Deploying Pivotal Cloud Foundry in Openstack Kilo – Part 1 of 3

This is the first part in a three part series s where I will share my experience installing Pivotal Cloud Foundry (PCF) on an open source version of an Openstack Kilo release. I have set up a rack with the following specifications:

- Server 1: Dell Poweredge 2950 III, 2X3.0GHz Quad Core, 32GB, 400GB 10K SAS and 1TB SATA
- Server 2: HP Proliant DL580 G5, 4X2.4GHz Six Core, 64 GB, 5X146GB 10K SAS
- Router 1: Netgear WNR2000 v2 Wireless-N Router
- Router 2: Comp USA wireless router

In this first blog post, I will explain:

1. Basic server configuration
2. Basic network configuration
3. Installation and configuration of the following Openstack services:
    a. Identity Service - Keystone
    b. Image Service - Glance
    c. Compute Service - Nova

Planning to Deploy PCF

Hardware Configuration

Server 1: ESXi Server 6.0. Controller and Network nodes are installed as guest VM in ESXi server.
Server 2: serves as compute and cinder nodes.

I have used Ubuntu 14.04 as base Operating System. The install procedure for installing openstack kilo is exhaustive yet impressive documentation. In the following sections, there are the documented steps that are applicable for my installation.

Controller Node Specifications

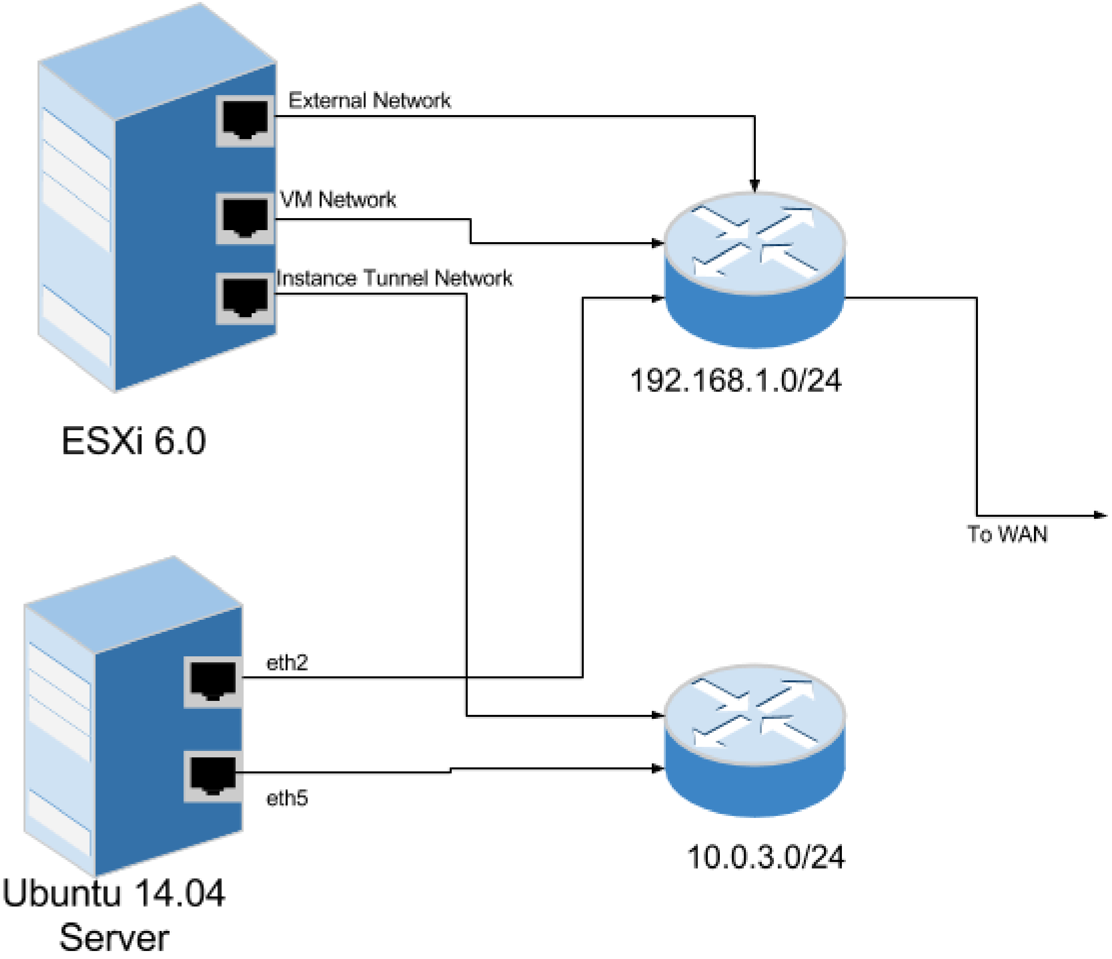
Virtual CPUs: 2
RAM: 2 GB
Hard Drive: 60 GB
No. cf Network Interfaces: 1

