Cisco Edition OpenStack

Admin Guide

Contents

Figures

Tables

No table of figures entries found.

1.0 Introduction

1.1 Preface

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 Compute provides a tool to orchestrate a cloud, including running instances, managing networks, 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. 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 web-based API.[2]

All components in OpenStack offer REST APIs that may be invoqued by user-firendly web-based user interfaces, Command Line Interfaces (CLIs) or customer-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) in the same system (sharing resources). Earlier versions of OpenStack used the term "project" instead of "tenant". Because of this legacy terminology, these terms are used interchangeably. The goal of this document is to provide a simple guide to help cloud computing administrators get familiar with Openstack administrative operations.

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

1.2 Audience

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

[1] Description from http://OpenStack.org/, 06/26/2012.

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

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 are 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.

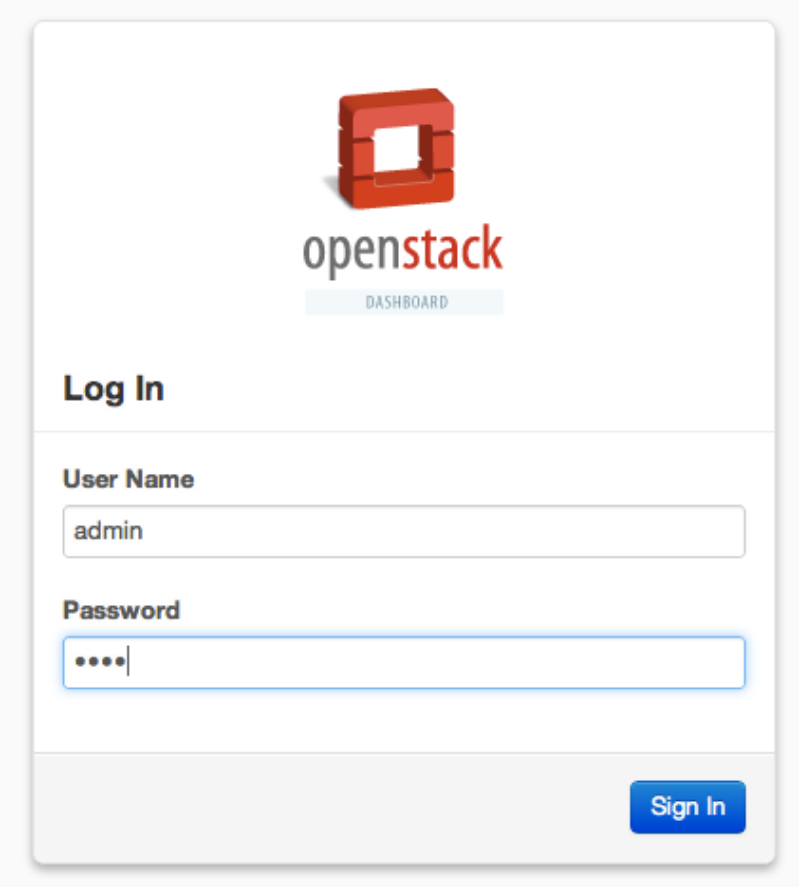
This includes OpenStack Administrative functions, such as Creating projects within OpenStack, creating new users, managing project quotas, etc. These items will be covered in a separate Cisco OpenStack Admin Guide document.

2.0 Web-Based OpenStack Admin Operations

Openstack offers a user-friendly UI known as the Dashboard (a reference UI implementation provided by the OpenStack Horizon project). The Dashboard offers two different views: the Admin System Panel and the Project Panel. The Admin System Panel is exclusively for administration activities such as creating projects or users and registering images. The Project Panel is specific to project administration (tenants); it helps project users manage compute servers and create new instances and volumes. This guide focuses on the Admin System Panel.

The OpenStack Dashboard is a web-based interface compatible with Firefox 13 (recommended), IE 7 and Crome 19. To acess the Dashboard simply open your browser and type the IP address of the Horizon host server. If DNS is enabled in your network the host name may be used instead. Figure 1 illustrates the OpenStack Dashboard Log In screen.

Figure 1. Openstack Dashboard Log In User Interface



2.1 Customizing the Dashboard

The OpenStack dashboard can be customized with changing the Site Title and Logo on dashboard User Interface.

Changing the Site Title

The site's title for the dashboard (i.e. Openstack) can be overwritten. To do this, add the attribute "SITE_BRANDING" to the local_settings.py file with the desired name. For example: SITE_BRANDING=<desired_name>
The local_settings.py file can be found in the Horizon directory path:
#/etc/openstack-dashboard/local_settings.py

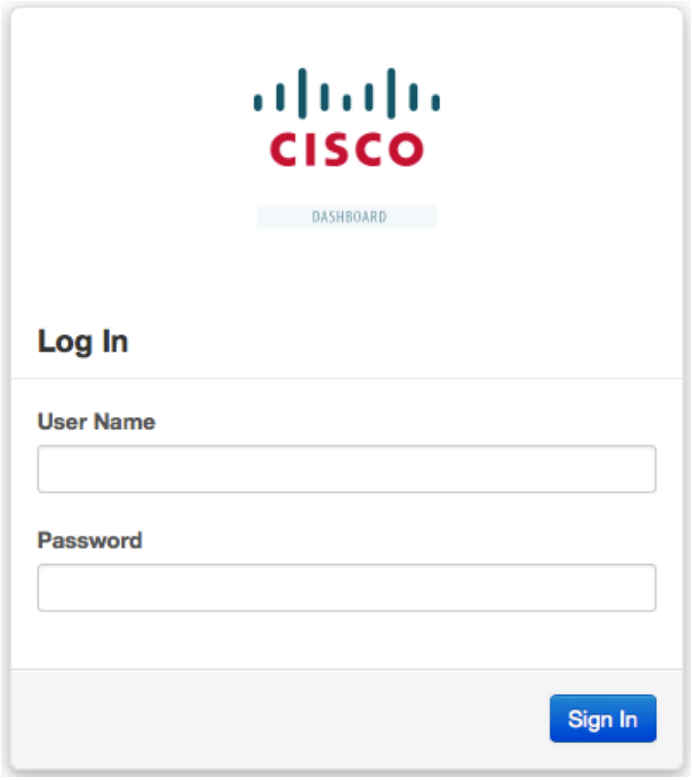
Changing the Logo

To override the OpenStack Logo image, replace the image in the following file:
/usr/share/openstack-dashboard/openstack_dashboard/static/dashboard/img/logo.png

The dimensions should be width: 108px and height: 121px. To activate the new logo, the package openstack-dashboard-ubuntu-theme should be unistalled by running:

apt-get remove openstack-dashboard-ubuntu-theme
The following screenshot shows the OpenStack login screen with a customized logo:

Figure 2. Customized Openstack Log In UI



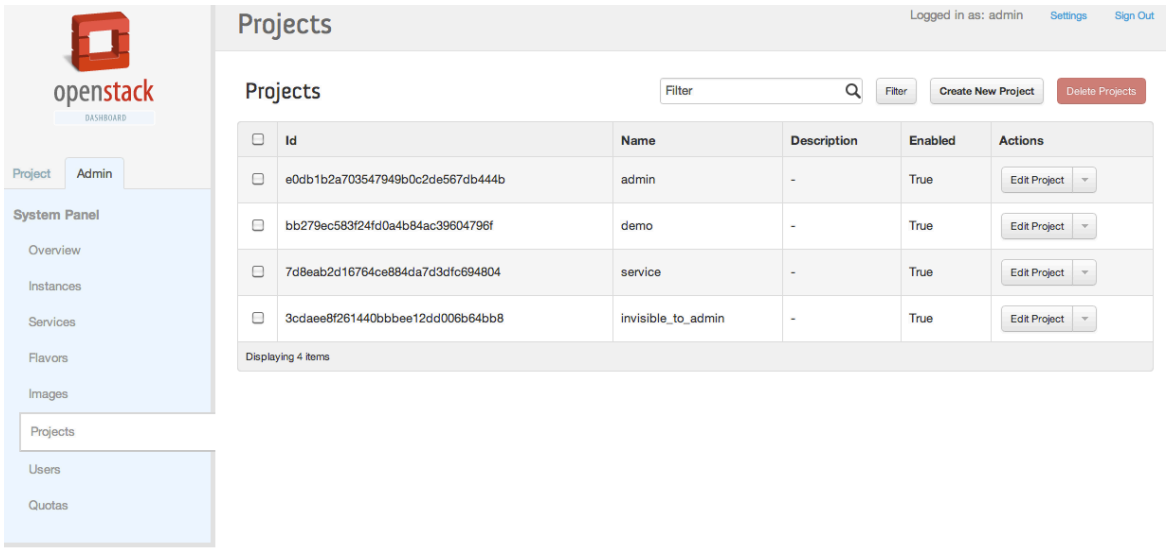
2.2 Managing Projects (Tenants)

OpenStack uses role-based access control to provide different access levels. By default, there are two roles: the admin role and the project-users role. Administrators need to create projects and project users.

Creating Projects

In order to create a new project, first select the section named "Project" on the system panel. Then select "Create New Project" on the right top corner, as shown in the following figure:

Figure 3. List of Projects Information



After filling in the required fields, the project is ready to be created by clicking the "Create Project" button as shown below.

