Install OpenStack on KVM-based virtual machines of Ubuntu 16.04 LTS server

Created by: Alvin Pei-Yan, Li

Abstract:
This installation will go through the OpenStack Identity, Image, Dashboard, and Compute services that can be successfully deployed across the nodes of the cloud on KVM-based virtual machines. At the end of this document, you should be able to launch the dashboard supported by Horizon by a web browser to navigate the installed and configured services. OpenStack Networking services (Neutron) can be installed further after you follow this document properly.

The host machine is supposed to be Ubuntu 16.04 LTS desktop.

<u>#: on root</u>

On host machine:

```
# apt update; apt upgrade -y
# apt -y install qemu-kvm libvirt-bin virtinst bridge-utils cpu-checker
# apt -y install virt-manager
# apt -y install virt-manager ssh-askpass-gnome —no-install-recommends
# apt -y install libguestfs-tools virt-top
# virt-manager
# wget http://releases.ubuntu.com/16.04.6/ubuntu-16.04.6-server-amd64.iso
```

Setup 1 bridge network for virtual machine on kvm:

On the graphic interface of Virtual Machine Manager, Edit
→ Connection Details
→ Network Interfaces
→ select "+" botton on the bottom-left corner
→ Interface type: Bridge
→ Name: br1, Start mode: onboot, Activate now: yes,
Choose interface(s) to bridge: enp3s0, IP settings: 192.168.40.23/24 (according to the enp3s0 IP address on the host machine that can be searched by the command: ip a) with gateway: 192.168.0.1 *or either* IPv4 DHCP
→ ok
→ finish.

Setup 3 virtual network for virtual machine on kvm:

On the graphic interface of Virtual Machine Manager, Edit
→ Connection Details
→ Virtual Networks
→ select "+" botton on the bottom-left corner
→ Network Name: openstack-net0 or openstack-net1 or openstack-net2
→ Enable IPv4 network address space definition: yes,
Network: 192.168.100.0/24 (for openstack-net0) or 192.168.110.0/24 (for openstack-net1) or 192.168.120.0/24 (for openstack-net2), Enable DHCPv4: yes
→ Choose IPv6 address space for the virtual network: no
→ Connected to a physical network: Isolated virtual network
→ finish

During the installation of ubuntu server on a created virtual machine, please be aware of the following steps:

1) Partitioning method: <u>Guided – use entire disk and set up LVM</u>
2) Write the changes to disks and configure LVM ? <u>Yes</u>
3) Write the changes to disks ? <u>Yes</u>
4) HTTP proxy information (blank for none): <u>____ → <continue> (Do nothing)</u>
5) How do you want to manage upgrades on this system? <u>No automatic updates</u>
6) Choose software to install: <u>[*]standard system utilities</u>

7) Install the GRUB boot loader to the master boot record? Yes

Install virtual machines on host (via QEMU/KVM of virt-manager) on 64-bit x86 processors:

1. Node Name: controller01
/ CPU:4 / 5120 MB / 100 GB / NIC1: Bridge br1, NIC2: openstack-net0, ...
2. Node Name: compute01
/ CPU:4 / 5120 MB / 100 GB / NIC1: Bridge br1, NIC2: openstack-net0, ...
3. Node Name: compute02
/ CPU:4 / 5120 MB / 100 GB / NIC1: Bridge br1, NIC2: openstack-net0, ...
4. Node Name: compute03
/ CPU:4 / 5120 MB / 100 GB / NIC1: Bridge br1, NIC2: openstack-net0, ...
5. Node Name: snat01
/ CPU:4 / 5120 MB / 100 GB / NIC1: Bridge br1, NIC2: openstack-net0, ...

The Virtual Network Interface Card (Virtual-NIC) can be added from the detail section of virtaul machine located at the upper strip on the window, once click the “detail” botton , select “Add Hardware” and “Network” to add a corresponding NIC.