Network Node Specifications

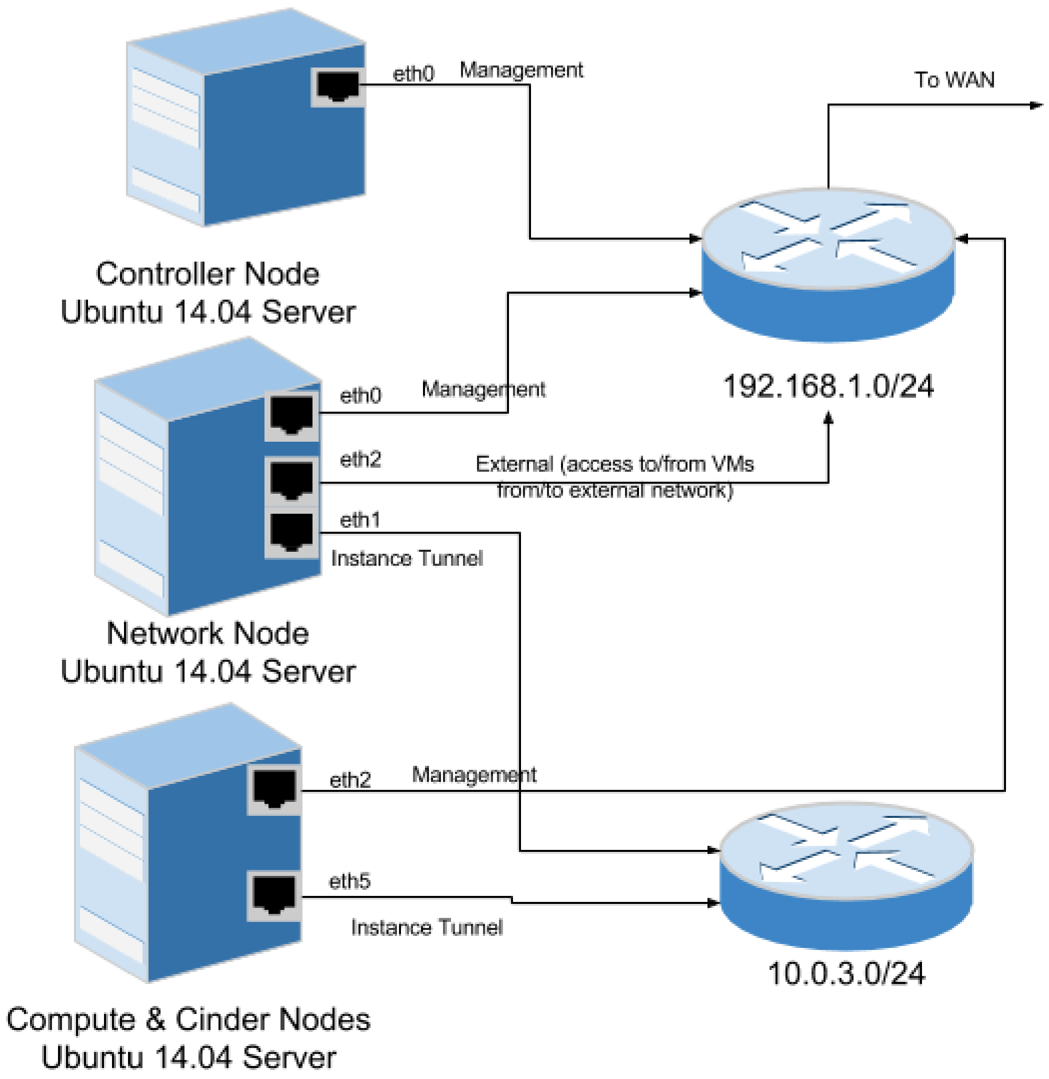
Virtual CPUs: 1
RAM: 1 GB
Hard Drive: 10 GB
No. of Network Interfaces: 3

For both controller and network nodes, I configured 2 GB for SWAP, 2 GB for boot partition and the rest for root partition.

Network Wiring - Hard Wired



Physical Network Architecture



CREDENTIALS

Openstack installation is pretty involved. It is advised to track the passwords for easy reference.

