /:   33.7% of 1.98GB  Users logged in:    0
 Memory usage: 8%           IP address for eth0: 10.4.0.8
 Swap usage:   0%

 Graph this data and manage this system at https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

Get cloud support with Ubuntu Advantage Cloud Guest
http://www.ubuntu.com/business/services/cloud

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@vm3:~$ 

```

3.3.8 Instance Snapshots

```

# To list all existing nova snapshot commands
nova help | grep snap
# To list all existing snap shot list (s) in a project
nova image-list
# To create a snap shot of an existing instance
nova image-create <instance_name> <snapshot_name>

```

Figure 51: Creating a Snapshot of an Instance

```
+ ubuntu@ubuntu:~/ssh$ nova image-list
+-----+-----+-----+-----+
| ID      | Name    | Status | Server   |
+-----+-----+-----+-----+
| 651d048a-b1a7-4af7-bbe4-a11ae7b8f420 | precise-amd64 | ACTIVE | |
| 75066f2f-7199-4519-932a-28ca65d8024c | cirros image  | ACTIVE |
| dde74670-6386-4dbb-92ef-f1dfef20ae01 | VM1_snapshot | ACTIVE | 703fe6da-5e46-4cd1-ba6f-b81073f8a612 |
+-----+-----+-----+-----+
ubuntu@ubuntu:~/ssh$ nova help image-create
usage: nova image-create [-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:
  --poll   Blocks while instance snapshots so progress can be reported.
ubuntu@ubuntu:~/ssh$ nova image-create VM3 SNAPofVM3
ubuntu@ubuntu:~/ssh$ nova image-list
+-----+-----+-----+-----+
| ID      | Name    | Status | Server   |
+-----+-----+-----+-----+
| 46881540-019c-455f-8c4f-b74dd89f8b28 | SNAPofVM3 | SAVING | f7c5cae2-ae43-4714-96b7-87970ee0f7da |
| 651d048a-b1a7-4af7-bbe4-a11ae7b8f420 | precise-amd64 | ACTIVE |
| 75066f2f-7199-4519-932a-28ca65d8024c | cirros image  | ACTIVE |
| dde74670-6386-4dbb-92ef-f1dfef20ae01 | VM1_snapshot | ACTIVE | 703fe6da-5e46-4cd1-ba6f-b81073f8a612 |
+-----+-----+-----+-----+
ubuntu@ubuntu:~/ssh$
```

This snapshot image can be used to create a new instance within this project.

3.3.9 Nova Volumes

```
# To list all existing nova volume commands
nova help | grep volume
# To list all existing volume list (s) in a project
nova volume-list
# To show volume details
nova volume-show <volume_id>
```

Figure 52: Nova Volume List

```
ubuntu@ubuntu:~/ssh$ nova volume-list
+-----+-----+-----+-----+-----+
| ID | Status | Display Name | Size | Volume Type | Attached to |
+-----+-----+-----+-----+-----+
| 2 | in-use | vol-mycompany | 1 | None | 703fe6da-5e46-4cd1-ba6f-b81073f8a612 |
+-----+-----+-----+-----+-----+
ubuntu@ubuntu:~/ssh$ nova volume-show 2
```

```
# To find out which VM this volume is attached to
nova show <instance_id>
```

Figure 53: Nova Volume Attached to Instance

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-STS:power_state	1
OS-EXT-STS:task_state	None
OS-EXT-STS:vm_state	active
accessIPv4	
accessIPv6	
config_drive	
created	2012-09-05T05:34:24Z
flavor	m1.small
hostId	bc7a1d4b8a5459d0e091f4b73e65194babf381e79be733766816887b
id	703fe6da-5e46-4cd1-ba6f-b81073f8a612
image	precise-amd64
key_name	myKey
metadata	{}
name	VM1
novanetwork network	10.4.0.6, 172.29.75.11
progress	0
status	ACTIVE
tenant_id	6d35d471937e4862997372b1a1d3f7ff
updated	2012-09-05T23:43:49Z
user_id	204b7ad6fe8a4e0fa4c81cab42b98d26

To create a new volume

nova volume-create --display_name <volume_name> <size>

Figure 54: Create a new Nova Volume