Unify the names of network interfaces on each node:

```
# nano /etc/default/grub
      Find:
      GRUB_CMDLINE_LINUX=””
      Modify it as:
      GRUB_CMDLINE_LINUX=”net.ifnames=0 biosdevname=0”
CTRL-X → Y

# grub-mkconfig -o /boot/grub/grub.cfg
# nano /etc/network/interfaces
      Change interface names:
      ens3 or enp3s0 (can check by # ifconfig -a for the name)→ eth0
      ens8 or enp5s0 (can check by # ifconfig -a for the name)→ eth1
      ens9 or enp8s0 (can check by # ifconfig -a for the name)→ eth2
      ...
# reboot
# ifconfig -a
      Verify that the network has been changed.
      Verify the network can be connected by: # ping -c 4 www.google.com
```

Interface configuration:

Update the network interface file on each host as follows:

```
# nano /etc/network/interfaces
```

For controller01 node:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.100.10
    netmask 255.255.255.0
```

For compute01 node:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
```

```
    address 192.168.100.11
    netmask 255.255.255.0
```

For compute02 node:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.100.12
    netmask 255.255.255.0
```

For compute03 node:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.100.13
    netmask 255.255.255.0
```

For snat01 node:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.100.14
    netmask 255.255.255.0
```

To activate the changes:

```
# ifdown --all; ifup --all
```

Install following packages in all nodes:

```
# apt update; apt install -y software-properties-common
# add-apt-repository cloud-archive:pike
# apt update; apt dist-upgrade -y
```

Setting the hostnames:

For controller01 node:

```
# hostnamectl set-hostname controller01
```

For compute01 node:

```
# hostnamectl set-hostname compute01
```

For compute02 node:

```
# hostnamectl set-hostname compute02
```

For compute03 node:

```
# hostnamectl set-hostname compute03
```

For snat01 node:

```
# hostnamectl set-hostname snat01
```

Update the /etc/hosts file on each node by adding the followings:

Note: comment out the line of 127.0.1.1

```
192.168.100.10 controller01.alvin.com controller01
192.168.100.11 compute01.alvin.com compute01
```

```
192.168.100.12 compute02.alvin.com compute02
192.168.100.13 compute03.alvin.com compute03
192.168.100.14 snat01.alvin.com snat01
```

Verify that the fully qualified host name is reflected:
At controller01 node:

```
# hostname –f
controller01.alvin.com
```

Reboot on each node.

To verify these ip addresses were defined properly.
Check on each node by:

```
# ping -c4 controller01
# ping -c4 compute01
# ping -c4 compute02
# ping -c4 compute03
# ping -c4 snat01
```

These “ping” should be working fine without missing any package.

Configuring the OpenStack repository
To enable the cloud archive repository, on all nodes:

```
# apt update; apt -y install software-properties-common
# add-apt-repositroy cloud-archive:pike
# apt update; apt -y dist-upgrade
```

Installing and configuring Network Time Protocol:
On all nodes:

```
# apt -y install chrony
```

On the controller node:

```
# nano /etc/chrony/chrony.conf
```

add the following line to this file to allow other nodes to synchronize their time against the controller:

```
      allow 192.168.100.0/24
```

Save and close the file.

On the other nodes:

```
# nano /etc/chrony/chrony.conf
```

comment out any pool lines and add the following line:

```
      # pool 2.debian.pool.ntp.org offline iburst
      server controller01 iburst
```

Save and close the file.

On all nodes:

```
# systemctl restart chrony
```

To check it configured properly on all nodes:

```
# chronyc sources
```

If it works fine, there will be a ‘^*’ sign at the beginning of each time server.

Rebooting all the nodes:

```
# reboot
```

Installing OpenStack:
On the controller node:

```
# apt -y install python-openstackclient
```

Installing and configuring the MySQL database server:
On the controller node:

```
# apt -y install mariadb-server python-pymysql
```

create and edit the configuration file

```
# nano /etc/mysql/mariadb.conf.d/99-openstack.cnf
```

add the following lines, the value for bind-address should be the management IP of the controller node:

```
[mysqld]
bind-address = 192.168.100.10
default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

Save and close the file, then restart the mysql server:

```
# systemctl restart mysql
```

The following command will begin the MySQL installation and configuration process:

```
# mysql_secure_installation
```

Answer [Y]es to the remaining questions to exit the configuration process.