User	Password	Description
root	dbPwd	MySQL database
openstack	rabbitPwd	Rabbit MQ Messaging system
keystone	keystoneDbPwd	database password for keystone database
glance	glanceDbPwd	database password for glance database

nova	novaDbPwd	database password for nova database
neutron	neutronDbPwd	database password for neutron database
cinder	cinderDbPwd	database password for cinder database
admin	adminPwd	admin user
glance	glancePwd	glance user
nova	novaPwd	nova user
neutron	neutronPwd	neutron user
cinder	cinderPwd	cinder user
demo	dempPwd	demo user for verifying operations

Prepare Servers to Install OpenStack Kilo

Installing the Ubuntu Server 14.04 is pretty straight forward. I generally allocate 2 GB for boot partition, one or two times the RAM for swap and the rest for root partition. For compute node, which has 64 GB RAM, I e allocated only 16 GB swap. Select to install OpenSSH server, which is not selected by default, during installation process.

1. **Configure Network Interfaces**

   Edit change the /etc/network/interfaces file to configure static ip addresses as follows.

   Controller Node

   ```
   auto eth0
   iface eth0 inet static
   address 192.168.1.8
   netmask 255.255.255.0
   gateway 192.168.1.1
   ```

   Network Node

   ```
   auto eth0
   iface eth0 inet static
   address 192.168.1.9
   network 192.168.1.0
   netmask 255.255.255.0
   gateway 192.168.1.1

   auto eth1
   iface eth1 inet static
   address 10.0.3.4
   ```

```
network 10.0.3.0
netmask 255.255.255.0

auto eth2
iface eth2 inet manual
       up ip link set dev $IFACE up
       down ip link set dev $IFACE down
```

Compute Node

```
auto eth2
iface eth2 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1

auto eth5
iface eth5 inet static
address 10.0.3.5
network 10.0.3.0
netmask 255.255.255.0
```

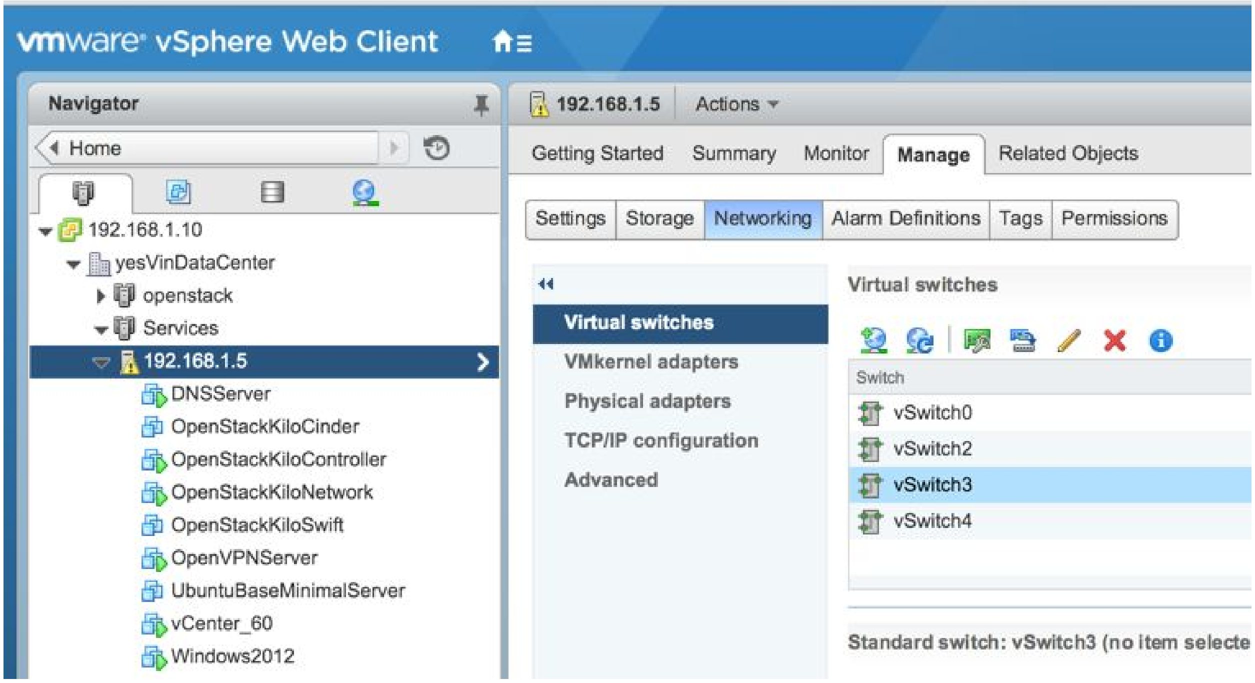
2. Configure hosts

Edit change the /etc/hosts file in all the nodes as follows

```
192.168.1.8    controller
192.168.1.9    network
192.168.1.2    compute1
```

