OpenStack ICE-house Deployment on Ubuntu Server 14.04 LTS Server

Draft - 1.0

This configuration guide includes Multi-node OpenStack deployment on Ubuntu Server 14.04 LTS (sever) platform. In my case Ubuntu servers are nested in VMware ESXi 5.5.

Setup requirements are as follows,

- 1 ESXi Server with minimum 64 GB memory, 1 TB hard drive and Intel cpu with min 8 cores
- 3 Ubuntu 14.04 LTS server edition VMs with Multiple NICS. Controller and Compute node with 2 NICs each and Network node with 3 NICS
- Internet connectivity

Note OpenStack deployment is also possible on ESXi server having lower memory. Example ESXi server with 32 Gb memory and 4 cpus however it is always better to have esxi host with 64GB Memory

Let's begin,

Download Ubuntu 14.04 LTS Server ISO from Internet and copy it on the data store of the ESX host on which Openstack host vms will get deployed.

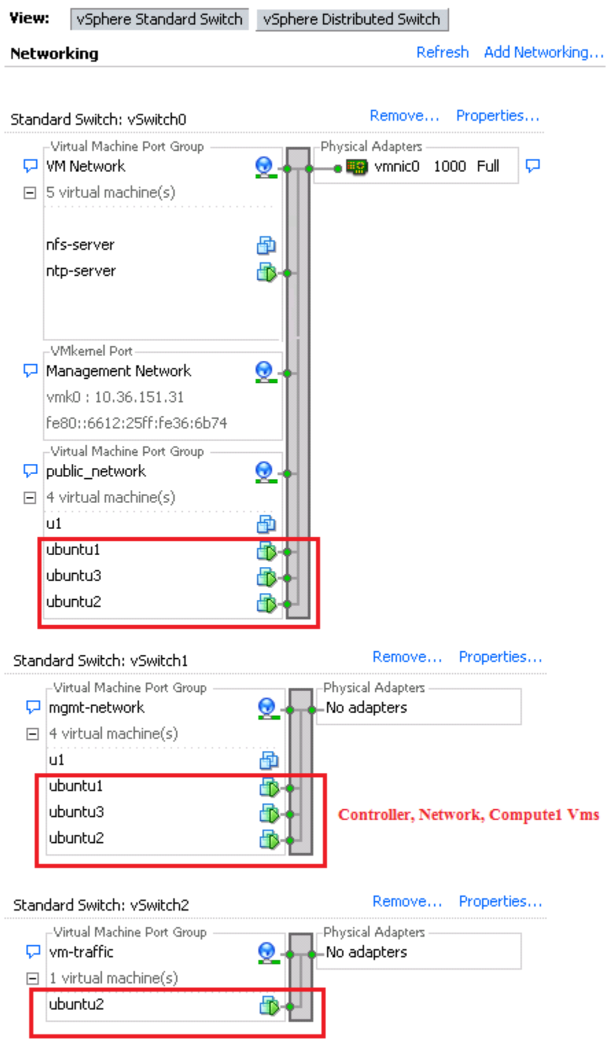
You may choose the latest Ubuntu version if any. Currently 14.0.4 LTS server is the latest version hence I will recommend to use the same version.

- **Create Controller VM** with 16GB memory, 100GB Hard drive, 2 NICS with VMXnet3 Adaptor type, min 2 vCpus with one core each and Install Ubuntu.

- **Create Network VM** with 12GB memory, 100GB Hard drive, 3 NICS with VMXnet3 Adaptor type, min 2 vCpus with one core each and install Ubuntu

- **Create Compute VM** with 16GB memory, 100GB Hard drive, 2 NICS with VMXnet3 Adaptor type, min 2 vCpus with one core each and install Ubuntu

Refer ESXi server network configuration port group mapping



View: vSphere Standard Switch | vSphere Distributed Switch

Networking — Refresh | Add Networking...

Standard Switch: vSwitch0 — Remove... | Properties...

Virtual Machine Port Group
VM Network
5 virtual machine(s)
nfs-server
ntp-server

Physical Adapters
vmnic0 1000 Full

VMkernel Port
Management Network
vmk0 : 10.36.151.31
fe80::6612:25ff:fe36:6b74

Virtual Machine Port Group
public_network
4 virtual machine(s)
u1
ubuntu1
ubuntu3
ubuntu2

Standard Switch: vSwitch1 — Remove... | Properties...

Virtual Machine Port Group
mgmt-network
4 virtual machine(s)
u1
ubuntu1
ubuntu3
ubuntu2