```
ubuntu@ubuntu:~/.ssh$ nova help volume-create
usage: nova volume-create [--snapshot_id <snapshot_id>]
                           [--display_name <display_name>]
                           [--display_description <display_description>]
                           [--volume_type <volume_type>]
                           <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)
  --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)
ubuntu@ubuntu:~/.ssh$ nova volume-create --display_name vol2-mycompany 2
ubuntu@ubuntu:~/.ssh$
ubuntu@ubuntu:~/.ssh$ nova volume-list
+---+-----+-----+-----+
| ID | Status | Display Name | Size | Volume Type | Attached to |
+---+-----+-----+-----+
| 2 | in-use | vol-mycompany | 1 | None | 703fe6da-5e46-4cd1-ba6f-b81073f8a612 |
| 3 | available | vol2-mycompany | 2 | None | |
+---+-----+-----+-----+
ubuntu@ubuntu:~/.ssh$
```

To attach a new volume to a particular instance

*nova volume-attach <instance_name> <volume_id> <name_of_device>***Figure 55: Attach New Nova Volume to Instance**

```
ubuntu@ubuntu:~/.ssh$ nova help volume-attach
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.
ubuntu@ubuntu:~/.ssh$ nova volume-attach VM3 3 /dev/vdc
ubuntu@ubuntu:~/.ssh$ nova volume-list
+---+-----+-----+-----+
| ID | Status | Display Name | Size | Volume Type | Attached to |
+---+-----+-----+-----+
| 2 | in-use | vol-mycompany | 1 | None | 703fe6da-5e46-4cd1-ba6f-b81073f8a612 |
| 3 | in-use | vol2-mycompany | 2 | None | f7c5cae2-ae43-4714-96b7-87970ee0f7da |
+---+-----+-----+-----+
ubuntu@ubuntu:~/.ssh$
```

3.3.10 Swift Containers

The

3.4 Image Management

This section will outline the steps to add images, list images and delete images.

The last two steps, list and delete images, are possible to do in the OpenStack Horizon dashboard. However, the current version of the Horizon dashboard in OpenStack Essex release does not support adding an image directly. This section will demonstrate how to accomplish all three functions (add, list and delete images) using OpenStack Glance client.

3.4.1 Adding Images

Images are files containing information about a virtual disk previously formatted and configured. Uploading images to the Image Service Glance can only be done through the command line client. Glance client needs to be installed in the user system and the user credentials file (openrc) should be sourced in order to execute any of the command herein mentioned. These steps are covered in Sections 3.1 and 3.2.

General Glance import options are:

```
# glance add name=<A name for the created image> \
is_public= <Optional, remove this switch for private images> \
disk_format= <ami|aki|ari|qcow2|raw> \
container_format= <ami|aki|ari|qcow2|bare> \
< (Your downloaded image path)
```

Two formats are supported by Glance:

- AMI (Amazon Machine Image)
- QCOW (Qemu Copy On Write)
- VMDK (Virtual Machine Disk) only in the “Monolithic File Flat” format.
<http://www.vmware.com/pdf/VirtualDiskManager.pdf>

Operating system images may be downloaded from a variety of free sources on the Internet or from your EC2 account.

Once you have your image, the first step is to identify the type of image you want to import. This information is usually included in the place you downloaded the image from. In case you have a raw iso image and want to convert it to a qcows image, you can take a look at the oz image builder (<https://github.com/rackerjoe/oz-image-build>). The repository also features links to various qcows2 images.

Once you have identified the image, follow the instructions below appropriate for your image type.

3.4.2 Adding AMI (Amazon Machine Image) Images

For the purpose of this install guide, test images are downloaded from this site;
<https://launchpad.net/cirros/+download>

STEP1: Download the AMI image from the link above and extract the contents:

```
%cd downloads
%tar -xvf cirros-0.3.0-x86_64-uec.tar.gz
```

Figure 56: AMI Image Download

The screenshot shows a terminal window titled "ubuntu@ubuntu: ~/Downloads". The user has navigated to the Downloads directory and extracted a tar.gz file named "cirros-0.3.0-x86_64-uec.tar.gz". After extraction, several files are listed in the directory:

	File Name	Owner	Size	Date	Type
-rw-r--r--	cirros-0.3.0-x86_64-blank.img	ubuntu	25165824	Oct 20 2011	File
-rw-----	cirros-0.3.0-x86_64-vmlinuz	ubuntu	4731440	Oct 20 2011	File
-rw-r--r--	cirros-0.3.0-x86_64-initrd	ubuntu	2254249	Oct 20 2011	File
-rw-rw-r--	cirros-0.3.0-x86_64-uec.tar.gz	ubuntu	6957349	Sep 11 00:27	File

Using The files required for the AMI format are the kernel (vmlinuz), ramdisk (initrd) and file disk (.img). The following steps will upload a test image known as Cirros. Cirros Image Files:

cirros-0.3.0-x86_64-blank.img
 cirros-0.3.0-x86_64-initrd
 cirros-0.3.0-x86_64-vmlinuz

From the Tenant Linux System, execute the following commands:

STEP2: Modify openrc and add the following entry (see Section 3.2). Make sure to source the openrc after the change.

```
export GLANCE_HOSTPORT=172.29.74.193
```

```
%source openrc
```

STEP3: Add AMI kernel image

```
% glance --url http://$GLANCE_HOSTPORT add name="cirros-kernel2" is_public=true
container_format=aki disk_format=aki < "cirros-0.3.0-x86_64-vmlinuz"
```

STEP4: Add AMI ram image

```
glance --url http://$GLANCE_HOSTPORT add name="cirros-ram2" is_public=true
container_format=ari disk_format=ari < "cirros-0.3.0-x86_64-initrd"
```

STEP5: Verify AMI Kernel and RAM Image upload

Figure 57: Verify AMI Kernel and RAM Image upload

ID	Name	Disk Format	Container Format	Size
e2c3c580-615e-4e01-b689-7f5599375b8d	cirros-kernel2	aki	aki	4731440

ID	Name	Disk Format	Container Format	Size
358cb1fc-ad6c-4140-a2a7-00b3b698c45e	cirros-ram2	ari	ari	2254249

STEP6: Add AMI file disk

```
glance --url http://$GLANCE_HOSTPORT add name="cirros" is_public=true
container_format=ami disk_format=ami kernel_id=<kernel_id> ramdisk_id=<ramdisk_id>
< "cirros-0.3.0-x86_64-blank.img"
```

Use the AMI kernel_id and ramdisk_id in STEP5.

STEP6: Verify AMI image Upload

Figure 58: Verify AMI Image Upload

ID	Name	Disk Format	Container Format	Size
e5c06a1b-c909-435b-93a7-95ec80d4bd5f	cirros2	ami	ami	25165824

You can also verify from the Horizon dashboard that the image “cirros2” was added in the list of images available for MyCompany Tenant.

3.4.3 Adding QCOW (Qemo Copy on Write) Images

STEP1: Download the QCOW image from the link below:

<https://launchpad.net/cirros/+download>

```
%cd downloads
%qemu-img info cirros-0.3.0-x86_64-disk.img
```

Figure 59: QCOW Image download

```
ubuntu@ubuntu:~/Downloads$ ls -lrt cirros-0.3.0-x86_64-disk.img
-rw-rw-r-- 1 ubuntu ubuntu 9761280 Sep 11 01:20 cirros-0.3.0-x86_64-disk.img
ubuntu@ubuntu:~/Downloads$ qemu-img info cirros-0.3.0-x86_64-disk.img
image: cirros-0.3.0-x86_64-disk.img
file format: qcow2
virtual size: 39M (41126400 bytes)
disk size: 9.3M
cluster size: 65536
ubuntu@ubuntu:~/Downloads$
```

STEP2: Add QCOW2 image. These images do not require any kernel or ramdisk images, you can simply import them as:

```
glance --url http://$GLANCE_HOSTPORT add name="cirros qcow2" is_public=true
container_format=bare disk_format=qcow2 < "cirros-0.3.0-x86_64-disk.img"
```

STEP3: Verify QCOW2 image upload:

Figure 60: QCOW2 Image upload

ID	Name	Disk Format	Container Format	Size
3e6b33d9-a4dc-4cbd-8763-6ea88789332c	cirros qcow2	qcow2	bare	9761280

3.4.4 Listing Images

The list of all the available images that a tenant can use can be retrieved using the following methods;

- 1/ Horizon dashboard as explained in Section 2.5
- 2/ Using Glance client
- 3/ Using Nova Client

The following command will display the available images using glance client:

Figure 61: Glance Image List

ID	Name	Disk Format	Container Format	Size
3e6b33d9-a4dc-4cbd-8763-6ea88789332c	cirros qcow2	qcow2	bare	9761280
e5c06a1b-c909-435b-93a7-95ec80d4bd5f	cirros2	ami	ami	25165824
358cb1fc-ad6c-4140-a2a7-00b3b698c45e	cirros-ram2	ari	ari	2254249
e2c3c580-615e-4e01-b689-7f5599375b8d	cirros-kernel2	aki	aki	4731440
a26aa9e2-92ab-4c4b-aa9c-301bbb9f3764	cirros	ami	ami	25165824
ecdabd0f-5e6d-4d86-8dbe-ddf997398303	cirros-ram	ari	ari	2254249
119e9b5a-aaba-4051-8d31-6132642420a7	cirros-kernel	aki	aki	4731440
46881540-019c-455f-8c4f-b74dd89f8b28	SNAofVM3	qcow2	ovf	738066432
dde74670-6386-4dbb-92ef-f1dfef20ae01	VM1_snapshot	qcow2	ovf	869335040
651d048a-b1a7-4af7-bbe4-allae7b8f420	precise-amd64	qcow2	ovf	233177088
Fetch next page? [Y/n] y				
75066f2f-7199-4519-932a-28ca65d8024c	cirros image	qcow2	bare	9761280
ubuntu@ubuntu:~/Downloads\$				

Note, the list above includes all images that was created by the Cloud admin and initially assigned to the “mycompany” tenant (e.g. “precise-amd64”), the snapshot images created as part of Section 2.8 / Section 3.3.8 (e.g. “VM1_snapshot”) and the AMI and QCOW2 image created in previous two sections 3.4.2 and 3.4.3.

Images can also be listed using nova client;

Figure 62: Nova Image List

ID	Name	Status	Server
119e9b5a-aaba-4051-8d31-6132642420a7	cirros-kernel	ACTIVE	
358cb1fc-ad6c-4140-a2a7-00b3b698c45e	cirros-ram2	ACTIVE	
3e6b33d9-a4dc-4cbd-8763-6ea88789332c	cirros qcow2	ACTIVE	
46881540-019c-455f-8c4f-b74dd89f8b28	SNAofVM3	ACTIVE	f7c5cae2-ae43-4714-96b7-87970ee0f7da
651d048a-b1a7-4af7-bbe4-allae7b8f420	precise-amd64	ACTIVE	
75066f2f-7199-4519-932a-28ca65d8024c	cirros image	ACTIVE	
a26aa9e2-92ab-4c4b-aa9c-301bbb9f3764	cirros	ACTIVE	
dde74670-6386-4dbb-92ef-f1dfef20ae01	VM1_snapshot	ACTIVE	703fe6da-5e46-4cd1-ba6f-b81073f8a612
e2c3c580-615e-4e01-b689-7f5599375b8d	cirros-kernel2	ACTIVE	
e5c06a1b-c909-435b-93a7-95ec80d4bd5f	cirros2	ACTIVE	
ecdabd0f-5e6d-4d86-8dbe-ddf997398303	cirros-ram	ACTIVE	
ubuntu@ubuntu:~/Downloads\$			

3.4.5 Deleting Images

Images can be deleted using the same three methods used in viewing them; Horizon Dashboard, glance client and nova client.

Figure 63: Deleting Images Using Dashboard

<input type="checkbox"/>	Image Name	Type	Status	Public	Container Format	Actions
<input type="checkbox"/>	cirros qcow2	Image	Active	Yes	BARE	<button>Launch</button> <button>Edit</button> <button>Delete Image</button>
<input type="checkbox"/>	cirros2	Image	Active	Yes	AMI	<button>Delete Image</button>
<input type="checkbox"/>	cirros	Image	Active	Yes	AMI	<button>Launch</button>
<input type="checkbox"/>	precise-amd64	Image	Active	Yes	OVF	<button>Launch</button>
<input type="checkbox"/>	cirros image	Image	Active	Yes	BARE	<button>Launch</button>

Displaying 5 items

Figure 64: Delete an image using Glance CLI