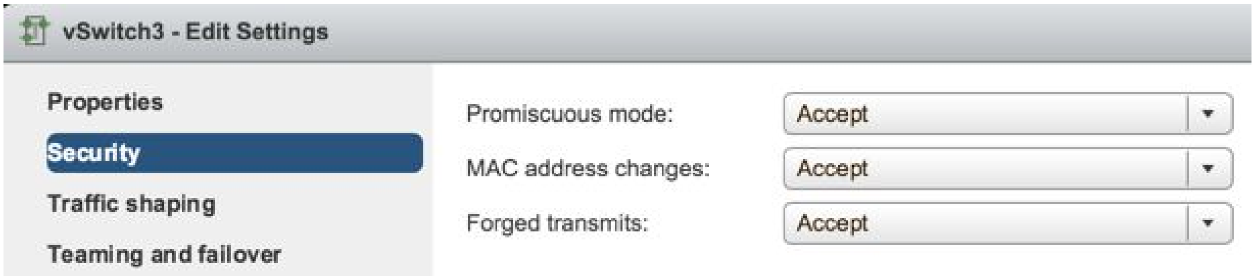
3. Enable promiscuous mode for External Network

a. Login to vCenter
b. Select the ESXi server host
c. Select Manage tab
d. Select network tab
e. Select the standard switch corresponding to external interface (eth2 in our case)



f. Click Edit and select security

g. Select Promicuous mode to Accept as shown below



Promiscuous mode: Accept

MAC address changes: Accept

Forged transmits: Accept

4. Configure NTP

Controller Server:

```
# apt-get install -y ntp
# vi /etc/ntp.conf
        at least on line that starts with server to get time from network
# service ntp restart
```

Other Servers:

```
# apt-get install -y ntp
# vi /etc/ntp.conf
        # add following line and disable all the lines that start with server
        server controller iburst
# service ntp restart
```

5. Add Openstack Kilo Repository to the source list

Do the following in all the servers

```
# apt-get install ubuntu-cloud-keyring
# echo "deb http://ubuntu-cloud.archive.canonical.com/ubuntu" \
"trusty-updates/kilo main" > /etc/apt/sources.list.d/cloudarchive-kilo.list
```

6. **Update the packages list and reboot the servers**
   Do the following in all the servers

```
# apt-get update
# apt-get upgrade
# reboot
```

Now that we have prepared the servers, they are ready for the installation of Openstack.

Install and Configure Controller Server

The installation procedure for all Openstack services typically has the following two steps:

1. Install and configure the management components in controller server; and
2. Install and configure the core components in appropriate servers, if they are not already installed in controller server.

Installation in controller server typically follows following pattern:

- Install DB components (database, user etc.)
- Install Openstack service modules
- Change the configuration files
- Populate database
- Create endpoints for the service
- Restart the openstack services for the service.

Below is how to install and configure of following components in controller server:

-

1. **Database - MySQL**
   Openstack uses a SQL database to store configuration and runtime information. Schemas are provided as part of Openstack packages. While other database options are available, I installed a MySQL database in controller node demonstration. Below is how to install, configure and secure MySQL.

```
# apt-get install -y mariadb-server python-mysqldb
       <Enter password for root user: dbPwd>
# vi /etc/mysql/conf.d/mysqld_openstack.cnf
       [mysqld]
       ...
       bind-address = 192.168.1.8
       default-storage-engine = innodb
       innodb_file_per_table
       collation-server = utf8_general_ci
       init-connect = 'SET NAMES utf8'
       character-set-server = utf8
# service mysql restart
# mysql_secure_installation
```

2. **Messaging Service - RabbitMQ**

Openstack uses message service like RabbitMQ or Qpid to communicate between all its services services. The following installs the RabbitMQ service:

```
# apt-get install -y rabbitmq-server
# rabbitmqctl add_user openstack rabbitPwd
# rabbitmqctl set_permissions openstack ".*" ".*" ".*"
```