Physical Adapters
No adapters

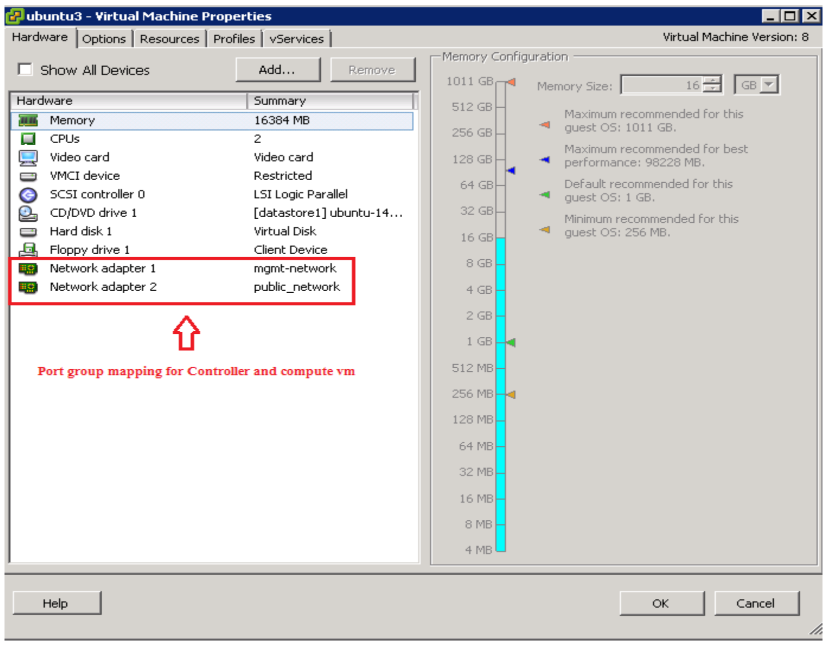
Controller, Network, Compute1 Vms

Standard Switch: vSwitch2 — Remove... | Properties...

Virtual Machine Port Group
vm-traffic
1 virtual machine(s)
ubuntu2

Physical Adapters
No adapters

Port group mapping screenshots for controller and compute



ubuntu3 - Virtual Machine Properties

Hardware | Options | Resources | Profiles | vServices — Virtual Machine Version: 8

Show All Devices — Add... | Remove

Hardware	Summary
Memory	16384 MB
CPUs	2
Video card	Video card
VMCI device	Restricted
SCSI controller 0	LSI Logic Parallel
CD/DVD drive 1	[datastore1] ubuntu-14...
Hard disk 1	Virtual Disk
Floppy drive 1	Client Device
Network adapter 1	mgmt-network
Network adapter 2	public_network

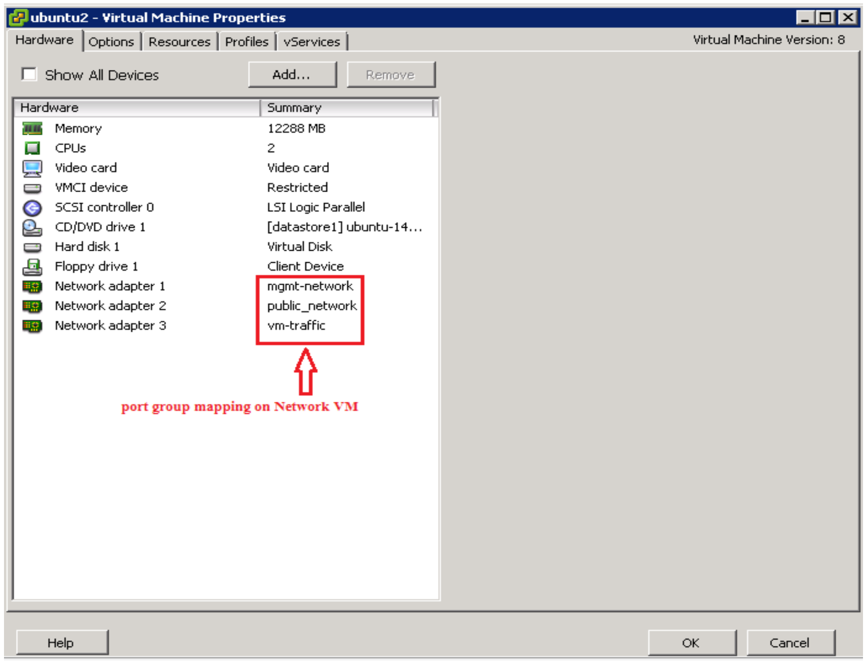
Port group mapping for Controller and compute vm

Memory Configuration

Memory Size: 16 GB

Maximum recommended for this guest OS: 1011 GB.
Maximum recommended for best performance: 98228 MB.
Default recommended for this guest OS: 1 GB.
Minimum recommended for this guest OS: 256 MB.

Port group mapping screenshots for controller and compute


ubuntu2 - Virtual Machine Properties
Hardware | Options | Resources | Profiles | vServices
Virtual Machine Version: 8
Show All Devices
Add...
Remove
Help
OK
Cancel
port group mapping on Network VM

Hardware	Summary
Memory	12288 MB
CPUs	2
Video card	Video card
VMCI device	Restricted
SCSI controller 0	LSI Logic Parallel
CD/DVD drive 1	[datastore1] ubuntu-14...
Hard disk 1	Virtual Disk
Floppy drive 1	Client Device
Network adapter 1	mgmt-network
Network adapter 2	public_network
Network adapter 3	vm-traffic

Ubuntu Installation Instructions

- Once the above blank vms are ready with the port group mapping, begin Ubuntu installation
- Select open-ssh server package on package selection menu
- Proceed with default installation options

Network Topology -

I have assigned the following IPs. User can use any IP pool however corporate / Public network should be available for internet access

Management Network - All openstack components will communicate using this network. In my case management network is 10.0.0.x/24

Corp/ Public Network - This network has internet connectivity. The IPs from this network will used for the following operations