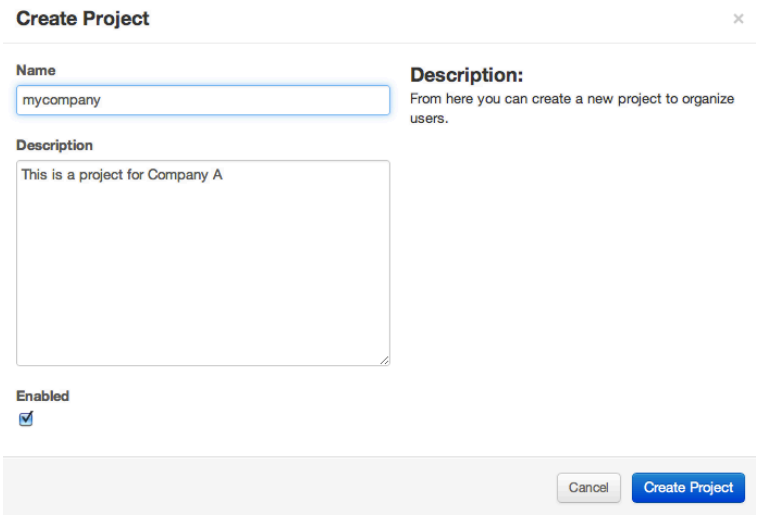


Figure 4. Required Project Information

Deleting Projects

On the same "Projects" section of the Admin Panel that we used for creating a project, select the project that needs to be deleted and click the "Delete Project" button located at the top right corner of the projects view.

2.3 Managing Users

The next step after creating the projects is to create users and assign them to their corresponding projects. There are two methods to onboard users. The first method is single user creation done manually on the Dashboard. The second method is the bulk user creation. We will cover both methods in this document. Let's get started with the first methodology.

Creating Users Accounts Manually

Select the "Users" section in the System Panel and click on "Create User" to create any number of users. The next figure shows a project with a number of users created.

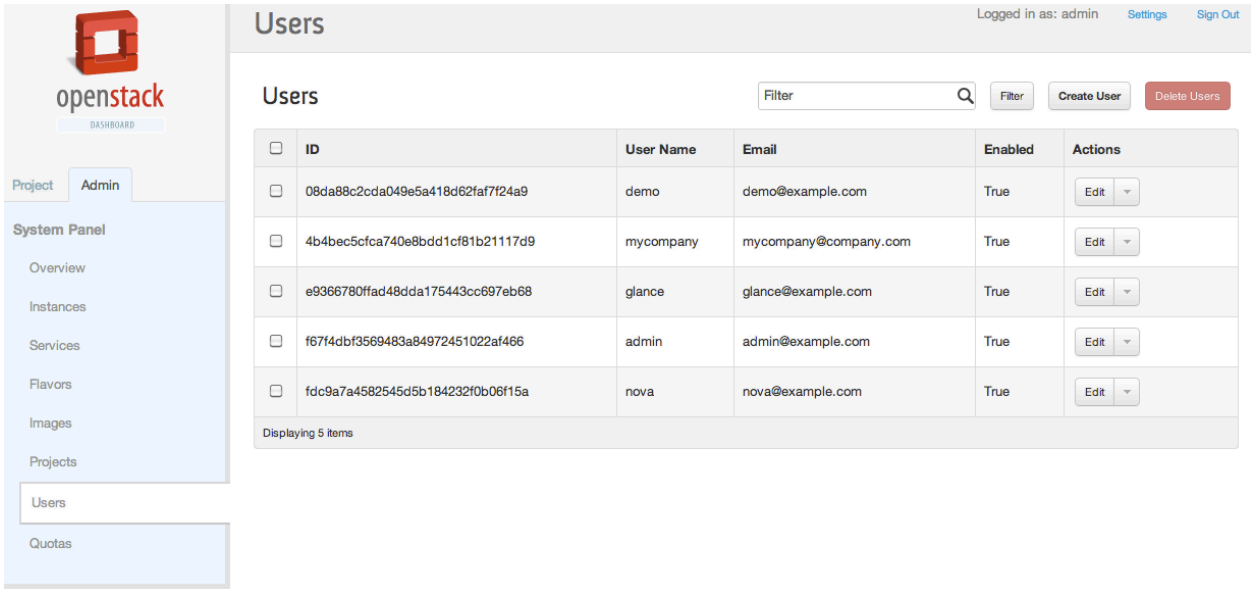


Figure 5. List of Users Created

When creating a user, one of the most important steps is to associate the user to a project. As the next screenshot shows, the project is defined in the "Primary Project" field. Once the user has been created, the only two actions available are "Delete" and "Disable". The "Delete" option completely deletes the user from the system. "Disable" denies any operation for that user but leaves the account in the system (Figure 7).

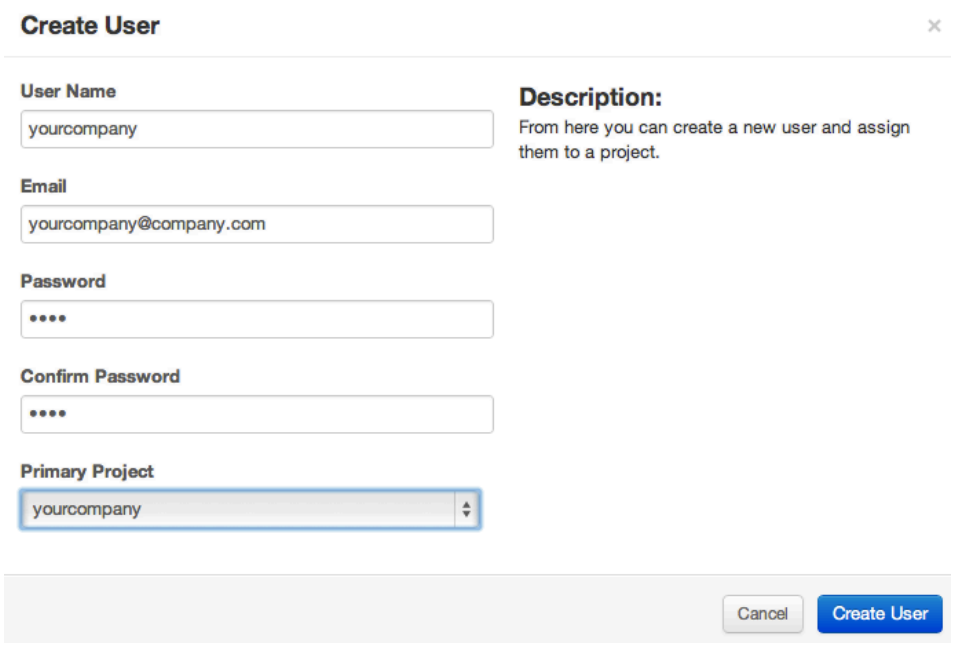


Figure 6. Required User Information

2.4 Instances Administration

OpenStack Operators can view the status of all instances (VMs) running in the Openstack deployment and, if needed, even terminate any of them regardless their corresponding ownership (projects). As the following figure shows, the "Instances" section in the System Panel allows viewing the tenant (project) that created the instance, the compute host where the instance is running, as well as the instance name, IP Address, Size, Status and State.

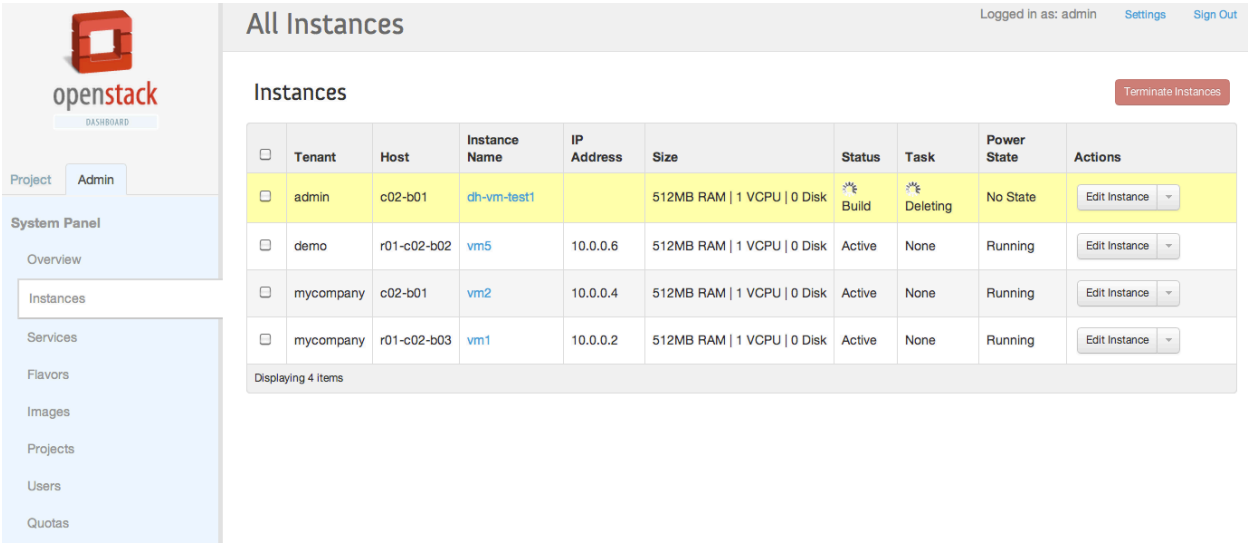


Figure 7. List of all running instances

There are a few actions available for the Cloud Operator to control instances. Selecting the "Edit Instances" button of the corresponding instance will display additional actions that can be performed on the Instance. Actions include opening a VNC console to the VM, reading the instances log, creating a snapshot of the instance, and pausing, suspending, rebooting or terminating instances (Figure 10).