```
ubuntu@ubuntu:~/Downloads$ glance help delete
glance delete [options] <ID>

Deletes an image from Glance
ubuntu@ubuntu:~/Downloads$ glance index name='cirros qcow2'
ID           Name          Disk Format   Container Format  Size
-----+-----+-----+-----+-----+
3e6b33d9-a4dc-4cbd-8763-6ea88789332c| cirros qcow2      qcow2        bare            9761280
ubuntu@ubuntu:~/Downloads$ 
ubuntu@ubuntu:~/Downloads$ 
ubuntu@ubuntu:~/Downloads$ 
ubuntu@ubuntu:~/Downloads$ glance delete 3e6b33d9-a4dc-4cbd-8763-6ea88789332c
Delete image 3e6b33d9-a4dc-4cbd-8763-6ea88789332c? [y/N] y
Deleted image 3e6b33d9-a4dc-4cbd-8763-6ea88789332c
ubuntu@ubuntu:~/Downloads$ 
ubuntu@ubuntu:~/Downloads$ 
ubuntu@ubuntu:~/Downloads$ glance index name='cirros qcow2'
ubuntu@ubuntu:~/Downloads$ 
```

Figure 65: Delete an image using nova CLI

```
ubuntu@ubuntu:~/Downloads$ nova image-list
+-----+-----+-----+-----+
| ID      | Name     | Status | Server |
+-----+-----+-----+-----+
| 119e9b5a-aaba-4051-8d31-6132642420a7 | cirros-kernel | ACTIVE | 
| 358cb1fc-ad6c-4140-a2a7-00b3b698c45e | cirros-ram2  | ACTIVE | 
| 46881540-019c-455f-8c4f-b74dd89f8b28 | SNApofVM3    | ACTIVE | f7c5cae2-ae43-4714-96b7-87970ee0f7da
| 651d048a-b1a7-4af7-bbe4-a11ae7b8f420 | precise-amd64 | ACTIVE | 
| 75066f2f-7199-4519-932a-28ca65d8024c | cirros image  | ACTIVE | 
| a26aa9e2-92ab-4c4b-aa9c-301bbb9f3764 | cirros       | ACTIVE | 
| dde74670-6386-4dbb-92ef-f1ddef20ae01 | VM1_snapshot | ACTIVE | 703fe6da-5e46-4cd1-ba6f-b81073f8a612
| e2c3c580-615e-4e01-b689-7f5599375b8d | cirros-kernel2 | ACTIVE | 
| e5c06a1b-c909-435b-93a7-95ec80d4bd5f | cirros2      | ACTIVE | 
| ecdabd0f-5e6d-4d86-8dbe-dfd997398303 | cirros-ram   | ACTIVE | 
+-----+-----+-----+-----+
ubuntu@ubuntu:~/Downloads$ nova help image-delete
usage: nova image-delete <image>

Delete an image. It should go without saying, but you can only delete images
you created.

Positional arguments:
  <image>  Name or ID of image.
ubuntu@ubuntu:~/Downloads$ nova image-delete e5c06a1b-c909-435b-93a7-95ec80d4bd5f
ubuntu@ubuntu:~/Downloads$ 
ubuntu@ubuntu:~/Downloads$ 
```

4.0 Appendix

4.1 Glossary

CIMC	- Cisco Integrated Management Controller
RDP	- Remote Desktop Protocol
KVM(Console)	-Keyboard Video and Mouse
KVM(Hypervisor)	- Open Source KVM hypervisor
LTS(Ubuntu)	-Long Term Support
FCOE	-Fiber-channel Over Ethernet
LOM	-LAN on Motherboard
UCS B Series	- UCS Chassis Blade Servers
UCS C Series	- UCS Rack Based C Series Servers

4.2 Caveats

Caveats and Technical notes if needed

4.3 References

- [1] Cisco-OpenStack Installation Guide
- [2] <http://glance.openstack.org/glanceapi.html>
- [3] OpenStack Essex Administration Guides (<http://docs.openstack.org/>)
- [4] OpenStack Essex API Guides (<http://docs.openstack.org/api/>)
- [5] OpenStack Essex Developer Documentation (<http://docs.openstack.org/developer/>)