Refer the following IP configuration chart

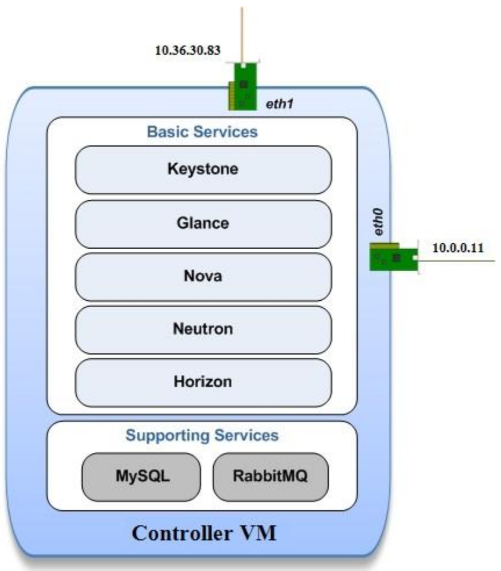
Node Type	Interface	Network Connection	IP address
Controller	Eth0	Management	10.0.0.11 /24
	Eth1	Corporate/ Public	10.36.30.83/16
Network	Eth0	Management	10.0.0.21/24
	Eth1	Corporate/ Public	10.36.30.84/16
	Eth2	VM-traffic	20.0.0.31
Compute1	Eth0	Management	10.0.0.31/24
	Eth1	Corporate/ Public	10.36.30.84/16
	Eth2	VM-traffic	20.0.0.41

Corporate/ Public Network has internet connectivity and it will be used to ssh Ubuntu VMs. Internet connection is also necessary to download OpenStack packages from Ubuntu repository.

Compute node is named as compute1 because it is possible to add more compute nodes as per the requirement. It is also possible to add multiple Network nodes.

Configure Controller Node

Here is a small diagram of basic services (keystone, glance, nova, neutron and horizon) that are installed on the Controller VM and also the supporting services such as MySql database, message broker (RabbitMQ), and NTP.



Let's Start preparing Controller VM

Following tasks will be performed in the following section,

- Upgrade Ubuntu OS along with repository update to get the latest package repos
- Install the NTP daemon
- Install MySQL server and configure
- Install RabbitMQ server
- Install and configure Keystone

Update and Upgrade your System

```
apt-get update -y && apt-get upgrade -y && apt-get dist-upgrade
```

Install NTP service (Network Time Protocol)

```
apt-get install -y ntp
```

Install MySQL

```
apt-get install -y mysql-server python-mysqldb
```

- Set the bind-address key to the management IP address of the controller node

```
vi /etc/mysql/my.cnf
bind-address = 10.0.0.11
```

- Under the [mysqld] section, set the following keys to enable InnoDB, UTF-8 character set, and UTF-8 collation by default

```
vi /etc/mysql/my.cnf
[mysqld]
default-storage-engine = innodb
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
```

- Restart the MySQL service:

```
service mysql restart
```

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

```
mysql_install_db
mysql_secure_installation
```

Component Introduction - RabbitMQ

- OpenStack requires a messaging service for internal communications. Between various components Messaging enables software applications to connect and scale. Applications can connect to each other, as components of a larger application, or to user devices and data. Typically RabbitMQ is used in Openstack environment.
- Default RabbitMQ port is 5672

- Install RabbitMQ (Message Queue)

```
apt-get install -y rabbitmq-server
```

Keystone installation and configuration

Component Introduction - Keystone (Identity service)

- Keystone is a framework for authentication and authorization for all the OpenStack services.
- Keystone handles API requests as well as providing configurable catalog, policy, token and identity services.
- It provides the ability to add users to groups (also known as tenants) and to manage permissions between users and groups. Permissions include the ability to launch and terminate instances.

▪ Install keystone packages

```
apt-get install -y keystone
```

▪ Create a MySQL database for keystone

```
mysql -u root -p

CREATE DATABASE keystone;
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY
'KEYSTONE_DBPASS';
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY
'KEYSTONE_DBPASS';

exit;
```

Note *'localhost' can be replaced by controller management ip 10.0.0.11*

▪ Remove Keystone SQLite database

```
rm /var/lib/keystone/keystone.db
```

▪ Edit /etc/keystone/keystone.conf

```
 vi /etc/keystone/keystone.conf

[database]
replace connection = sqlite:////var/lib/keystone/keystone.db by
connection = mysql://keystone:KEYSTONE_DBPASS@10.0.0.11/keystone

[DEFAULT]
```

```
admin_token=ADMIN
log_dir=/var/log/keystone
```

- Restart the identity service then synchronize the database

```
service keystone restart
keystone-manage db_sync
```

- Check synchronization

```
mysql -u root -p keystone
show TABLES;
```

- Define users, tenants, and roles

```
export OS_SERVICE_TOKEN=ADMIN
export OS_SERVICE_ENDPOINT=http://10.0.0.11:35357/v2.0
```

```
#Create an administrative user
keystone user-create --name=admin --pass=admin_pass --email=admin@domain.com
keystone role-create --name=admin
keystone tenant-create --name=admin --description="Admin Tenant"
keystone user-role-add --user=admin --tenant=admin --role=admin
keystone user-role-add --user=admin --role=_member_ --tenant=admin

#Create a normal user
keystone user-create --name=demo --pass=demo_pass --email=demo@domain.com
keystone tenant-create --name=demo --description="Demo Tenant"
keystone user-role-add --user=demo --role=_member_ --tenant=demo

#Create a service tenant
keystone tenant-create --name=service --description="Service Tenant"
```

