**Openstack Installation for controller and compute on Xen Hypervisor**

Version - 1.1

[1. OpenStack packages](#h.ptje9y3wk9g0)

[2. Basic Operating System Configuration](#h.6d53slt2sjit)

[MySQL DB Setup - Controller](#h.84fj4nk2d342)

[MySQL DB Setup - Node](#h.boy5ttu64acp)

[Messaging Server(RabbitMQ) Setup](#h.yr36wif9cv23)

[3. Identity Service Setup & Configuration](#h.kuezej9usa2)

[Install the Identity Service](#h.wz5wo7tn0tc2)

[Define users, tenants, and roles](#h.kkrefhqv9r7m)

[Define service and API endpoint](#h.v4sqy3m2zlwb)

[Verify Identity Service Installation](#h.r7s25a6wpab)

[4. Image Service Setup & Configuration](#h.3atnavf6r1xf)

[Install the Image Service](#h.mzm3f7peu2jq)

[Define user and roles](#h.or7el1iz91fb)

[Define service and API endpoint](#h.m9yuejedw99a)

[Verify Image Service Installation](#h.jf34hkf1va7b)

[5. Compute Controller Service Setup & Configuration](#h.p7acdtz3ovy2)

[Install Compute Controller Services](#h.8r3dmi89rwdc)

[Define user and roles](#h.c6ys6517l684)

[Define service and API endpoint](#h.1ct10km7enml)

[Verify Compute Service Installation](#h.ok4e12yyt78a)

[6. Setup Xen Hypervisor and Ubuntu 12.04 domU:](#h.ctnoa6pmdcco)

[7. Nova Compute and Nova Network Node Setup and Configuration:](#h.ea6un75zhm5v)

[Launch an Instance:](#h.dj4k6f4haiil)

[8. Dashboard Setup & Configuration](#h.2u7tmc7z8pyd)

[Add the Dashboard](#h.o07mrg6vf3y3)

[9. Block Storage Service Setup & Configuration](#h.8h6u87491jb5)

[Install a Block Storage Service Controller](#h.drwjm1or1oxy)

[Define user and roles](#h.aoxlosp8vg05)

[Define service and API endpoint](#h.7hy6il1fmt62)

[Annex 1](#h.bv6t05g62p0j)

[Annex 2](#h.hxiwxuf2zeq7)

[References](#h.frx7t7m1dniy)

Here I am assuming my Controller Machine IP as 10.35.34.207, Compute machine IP as 10.35.34.208 and Hypervisor IP as 10.35.34.13. Please update the IPs as per your environment while following this document.

# 1. OpenStack packages

**Use the Ubuntu Cloud Archive for Havana**

* Install the Ubuntu Cloud Archive for Havana

**# apt-get install python-software-properties**

**# add-apt-repository cloud-archive:havana**

* Update the package database, upgrade your system, and reboot

**# apt-get update && apt-get dist-upgrade**

**# reboot**

# 2. Basic Operating System Configuration

## MySQL DB Setup - Controller

* Install mysql packages on controller

**# apt-get install python-mysqldb mysql-server**

* Edit /etc/mysql/my.cnf and set the bind-address to the IP address of the controller

**bind-address = 10.35.34.207**

* Restart mysql service

**# service mysql restart**

* Delete the anonymous users that are created when the database is first started

**# mysql\_install\_db**

**# mysql\_secure\_installation**

This command presents a number of options for you to secure your database installation. Respond yes to all prompts unless you have a good reason to do otherwise.

## 

## MySQL DB Setup - Node

On all nodes other than the controller node, install the MySQL client and the MySQL Python library on any system that does not host a MySQL database:

**# apt-get install python-mysqldb**

## Messaging Server(RabbitMQ) Setup

* On the controller node, install the messaging queue server RabbitMQ

**# apt-get install rabbitmq-server**

* Change the default guest password of RabbitMQ

**# rabbitmqctl change\_password guest <passwd>**

# 

# 3. Identity Service Setup & Configuration

## Install the Identity Service

* Install the OpenStack Identity Service on the controller node

**# apt-get install keystone**

* Edit /etc/keystone/keystone.conf and change the [sql] section

**[sql]**

**....**

**connection = mysql://keystone:<passwd>@10.35.34.207/keystone**

**....**

* Delete the keystone.db file created in the /var/lib/keystone/ directory so that it does not get used