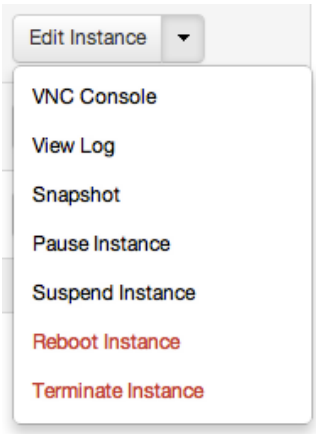


Figure 8. Available operations per instance

2.5 Services

OpenStack is a set of different software components that work together based on a messaging infrastructure. These components, known as services, may run in the same host or may be distributed. The "Services" section of the Admin System Panel provides a consolidated view of each available service regardless of where the service is located. It allows operators to view services status and also provides search capabilities. Figure 11 shows an example Services screen.

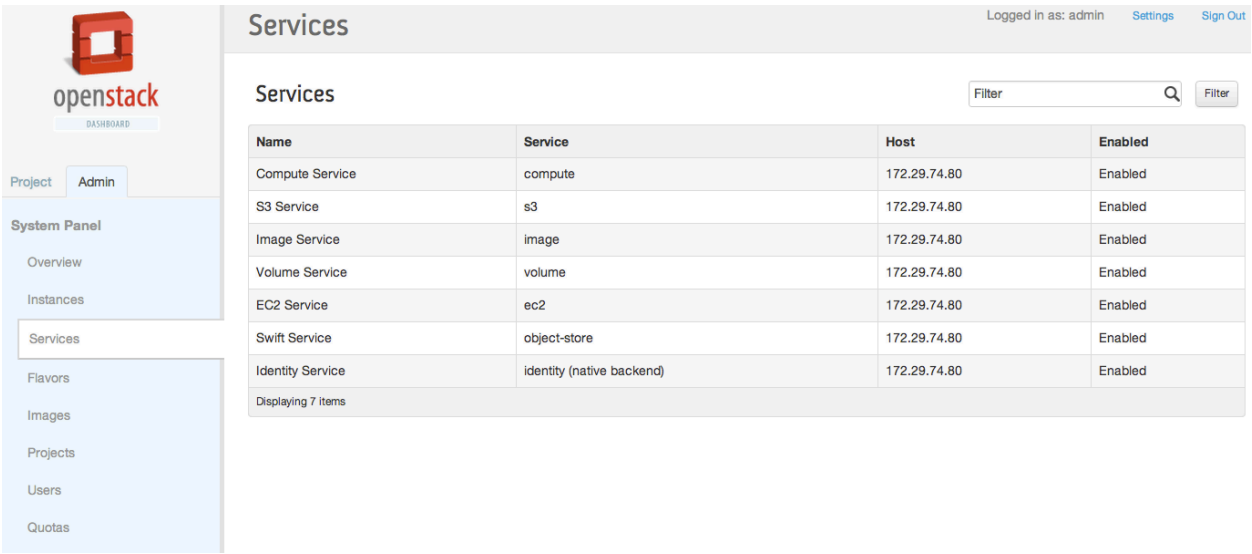


Figure 9. List of Services and Status

2.6 System Quotas

Quotas define the number of resources available for each project. For example, operators may define quotas to limit the size and capacity of projects. As shown below, the "Quotas" section in the System Panel displays the default quota values. Note that the "Quotas" section is read-only (Figure 12).

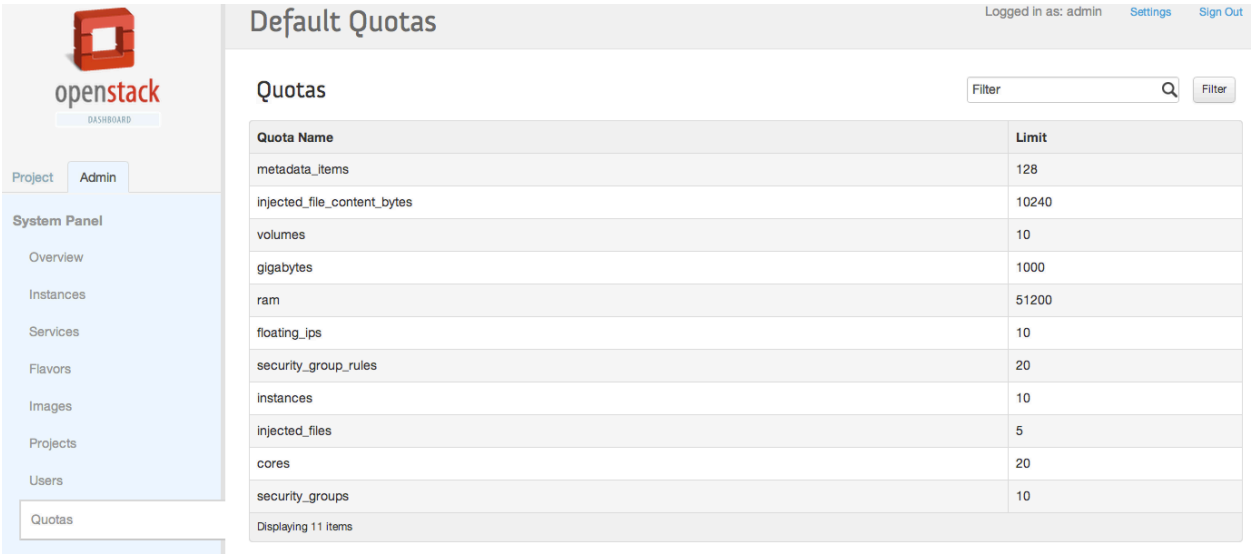


Figure 10. Default OpenStack Essex Quotas

Modifying System Quotas

The system quotas are customizable per project. Operators may change these quotas from the "Projects" section by selecting "Edit Project" followed by the "Modify Quotas" option. The next screenshot shows these steps.

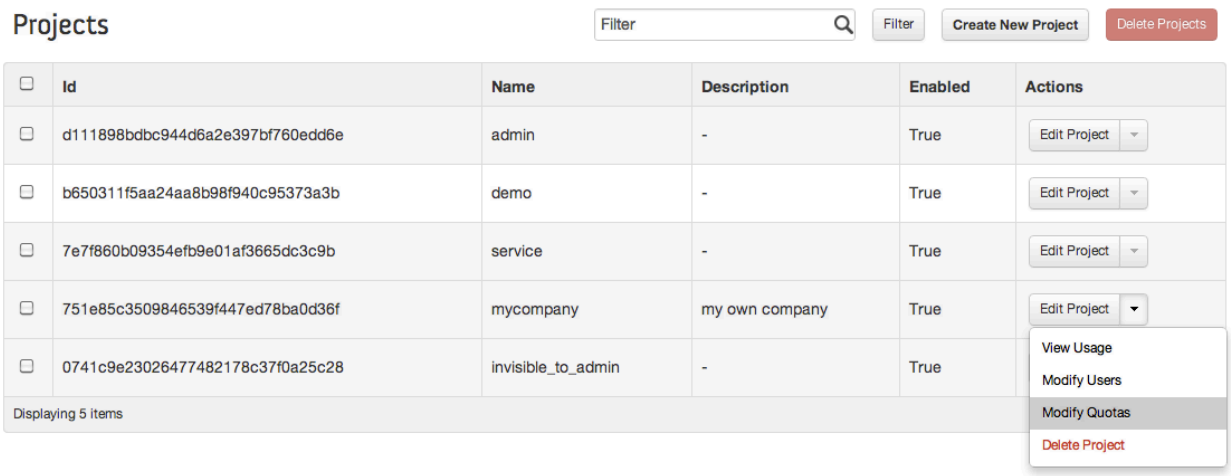


Figure 11. Available operations per project

The UI will open a new screen where operators can modify the default quotas. The default quotas do not need to be modified unless cloud operators specifically want to assign different values. Common quota modifications include increasing/decreasing the number of floating IPs for a Tenant or modifying the maximum size of the volumes that can be created by users. The various quotas that can be modified are shown in the figure below.

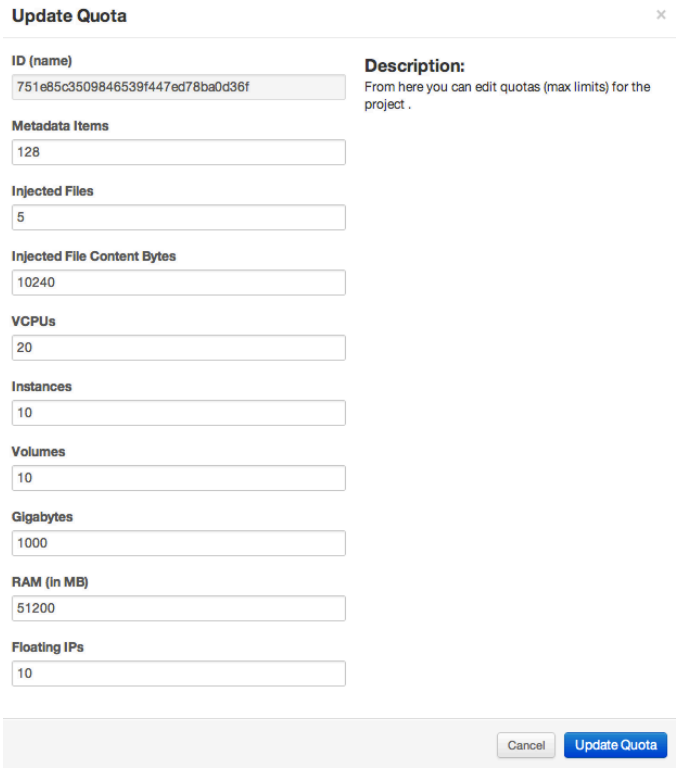


Figure 12. Quotas customization

2.7 Images

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. It 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]. The Dashboard provides an overview of previously uploaded images and lets users delete any of the images uploaded by them. Images owned by the Administrator cannot be deleted or modified by users.