- Define services and API endpoints

```
keystone service-create --name=keystone --type=identity --
description="OpenStack Identity"

keystone endpoint-create --service-id(service id which was generated by the above
command) --publicurl=http://10.36.30.83:5000/v2.0
internalurl=http://10.0.0.11:5000/v2.0
--adminurl=http://10.0.0.11:35357/v2.0
```

Note *for example, keystone endpoint-create --service-id15c11a23667e427e91bc31335b45f4bd publicurl=http://10.36.30.83:5000/v2.0 internalurl=http://10.0.0.11:5000/v2.0 --adminurl=http://10.0.0.11:35357/v2.0*

- Create a simple credential file

```
vi creds
#Paste the following:
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_AUTH_URL=http://10.36.30.83:5000/v2.0/
```

```
vi admin_creds
#Paste the following:
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://10.0.0.11:35357/v2.0
```

Test Keystone

- Clear the values in the OS_SERVICE_TOKEN and OS_SERVICE_ENDPOINT environment variables

```
unset OS_SERVICE_TOKEN OS_SERVICE_ENDPOINT
```

- Request a authentication token

```
keystone --os-username=admin --os-password=admin_pass --os-auth-
url=http://10.0.0.11:35357/v2.0 token-get
```

- Load credential admin file

```
source admin_creds

keystone token-get
```

- Load credential file

```
source creds

keystone user-list
keystone user-role-list --user admin --tenant admin
```

Glance installation and configuration

Component Introduction - Glance

- The OpenStack Image Service called Glance provides discovery, registration and delivery services for disk and server images.
- It has the ability to copy (or snapshot) a server image and then to store it promptly. Stored images then can be used as templates to get new servers up and running quickly, and can also be used to store and catalog unlimited backups.
- Virtual-machine images can be stored in various locations, including simple file systems and object-storage systems such as OpenStack Object Storage (code named Swift).
- Glance includes **Glance-api, Glance-registry, Database, Storage repository**

- Install Glance packages

```
    apt-get install -y glance python-glanceclient
```

- Create a MySQL database for Glance

```
mysql -u root -p

CREATE DATABASE glance;

GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY
GLANCE_DBPASS';

GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY
'GLANCE_DBPASS';



exit;
```

- Configure service user and role

```
keystone user-create --name=glance --pass=service_pass --email=glance@domain.com

keystone user-role-add --user=glance --tenant=service --role=admin
```

- Register the service and create the endpoint

```
keystone service-create --name=glance --type=image --description="OpenStack Image Service"

   keystone endpoint-create    --service-id= (service id which was generated by the above command)

   --publicurl=http://10.36.30.83:9292 --internalurl=http://10.0.0.11:9292 --adminurl=http://10.0.0.11:9292
```

Note *for example, keystone endpoint-create --service-id15c11a23667e427e91bc31335b45f4bd --publicurl=http://10.36.30.83:9292 --internalurl=http://10.0.0.11:9292 --adminurl=http://10.0.0.11:9292*

- Update /etc/glance/glance-api.conf

```
  vi /etc/glance/glance-api.conf

    [database]

    replace sqlite_db = /var/lib/glance/glance.sqlite with
```

```
   connection = mysql://glance:GLANCE_DBPASS@10.0.0.11/glance


[DEFAULT]

   rpc_backend = rabbit

   rabbit_host = 10.0.0.11


   [keystone_authtoken]

   auth_uri = http://10.0.0.11:5000

   auth_host = 10.0.0.11

   auth_port = 35357

   auth_protocol = http

   admin_tenant_name = service

   admin_user = glance

   admin_password = service_pass


   [paste_deploy]

   flavor = keystone
```

- Update /etc/glance/glance-registry.conf

```
   vi /etc/glance/glance-registry.conf


   [database]

   replace sqlite_db = /var/lib/glance/glance.sqlite with:

   connection = mysql://glance:GLANCE_DBPASS@10.0.0.11/glance


   [keystone_authtoken]

   auth_uri = http://10.0.0.11:5000

   auth_host = 10.0.0.11

   auth_port = 35357

   auth_protocol = http

   admin_tenant_name = service
```

```
    admin_user = glance

    admin_password = service_pass


    [paste_deploy]

    flavor = keystone
```

- Restart the glance-api and glance-registry services

```
  service glance-api restart; service glance-registry restart
```

- Synchronize the glance database

```
glance-manage db_sync
```

- Test Glance, upload the cirros cloud image

```
source creds

glance image-create --name "cirros-0.3.2-x86_64" --is-public true \

--container-format bare --disk-format qcow2 \

--location http://cdn.download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86_64-
disk.img
```

```
glance image-list
```

Install the compute Service (Nova)

Component Introduction – Nova

- Nova, also known as OpenStack Compute, is the software that controls your Infrastructure as as Service (IaaS) cloud computing platform. It is similar in scope to Amazon EC2 and Rackspace Cloud Servers. Nova 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 API.
- Nova has a concept of Fixed IPs and Floating IPs. Fixed IPs are assigned to an instance on creation and stay the same until the instance is explicitly terminated. Floating ips are ip addresses that can be dynamically associated with an instance. This address can be disassociated and associated with another instance at any time.