3. Install and Configure Identity Service - Keystone

Keystone is used to provision users and their access levels. It also provisions the list of available Openstack services and their API endpoints. Keystone handles all of the authentication and authorization requests for all Openstack services. The following scripts will install and configure Keystone:

```
$ mysql -u root -p
       MariaDB [(none)]> CREATE DATABASE keystone;
       MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO
       'keystone'@'localhost' IDENTIFIED BY 'keystoneDbPwd';
       MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%'
       IDENTIFIED BY 'keystoneDbPwd';
       MariaDB [(none)]> exit
# echo "manual" > /etc/init/keystone.override
# apt-get install -y keystone python-openstackclient apache2 libapache2-mod-wsgi
memcached python-memcache
# openssl rand -hex 10
       9aede9647725ce0aac20
# vi /etc/keystone/keystone.conf
       [DEFAULT]
       ...
       admin_token = 9aede9647725ce0aac20
       verbose = True

       [database]
       ...
       connection = mysql://keystone:keystoneDbPwd@controller/keystone

       [memcache]
       ...
       servers = localhost:11211

       [token]
       ...
       provider = keystone.token.providers.uuid.Provider
       driver = keystone.token.persistence.backends.memcache.Token

       [revoke]
       ...
       driver = keystone.contrib.revoke.backends.sql.Revoke
# su -s /bin/sh -c "keystone-manage db_sync" keystone
# vi /etc/apache2/apache2.conf
       ServerName  controller
# vi /etc/apache2/sites-available/wsgi-keystone.conf
```

```
        Note: This is new file
        Listen 5000
        Listen 35357

        <VirtualHost *:5000>
           WSGIDaemonProcess keystone-public processes=5 threads=1 user=keystone
        display-name=%{GROUP}
           WSGIProcessGroup keystone-public
           WSGIScriptAlias / /var/www/cgi-bin/keystone/main
           WSGIApplicationGroup %{GLOBAL}
           WSGIPassAuthorization On
           <IfVersion >= 2.4>
             ErrorLogFormat "%{cu}t %M"
           </IfVersion>
           LogLevel info
           ErrorLog /var/log/apache2/keystone-error.log
           CustomLog /var/log/apache2/keystone-access.log combined
        </VirtualHost>

        <VirtualHost *:35357>
           WSGIDaemonProcess keystone-admin processes=5 threads=1 user=keystone
        display-name=%{GROUP}
           WSGIProcessGroup keystone-admin
           WSGIScriptAlias / /var/www/cgi-bin/keystone/admin
           WSGIApplicationGroup %{GLOBAL}
           WSGIPassAuthorization On
           <IfVersion >= 2.4>
             ErrorLogFormat "%{cu}t %M"
           </IfVersion>
           LogLevel info
           ErrorLog /var/log/apache2/keystone-error.log
           CustomLog /var/log/apache2/keystone-access.log combined
        </VirtualHost>
# ln -s /etc/apache2/sites-available/wsgi-keystone.conf
/etc/apache2/sites-enabled
# mkdir -p /var/www/cgi-bin/keystone
# curl
http://git.openstack.org/cgit/openstack/keystone/plain/httpd/keystone.py?h=stabl
e/kilo | tee /var/www/cgi-bin/keystone/main /var/www/cgi-bin/keystone/admin
# chown -R keystone:keystone /var/www/cgi-bin/keystone
# service apache2 restart
# rm -f /var/lib/keystone/keystone.db
$ export OS_TOKEN=9aede9647725ce0aac20
$ export OS_URL=http://controller:35357/v2.0
$ openstack service create --name keystone --description "OpenStack Identity"
identity
$ openstack endpoint create \
        --publicurl http://controller:5000/v2.0 \
        --internalurl http://controller:5000/v2.0 \
        --adminurl http://controller:35357/v2.0 \
        --region RegionOne \
        identity
$ openstack project create --description "Admin Project" admin
$ openstack user create --password-prompt admin
       Password: adminPwd
$ openstack role create admin
$ openstack role add --project admin --user admin admin
$ openstack project create --description "Service Project" service
$ openstack project create --description "Demo Project" demo
$ openstack user create --password-prompt demo
       Password: demoPwd
$ openstack role create user
$ openstack role add --project demo --user demo user
$ vi admin-openrc.sh
       export OS_PROJECT_DOMAIN_ID=default
       export OS_USER_DOMAIN_ID=default
       export OS_PROJECT_NAME=admin
       export OS_TENANT_NAME=admin
       export OS_USERNAME=admin
       export OS_PASSWORD=adminPwd
       export OS_AUTH_URL=http://controller:35357/v3
$ vi demo-openrc.sh
       export OS_PROJECT_DOMAIN_ID=default
       export OS_USER_DOMAIN_ID=default
       export OS_PROJECT_NAME=demo
       export OS_TENANT_NAME=demo
       export OS_USERNAME=demo
       export OS_PASSWORD=demoPwd
       export OS_AUTH_URL=http://controller:5000/v3

$ source admin-openrc.sh
$ openstack token issue
```