**# rm /var/lib/keystone/keystone.db**

* Create a keystone database

**# mysql -u root -p**

**mysql> CREATE DATABASE keystone;**

**mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY '<passwd>';**

**mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY '<passwd>';**

**mysql> quit**

* Create the database tables for the Identity Service

**# keystone-manage db\_sync**

* Use openssl to generate a random token and store it in the configuration file

**# openssl rand -hex 10**

**7c9a33aeae6a34ad6ff1**

* Edit /etc/keystone/keystone.conf and change the [DEFAULT] section, replacing ADMIN\_TOKEN with the results of the command

**[DEFAULT]**

**....**

**# A "shared secret" between keystone and other openstack services**

**admin\_token = 7c9a33aeae6a34ad6ff1**

**....**

* Restart the keystone service

**# service keystone restart**

## Define users, tenants, and roles

* set OS\_SERVICE\_TOKEN, as well as OS\_SERVICE\_ENDPOINT to specify where the Identity Service is running

**# export OS\_SERVICE\_TOKEN=7c9a33aeae6a34ad6ff1**

**# export OS\_SERVICE\_ENDPOINT=http://10.35.34.207:35357/v2.0**

* Create a tenant for an administrative user and a tenant for other OpenStack services to use

**# keystone tenant-create --name=admin --description="Admin Tenant"**

**# keystone tenant-create --name=service --description="Service Tenant"**

* Create an administrative user called admin

**# keystone user-create --name=admin --pass=<passwd>**

* Create a role for administrative tasks called admin

**# keystone role-create --name=admin**

* Add roles to users

**# keystone user-role-add --user=admin --tenant=admin --role=admin**

## Define service and API endpoint

* Create a service entry for the Identity Service

**# keystone service-create --name=keystone --type=identity --description="Keystone Identity Service"**

* Specify an API endpoint for the Identity Service by using the returned service ID in above command output

**# keystone endpoint-create \**

**--service-id=7cfc6f0d6bff41cba6a0437a8977e3ee \**

**--publicurl=http://10.35.34.207:5000/v2.0 \**

**--internalurl=http://10.35.34.207:5000/v2.0 \**

**--adminurl=http://10.35.34.207:35357/v2.0**

## Verify Identity Service Installation

* Unset the OS\_SERVICE\_TOKEN & OS\_SERVICE\_ENDPOINT variables

**# unset OS\_SERVICE\_TOKEN OS\_SERVICE\_ENDPOINT**

If not unset, will face the issues while trying next command

* Request an authentication token using the admin user and the password

**# keystone --os\_username=admin --os\_password=<passwd> --os\_auth\_url=http://10.35.34.207:35357/v2.0 token-get**

* Set up a keystonerc file with the admin credentials and admin endpoint

**export OS\_USERNAME=admin**

**export OS\_PASSWORD=<passwd>**

**export OS\_TENANT\_NAME=admin**

**export OS\_AUTH\_URL=http://10.35.34.207:35357/v2.0**

* Source this file to read in the environment variables

**# source keystonerc**

* Verify the keystone identity service

**# keystone token-get**

The command returns a token and the ID of the specified tenant

# 4. Image Service Setup & Configuration

## Install the Image Service

* Install the Image Service on the controller node

**# apt-get install glance python-glanceclient python-cinderclient\* sheepdog\***

* Edit /etc/glance/glance-api.conf and /etc/glance/glance-registry.conf and change the [DEFAULT] section

**[DEFAULT]**

**....**

**# SQLAlchemy connection string for the reference implementation**

**# registry server. Any valid SQLAlchemy connection string is fine.**

**sql\_connection = mysql://glance:<passwd>@10.35.34.207/glance**

**....**

* Delete the glance.sqlite file created in the /var/lib/glance/ directory so that it does not get used

**# rm /var/lib/glance/glance.sqlite**

* Create a glance database user

**# mysql -u root -p**

**mysql> CREATE DATABASE glance;**

**mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY '<passwd>';**

**mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY '<passwd>';**

**mysql> quit**

* Create the database tables for the Image Service

**# glance-manage db\_sync**

**Note : \* -** Installation of these packages is required as glance fails to work without these. These packages may or may not be documented in the Openstack Documentation online.