- Install nova packages

```
apt-get install -y nova-api nova-cert nova-conductor nova-consoleauth \

nova-novncproxy nova-scheduler python-novaclient
```

- Create a Mysql database for Nova

```
mysql -u root -p



CREATE DATABASE nova;

GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY
'NOVA_DBPASS';

GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY 'NOVA_DBPASS';



exit;
```

- Configure service user and role

```
keystone user-create --name=nova --pass=service_pass --email=nova@domain.com

keystone user-role-add --user=nova --tenant=service --role=admin
```

- Register the service and create the endpoint

```
keystone service-create --name=nova --type=compute --description="OpenStack
Compute"

keystone endpoint-create --service-id= (service id which was generated by the above command) --
publicurl=http://10.36.30.83:8774/v2/%\(tenant_id\)s --
internalurl=http://10.0.0.11:8774/v2/%\(tenant_id\)s --
adminurl=http://10.0.0.11:8774/v2/%\(tenant_id\)s
```

for example, keystone endpoint-create --service-id15c11a23667e427e91bc31335b45f4bd --publicurl=http://10.36.30.83:8774/v2/%\(tenant_id\)s --internalurl=http://10.0.0.11:8774/v2/%\(tenant_id\)s --adminurl=http://10.0.0.11:8774/v2/%\(tenant_id\)s

- Edit the /etc/nova/nova.conf

```
vi /etc/nova/nova.conf


[database]

connection = mysql://nova:NOVA_DBPASS@controller/nova


[DEFAULT]

rpc_backend = rabbit

rabbit_host = controller

my_ip = 10.0.0.11

vncserver_listen = 10.0.0.11

vncserver_proxyclient_address = 10.0.0.11

auth_strategy = keystone


[keystone_authtoken]

auth_uri = http://10.0.0.11:5000

auth_host = 10.0.0.11

auth_port = 35357

auth_protocol = http

admin_tenant_name = service

admin_user = nova

admin_password = service_pass
```

- Remove Nova SQLite database

```
    rm /var/lib/nova/nova.sqlite
```

- Synchronize your database

```
  nova-manage db sync
```

- Restart nova-* services

```
service nova-api restart

service nova-cert restart

service nova-conductor restart

service nova-consoleauth restart

service nova-novncproxy restart

service nova-scheduler restart
```

- nova-manage service list

To verify your configuration, list available images

```
source creds

nova image-list
```

Install Neutron

Component Introduction – Neutron

- Neutron(formerly Quantum) is a system for managing networks and IP addresses. Like other aspects of the cloud operating system, it can be used by administrators and users to increase the value of existing data center assets. Neutron ensures the network will not be the bottleneck or limiting factor in a cloud deployment and gives users real self-service, even over their network configurations.
- Neutron provides networking models for different applications or user groups. Standard models include flat networks or VLANs for separation of servers and traffic. Neutron manages IP addresses, allowing for dedicated static IPs or DHCP. Floating IPs allow traffic to be dynamically re routed to any of your compute resources, which allows you to redirect traffic during maintenance or in the case of failure.
- Users can create their own networks, control traffic and connect servers and devices to one or more networks.

- Install the Neutron server and the OpenVSwitch packages

```
  apt-get install -y neutron-server neutron-plugin-ml2
```

- Create a MySql database for Neutron

```
mysql -u root -p

CREATE DATABASE neutron;

GRANT ALL PRIVILEGES ON neutron.* TO neutron@'localhost' IDENTIFIED BY
'NEUTRON_DBPASS';

GRANT ALL PRIVILEGES ON neutron.* TO neutron@'%' IDENTIFIED BY
'NEUTRON_DBPASS';

exit;
```

- Configure service user and role

```
keystone user-create --name=neutron --pass=service_pass --
email=neutron@domain.com

keystone user-role-add --user=neutron --tenant=service --role=admin
```

- Register the service and create the endpoint

```
keystone service-create --name=neutron --type=network --
description="OpenStack Networking"



keystone endpoint-create --service-id=(service id which was generated by the above command)

--publicurl=http://10.36.30.83:9696 --internalurl=http://10.0.0.11:9696

--adminurl=http://10.0.0.11:9696
```

Note

for example, keystone endpoint-create --service-id15c11a23667e427e91bc31335b45f4bd
```
--publicurl=http://10.36.30.83:9696 --internalurl=http://10.0.0.11:9696 --
adminurl=http://10.0.0.11:9696
```

- Update /etc/neutron/neutron.conf

```
  vi /etc/neutron/neutron.conf



    [database]
    replace connection = sqlite:////var/lib/neutron/neutron.sqlite with

    connection = mysql://neutron:NEUTRON_DBPASS@10.0.0.11/neutron

[DEFAULT]

replace  core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin with

core_plugin = ml2

service_plugins = router

allow_overlapping_ips = True

auth_strategy = keystone

rpc_backend = neutron.openstack.common.rpc.impl_kombu

rabbit_host = 10.0.0.11

notify_nova_on_port_status_changes = True

notify_nova_on_port_data_changes = True

nova_url = http://10.0.0.11:8774/v2

nova_admin_username = nova

# with the output of this command (keystone tenant-list | awk '/ service / {
print $2 }')

nova_admin_tenant_id = (put SERVICE_TENANT_ID here)

#service tenant id can be found with the help of this command “keystone
tenant-list”

nova_admin_password = service_pass

nova_admin_auth_url = http://10.0.0.11:35357/v2.0



[keystone_authtoken]

auth_uri = http://10.0.0.11:5000

auth_host = 10.0.0.11

auth_port = 35357

auth_protocol = http

admin_tenant_name = service

admin_user = neutron

admin_password = service_pass
```

Note *If you get an error message “Expecting an auth URL via either --os-auth-url or env[OS_AUTH_URL”] after typing* keystone tenant-list command, you probably need to make sure that the /cred file is updated correctly.