4. **Install and Configure Image Service - Glance** The image service provisions the Openstack images to compute nodes when virtual machines are created. It also stores the metadata associated with images, like minimum configuration for the virtual machines based on the image, OS type etc. The following scripts configure image service for Openstack:

```
$ mysql -u root -p
       MariaDB [(none)]> CREATE DATABASE glance;
       MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO
       'glance'@'localhost' IDENTIFIED BY 'glanceDbPwd';
       MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%'
       IDENTIFIED BY 'glanceDbPwd';
       MariaDB [(none)]> exit
$ source admin-openrc.sh
$ openstack user create --password-prompt glance
       Password:glancePwd
$ openstack role add --project service --user glance admin
$ openstack service create --name glance --description "OpenStack Image service"
image
$ openstack endpoint create \
               --publicurl http://controller:9292 \
               --internalurl http://controller:9292 \
               --adminurl http://controller:9292 \
               --region RegionOne \
               image
# apt-get install -y glance python-glanceclient
# vi /etc/glance/glance-api.conf
       [DEFAULT]
       …
       notification_driver = noop
       verbose = True

       [database]
       …
       connection = mysql://glance:glanceDbPwd@controller/glance

       [keystone_authtoken]
       …
       revocation_cache_time = 10
       auth_uri = http://controller:5000
       auth_url = http://controller:35357
       auth_plugin = password
       project_domain_id = default
       user_domain_id = default
       project_name = service
       username = glance
       password = glancePwd

       [paste_deploy]
       …
       flavor = keystone

       [glance_store]
       …
       default_store = file
       filesystem_store_datadir = /var/lib/glance/images/

# vi /etc/glance/glance-registry.conf
       [DEFAULT]
       …
       notification_driver = noop
       verbose = True

       [database]
       …
       connection = mysql://glance:glanceDbPwd@controller/glance
```

```
    [keystone_authtoken]
    ...
    auth_uri = http://controller:5000
    auth_url = http://controller:35357
    auth_plugin = password
    project_domain_id = default
    user_domain_id = default
    project_name = service
    username = glance
    password = glancePwd

    [paste_deploy]
    ...
    flavor = keystone
# su -s /bin/sh -c "glance-manage db_sync" glance
# for a in glance-registry glance-api; do service $a restart; done
# rm -f /var/lib/glance/glance.sqlite
$ echo "export OS_IMAGE_API_VERSION=2" | tee -a admin-openrc.sh demo-openrc.sh
```

5. **Install and Configure Compute Services (Management/Configuration Components Only) - Nova**

   Compute services are the core of any IaaS layer, IMO. All other services are setup with one thing in mind - to efficiently provision and manage virtual machines. Openstack compute services interact with identity, image, network and storage services to provision virtual machines. Compute services in Openstack are a collection of several modules. For this installation, all the services except nova-compute (which is responsible for creating and deleting the virtual machine through hypervisor API calls) runs on controller node. The following scripts install and configure nova management modules:

```
$ mysql -u root -p
    MariaDB [(none)]> CREATE DATABASE nova;
    MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost'
    IDENTIFIED BY 'novaDbPwd';
    MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%'
    IDENTIFIED BY 'novaDbPwd';
    MariaDB [(none)]> exit
$ source admin-openrc.sh
$ openstack user create --password-prompt nova
    <Enter novaPwd for password>
$ openstack role add --project service --user nova admin
$ openstack service create --name nova --description "OpenStack Compute" compute
$ openstack endpoint create \
    --publicurl http://controller:8774/v2/%\(tenant_id\)s \
    --internalurl http://controller:8774/v2/%\(tenant_id\)s \
    --adminurl http://controller:8774/v2/%\(tenant_id\)s \
    --region RegionOne \
    compute
# apt-get install -y nova-api nova-cert nova-conductor nova-consoleauth
nova-novncproxy nova-scheduler python-novaclient
```

```
# vi /etc/nova/nova.conf
       [DEFAULT]
       …
       rpc_backend = rabbit
       auth_strategy = keystone
       my_ip = 192.168.1.8
       vncserver_listen = 192.168.1.8
       vncserver_proxyclient_address = 192.168.1.8

       [database]
       …
       connection = mysql://nova:novaDbPwd@controller/nova

       [oslo_messaging_rabbit]
       …
       rabbit_host = controller
       rabbit_userid = openstack
       rabbit_password = rabbitPwd

       [keystone_authtoken]
       …
       auth_uri = http://controller:5000
       auth_url = http://controller:35357
       auth_plugin = password
       project_domain_id = default
       user_domain_id = default
       project_name = service
       username = nova
       password = novaPwd

       [glance]
       …
       host = controller

       [oslo_concurrency]
       …
       lock_path = /var/lib/nova/tmp
# su -s /bin/sh -c "nova-manage db sync" nova
# for a in nova-api nova-cert nova-consoleauth nova-scheduler nova-conductor
nova-novncproxy; do service $a restart; done
# rm -f /var/lib/nova/nova.sqlite
```

Install and Configure Compute Server

The compute server is the beefiest server in Openstack deployment. It receives the request for creating or deleting virtual machines, interacts with hypervisor API and fulfills the requests. I will use libvirt for KVM (default for openstack) as our hypervisor:

```
# apt-get install -y nova-compute sysfsutils


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

```
       [DEFAULT]
       rpc_backend = rabbit
       auth_strategy = keystone
       my_ip = 192.168.1.2
       vnc_enabled = True
       vncserver_listen = 0.0.0.0
       vncserver_proxyclient_address = 192.168.1.2
       novncproxy_base_url = http://controller:6080/vnc_auto.html