Installing and configuring the messaging server:

On the controller node:

```
# apt -y install rabbitmq-server
```

Add a user to RabbitMQ named openstack with a password of rabbit:

```
# rabbitmqctl add_user openstack rabbit
```

Set RabbitMQ permissions for the openstack user:

```
# rabbitmqctl set_permissions openstack “.*” “.*” “.*”
```

Installing and configuring memcached:

On the controller node:

```
# apt -y install memcached python-memcache
# nano /etc/memcached.conf
```

At this file, replace the default listener address with IP address of the controller node as:

```
#-l 127.0.0.1
-l 192.168.100.10
```

Restart the memcached service:

```
# systemctl restart memcached
```

Installing and configuring the identity service:

On the controller node:

configuring the database using the mysql client, create the Keystone database:

```
# mysql
MariabDB [(none)] > create database keystone;
MariabDB [(none)] > grant all privileges on keystone.* ‘keystone’@’localhost’
identified by ‘keystone’;
MariabDB [(none)] > grant all privileges on keystone.* ‘keystone’@’%’ identified
by ‘keystone’;
```

Installing Keystone:

On the controller node:

```
# apt -y install keystone apache2 libapache2-mod-wsgi
```

Update the [database] section in the /etc/keystone/keystone.conf file. In this installation, the username and password will be keystone. Overwrite the existing connection string with the following value on one line:

```
# nano /etc/keystone/keystone.conf
[database]
…
connection = mysql+pymysql://keystone:keystone@controller01/keystone
```

Update the [token] section in the same file accordingly:

```
[token]
…
```

```
provider = fernet
```

Save and close the file.

Populate the Keystone database:

```
# su -s /bin/sh -c “keystone-manage db_sync” keystone
# keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
# keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

Bootstrap the Identity service:

```
# keystone-manage bootstrap --bootstrap-password openstack --bootstrap-admin-url http://controller01:35357/v3/ --bootstrap-internal-url http://controller01:5000/v3/ --bootstrap-public-url http://controller01:5000/v3/ --bootstrap-region-id RegionOne
```

Configuring the Apache HTTP server:

On the controller node:

```
# sed -i ‘1s/^/ServerName controller01\n&/’ /etc/apache2/apache2.conf
# systemctl restart apache2
```

Setting enviroment variables:

On the controller node:

The following commands will create a file named adminrc containing environment variables for the admin user:

```
# cat >> ~/adminrc <<EOF
export OS_PROJECT_DOMAIN_NAME=default
export OS_USER_DOMAIN_NAME=default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=openstack
export OS_AUTH_URL=http://controller01:35357/v3
export OS_IDENTITY_API_VERSION=3
EOF
```

The following commands will create a file named demorc containing environment variables for the demo user:

```
# cat >> ~/demorc <<EOF
export OS_PROJECT_DOMAIN_NAME=default
export OS_USER_DOMAIN_NAME=default
export OS_PROJECT_NAME=demo
export OS_USERNAME=demo
export OS_PASSWORD=demo
export OS_AUTH_URL=http://controller01:35357/v3
export OS_IDENTITY_API_VERSION=3
EOF
```

Use the source command to load the environment variables from the file:

```
# source ~/adminrc
```

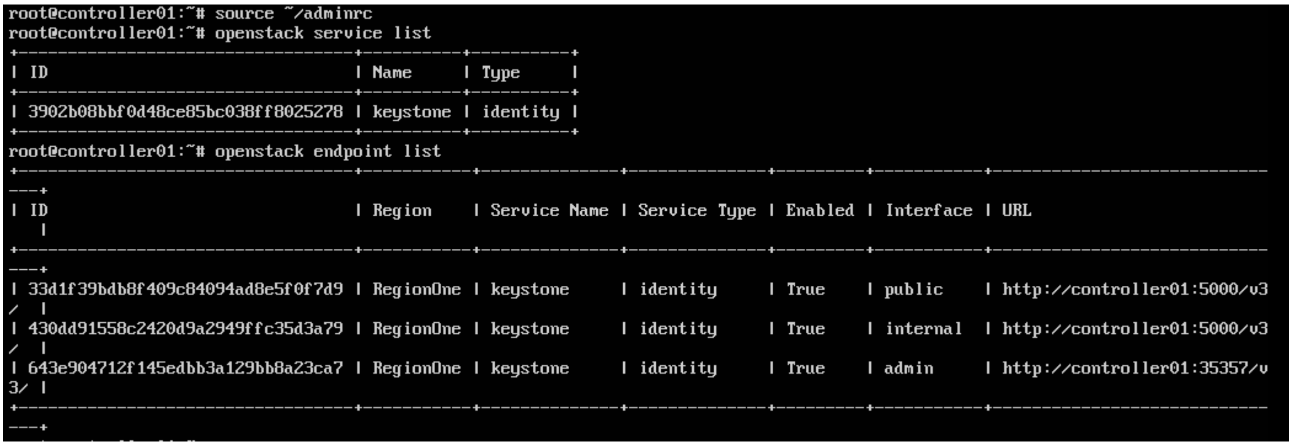
Defining services and API endpoints in keystone:

The Openstack Identity service and endpoint were created during the bootstrap process earlier. To verify the objects were created using the ‘openstack service list’ and ‘openstack endpoint list’ commands as shown here:

On the controller node:

```
# openstack service list
# openstack endpoint list
```

You should see something similar to this:



```
root@controller01:~# source ~/adminrc
root@controller01:~# openstack service list
+----------------------------------+----------+----------+
| ID                               | Name     | Type     |
+----------------------------------+----------+----------+
| 3902b08bbf0d48ce85bc038ff8025278 | keystone | identity |
+----------------------------------+----------+----------+
root@controller01:~# openstack endpoint list
+----------------------------------+-----------+--------------+--------------+---------+-----------+----------------------------
---+
| ID                               | Region    | Service Name | Service Type | Enabled | Interface | URL
   |
+----------------------------------+-----------+--------------+--------------+---------+-----------+----------------------------
---+
| 33d1f39bdb8f409c84094ad8e5f0f7d9 | RegionOne | keystone     | identity     | True    | public    | http://controller01:5000/v3
/  |
| 430dd91558c2420d9a2949ffc35d3a79 | RegionOne | keystone     | identity     | True    | internal  | http://controller01:5000/v3
/  |
| 643e904712f145edbb3a129bb8a23ca7 | RegionOne | keystone     | identity     | True    | admin     | http://controller01:35357/v
3/ |
+----------------------------------+-----------+--------------+--------------+---------+-----------+----------------------------
---+
```

Defining users, projects, and roles in Keystone:

On the controller node:

```
# openstack project create --description “Service Project” service
# openstack project create --description “Demo Project” demo
# openstack user create demo --password=demo
# openstack role create user
# openstack role add --project demo --user demo user
```

Installing and configuring the image service:

On the controller node:

```
# mysql
MariabDB [(none)] > create database glance;
MariabDB [(none)] > grant all privileges on glance.* to ‘glance’@’localhost’
identified by ‘glance’;
MariabDB [(none)] > grant all privileges on glance.* to ‘glance’@’%’ identified
by ‘glance’;
MariabDB [(none)] > quit;
```

Defining the Glance user, service, and endpoints:

```
# openstack user create glance --domain default --password=glance
# openstack role add --project service --user glance admin
# openstack service create --name glance --desrcription “OpenStack Image” image
# openstack endpoint create --region RegionOne image public
http://controller01:9292
# openstack endpoint create --region RegionOne image internal
http://controller01:9292
# openstack endpoint create --region RegionOne image admin
http://controller01:9292
```

Installing and configuring Glance components:

On the controller node:

```
# apt install glance
# nano /etc/glance/glance-api.conf
```

Update the database connection string as:

```
[database]
…
connection = mysql+pymysql://glance:glance@controller01/glance
```

Update the [keystone_authtoken] settings as:

```
[keystone_authtoken]
…
auth_uri = http://controller01:5000
auth_url = http://controller01:35357
memcached_servers = controller01:11211
auth_type = password
user_domain_name = default
project_domain_name = default
project_name = service
```

```
username = glance
password = glance
```

Update the [paste_deploy] settings as:

```
[paste_deploy]
…
flavor = keystone
```

Update the [glance_store] settings as:

```
[glance_store]
…
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images
```

Save and close the file.