Following are the cred file entries

export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin_pass
export OS_AUTH_URL="http://192.168.100.51:5000/v2.0/"d type “source creds”

Configure Modular Layer 2 (ML2) plug-in

Component Introduction – ML2 Plugin for Neutron

- Starting with Havana release, openvswitch and linuxbridge plugins are deprecated. Modular Layer 2 (ML2) plugin replaces these plugins. L2 agents work with ML2 plugin and continue to work with the deprecated monolithic plugins.

- Edit ml2_conf.ini

```
vi /etc/neutron/plugins/ml2/ml2_conf.ini

[ml2]

type_drivers = gre

tenant_network_types = gre

mechanism_drivers = openvswitch



[ml2_type_gre]

tunnel_id_ranges = 1:1000



[securitygroup]

firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver

enable_security_group = True
```

- Configure Compute to use Networking

  add in /etc/nova/nova.conf

```
vi /etc/nova/nova.conf


[DEFAULT]

network_api_class=nova.network.neutronv2.api.API

neutron_url=http://10.0.0.11:9696

neutron_auth_strategy=keystone

neutron_admin_tenant_name=service

neutron_admin_username=neutron

neutron_admin_password=service_pass

neutron_admin_auth_url=http://10.0.0.11:35357/v2.0

libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver

linuxnet_interface_driver=nova.network.linux_net.LinuxOVSInterfaceDriver

firewall_driver=nova.virt.firewall.NoopFirewallDriver

security_group_api=neutron
```

- Restart the Compute services

```
service nova-api restart

service nova-scheduler restart

service nova-conductor restart
```

- Restart the Networking service

```
service neutron-server restart
```

Install the dashboard Service (Horizon)

Component Introduction - Horizon

- Horizon is a Django-based project aimed at providing a complete OpenStack Dashboard.
- Horizon is the canonical implementation of OpenStack's Dashboard, which provides a web based user interface to OpenStack services including Nova, Swift etc.

- Install the required packages

```
apt-get install -y apache2 memcached libapache2-mod-wsgi openstack-dashboard
```

- You can remove the openstack-dashboard-ubuntu-theme package

```
apt-get remove -y --purge openstack-dashboard-ubuntu-theme
```

- Edit /etc/openstack-dashboard/local_settings.py

```
vi /etc/openstack-dashboard/local_settings.py
ALLOWED_HOSTS = ['localhost', '10.36.30.83']
OPENSTACK_HOST = "10.0.0.11"
```

- Reload Apache and memcached

```
service apache2 restart; service memcached restart
```

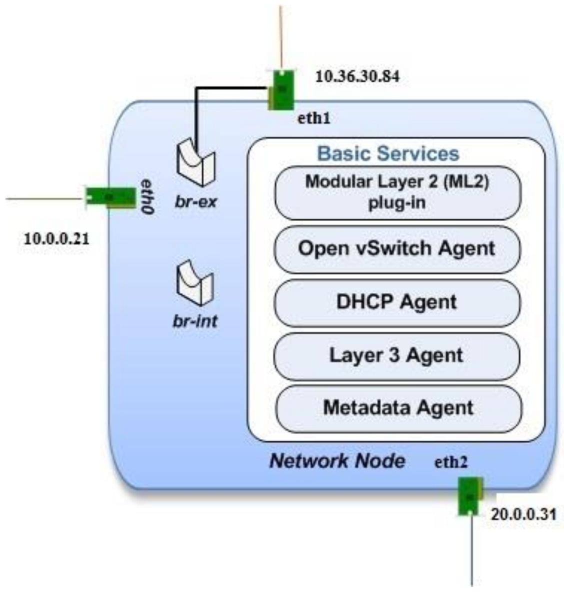
Note *If you have this error: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message"*

Solution: Edit /etc/apache2/apache2.conf

```
vi /etc/apache2/apache2.conf
Add the following new line end of file:
ServerName localhost
```

Reload Apache and memcached

Configure Network Node



Update and Upgrade your System

```
apt-get update -y && apt-get upgrade -y && apt-get dist-upgrade
```

Install NTP service

```
apt-get install -y ntp
```

Set your network node to follow up your conroller node

```
  sed -i 's/server ntp.ubuntu.com/server 10.0.0.11/g' /etc/ntp.conf
```

Restart NTP service

```
  service ntp restart
```

Install other services

```
apt-get install -y vlan bridge-utils
```

Edit /etc/sysctl.conf to contain the following

```
  vi /etc/sysctl.conf
  net.ipv4.ip_forward=1
```

```
    net.ipv4.conf.all.rp_filter=0
    net.ipv4.conf.default.rp_filter=0
```

Implement the changes

```
    sysctl -p
```

Install the Networking components

```
apt-get install -y neutron-plugin-ml2 neutron-plugin-openvswitch-agent dnsmasq
neutron-l3-agent neutron-dhcp-agent
```