## Define user and roles

* set OS\_SERVICE\_TOKEN, as well as OS\_SERVICE\_ENDPOINT to specify where the Identity Service is running

**# export OS\_SERVICE\_TOKEN=7c9a33aeae6a34ad6ff1**

**# export OS\_SERVICE\_ENDPOINT=http://10.35.34.207:35357/v2.0**

* Create a glance user that the Image Service can use to authenticate with the Identity Service:

**# keystone user-create --name=glance --pass=<passwd>**

* Use the service tenant and give the user the admin role

**# keystone user-role-add --user=glance --tenant=service --role=admin**

* Edit /etc/glance/glance-api.conf and /etc/glance/glance-registry.conf and change the [keystone\_authtoken] section

**....**

**[keystone\_authtoken]**

**auth\_host = 10.35.34.207**

**auth\_port = 35357**

**auth\_protocol = http**

**admin\_tenant\_name = service**

**admin\_user = glance**

**admin\_password = <passwd>**

* Add the credentials to the /etc/glance/glance-api-paste.ini and /etc/glance/glance-registry-paste.ini files

**....**

**[filter:authtoken]**

**paste.filter\_factory = keystone.middleware.auth\_token:filter\_factory**

**auth\_host = 10.35.34.207**

**delay\_auth\_decision = true**

**admin\_tenant\_name = service**

**admin\_user = glance**

**admin\_password = <passwd>**

**flavor=keystone**

**....**

## Define service and API endpoint

* Register the Image Service with the Identity Service

**# keystone service-create --name=glance --type=image --description="Glance Image Service"**

* Use the id property returned for the service to create the endpoint

**# keystone endpoint-create \**

**--service\_id=461efefebfba47df98ab9a6f8dc80502 \**

**--publicurl=http://10.35.34.207:9292 \**

**--internalurl=http://10.35.34.207:9292 \**

**--adminurl=http://10.35.34.207:9292**

* Restart the glance service with its new settings

**# service glance-registry restart**

**# service glance-api restart**

## Verify Image Service Installation

* To verify glance

**# mkdir /root/images**

**# cd /root/images**

**# wget** [**http://cdn.download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86\_64-disk.img**](http://cdn.download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86_64-disk.img)

* Upload the image to the Image Service

**# glance image-create --name="Cirros" --disk-format=qcow2 --container-format=bare --is-public=true < cirros-0.3.1-x86\_64-disk.img**

* Confirm that the image was uploaded and display its attributes

**# glance image-list**

# 5. Compute Controller Service Setup & Configuration

## Install Compute Controller Services

* Install the Compute packages

**# apt-get install nova-novncproxy novnc nova-api nova-ajax-console-proxy nova-cert nova-conductor nova-consoleauth nova-doc nova-scheduler python-novaclient**

* Edit the /etc/nova/nova.conf file and add these lines to the [database] and [keystone\_authtoken] sections:

**....**

**[database]**

**# The SQLAlchemy connection string used to connect to the database**

**connection = mysql://nova:<passwd>@10.35.34.207/nova**

**....**

**....**

**[keystone\_authtoken]**

**auth\_host = 10.35.34.207**

**auth\_port = 35357**

**auth\_protocol = http**

**admin\_tenant\_name = service**

**admin\_user = nova**

**admin\_password = <passwd>**

**....**

* Configure the Compute Service to use the RabbitMQ message broker by setting these configuration keys in the [DEFAULT] configuration group of the /etc/nova/nova.conf file

**....**

**rpc\_backend = nova.rpc.impl\_kombu**

**rabbit\_host = 10.35.34.207**

**rabbit\_password = <passwd>**

**....**

* Delete the nova.sqlite file created in the /var/lib/nova/ directory

**# rm /var/lib/nova/nova.sqlite**

* Create a nova database user

**# mysql -u root -p**

**mysql> CREATE DATABASE nova;**

**mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY '<passwd>';**

**mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY '<passwd>';**

**mysql> quit**

* Create the database tables for the Compute Service

**# nova-manage db sync**

* Edit the /etc/nova/nova.conf file, Set the my\_ip, vncserver\_listen, and vncserver\_proxyclient\_address configuration options to the IP address of the controller node