       [database]
       …
       connection = mysql://nova:novaDbPwd@controller/nova

       [oslo_messaging_rabbit]
       rabbit_host = controller
       rabbit_userid = openstack
       rabbit_password = rabbitPwd

       [keystone_authtoken]
       auth_uri = http://controller:5000
       auth_url = http://controller:35357
       auth_plugin = password
       project_domain_id = default
       user_domain_id = default
       project_name = service
       username = nova
       password = novaPwd

       [glance]
       host = controller

       [oslo_concurrency]
       lock_path = /var/lib/nova/tmp
# service nova-compute restart
# rm -f /var/lib/nova/nova.sqlite
```

Verify the configuration

Verify the identity service with the following:

```
$ source admin-openrc.sh
//check to see if keystone can issue tokens
$ openstack token issue
       +------------+----------------------------------+
       | Field      | Value                            |
       +------------+----------------------------------+
        | expires    | 2015-12-02T01:36:47.917274Z      |
        | id         | 62cfb210b5c74b50af1c18013d4a712c |
        | project_id | 9ae23c9777b444799424d4473052394a |
        | user_id    | 4bb4e8d2863342a089f0fb7b3969db91 |
        +------------+----------------------------------+
//check list of installed openstack services
$ openstack service list
        +----------------------------------+----------+----------+
        | ID                               | Name     | Type     |
        +----------------------------------+----------+----------+
        | 01fb9ffdf46a4164b053ab7e6d51fd3c | keystone | identity |
        | 6c1962782cf944139723773644282b7b | glance   | image    |
        | bc8d697735c44783bdf62dc08de88702 | nova     | compute  |
        +----------------------------------+----------+----------+
//Verify keystone service is enabled
$ openstack service show keystone
        +-------------+----------------------------------+
        | Field       | Value                            |
        +-------------+----------------------------------+
        | description | OpenStack Identity               |
        | enabled     | True                             |
        | id          | 01fb9ffdf46a4164b053ab7e6d51fd3c |
        | name        | keystone                         |
        | type        | identity                         |
        +-------------+----------------------------------+

//verify that endpoints for identity, image and compute services are present
$ openstack endpoint list
        +----------------------------------+-----------+--------------+--------------+
        | ID                               | Region    | Service Name | Service Type |
        +----------------------------------+-----------+--------------+--------------+
        | 6dd2c110b46b4599832f3a7041e77fa5 | RegionOne | glance       | image        |
        | bcefb4de1fb84057938115b2b2458a91 | RegionOne | keystone     | identity     |
        | fba191a9e27840d0bdaf9d4b4a40f350 | RegionOne | nova         | compute      |
        +----------------------------------+-----------+--------------+--------------+
//verify keystone endpoint is enabled
$ openstack endpoint show keystone
        +--------------+----------------------------------+
        | Field        | Value                            |
        +--------------+----------------------------------+
        | adminurl     | http://controller:35357/v2.0     |
        | enabled      | True                             |
        | id           | bcefb4de1fb84057938115b2b2458a91 |
        | internalurl  | http://controller:5000/v2.0      |
        | publicurl    | http://controller:5000/v2.0      |
        | region       | RegionOne                        |
        | service_id   | 01fb9ffdf46a4164b053ab7e6d51fd3c |
        | service_name | keystone                         |
        | service_type | identity                         |
        +--------------+----------------------------------+
//list users
$ openstack user list
        +----------------------------------+---------+
        | ID                               | Name    |
        +----------------------------------+---------+
        | 00a48ee2f54b48f2bfacd4b597e4f581 | glance  |
        | 20e5fc8bf6d74893983b2f3f1698c216 | nova    |
        | 4bb4e8d2863342a089f0fb7b3969db91 | admin   |
        | 9af96d0953a34b8cb33e441b5c099e69 | demo    |
        +----------------------------------+---------+
//list projects/tenants
$ openstack project list
        +----------------------------------+------------+
        | ID                               | Name       |
        +----------------------------------+------------+
        | 1c1d5850adff482581e4ae41ac44ebca | service    |
        | 46519d6186e24cc7bdaf106b764a0dcb | demo       |
        | 9ae23c9777b444799424d4473052394a | admin      |
        +----------------------------------+------------+
//list roles
$ openstack role list
        +----------------------------------+-------+
        | ID                               | Name  |
        +----------------------------------+-------+
        | adff327dd29440e7be28f3f3b3383971 | admin |
        | e060a97ef2664f40afc3771db107ea29 | user  |
        +----------------------------------+-------+
```

Verify the image service with the following:

```
$ source admin-openrc.sh
//verify image service is enabled
$ openstack service show image
        +-------------+----------------------------------+
        | Field       | Value                            |
        +-------------+----------------------------------+
        | description | OpenStack Image service          |
        | enabled     | True                             |
        | id          | 6c1962782cf94413972377364428 2b7b |
        | name        | glance                           |
        | type        | image                            |
        +-------------+----------------------------------+
//verify image endpoint is enabled
$ openstack endpoint show image
        +--------------+----------------------------------+
        | Field        | Value                            |
        +--------------+----------------------------------+
        | adminurl     | http://controller:9292           |
        | enabled      | True                             |
        | id           | 6dd2c110b46b4599832f3a7041e77fa5 |
        | internalurl  | http://controller:9292           |
    | publicurl    | http://controller:9292           |
    | region       | RegionOne                        |
    | service_id   | 6c1962782cf944139723773644282b7b |
    | service_name | glance                           |
    | service_type | image                            |
    +--------------+----------------------------------+
$ mkdir /tmp/images
$ wget -P /tmp/images
http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img


// Create cirros image
$ glance image-create --name "cirros-0.3.4-x86_64" --file
/tmp/images/cirros-0.3.4-x86_64-disk.img \
  --disk-format qcow2 --container-format bare --is-public true --progress
//list the image
$ glance image-list
+--------------------------------------+--------------------------------------------+
| ID                                   | Name                                       |
+--------------------------------------+--------------------------------------------+
| 1a79ee5c-f255-445b-bec5-1716dbf9d567 | cirros-0.3.4-x86_64                        |
+--------------------------------------+--------------------------------------------+
$ rm -r /tmp/images
```

Verify the compute service with the following:

```
$ source admin-openrc.sh
//verify image service is enabled
$ openstack service show compute
    +-------------+----------------------------------+
    | Field       | Value                            |
    +-------------+----------------------------------+
    | description | OpenStack Compute                |
    | enabled     | True                             |
    | id          | bc8d697735c44783bdf62dc08de88702 |
    | name        | nova                             |
    | type        | compute                          |
    +-------------+----------------------------------+
//verify image endpoint is enabled
$ openstack endpoint show compute
    +--------------+-----------------------------------------+
    | Field        | Value                                   |
    +--------------+-----------------------------------------+
    | adminurl     | http://controller:8774/v2/%(tenant_id)s |
    | enabled      | True                                    |
    | id           | fba191a9e27840d0bdaf9d4b4a40f350        |
    | internalurl  | http://controller:8774/v2/%(tenant_id)s |
    | publicurl    | http://controller:8774/v2/%(tenant_id)s |
    | region       | RegionOne                             |
    | service_id   | bc8d697735c44783bdf62dc08de88702      |
    | service_name | nova                                  |
    | service_type | compute                               |
    +--------------+---------------------------------------+
//list running nova services in all the hosts
$ nova service-list
    +----+------------------+------------+----------+----------+-------+----------------------------+-----------------+
    | Id | Binary           | Host       | Zone     | Status   | State | Updated_at                 | Disabled Reason |
    +----+------------------+------------+----------+----------+-------+----------------------------+-----------------+
    | 1  | nova-cert        | controller | internal | enabled  | up    | 2015-12-02T01:10:33.000000 | -               |
    | 2  | nova-consoleauth | controller | internal | enabled  | up    | 2015-12-02T01:10:33.000000 | -               |
    | 3  | nova-scheduler   | controller | internal | enabled  | up    | 2015-12-02T01:10:34.000000 | -               |
    | 4  | nova-conductor   | controller | internal | enabled  | up    | 2015-12-02T01:10:33.000000 | -               |
    | 7  | nova-compute     | compute1   | nova     | enabled  | up    | 2015-12-02T01:10:32.000000 | -               |
    +----+------------------+------------+----------+----------+-------+----------------------------+-----------------+
//list images
$ nova image-list
    | ID                                   | Name                                      | Status | Server |
    +--------------------------------------+-------------------------------------------+--------+--------+
    | 1a79ee5c-f255-445b-bec5-1716dbf9d567 | cirros-0.3.4-x86_64                       | ACTIVE |        |
    +--------------------------------------+-------------------------------------------+--------+--------+
```

What's Next? The Networking Service

There is one more important component that needs to be configured before we can call Openstack installation minimally operational and create virtual machines in compute server. That is the networking service, which comes in two flavors:

1. **Nova network or legacy networking -** which basically provides flat address space for all virtual machines.
2. **Neutron Networking -** provides enterprise-class networking capabilities.

Remember that our goal is to install PCF on top of Openstack. PCF (and open source cloud foundry) uses neutron networking APIs in its CPI. The Cloud Foundry CPI also relies on Openstack storage services. We will also need the Openstack dashboard to prepare our Openstack to install PCF. So the next step to prepare is to install and configure the network service (Neutron), the block storage service (Cinder) and the dashboard (Horizon), which will be topic for the next post in the series