Update /etc/neutron/neutron.conf

```
    vi /etc/neutron/neutron.conf

    [DEFAULT]
    auth_strategy = keystone
    rpc_backend = neutron.openstack.common.rpc.impl_kombu
    rabbit_host = 10.0.0.11
    replace  core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin with
    core_plugin = ml2
    service_plugins = router
    allow_overlapping_ips = True

    [keystone_authtoken]
    auth_uri = http://10.0.0.11:5000
    auth_host = 10.0.0.11
    auth_port = 35357
    auth_protocol = http
    admin_tenant_name = service
    admin_user = neutron
  admin_password = service_pass
```

Edit the /etc/neutron/l3_agent.ini

```
  vi /etc/neutron/l3_agent.ini

    [DEFAULT]
    interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
    use_namespaces = True
```

Edit the /etc/neutron/dhcp_agent.ini

```
    vi /etc/neutron/dhcp_agent.ini

    [DEFAULT]
    interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
    dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
    use_namespaces = True
```

Edit the /etc/neutron/metadata_agent.ini

```
    vi /etc/neutron/metadata_agent.ini

    [DEFAULT]
    auth_url = http://10.0.0.11:5000/v2.0
    auth_region = regionOne
    admin_tenant_name = service
    admin_user = neutron
    admin_password = service_pass
    nova_metadata_ip = 10.0.0.11
    metadata_proxy_shared_secret = helloOpenStack
```

SSh to Controller VM and edit nova.conf

```
    vi /etc/nova/nova.conf

    [DEFAULT]
    service_neutron_metadata_proxy = true
    neutron_metadata_proxy_shared_secret = helloOpenStack

    service nova-api restart
```

Now Switch back to Network VM and edit the following file /etc/neutron/plugins/ml2/ml2_conf.ini

```
  vi /etc/neutron/plugins/ml2/ml2_conf.ini

    [ml2]
    type_drivers = gre
    tenant_network_types = gre
    mechanism_drivers = openvswitch

    [ml2_type_gre]
    tunnel_id_ranges = 1:1000

    [ovs]
    local_ip = 20.0.0.31
    tunnel_type = gre
    enable_tunneling = True

    [securitygroup]
    firewall_driver =
neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
    enable_security_group = True
```

Restart openVSwitch

```
    service openvswitch-switch restart
```

Create the bridges:

```
    #br-int will be used for VM integration
    ovs-vsctl add-br br-int
```

```
    #br-ex is used to make to VM accessible from the internet
    ovs-vsctl add-br br-ex
```

Add the eth2 to the br-ex

#Internet connectivity will be lost after this step but this won't affect OpenStack's work

```
    ovs-vsctl add-port br-ex eth2
```

Edit /etc/network/interfaces

```
    vi /etc/network/interfaces

    # The public network interface
    auto eth2
    iface eth2 inet manual
    up ifconfig $IFACE 0.0.0.0 up
    up ip link set $IFACE promisc on
    down ip link set $IFACE promisc off
    down ifconfig $IFACE down

    auto br-ex
    iface br-ex inet static
    address 10.36.30.83
    netmask 255.255.0.0
    gateway 10.36.0.1
    dns-nameservers 8.8.8.8
```

Restart network

```
    ifdown eth2 && ifup eth2

    ifdown br-ex && ifup br-ex
```

Restart all neutron services

```
service neutron-plugin-openvswitch-agent restart
service neutron-dhcp-agent restart
service neutron-l3-agent restart
service neutron-metadata-agent restart
service dnsmasq restart
```

Check status

```
    service neutron-plugin-openvswitch-agent status
    service neutron-dhcp-agent status
    service neutron-l3-agent status
    service neutron-metadata-agent status
    service dnsmasq status
```

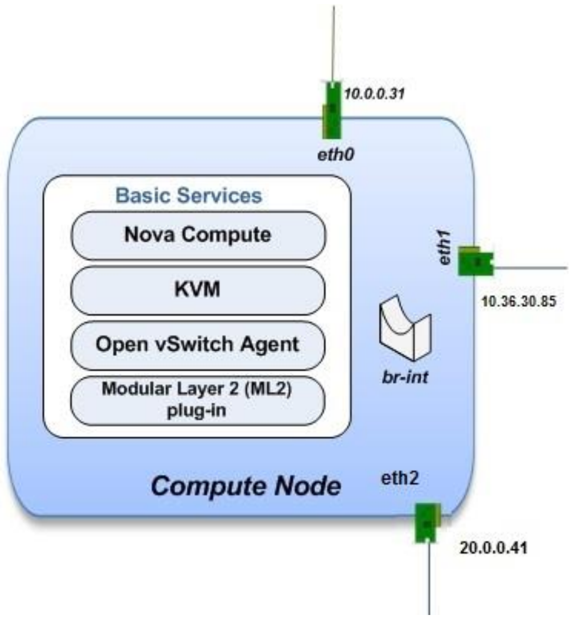
Create a simple credential file

```
    vi creds
    #Paste the following:
    export OS_TENANT_NAME=admin
    export OS_USERNAME=admin
    export OS_PASSWORD=admin_pass
    export OS_AUTH_URL="http://10.36.30.83:5000/v2.0/"
```