**....**

**my\_ip=10.35.34.207**

**vncserver\_listen=10.35.34.207**

**vncserver\_proxyclient\_address=10.35.34.207**

**....**

## Define user and roles

* Create a nova user that Compute uses to authenticate with the Identity Service

**# keystone user-create --name=nova --pass=<passwd>**

* Use the service tenant and give the user the admin role

**# keystone user-role-add --user=nova --tenant=service --role=admin**

* Edit the [DEFAULT] section in the /etc/nova/nova.conf file to add the following key

**....**

**auth\_strategy=keystone**

**....**

* Add the credentials to the /etc/nova/api-paste.ini file, Add these options to the [filter:authtoken] section

**....**

**[filter:authtoken]**

**paste.filter\_factory = keystoneclient.middleware.auth\_token:filter\_factory**

**auth\_host = 10.35.34.207**

**auth\_port = 35357**

**auth\_protocol = http**

**admin\_tenant\_name = service**

**admin\_user = nova**

**admin\_password = <passwd>**

**....**

## Define service and API endpoint

* Register Compute with the Identity Service

**# keystone service-create --name=nova --type=compute --description="Nova Compute service”**

* Use the id property that is returned to create the endpoint

**# keystone endpoint-create \**

**--service-id=b1dcbb4a2f6041fc802efefb9b6cc76d \**

**--publicurl=http://10.35.34.207:8774/v2/%\(tenant\_id\)s \**

**--internalurl=http://10.35.34.207:8774/v2/%\(tenant\_id\)s \**

**--adminurl=http://10.35.34.207:8774/v2/%\(tenant\_id\)s**

* Restart Compute services

**# service nova-api restart**

**# service nova-cert restart**

**# service nova-consoleauth restart**

**# service nova-scheduler restart**

**# service nova-conductor restart**

**# service nova-novncproxy restart**

## Verify Compute Service Installation

* To verify your configuration

**# nova image-list**

Will list the available images

# 6. Setup Xen Hypervisor and Ubuntu 12.04 domU:

Please use either of the following Titled documents for setting up Xen Hypervisor and Ubuntu 12.04 domU instance on it :

1. For diskless dom0 setup of xen, use:

**“Diskless booting of dom0 in Xen hypervisor with driver domain installation”**

2. For normal xen setup (without diskless setup and driver domain isolation), use:

**“Dom0 setup on Xen Hypervisor for RPCore”**

The following Nova Compute setup is to be done on a domU instance on the xen hypervisor:

**Assumption:** The domU machine is running with Ubuntu 12.04 OS.

# 7. Nova Compute and Nova Network Node Setup and Configuration:

* Install the following packages for Compute Node :

**# apt-get install nova-compute python-guestfs**

When prompted to create a **supermin** appliance, respond yes.

* Make the current kernel readable as follows :

**# dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-$(uname -r)**

For more info see this :<https://bugs.launchpad.net/ubuntu/+source/linux/+bug/759725>

* To also enable this override for all future kernel updates, create the file /etc/kernel/postinst.d/statoverride containing:

**#!/bin/sh**

**version="$1"**

**# passing the kernel version is required**

**[ -z "${version}" ] && exit 0**

**dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-${version}**

* Remember to make the file executable:

**# chmod +x /etc/kernel/postinst.d/statoverride**

* Remove the SQLite database created by the packages:

**#rm /var/lib/nova/nova.sqlite**

* Edit the /etc/nova/nova.conf configuration file and add these lines to the appropriate sections:

**...**

**[DEFAULT]**

**...**

**auth\_strategy=keystone**

**...**

**[database]**

**# The SQLAlchemy connection string used to connect to the database**

**connection = mysql://nova:<passwd>@10.35.34.207/nova**

* Configure the Compute Service to use the RabbitMQ message broker by setting these configuration keys in the [DEFAULT] configuration group of the /etc/nova/nova.conf file:

**rpc\_backend = nova.rpc.impl\_kombu**

**rabbit\_host = 10.35.34.207**

**rabbit\_password = <passwd>**

* Configure Compute to provide remote console access to instances. Edit /etc/nova/nova.conf and add the following keys under the [DEFAULT] section:

**[DEFAULT]**

**...**

**my\_ip=10.35.34.208**

**vnc\_enabled=True**

**vncserver\_listen=0.0.0.0**

**vncserver\_proxyclient\_address=10.35.34.208**

**novncproxy\_base\_url=**[**http://10.35.34.207:6080/vnc\_auto.html**](http://10.35.34.207:6080/vnc_auto.html)

(Assuming that the compute node IP is 10.35.34.208 and is accessible to the Controller)

* Specify the host that runs the Image Service. Edit /etc/nova/nova.conf file and add these lines to the [DEFAULT] section:

**[DEFAULT]**

**...**

**glance\_host=10.35.34.207**

* Edit the /etc/nova/api-paste.ini file to add the credentials to the [filter:authtoken] section:

**[filter:authtoken]**

**paste.filter\_factory = keystoneclient.middleware.auth\_token:filter\_factory**

**auth\_host = 10.35.34.207**

**auth\_port = 35357**

**auth\_protocol = http**

**admin\_tenant\_name = service**

**admin\_user = nova**

**admin\_password = <passwd>**

* Add the following xenapi related attributes to /etc/nova/nova-compute.conf :

**[DEFAULT]**

**...**

**# Xen settings**

**connection\_type=xenapi**

**compute\_driver=xenapi.XenAPIDriver**

**xenapi\_connection\_url=http://10.35.34.13**

**xenapi\_connection\_username=root**

**xenapi\_connection\_password=<passwd>**

**xenapi\_vif\_driver=nova.virt.xenapi.vif.XenAPIBridgeDriver**

**xenapi\_proxy\_connection\_url=**[**http://10.35.34.13:8080**](http://10.35.34.13:8080/)

(Assumption : 10.35.34.13 is the xen-hypervisor's ip)

* Setup the networking for the compute node as follows :

Enable promiscuous mode on eth0:

**# ip link set eth1 promisc on**

* Install the compute networking packages as follows:

**# apt-get install nova-network nova-api-metadata**

* Edit the nova.conf file to define the networking mode:

Edit the /etc/nova/nova.conf file and add these lines to the [DEFAULT] section:

**network\_manager=nova.network.manager.FlatDHCPManager**

**xenapi\_vif\_driver=nova.virt.xenapi.vif.XenAPIBridgeDriver**

**network\_size=253**

**allow\_same\_net\_traffic=False**

**multi\_host=True**

**send\_arp\_for\_ha=True**

**fixed\_range=10.35.34.0/24**

**share\_dhcp\_address=True**

**force\_dhcp\_release=True**

**flat\_network\_bridge=xenbr0**

**flat\_interface=eth0**

**public\_interface=eth0**

**flat\_injected=False**

**network\_host=10.35.34.208**

**firewall\_driver=nova.virt.xenapi.firewall.Dom0IptablesFirewallDriver**

* Restart the Compute and Network service.

**# service nova-compute restart**

**# service nova-network restart**

* Source this file to read in the environment variables

**# source keystonerc (This can be copied over from the controller server)**

* Check if python-novaclient is installed :

**# dpkg –-list | grep python-novaclient.**

* Run the **nova network-create** command on the controller:

**# nova network-create vmnet --fixed-range-v4=10.35.34.0/24 --bridge-interface=xenbr0 --multi-host=T**

**Note :** Refer the Annex 2 for a sample nova.conf for compute.

## Launch an Instance:

Once the setup is complete, you can launch an instance and validate the setup as follows :

* Generate a key-pair:

**$ ssh-keygen**

**$ cd .ssh**

**$ nova keypair-add --pub\_key id\_rsa.pub mykey**

* View available keypairs:

**$ nova keypair-list**

+--------+-------------------------------------------------+

| Name | Fingerprint |

+--------+-------------------------------------------------+

| mykey | b0:18:32:fa:4e:d4:3c:1b:c4:6c:dd:cb:53:29:13:82 |

+--------+-------------------------------------------------+

* Check the default flavors that are available to you :

**$ nova flavor-list**

+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+

| ID | Name | Memory\_MB | Disk | Ephemeral | Swap | VCPUs | RXTX\_Factor | Is\_Public |