Overview Images

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.

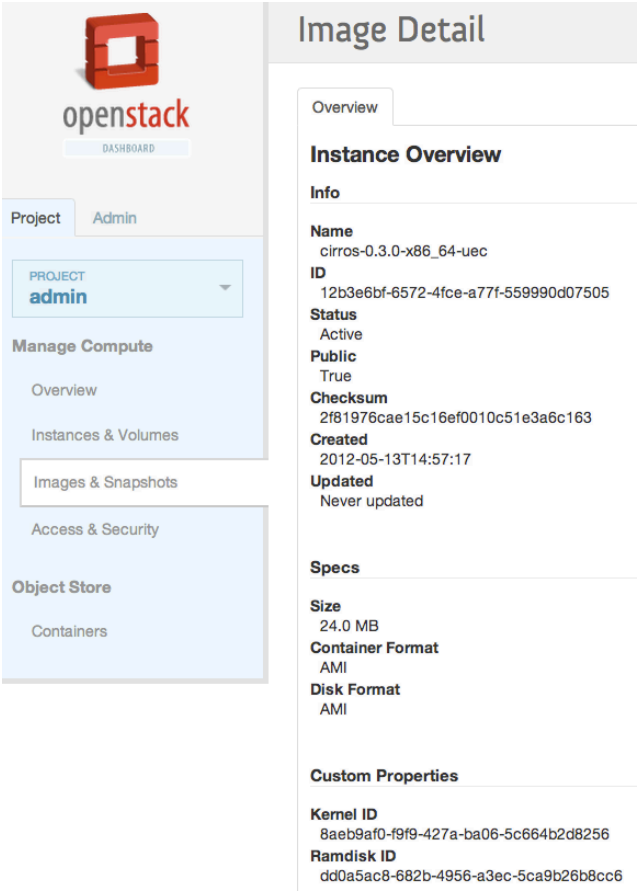


Figure 13. Glance Images details

Deleting Images

In the "Images" section, select the name of the image that should be deleted and click the red "Delete Images" button in the top right corner (Figure 17).

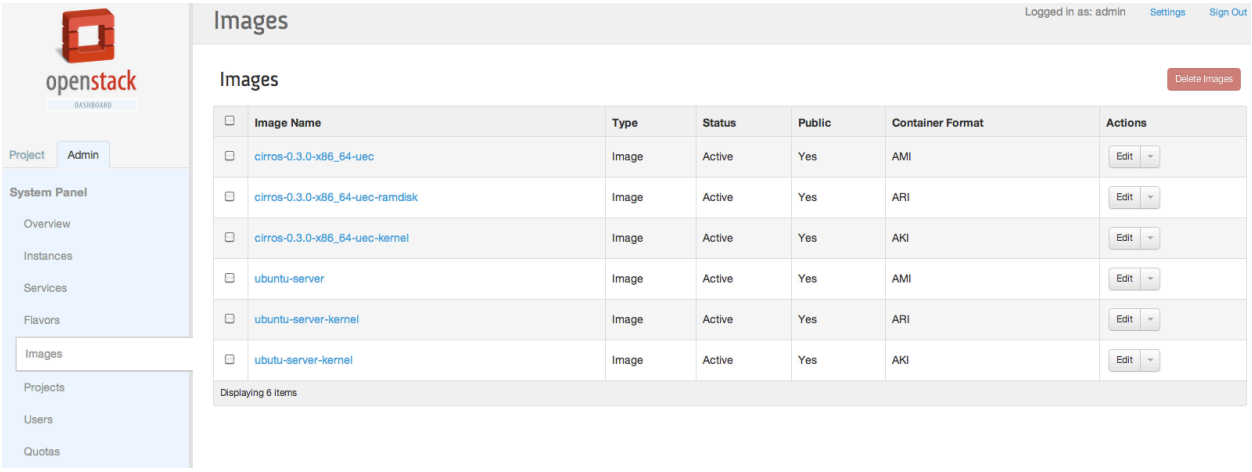


Figure 14. List of Glance Images Available

3.0 CLI-Based OpenStack 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 admin/user system. They are required mostly for performing certain operations that are not included in the Dashboard.
Installing any OpenStack Service client is done by the installing command as follows:

Installating Nova Client:

sudo apt-get install python-novaclient

Installating Glance Client:

sudo apt-get install glance-client

Installating Keystone Client:

sudo apt-get install python-keystoneclient

Installating Swift Client:

sudo apt-get install python-swift

Clients include a help section that lists the available sub-commands and optional arguments. The Nova client will be:

 # nova help

Command-line interface to the OpenStack Nova API.

Positional arguments:
<subcommand>
absolute-limits Print a list of absolute limits for a user
actions Retrieve server actions.
add-fixed-ip Add new IP address to network.
add-floating-ip Add a floating IP address to a server.
aggregate-add-host Add the host to the specified aggregate.
aggregate-create Create a new aggregate with the specified details.
aggregate-delete Delete the aggregate by its id.
aggregate-details Show details of the specified aggregate.
aggregate-list Print a list of all aggregates.
aggregate-remove-host
Remove the specified host from the specfied aggregate.
aggregate-set-metadata
Update the metadata associated with the aggregate.
aggregate-update Update the aggregate's name and optionally
availability zone.
boot Boot a new server.
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 a server.
describe-resource Show details about a resource
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
flavor-create   Create a new flavor
flavor-delete   Delete a specific flavor
flavor-list     Print a list of available 'flavors' (sizes of
                servers).
floating-ip-create  Allocate a floating IP for the current tenant.
floating-ip-delete  De-allocate a floating IP.
floating-ip-list    List floating ips for this tenant.
floating-ip-pool-list
                List all floating ip pools.
get-vnc-console     Get a vnc console to a server.
host-action     Perform a power action on a host.
host-update     Update host settings.
image-create    Create a new image by taking a snapshot of a running
                server.
image-delete    Delete an image.
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.
keypair-add     Create a new key pair for use with instances
keypair-delete  Delete keypair by its id
keypair-list    Print a list of keypairs for a user
list            List active servers.
live-migration  Migrates a running instance to a new machine.
lock            Lock a server.
meta            Set or Delete metadata on a server.
migrate         Migrate a server.
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-show      List the quotas for a tenant.
quota-update    Update the quotas for a tenant.
rate-limits     Print a list of rate limits for a user
reboot          Reboot a server.
rebuild         Shutdown, re-image, and re-boot a server.
remove-fixed-ip     Remove an IP address from a server.
remove-floating-ip  Remove a floating IP address from a server.
rename          Rename a server.
rescue          Rescue a server.
resize          Resize a server.
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.
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.
show Show details about the given server.
ssh SSH into a server.
suspend Suspend a server.
unlock Unlock a server.
unpause Unpause a server.
unrescue Unrescue a server.
usage-list List usage data for all tenants
volume-attach Attach a volume to a server.
volume-create Add a new volume.
volume-delete Remove a volume.
volume-detach Detach a volume from a server.
volume-list List all the volumes.
volume-show Show details about a volume.
volume-snapshot-create
Add a new snapshot.
volume-snapshot-delete
Remove a snapshot.
volume-snapshot-list
List all the snapshots.
volume-snapshot-show
Show details about a snapshot.
volume-type-create Create a new volume type.
volume-type-delete Delete a specific flavor
volume-type-list Print a list of available 'volume types'.
x509-create-cert Create x509 cert for a user in tenant
x509-get-root-cert Fetches the x509 root cert.
bash-completion Prints all of the commands and options to stdout so
that the
help Display help about this program.

```
Optional arguments:
 --debug            Print debugging output
 --os_username OS_USERNAME
                    Defaults to env[OS_USERNAME].
 --os_password OS_PASSWORD
                    Defaults to env[OS_PASSWORD].
 --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].
 --service_type SERVICE_TYPE
                    Defaults to compute for most actions
 --service_name SERVICE_NAME
                    Defaults to env[NOVA_SERVICE_NAME]
 --endpoint_type ENDPOINT_TYPE
                    Defaults to env[NOVA_ENDPOINT_TYPE] or publicURL.
 --os_compute_api_version OS_COMPUTE_API_VERSION
                     Accepts 1.1, defaults to env[OS_COMPUTE_API_VERSION].
 --username USERNAME   Deprecated
 --region_name REGION_NAME
                    Deprecated
 --apikey APIKEY, --password APIKEY
                    Deprecated
 --projectid PROJECTID, --tenant_name PROJECTID
                    Deprecated
 --url URL, --auth_url URL
                    Deprecated

See "nova help COMMAND" for help on a specific command.
```

3.1 User Credentials

Credentials for the admin account are set-up during the OpenStack installation process. Those credentials will allow administrator to log in to OpenStack dashboard and perform most of the administrative operations such as creating projects or users, uploading images, and controlling quotas. There are certain operations that are not yet available in the OpenStack dashboard thus, the Administrator may need to use some of the CLI-based OpenStack clients from the terminal console.

Prior to use any of those clients, administrator 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.[3] Environmental variables can be accessed from the Dashboard. On the top right corner, select the "Settings" link. Then, select the section named "OpenStack Credentials" on the left panel as shown in Figure 3 and click over the Download RC File bottom.