Check Neutron agents

```
    source creds
    neutron agent-list
```

Configure Compute Node



- Update and Upgrade your System

```
apt-get update -y && apt-get upgrade -y && apt-get dist-upgrade
```

- Install ntp service

```
apt-get install -y ntp

   Set the compute node to follow up your conroller node:

   sed -i 's/server ntp.ubuntu.com/server 10.0.0.11/g' /etc/ntp.conf
```

- Restart NTP service

```
service ntp restart
```

- Check that your hardware supports virtualization

```
apt-get install -y cpu-checker
   kvm-ok
```

- Install and configure kvm

```
   apt-get install -y kvm libvirt-bin pm-utils
```

- Install the Compute packages

```
apt-get install -y nova-compute-kvm python-guestfs
```

- Make the current kernel readable

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

- Enable this override for all future kernel updates, create the file /etc/kernel/postinst.d/statoverride containing

```
vi /etc/kernel/postinst.d/statoverride
   #!/bin/sh
   version="$1"
   # passing the kernel version is required
   [ -z "${version}" ] && exit 0
   dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-${version}
```

- Make the file executable

```
chmod +x /etc/kernel/postinst.d/statoverride
```

- Modify the /etc/nova/nova.conf like this

```
vi /etc/nova/nova.conf
   [DEFAULT]
   auth_strategy = keystone
   rpc_backend = rabbit
   rabbit_host = 10.0.0.11
   my_ip = 10.0.0.31
   vnc_enabled = True
   vncserver_listen = 0.0.0.0
   vncserver_proxyclient_address = 10.0.0.31
   novncproxy_base_url = http://10.36.30.83:6080/vnc_auto.html
   glance_host = 10.0.0.11
   vif_plugging_is_fatal=false
   vif_plugging_timeout=0

   [database]
   connection = mysql://nova:NOVA_DBPASS@10.0.0.11/nova

   [keystone_authtoken]
   auth_uri = http://10.0.0.11:5000
   auth_host = 10.0.0.11
   auth_port = 35357
   auth_protocol = http
   admin_tenant_name = service
   admin_user = nova
   admin_password = service_pass
```

- Delete /var/lib/nova/nova.sqlite file

```
   rm /var/lib/nova/nova.sqlite
```

- Restart nova-compute services

```
service nova-compute restart
```

- Edit /etc/sysctl.conf to contain the following

```
 vi /etc/sysctl.conf
   net.ipv4.ip_forward=1
   net.ipv4.conf.all.rp_filter=0
   net.ipv4.conf.default.rp_filter=0
```

- Implement the changes

```
   sysctl -p
```

- Install the Networking components

```
 apt-get install -y neutron-common neutron-plugin-ml2 neutron-plugin-
openvswitch-agent
```

- Update /etc/neutron/neutron.conf

```
 vi /etc/neutron/neutron.conf

   [DEFAULT]
   auth_strategy = keystone
   replace  core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin with
   core_plugin = ml2
   service_plugins = router
   allow_overlapping_ips = True
```

```
   rpc_backend = neutron.openstack.common.rpc.impl_kombu
   rabbit_host = 10.0.0.11

   [keystone_authtoken]
   auth_uri = http://10.0.0.11:5000
   auth_host = 10.0.0.11
   auth_port = 35357
   auth_protocol = http
   admin_tenant_name = service
   admin_user = neutron
   admin_password = service_pass
```

- Configure the Modular Layer 2 (ML2) plug-in

```
 vi /etc/neutron/plugins/ml2/ml2_conf.ini

   [ml2]
   type_drivers = gre
   tenant_network_types = gre
   mechanism_drivers = openvswitch

   [ml2_type_gre]
   tunnel_id_ranges = 1:1000

   [ovs]
   local_ip = 20.0.0.41
   tunnel_type = gre
   enable_tunneling = True

   [securitygroup]
   firewall_driver =
neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
   enable_security_group = True
```

- Restart the OVS service

```
 service openvswitch-switch restart
```

- Create the bridges

```
 #br-int will be used for VM integration
    ovs-vsctl add-br br-int
```

- Edit /etc/nova/nova.conf

```
 vi /etc/nova/nova.conf

    [DEFAULT]
    network_api_class = nova.network.neutronv2.api.API
    neutron_url = http://10.0.0.11:9696
    neutron_auth_strategy = keystone
    neutron_admin_tenant_name = service
    neutron_admin_username = neutron
    neutron_admin_password = service_pass
    neutron_admin_auth_url = http://10.0.0.11:35357/v2.0
    linuxnet_interface_driver =
nova.network.linux_net.LinuxOVSInterfaceDriver
    firewall_driver = nova.virt.firewall.NoopFirewallDriver
    security_group_api = neutron

    Edit /etc/nova/nova-compute.conf with the correct hypervisor type (set to
qemu if using virtualbox for example, kvm is default)

    vi /etc/nova/nova-compute.conf

    [DEFAULT]
    compute_driver=libvirt.LibvirtDriver
    [libvirt]
    virt_type=qemu
```

- Restart nova-compute services

```
 service nova-compute restart
```

- Restart the Open vSwitch (OVS) agent

```
 service neutron-plugin-openvswitch-agent restart
```

- Check whether Nova is running

```
 nova-manage service list
```

That’s It ! Now your Openstack Cloud is UP and running.