```
# nano /etc/glance/glance-registry.conf
```

Update the database connection string as:

```
[database]
…
connection = mysql+pymysql://glance:glance@controller01/glance
```

Update the [keystone_authtoken] settings as:

```
[keystone_authtoken]
…
auth_uri = http://controller01:5000
auth_url = http://controller01:35357
memcached_servers = controller01:11211
auth_type = password
user_domain_name = default
project_domain_name = default
project_name = service
username = glance
password = glance
```

Update the [paste_deploy] settings as:

```
[paste_deploy]
…
flavor = keystone
```

Save and close the file.

Populate the Glance database:

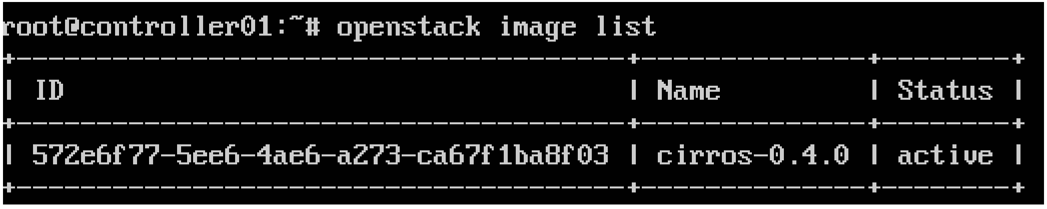
```
# su -s /bin/sh -c “glance-manage db_sync” glance
# systemctl restart glance-registry glance-api
```

Verifying the Glance image service installation:

```
# source ~/adminrc
# mkdir /tmp/images
# wget -P /tmp/images http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img
# openstack image create “cirros-0.4.0” --file /tmp/images/cirros-0.4.0-x86_64-disk.img --disk-format qcow2 --container-format bare --public
```

```
# openstack image list
```

You should see something like this:



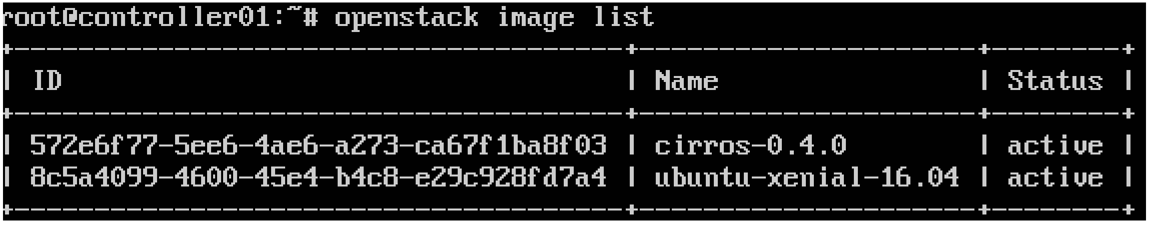
```
root@controller01:~# openstack image list
+--------------------------------------+--------------+--------+
| ID                                   | Name         | Status |
+--------------------------------------+--------------+--------+
| 572e6f77-5ee6-4ae6-a273-ca67f1ba8f03 | cirros-0.4.0 | active |
+--------------------------------------+--------------+--------+
```

Installing additional images:

```
# wget -P /tmp/images http://cloud-images.ubuntu.com/xenial/current/xenial-server-cloudimg-amd64-disk1.img
# openstack image create “ubuntu-xenial-16.04” --file /tmp/images/xenial-server-cloudimg-amd64-disk1.img --disk-format qcow2 --container-format bare --public

# openstack image list
```

You should see something like this:



```
root@controller01:~# openstack image list
+--------------------------------------+---------------------+--------+
| ID                                   | Name                | Status |
+--------------------------------------+---------------------+--------+
| 572e6f77-5ee6-4ae6-a273-ca67f1ba8f03 | cirros-0.4.0        | active |
| 8c5a4099-4600-45e4-b4c8-e29c928fd7a4 | ubuntu-xenial-16.04 | active |
+--------------------------------------+---------------------+--------+
```

Installing and configuring the Compute service:

On the controller node:

```
# mysql
MariaDB [(none)] > create database nova;
MariaDB [(none)] > create database nova_api;
MariaDB [(none)] > create database nova_cell0;
MariaDB [(none)] > grant all privileges on nova.* to ‘nova’@’localhost’ identified by ‘nova’;
MariaDB [(none)] > grant all privileges on nova.* to ‘nova’@’%’ identified by ‘nova’;
MariaDB [(none)] > grant all privileges on nova_api.* to ‘nova’@’localhost’ identified by ‘nova’;
MariaDB [(none)] > grant all privileges on nova_api.* to ‘nova’@’%’ identified by ‘nova’;
MariaDB [(none)] > grant all privileges on nova_cell0.* to ‘nova’@’localhost’ identified by ‘nova’;
MariaDB [(none)] > grant all privileges on nova_cell0.* to ‘nova’@’%’ identified by ‘nova’;
MariaDB [(none)] > quit;

# source ~/adminrc
# openstack user create nova --domain default --password=nova
# openstack user create placement --domain default --password=placement
# openstack role add --project service --user nova admin
# openstack role add --project service --user placement admin
# openstack service create --name nova --description “OpenStack Compute” compute
# openstack service create --name placement --description “Placement API” placement
# openstack endpoint create --region RegionOne compute public http://controller01:8774/v2.1
# openstack endpoint create --region RegionOne compute internal http://controller01:8774/v2.1
# openstack endpoint create --region RegionOne compute admin http://controller01:8774/v2.1
# openstack endpoint create --region RegionOne placement public http://controller01:8778
# openstack endpoint create --region RegionOne placement internal http://controller01:8778
# openstack endpoint create --region RegionOne placement admin http://controller01:8778
```

Installing and configuring controller node components:

On the controller node:

```
# apt -y install nova-api nova-conductor nova-consoleauth nova-novncproxy nova-scheduler nova-placement-api
# nano /etc/nova/nova.conf
```

Update the [database] and [api_database] sections of the Nova configuration file as:

```
[database]
…
connection = mysql+pymysql://nova:nova@controller01/nova
[api_database]
…
connection = mysql+pymysql://nova:nova@controller01/nova_api
```

Update the [DEFAULT] section as:

```
[DEFAULT]
…
transport_url = rabbit://openstack:rabbit@controller01
my_ip = 192.168.100.10
```

Update the [vnc] section as:

```
…
enabled = true
vncserver_listen = 192.168.100.10
vncserver_proxyclient_address = 192.168.100.10
```

Update the [api] and [keystone_authtoken] sections as:

```
[api]
…
auth_strategy = keystone

[keystone_authtoken]
…
auth_uri = http://controller01:5000
auth_url = http://controller01:35357
memcached_servers = controller01:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = nova
```

Update the [glance] section as:

```
[glance]
…
api_servers = http://controller01:9292
```

Update the [oslo_concurrency] section as:

```
[oslo_concurrency]
…
lock_path = /var/lib/nova/tmp
```

Update the [placement] section as:

```
[placement]
…
os_region_name = RegionOne
auth_url = http://controller01:35357/v3
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = placement
password = placement
```

Save and close the file.

Populate the these databases through:

```
# su -s /bin/sh -c “nova-manage api_db sync” nova
# su -s /bin/sh -c “nova-manage cell_v2 map_cell0” nova
# su -s /bin/sh -c “nova-manage cell_v2 create_cell --name=cell1 --verbose” nova
# su -s /bin/sh -c “nova-manage db sync” nova
# systemctl restart nova-api nova-consoleauth nova-scheduler nova-conductor
nova-novncproxy
```

Installing and configuring compute node components:

On the compute nodes:

```
# apt install nova-compute
# nano /etc/nova/nova.conf
```

Update the [api] and [keystone_authtoken] sections as:

```
[api]
…
auth_strategy = keystone

[keystone_authtoken]
…
auth_uri = http://controller01:5000
auth_url = http://controller01:35357
memcached_servers = controller01:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = nova
```

Update the [DEFAULT] and [vnc] sections respectively as:

for compute01:

```
[DEFAULT]
…
transport_url = rabbit://openstack:rabbit@controller01
my_ip = 192.168.100.11