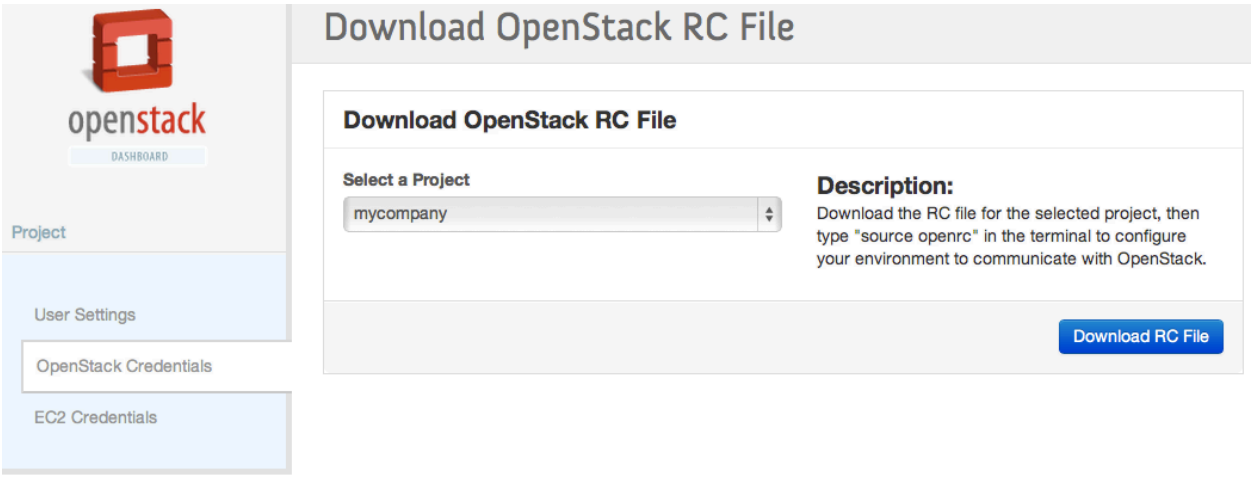


Figure 15. Download section for OpenStack credentials

Open a console terminal in your 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. 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.

Sourcing User Credentials OpenStack File:

source /PATH_TO_FILE/openrc

[3] http://wiki.openstack.org/UnifiedCLI/Authentication

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 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.4 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 we referred to earlier, but in the "EC2 Credentials" section (Figure 4).

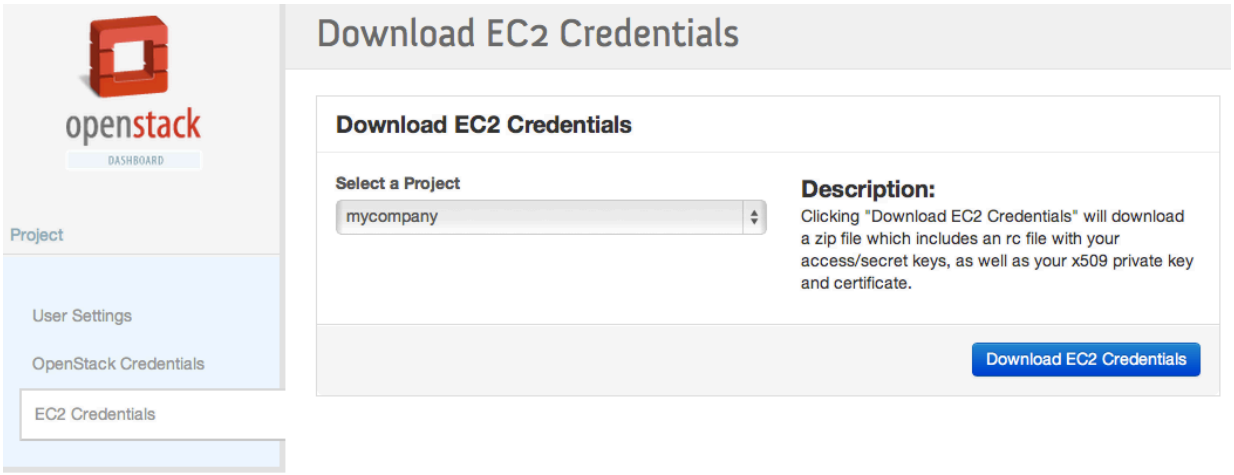


Figure 16. Download section for OpenStack credentials

Sourcing User Credentials EC2 File:

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

[4] http://open.eucalyptus.com/wiki/Euca2oolsUsing

Creating Users Accounts Automatically

The multi-user creation method relies on a python code and a csv file. This code basically creates OpenStack users and tenants based on a CSV file. If run on the same host as Keystone, the admin token will automatically be grabbed from /etc/keystone/keystone.conf (if you have access to this

file). If run on a different host, you can pass credentials for Keystone by setting and exporting the SERVICE_TOKEN, SERVICE_ENDPOINT and all user environments included in the openrc file as mentioned on section 4.1.

This code will read in the csv file and import users using the Keystone client. In order to be compliant always check the following reference [6] for updates on both files. The admin must have keystone-client installed on the system where the code will be executed.

Installating Keystone Client:

sudo apt-get install python-keystoneclient

Getting and Running the code:

git clone https://github.com/CiscoSystems/cisco-openstack-docs

cd keystone_user_onboard

python openstack-load-users-from-csv.py csv_file

It will automatically create tenants and users (same name) as entries in the CSV file.

General Keystone tenant-create options are:

keystone tenant-create --name <tenant-name>

[--description <tenant-description>]

[--enabled <true|false>]

Create new tenant

Optional 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 true)

The script expects an input file in csv format with the following schema:

FirstName,LastName,user-id,email-address

Here is a sample of the CSV format

John,Doe,jdoe,jdoe@company.com

Jane,Roe,jroe,jroe@company.com

...

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.

Installating Glance Client:

sudo apt-get install glance-client

Sourcing User Credentials File:

source/PATH_TO_FILE/openrc

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 qcow 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 qcow2 images.

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

Adding AMI (Amazon Machine Image) Images

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 cloud controller execute the following commands:

ADMIN_USER=admin

ADMIN_TENANT=admin

TOKEN=$(keystone --os_tenant_name $ADMIN_TENANT --os_username $ADMIN_USER --os_password $ADMIN_PASSWORD --os_auth_url http://$KEYSTONE_IP:5000/v2.0 token-get | grep ' id ' | get_field 2)

KERNEL_ID=$(glance --url http://$GLANCE_HOSTPORT add name="$IMAGE_NAME-kernel" is_public=true container_format=aki disk_format=aki < "cirros-0.3.0-x86_64-vmlinuz" | grep ' id ' | get_field 2)

RAMDISK_ID=$(glance –url=http://$GLANCE_HOSTPORT image add name="$IMAGE_NAME-ramdisk" is_public=true container_format=ari disk_format=ari < "cirros-0.3.0-x86_64-initrd" | grep ' id ' | get_field 2)

glance --os-auth-token=$TOKEN –-url=http://$GLANCE_HOSTPORT image 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"

Adding QCOW (Qemu Copy On Write) Images

These images do not require any kernel or ramdisk images, you can simply import them as:
ADMIN_USER=admin
ADMIN_TENANT=admin
TOKEN=$(keystone --os_tenant_name $ADMIN_TENANT --os_username $ADMIN_USER --os_password $ADMIN_PASSWORD --os_auth_url http://$KEYSTONE_IP:5000/v2.0 token-get | grep ' id ' | get_field 2)

glance --os-auth-token=$TOKEN –-url=http://$GLANCE_HOSTPORT image add name="cirros" is_public=true container_format=bare disk_format qcow2 < centos60_x86_64.qcow2

Listing Images

The list of all the available images can be retrieved as explained in section 2.6.1 “Overview Images” or using Glance APIs.
The following command will display the available images.

glance index

ID	Name	Disk Format	Container Format	Size
c2d9ed77-4fa6-409d-82c0-fb7fbe89f12e	cirros-0.3.0-x86_64-uec-ramdis	ari	ari	2254249
9057836a-ca1f-40bf-aa14-0751aa688c4d	cirros-0.3.0-x86_64-uec	ami	ami	25165824
0619452a-41bc-48de-b286-864463d82839	cirros-0.3.0-x86_64-uec-kernel	aki	aki	4731440
0401671e-e224-4e88-b9eb-e8bfeb993a5e	ubuntu-server	ami	ami	25165824
cfcd68e3-5665-4865-b1ee-f5aa5243a5a0	ubutu-server-kernel	aki	aki	4404752
4f9b3066-6fa8-40f4-8346-293e6ca27b08	ubuntu-server-kernel	ari	ari	5882349

Figure 17. Listing of all available images

Using the nova client is also possible:

nova image-list

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

```
|                  ID                  |               Name               | Status | Server |
+--------------------------------------+----------------------------------+--------+--------+
| 0401671e-e224-4e88-b9eb-e8bfeb993a5e | ubuntu-server                    | ACTIVE |        |
| 0619452a-41bc-48de-b286-864463d82839 | cirros-0.3.0-x86_64-uec-kernel   | ACTIVE |        |
| 4f9b3066-6fa8-40f4-8346-293e6ca27b08 | ubuntu-server-kernel             | ACTIVE |        |
| 9057836a-ca1f-40bf-aa14-0751aa688c4d | cirros-0.3.0-x86_64-uec          | ACTIVE |        |
| c2d9ed77-4fa6-409d-82c0-fb7fbe89f12e | cirros-0.3.0-x86_64-uec-ramdisk  | ACTIVE |        |
| cfcd68e3-5665-4865-b1ee-f5aa5243a5a0 | ubutu-server-kernel              | ACTIVE |        |
+--------------------------------------+----------------------------------+--------+--------+
```

3.2 Floating IPs

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. Administrator should create "pools" of floating IPs based on the network values where the Cloud OpenSatck is being connected. OpenStack Compute uses Network Address Translation (NAT) to assign floating IPs to virtual instances.