+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+

| 1 | m1.tiny | 512 | 1 | 0 | | 1 | 1.0 | True |

| 2 | m1.small | 2048 | 20 | 0 | | 1 | 1.0 | True |

| 3 | m1.medium | 4096 | 40 | 0 | | 2 | 1.0 | True |

| 4 | m1.large | 8192 | 80 | 0 | | 4 | 1.0 | True |

| 5 | m1.xlarge | 16384 | 160 | 0 | | 8 | 1.0 | True |

+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+

* Get ID of the image that you added earlier :

**$ nova image-list**

+--------------------------------------+--------------+--------+--------+

| ID | Name | Status | Server |

+--------------------------------------+--------------+--------+--------+

| 9e5c2bee-0373-414c-b4af-b91b0246ad3b | CirrOS 0.3.1 | ACTIVE | |

+--------------------------------------+--------------+--------+--------+

* To use SSH and ping, you must configure security group rules :

**# nova secgroup-add-rule default tcp 22 22 0.0.0.0/0**

**# nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0**

* Now we can launch the instance using the following syntax :

**$ nova boot --flavor flavorType --key\_name keypairName --image ID newInstanceName**

* For example:

**$ nova boot --flavor 1 --key\_name mykey --image 9e5c2bee-0373-414c-b4af-b91b0246ad3b --security\_group default cirrOS**

+--------------------------------------+--------------------------------------+

| Property | Value |

+--------------------------------------+--------------------------------------+

| OS-EXT-STS:task\_state | scheduling |

| image | CirrOS 0.3.1 |

| OS-EXT-STS:vm\_state | building |

| OS-EXT-SRV-ATTR:instance\_name | instance-00000001 |

| OS-SRV-USG:launched\_at | None |

| flavor | m1.tiny |

| id | 3bdf98a0-c767-4247-bf41-2d147e4aa043 |

| security\_groups | [{u'name': u'default'}] |

| user\_id | 530166901fa24d1face95cda82cfae56 |

| OS-DCF:diskConfig | MANUAL |

| accessIPv4 | |

| accessIPv6 | |

| progress | 0 |

| OS-EXT-STS:power\_state | 0 |

| OS-EXT-AZ:availability\_zone | nova |

| config\_drive | |

| status | BUILD |

| updated | 2013-10-10T06:47:26Z |

| hostId | |

| OS-EXT-SRV-ATTR:host | None |

| OS-SRV-USG:terminated\_at | None |

| key\_name | mykey |

| OS-EXT-SRV-ATTR:hypervisor\_hostname | None |

| name | cirrOS |

| adminPass | DWCDW6FnsKNq |

| tenant\_id | e66d97ac1b704897853412fc8450f7b9 |

| created | 2013-10-10T06:47:23Z |

| os-extended-volumes:volumes\_attached | [] |

| metadata | {} |

+--------------------------------------+--------------------------------------+

As this setup is complete, we need to export this image to a file on dom0 and copy it to the USB for disklessdom0 setup to complete. Use the following steps to export it to a file:

* Shutdown the domU (compute node) instance as follows:

**# xe vm-shutdown uuid=<nova-compute-domU-uuid>**

* Export the image as follows :

**# xe vm-export uuid=<nova-compute-domU-uuid> filename=/home/<filename>** **compress=true**

This will take some time to complete. Once done copy this file to the USB and gives its path in NOVA\_VM\_IMAGE\_PATH in the config file in disklessdom0 setup.

# 

# 

# 8. Dashboard Setup & Configuration

## Add the Dashboard

* Install the dashboard on the node that can contact the Identity Service as root

**# apt-get install memcached libapache2-mod-wsgi openstack-dashboard**

* Remove the openstack-dashboard-ubuntu-theme package

**# apt-get remove --purge openstack-dashboard-ubuntu-theme**

* Modify the value of CACHES['default']['LOCATION'] in /etc/openstack-dashboard/local\_settings.py to match the ones set in /etc/memcached.conf. Open /etc/openstack-dashboard/local\_settings.py and look for this line

**....**

**CACHES = {**

**'default': {**

**'BACKEND' : 'django.core.cache.backends.memcached.MemcachedCache',**

**'LOCATION' : '127.0.0.1:11211'**

**}**

**}**

**....**

* Update the ALLOWED\_HOSTS in local\_settings.py to include the addresses you wish to access the dashboard from. Edit /etc/openstack-dashboard/local\_settings.py

**....**

**ALLOWED\_HOSTS = '\*'**

**....**

* This guide assumes that you are running the Dashboard on the controller node. You can easily run the dashboard on a separate server, by changing the appropriate settings in local\_settings.py. Edit /etc/openstack-dashboard/local\_settings.py and change OPENSTACK\_HOST to the hostname of your Identity Service

**....**

**OPENSTACK\_HOST = "controller"**

**....**

* Start the Apache web server and memcached

**# service apache2 restart**

**# service memcached restart**

* Access the dashboard at<http://10.35.34.207/horizon>

# 

# 

# 

# 9. Block Storage Service Setup & Configuration

## Install a Block Storage Service Controller

* Install the appropriate packages for the Block Storage Service on controller

**# apt-get install cinder-api cinder-scheduler**

* Edit the /etc/cinder/cinder.conf file and add the following key under the [database] section

**....**

**[database]**

**connection = mysql://cinder:<passwd>@10.35.34.207/cinder**

**....**

* Create a cinder database user

**# mysql -u root -p**

**mysql> CREATE DATABASE cinder;**

**mysql> GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'localhost' IDENTIFIED BY '<passwd>';**

**mysql> GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'%' IDENTIFIED BY '<passwd>';**

**mysql> quit**

* Create the database tables for the Block Storage Service

**# cinder-manage db sync**

## Define user and roles

* Create a cinder user, the Block Storage Service uses this user to authenticate with the Identity Service

**# keystone user-create --name=cinder –pass=<passwd>**

* Use the service tenant and give the user the admin role

**# keystone user-role-add --user=cinder --tenant=service –role=admin**

* Add the credentials to the file /etc/cinder/api-paste.ini. Open the file in a text editor and locate the section [filter:authtoken]. Set the following options:

**....**

**[filter:authtoken]**

**paste.filter\_factory = keystoneclient.middleware.auth\_token:filter\_factory**

**auth\_host = 10.35.34.207**

**auth\_port = 35357**

**auth\_protocol = http**

**admin\_tenant\_name = service**

**admin\_user = cinder**

**admin\_password = <passwd>**

**....**

* Configure Block Storage to use the RabbitMQ message broker by setting these configuration keys in the [DEFAULT] configuration group of the /etc/cinder/cinder.conf file

**....**

**rpc\_backend = cinder.openstack.common.rpc.impl\_kombu**

**rabbit\_host = 10.35.34.207**

**rabbit\_port = 5672**

**rabbit\_userid = guest**

**rabbit\_password = <passwd>**

**....**

## Define service and API endpoint

* Register the Block Storage Service with the Identity Service

**# keystone service-create --name=cinder --type=volume --description="Cinder Volume Service"**

* Use the id property returned to create the endpoint

**# keystone endpoint-create \**

**--service-id=8d7cbda9638945f19a4d5a667adf1258 \**

**--publicurl=http://10.35.34.207:8776/v1/%\(tenant\_id\)s \**

**--internalurl=http://10.35.34.207:8776/v1/%\(tenant\_id\)s \**

**--adminurl=http://10.35.34.207:8776/v1/%\(tenant\_id\)s**

* Also register a service and endpoint for version 2 of the Block Storage Service API

**# keystone service-create --name=cinderv2 --type=volumev2 --description="Cinder Volume Service V2"**

* Use the id property returned to create the endpoint

**# keystone endpoint-create \**

**--service-id=509a3c9295df4404962d599508db21c6 \**

**--publicurl=http://10.35.34.207:8776/v1/%\(tenant\_id\)s \**

**--internalurl=http://10.35.34.207:8776/v1/%\(tenant\_id\)s \**

**--adminurl=http://10.35.34.207:8776/v1/%\(tenant\_id\)s**

* Restart the cinder service with its new settings

**# service cinder-scheduler restart**

**# service cinder-api restart**

**Note :** Refer the Annex 1 for a sample nova.conf for controller.

# 

# Annex 1

Following is the /etc/nova/nova.conf sample for openstack controller instance :