Creating a Pool of Floating IP Addresses

Nova maintains a list of floating IP addresses that are available for assigning to instances. Use the "nova-manage floating create" command to add entries to this list, as root.5

nova-manage floating create --ip_range=192.168.100.1/24

Deleting a Pool of Floating IP Addresses

nova-manage floating delete --ip_range=192.168.100.1/24

Associating a Floating IP to an Instance

Adding a floating IP to an instance is a two steps process:

Allocate a floating IP address from the list of available addresses.
Add an allocated floating IP address to a running instance.

Example:

nova floating-ip-create

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

[5] Adapted from http://docs.openstack.org/trunk/openstack-compute/admin/content/, 06/26/2012.

```
|        Ip       | Instance Id | Fixed Ip | Pool |
+-----------------+-------------+----------+------+
|   192.168.100.2 | None        | None     |      |
+-----------------+-------------+----------+------+
```

nova add-floating-ip <image_id> 192.168.100.2

Deleting a Floating IP

If the instance no longer needs a public address, remove the floating IP address from the instance and de-allocate the address:

nova remove-floating-ip <image_id> 192.168.100.2

nova floating-ip-delete 192.168.100.2

Automatically adding floating IPs

The nova-network service can be configured to automatically allocate and assign a floating IP address to virtual instances when they are launched. Add the following line to nova.conf and restart the nova-network service

auto_assign_floating_ip=True

4.0 Compute Node On-Boarding

1. Login to the Management Node and edit the cobbler-node.pp file.

```
root@oscomp-ch2-a01:~#   vi   /etc/puppet/manifests/cobbler-
node.pp
```

Page down to the bottom of the file and create a new node definition between "}" and "#Repeat as necessary". Here is an example node definition for Compute Node 11 (replace the parameters such as mac, ip, power_user and password accordingly):

```
cobbler::node { "oscomp-ch2-a11":
 mac => "F0:F7:55:AB:3D:B8",
 profile => "precise-x86_64-auto",
```

```
 ip => "69.252.239.149",
 domain => "sys.domain.net",
 preseed   =>   "/etc/cobbler/preseeds/ucs-preseed-compute-
lsi",
 power_address => "69.252.239.239",
 power_type => "ipmitool",
 power_user => "ucs_admin_user",
 power_password => "ucs_admin_password",
 }
```

All Compute Nodes have been previously defined in the Puppet site.pp file (/etc/puppet/manifests/site.pp). Therefore, no additional node definitions are required on site.pp. The site.pp Compute Node definitions below have been included for reference purposes only. The regular expression (0[2-9]|1[0-9]|20) is matching hosts oscomp-ch2-a0 and oscomp-ch2-a02 - oscomp-ch2-a20.

Build your compute nodes

```
node /oscomp-ch2-a(0[2-9]|1[0-9]|20)/ inherits flat_dhcp {
```

Needed to address a short term failure in nova-volume management - bug has been filed

```
  class { 'nova::compute::file_hack': }

  class { 'openstack::compute':
    private_interface  => $private_interface,
    internal_address   => $ipaddress_eth0,
    glance_api_servers                                    =>
"${controller_node_internal}:9292",
    rabbit_host        => $controller_node_internal,
    rabbit_password    => $rabbit_password,
    rabbit_user        => $rabbit_user,
    sql_connection     => $sql_connection,
    vncproxy_host      => $controller_node_internal,
    verbose            => $verbose,
    manage_volumes     => true,
 }
}
```

2. Run the puppet apply command on the site.pp file after modifying cobbler-node.pp.

Use the –v (Verbose) command option to view the actions puppet performs. You should observe the node (oscomp-ch2-a11.sys.domain.net in this example) being added to Cobbler. You should also observe the cobbler-sync ***TASK COMPLETE*** and a finished catalog run.

```
root@oscomp-ch2-a01:~#         puppt         apply         -v
/etc/puppet/manifests/site.pp
…
notice:                            /Stage[main]//Node[cobbler-
node]/Cobbler::Node[oscomp-ch2-a11]/Host[oscomp-ch2-
a11.sys.domain.net]/ensure: created
info:                     FileBucket                    adding
{md5}a18f083f0e8e4a1e98cacb3e17b0ecff
….
notice:                            /Stage[main]//Node[cobbler-
node]/Cobbler::Node[oscomp-ch2-a11]/Exec[cobbler-add-node-
oscomp-ch2-a11]/returns: executed successfully
info:                              /Stage[main]//Node[cobbler-
node]/Cobbler::Node[oscomp-ch2-a11]/Exec[cobbler-add-node-
oscomp-ch2-a11]: Scheduling refresh of Exec[cobbler-sync]
……..
notice:     /Stage[main]/Cobbler/Exec[cobbler-sync]/returns:
*** TASK COMPLETE ***
notice:  /Stage[main]/Cobbler/Exec[cobbler-sync]:  Triggered
'refresh' from 12 events
notice: Finished catalog run in 13.49 seconds
```

3. After the puppet run has completed, you should see the newly added system being managed by Cobbler:

```
root@oscomp-ch2-a01:~# cobbler system list
   oscomp-ch2-a02
   oscomp-ch2-a03
   oscomp-ch2-a04
   oscomp-ch2-a05
   oscomp-ch2-a06
   oscomp-ch2-a07
   oscomp-ch2-a08
   oscomp-ch2-a09
   oscomp-ch2-a10
   oscomp-ch2-a11
   osctrl-ch2-a01
   osctrl-ch2-a02
```

Now that the system is being managed by Cobbler, run the clean_node.sh script located at /root/os-docs/examples/.

The script uses ipmitool to power off/on the server and clears the SSH key and Puppet certificate. You should see *** TASK COMPLETE *** messages if IPMI properly reboots the server.

```
root@oscomp-ch2-a01:~# ~/clean_node.sh oscomp-ch2-a11
task started: 2012-07-06_032550_power
task started (id=Power management (off), time=Fri Jul 6
03:25:50 2012)
cobbler power configuration is:
      type   : ipmitool
      address: 69.252.239.239
      user   : cisco
      id     :
-  /usr/bin/ipmitool -H "69.252.239.239" -U "cisco" -P
"more+han8char5" power "off"
running:   ['/bin/sh',   '-c',   u'/usr/bin/ipmitool   -H
"69.252.239.239"     -U     "ucs_admin_user"     -P     "
ucs_admin_password " power "off"']
received on stdout: Chassis Power Control: Down/Off

received on stderr:
*** TASK COMPLETE ***
task started: 2012-07-06_032557_power
task started (id=Power management (on), time=Fri Jul 6
03:25:57 2012)
cobbler power configuration is:
      type   : ipmitool
      address: 69.252.239.239
      user   : cisco
      id     :
-  /usr/bin/ipmitool -H "69.252.239.239" -U "cisco" -P
"more+han8char5" power "on"
running:   ['/bin/sh',   '-c',   u'/usr/bin/ipmitool   -H
"69.252.239.239"     -U     "ucs_admin_user"     -P     "
ucs_admin_password "  power "on"']
received on stdout:
received on stderr: Error: Unable to establish LAN session
Unable to set Chassis Power Control to Up/On

running:   ['/bin/sh',   '-c',   u'/usr/bin/ipmitool   -H
"69.252.239.239"    -    U    "ucs_admin_user"    -P    "
ucs_admin_password "  power "on"']
received on stdout: Chassis Power Control: Up/On

received on stderr:
*** TASK COMPLETE ***
```

Note: You will receive the following error message if the server has not previously been managed by Puppet:

```
err: Could not call revoke: Could not find a serial number
for oscomp-ch2-a11.sys.domain.net
```

The server should then PXE boot and begin the Ubuntu Precise installation. When the Ubuntu installation is complete, the server will reboot from the Hard Disk. Optionally, you can login into the CIMC of the server and launch the Virtual KVM Console to view the progress of the Ubuntu installation.

After the server reboots, the puppet agent will register with the Puppet Master (Management Node: oscomp-ch2-a01.sys.domainq.net) and configure the newly added server. You should then be able to SSH to the server. Note: Ping is disabled from the VPN and possibly from other network locations.

You can check the status of the puppet agent run from /var/log/syslog. The puppet run should end with the following message:

```
notice: Finished catalog run in x# of seconds
```

When the puppet run is complete, you should see the openrc file installed at /root (you must be logged-in as root user). Use the openrc file to source your OpenStack environmental variables.

```
root@oscomp-ch2-a05:~# source openrc
```

After sourcing the openrc file, you should be able to run commands against the OpenStack API endpoints. Ensure that the nova-compute and nova-volume services have ☺ under State.

```
root@oscomp-ch2-a05:~# nova-manage service list
Binary           Host                                 Zone
Status     State    Updated_At
nova-volume      oscomp-ch2-a11    nova             enabled
:-)       2012-07-06 04:48:06
nova-compute      oscomp-ch2-a11    nova             enabled
:-)       2012-07-06 04:48:02
```

Troubleshooting

If your node does not appear in the cobbler system list, verify all the settings in your cobbler-node.pp node definition.

If you are unable to access your node through SSH, use the CIMC Virtual KVM to check the status of the Ubuntu installation.

If the Partition Disks screen appears showing a disk other than sdd, the server boot order is most likely incorrect. The following figure demonstrates the RAID Controller appearing as sdc instead of sdd to the Ubuntu installer:

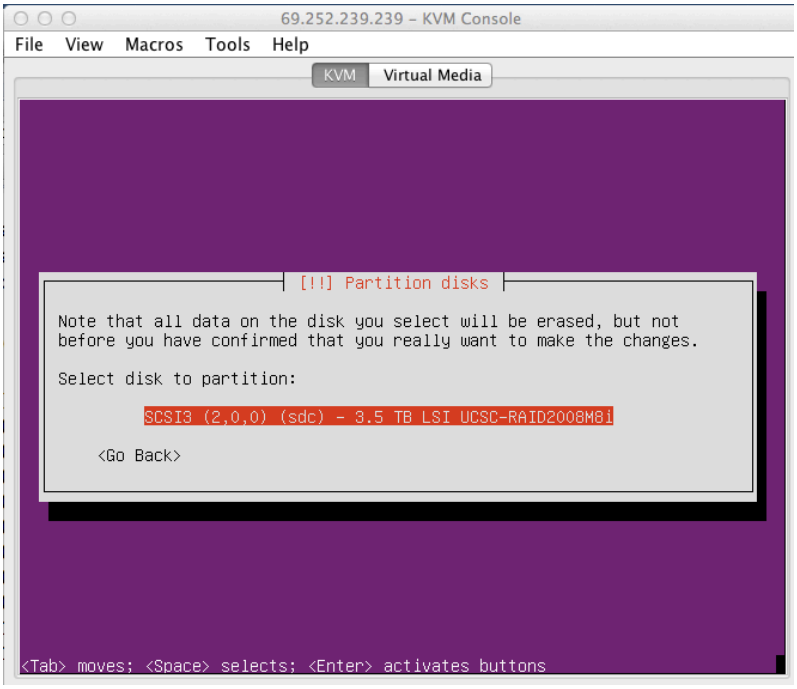


Figure 18. Ubuntu Partition Disks Screen

The boot order can be viewed from Server Tab > BIOS of the CIMC. The desired boot order should look like the following figure:

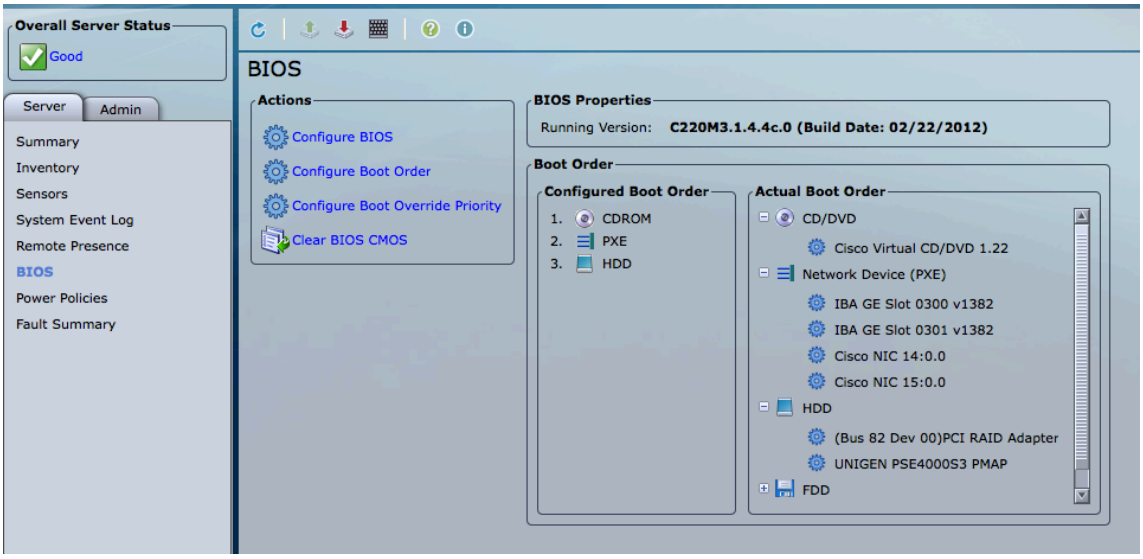


Figure 19. UCS CIMC Boot Order

Make sure you pay attention to the Actual Boot Order and not the Configured Boot Order in the BIOS.

EC2 API Comparison Matrix

This section attempts to enumerate OpenStack's compatibility with the Amazon EC2 API. This information is based on OpenStack Essex release and EC2 (API Version 2012-03-01). [6]

General API Support

General Features	OpenStack
EC2 Query API	Y
EC2 Soap API	N
OpenStack API / Rackspace API	Y
SSL Between Components	N
Horizontal Component Scalability	Y
Web-based UI	Y
Command line interface	Y

[6] For more information, refer to the community wiki: http://wiki.openstack.org/Nova/APIFeatureComparison

Amazon EC2 High Level Feature Support

EC2 feature	OpenStack
Shared AMIs	Y
Parameterized launch (user-data)	Y
Instance metadata	Y
Public AMI's	Y
Launch/Terminate Instance	Y
Reboot Instance	Y
Start/Stop Persisted Instance	Y
Retrieve Console Output	Y
Paid AMI's	N
Multiple Instance Types	Y
Instance Launch Time	Y
Elastic IP's	Y
Availability Zones	Y
Region Support	Y
User selectable kernels	Y
Elastic Block Store	Y
Booting without a ramdisk	Y
Windows Support	Y
Reserved Instances	N
Auto Scaling	N
Elastic Load Balancing	N
CloudWatch	N
Virtual Private Cloud (IPSec)	N
Shared Snapshots	N
AMI's backed by EBS	Y
Spot Instances	N
Sticky session in Elastic LB	N
Specify IP address for inst in VPC	N
Tags	N
Filters	N
Idempotent RunInstance Calls	N

Import keypair	Y
SSL termination	N

Amazon EC2 API Compatability

EC2 API method	OpenStack
AllocateAddress	Y
AssociateAddress	Y
AttachVolume	Y
AuthorizeSecurityGroupIngress	Y
BundleInstance	N
CancelBundleTask	N
CancelSpotInstanceRequests	N
ConfirmProductInstance	N
CreateImage	N
CreateKeyPair	Y
CreatePlacementGroup	N
CreateSecurityGroup	Y
CreateSnapshot	Y
CreateSpotDatafeedSubscription	N
CreateTags	N
CreateVolume	Y
DeleteKeyPair	Y
DeletePlacementGroup	N
DeleteSecurityGroup	Y
DeleteSnapshot	Y
DeleteSpotDatafeedSubscription	N
DeleteTags	N
DeleteVolume	Y
DeregisterImage	Y
DescribeAddresses	Y
DescribeAvailabilityZones	Y
DescribeBundleTasks	N
DescribeImageAttribute	Y
DescribeImages	Y

DescribeInstanceAttribute	N
DescribeInstances	Y
DescribeKeyPairs	Y
DescribePlacementGroups	N
DescribeRegions	Y
DescribeReservedInstances	N
DescribeReservedInstancesOfferings	N
DescribeSecurityGroups	Y
DescribeSnapshotAttribute	N
DescribeSnapshots	Y
DescribeSpotDatafeedSubscription	N
DescribeSpotInstanceRequests	N
DescribeSpotPriceHistory	N
DescribeTags	N
DescribeVolumes	Y
DetachVolume	Y
DisassociateAddress	Y
GetConsoleOutput	Y
GetPasswordData	N
ImportKeyPair	Y
ModifyImageAttribute	Y
ModifyInstanceAttribute	N
ModifySnapshotAttribute	N
MonitorInstances	N
PurchaseReservedInstancesOffering	N
RebootInstances	Y
RegisterImage	Y
ReleaseAddress	Y
RequestSpotInstances	N
ResetImageAttribute	N
ResetInstanceAttribute	N
ResetSnapshotAttribute	N
RevokeSecurityGroupIngress	Y
RunInstances	Y

StartInstances	Y
StopInstances	Y
TerminateInstances	Y
UnmonitorInstances	N

Support

Any issues or discrepancies above please mail: eperdomo@cisco.com (Edgar Magana) and openstack-support@cisco.com

Related OpenStack Documentation
==========

Openstack Essex Administration Guides
http://docs.openstack.org/

Openstack Essex API Guides
http://docs.openstack.org/api/

Openstack Essex Developer Documentation
http://docs.openstack.org/developer/

Glossary

The following list describes acronyms and definitions for terms used throughout this document:

1. **AMI**: Amazon Machine Image
2. **API**: Application Programming Interface
3. **DNS**: Domain Name System
4. **EC2**: Elastic Compute Cloud
5. **FO**: Failover. A term used to indicate a failed active node is taken out of service and replaced automatically by another dedicated node that was previously in a standby mode of operation.
6. **HA**: High Availability

7. **IaaS**: Infrastructure as a Service
8. **iSCSI**: Internet Small Computer System Interface
9. **LVM**: Logical Volume Manager
10. **QCOW**: Qemu Copy On Write
11. **QEMU** stands for "Quick EMUlator
12. **SSL**: Secure Socket Layer
13. **UI**: User Interface
14. **VPC**: Virtual Private Cloud

4.0 Appendix

Hardware Monitoring OIDs

Component	OID	Response Values
Faults		
cucsFaultEntry	1.3.6.1.4.1.9.9.719.1.1.1.1	list
cucsFaultDn	1.3.6.1.4.1.9.9.719.1.1.1.1.2	list
cucsFaultRn	1.3.6.1.4.1.9.9.719.1.1.1.1.3	list (Append Fault Id from cucsFaultDn for single return: STRING)
cucsFaultAffectedObjectId	1.3.6.1.4.1.9.9.719.1.1.1.1.4	list (Append Fault Id from cucsFaultDn for single return: OID)
cucsFaultAffectedObjectDn	1.3.6.1.4.1.9.9.719.1.1.1.1.5	list (Append Fault Id from cucsFaultDn for single return: STRING)
cucsFaultProbableCause	1.3.6.1.4.1.9.9.719.1.1.1.1.7	list (Append Fault Id from cucsFaultDn for single return: INTEGER)
cucsFaultCode	1.3.6.1.4.1.9.9.719.1.1.1.1.9	list (Append Fault Id from cucsFaultDn for single return: INTEGER)
cucsFaultCreationTime	1.3.6.1.4.1.9.9.719.1.1.1.1.10	list (Append Fault Id from cucsFaultDn for single return: HEX-STRING)
cucsFaultDescription	1.3.6.1.4.1.9.9.719.1.1.1.1.11	list (Append Fault Id from cucsFaultDn for single return: STRING)
cucsFaultSeverity	1.3.6.1.4.1.9.9.719.1.1.1.1.20	list (Append Fault Id from cucsFaultDn for single return: INTEGER)
cucsFaultType	1.3.6.1.4.1.9.9.719.1.1.1.1.22	list (Append Fault Id

		from cucsFaultDn for single return: INTEGER)