[DEFAULT]

max\_kernel\_ramdisk\_size=1073741824

dhcpbridge\_flagfile=/etc/nova/nova.conf

dhcpbridge=/usr/bin/nova-dhcpbridge

logdir=/var/log/nova

state\_path=/var/lib/nova

lock\_path=/var/lock/nova

force\_dhcp\_release=True

iscsi\_helper=tgtadm

#libvirt\_use\_virtio\_for\_bridges=True

#connection\_type=libvirt

root\_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

verbose=True

ec2\_private\_dns\_show\_ip=True

api\_paste\_config=/etc/nova/api-paste.ini

volumes\_path=/var/lib/nova/volumes

enabled\_apis=metadata,ec2,osapi\_compute

rpc\_backend = nova.rpc.impl\_kombu

rabbit\_host = <controller-ip>

rabbit\_password = <passwd>

my\_ip=<controller-ip>

vncserver\_listen=<controller-ip>

vncserver\_proxyclient\_address=<controller-ip>

auth\_strategy=keystone

multi\_host=True

# #

[database]

# The SQLAlchemy connection string used to connect to the database

connection = mysql://nova:<passwd>@<controller-ip>/nova

[keystone\_authtoken]

auth\_host = <controller-ip>

auth\_port = 35357

auth\_protocol = http

admin\_tenant\_name = service

admin\_user = nova

admin\_password = <passwd>

# Annex 2

/etc/nova/nova.conf sample for nova-compute instance :

[DEFAULT]

max\_kernel\_ramdisk\_size=2073741824

dhcpbridge\_flagfile=/etc/nova/nova.conf

dhcpbridge=/usr/bin/nova-dhcpbridge

logdir=/var/log/nova

state\_path=/var/lib/nova

lock\_path=/var/lock/nova

force\_dhcp\_release=True

iscsi\_helper=tgtadm

#libvirt\_use\_virtio\_for\_bridges=True

root\_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

verbose=True

ec2\_private\_dns\_show\_ip=True

api\_paste\_config=/etc/nova/api-paste.ini

volumes\_path=/var/lib/nova/volumes

enabled\_apis=metadata,ec2,osapi\_compute

#enabled\_apis=metadata

metadata\_host = <controller-ip>

auth\_strategy=keystone

rpc\_backend = nova.rpc.impl\_kombu

rabbit\_host = <controller-ip>

rabbit\_password = <passwd>

my\_ip=<compute-ip>

vnc\_enabled=True

vncserver\_listen=0.0.0.0

vncserver\_proxyclient\_address=<compute-ip>

novncproxy\_base\_url=http://<controller-ip>:6080/vnc\_auto.html

glance\_host=<controller-ip>

glance\_api\_servers=<controller-ip>:9292

image\_service=nova.image.glance.GlanceImageService

compute\_scheduler\_driver=nova.scheduler.simple.SimpleScheduler

nova\_url=http://<controller-ip>:8774/v1.1/

root\_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf

# Network

network\_manager=nova.network.manager.FlatDHCPManager

xenapi\_vif\_driver=nova.virt.xenapi.vif.XenAPIBridgeDriver

network\_size=253

allow\_same\_net\_traffic=False

multi\_host=True

send\_arp\_for\_ha=True

share\_dhcp\_address=True

force\_dhcp\_release=True

flat\_network\_bridge=xenbr0

flat\_interface=eth1

public\_interface=eth0

flat\_injected=False

network\_host=<nova-network instance ip>

firewall\_driver=nova.virt.xenapi.firewall.Dom0IptablesFirewallDriver

#firewall\_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

[database]

connection = mysql://nova:<passwd>@<controller-ip>/nova

# Xen settings

connection\_type=xenapi

compute\_driver=xenapi.XenAPIDriver

xenapi\_connection\_url=http://<xen-server-ip>

xenapi\_connection\_username=root

xenapi\_connection\_password=<root-pass>

xenapi\_vif\_driver=nova.virt.xenapi.vif.XenAPIBridgeDriver

xenapi\_proxy\_connection\_url=http://<xen-server-ip>:8080

# References

* Openstack Havana Install Documentation - http://docs.openstack.org/havana/install-guide/install/apt/content/