CPU Utilization		
cucsProcessorUnitPerf	1.3.6.1.4.1.9.9.719.1.41.9.1.11	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)

Power Events		
cucsEquipmentPsuOperState	1.3.6.1.4.1.9.9.719.1.15.56.1.7	unknown(0), operable(1), inoperable(2), degraded(3), poweredOff(4), powerProblem(5), removed(6), voltageProblem(7), thermalProblem(8), performanceProblem(9), accessibilityProblem(10), identityUnestablishable(11), biosPostTimeout(12), disabled(13), malformedFru(14), fabricConnProblem(51), fabricUnsupportedConn(52), config(81), equipmentProblem(82), decomissioning(83), chassisLimitExceeded(84), notSupported(100), discovery(101), discoveryFailed(102), identify(103), postFailure(104), upgradeProblem(105), peerCommProblem(106), autoUpgrade(107)
cucsEquipmentPsuOperability	1.3.6.1.4.1.9.9.719.1.15.56.1.8	unknown(0), operable(1), inoperable(2), degraded(3), poweredOff(4), powerProblem(5), removed(6), voltageProblem(7), thermalProblem(8), performanceProblem(9), accessibilityProblem(10), identityUnestablishable(11), biosPostTimeout(12), disabled(13), malformedFru(14), fabricConnProblem(51), fabricUnsupportedConn(52), config(81), equipmentProblem(82), decomissioning(83),

		chassisLimitExceeded(84), notSupported(100), discovery(101), discoveryFailed(102), identify(103), postFailure(104), upgradeProblem(105), peerCommProblem(106), autoUpgrade(107)
cucsEquipmentPsuThermal	1.3.6.1.4.1.9.9.719.1.15.56.1.14	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)
cucsEquipmentPsuPower	1.3.6.1.4.1.9.9.719.1.15.56.1.10	unknown(0), on(1), test(2), off(3), online(4), offline(5), offduty(6), degraded(7), powerSave(8), error(9), ok(10), failed(11), notSupported(100)

Fan Enclosure Event		
cucsEquipmentFanOperState	1.3.6.1.4.1.9.9.719.1.15.12.1.9	unknown(0), operable(1), inoperable(2), degraded(3), poweredOff(4), powerProblem(5), removed(6), voltageProblem(7), thermalProblem(8), performanceProblem(9), accessibilityProblem(10), identityUnestablishable(11), biosPostTimeout(12), disabled(13), malformedFru(14), fabricConnProblem(51), fabricUnsupportedConn(52), config(81),

		equipmentProblem(82), decomissioning(83), chassisLimitExceeded(84), notSupported(100), discovery(101), discoveryFailed(102), identify(103), postFailure(104), upgradeProblem(105), peerCommProblem(106), autoUpgrade(107)
cucsEquipmentFanOperability	1.3.6.1.4.1.9.9.719.1.15.12.1.10	unknown(0), operable(1), inoperable(2), degraded(3), poweredOff(4), powerProblem(5), removed(6), voltageProblem(7), thermalProblem(8), performanceProblem(9), accessibilityProblem(10), identityUnestablishable(11), biosPostTimeout(12), disabled(13), malformedFru(14), fabricConnProblem(51), fabricUnsupportedConn(52), config(81), equipmentProblem(82), decomissioning(83), chassisLimitExceeded(84), notSupported(100), discovery(101), discoveryFailed(102), identify(103), postFailure(104), upgradeProblem(105), peerCommProblem(106), autoUpgrade(107)
cucsEquipmentFanPerf	1.3.6.1.4.1.9.9.719.1.15.12.1.11	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)
cucsEquipmentFanPower	1.3.6.1.4.1.9.9.719.1.15.12.1.12	unknown(0), on(1),

		test(2), off(3), online(4), offline(5), offduty(6), degraded(7), powerSave(8), error(9), ok(10), failed(11), notSupported(100)
cucsEquipmentFanThermal	1.3.6.1.4.1.9.9.719.1.15.12.1.16	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)

Memory Events		
cucsMemoryUnitOperState	1.3.6.1.4.1.9.9.719.1.30.11.1.13	unknown(0), operable(1), inoperable(2), degraded(3), poweredOff(4), powerProblem(5), removed(6), voltageProblem(7), thermalProblem(8), performanceProblem(9), accessibilityProblem(10), identityUnestablishable(11), biosPostTimeout(12), disabled(13), malformedFru(14), fabricConnProblem(51), fabricUnsupportedConn(52), config(81), equipmentProblem(82), decomissioning(83), chassisLimitExceeded(84), notSupported(100), discovery(101), discoveryFailed(102), identify(103), postFailure(104), upgradeProblem(105), peerCommProblem(106), autoUpgrade(107)

cucsMemoryUnitOperability	1.3.6.1.4.1.9.9.719.1.30.11.1.14	unknown(0), operable(1), inoperable(2), degraded(3), poweredOff(4), powerProblem(5), removed(6), voltageProblem(7), thermalProblem(8), performanceProblem(9), accessibilityProblem(10), identityUnestablishable(11), biosPostTimeout(12), disabled(13), malformedFru(14), fabricConnProblem(51), fabricUnsupportedConn(52), config(81), equipmentProblem(82), decomissioning(83), chassisLimitExceeded(84), notSupported(100), discovery(101), discoveryFailed(102), identify(103), postFailure(104), upgradeProblem(105), peerCommProblem(106), autoUpgrade(107)
cucsMemoryUnitPerf	1.3.6.1.4.1.9.9.719.1.30.11.1.15	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)
cucsMemoryUnitPower	1.3.6.1.4.1.9.9.719.1.30.11.1.16	unknown(0), on(1), test(2), off(3), online(4), offline(5), offduty(6), degraded(7), powerSave(8), error(9), ok(10), failed(11), notSupported(100)

cucsMemoryUnitThermal	1.3.6.1.4.1.9.9.719.1.30.11.1.22	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)

Processor Device Event		
cucsProcessorUnitOperState	1.3.6.1.4.1.9.9.719.1.41.9.1.9	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)
cucsProcessorUnitOperability	1.3.6.1.4.1.9.9.719.1.41.9.1.10	unknown(0), operable(1), inoperable(2), degraded(3), poweredOff(4), powerProblem(5), removed(6), voltageProblem(7), thermalProblem(8), performanceProblem(9), accessibilityProblem(10), identityUnestablishable(11), biosPostTimeout(12), disabled(13), malformedFru(14), fabricConnProblem(51), fabricUnsupportedConn(52), config(81), equipmentProblem(82), decomissioning(83), chassisLimitExceeded(84), notSupported(100), discovery(101), discoveryFailed(102), identify(103), postFailure(104), upgradeProblem(105), peerCommProblem(106), autoUpgrade(107)
cucsProcessorUnitPerf	1.3.6.1.4.1.9.9.719.1.41.9.1.11	unknown(0), ok(1),

		upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)
cucsProcessorUnitPower	1.3.6.1.4.1.9.9.719.1.41.9.1.12	unknown(0), on(1), test(2), off(3), online(4), offline(5), offduty(6), degraded(7), powerSave(8), error(9), ok(10), failed(11), notSupported(100)
cucsProcessorUnitThermal	1.3.6.1.4.1.9.9.719.1.41.9.1.19	unknown(0), ok(1), upperNonRecoverable(2), upperCritical(3), upperNonCritical(4), lowerNonCritical(5), lowerCritical(6), lowerNonRecoverable(7), notSupported(100)

Services Monitoring Openstack

Controller Node
nova-api
nova-cert
nova-network
nova-scheduler
nova-novncproxy
nova-consoleauth
nova-volume
keystone-all
glance-registry
glance-api
dnsmasq
libvirtd
apache2
ntp
lvm2
rabbitmq
memcached
mysql

Compute Node
nova-compute
libvirtd
ntp
lvm2

Storage Node
swift-container-updater
swift-container-auditor
swift-container-server
swift-container-sync
swift-container-replicator
swift-object-replicator

swift-object-auditor
swift-object-updater
swift-object-server
swift-account-server
swift-account-reaper
swift-account-auditor
swift-account-replicator
swift-proxy-server

References

[1] CTO-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/)
[6] https://github.com/CiscoSystems/cisco-openstack-docs/tree/master/keystone_passwd

4.2 Caveats

Caveats and Technical notes if needed

4.3 References

1. Wikipedia- Disk Partitioning:
   http://en.wikipedia.org/wiki/Disk_partitioning#PC_partition_types
2. Cisco System cisco-openstack-docs GitHub Site:
   https://github.com/CiscoSystems/cisco-openstack-docs
3. Ubuntu 12.04 Server Guide- Advanced Installation:
   https://help.ubuntu.com/12.04/serverguide/C/advanced-installation.html
4. Openstack.org:
   http://www.openstack.org/