[vnc]
…
vncserver_proxyclient_address = 192.168.100.11
enabled = True
vncserver_listen = 0.0.0.0
novncproxy_base_url = http://controller01:6080/vnc_auto.html
```

for compute02:

```
[DEFAULT]
…
transport_url = rabbit://openstack:rabbit@controller01
my_ip = 192.168.100.12

[vnc]
…
vncserver_proxyclient_address = 192.168.100.12
enabled = True
vncserver_listen = 0.0.0.0
novncproxy_base_url = http://controller01:6080/vnc_auto.html
```

for compute03:

```
[DEFAULT]
…
transport_url = rabbit://openstack:rabbit@controller01
my_ip = 192.168.100.13

[vnc]
```

```
…
vncserver_proxyclient_address = 192.168.100.13
enabled = True
vncserver_listen = 0.0.0.0
novncproxy_base_url = http://controller01:6080/vnc_auto.html
```

On the compute nodes:

Update the [glance] section as:

```
[glance]
…
api_servers = http://controller01:9292
```

Update the [oslo_concurrency] section as:

```
[oslo_concurrency]
…
lock_path = /var/lib/nova/tmp
```

Update the [placement] section as:

```
[placement]
…
os_region_name = RegionOne
auth_url = http://controller01:35357/v3
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = placement
password = placement
```

Save and close the file.

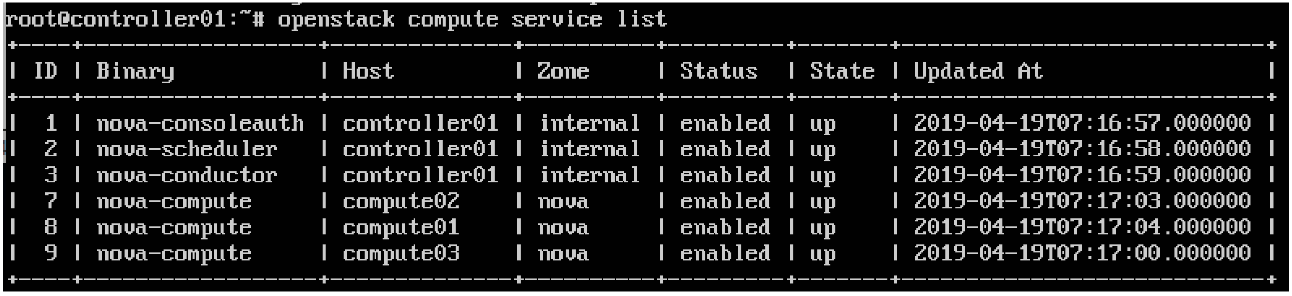
Restart the nova-compute service on all compute nodes:

```
# systemctl restart nova-compute
```

On the controller node:

```
# openstack compute service list
```

You should see something like this:



```
root@controller01:~# openstack compute service list
+----+------------------+--------------+----------+---------+-------+----------------------------+
| ID | Binary           | Host         | Zone     | Status  | State | Updated At                 |
+----+------------------+--------------+----------+---------+-------+----------------------------+
|  1 | nova-consoleauth | controller01 | internal | enabled | up    | 2019-04-19T07:16:57.000000 |
|  2 | nova-scheduler   | controller01 | internal | enabled | up    | 2019-04-19T07:16:58.000000 |
|  3 | nova-conductor   | controller01 | internal | enabled | up    | 2019-04-19T07:16:59.000000 |
|  7 | nova-compute     | compute02    | nova     | enabled | up    | 2019-04-19T07:17:03.000000 |
|  8 | nova-compute     | compute01    | nova     | enabled | up    | 2019-04-19T07:17:04.000000 |
|  9 | nova-compute     | compute03    | nova     | enabled | up    | 2019-04-19T07:17:00.000000 |
+----+------------------+--------------+----------+---------+-------+----------------------------+
```

Installing the OpenStack Dashboard:

On the controller node:

```
# apt -y install openstack-dashboard
# nano /etc/openstack-dashboard/local_setting.py
```

Update several as:

```
OPENSTACK_HOST = “controller01”
OPENSTACK_API_VERSIONS = {
    “identity”: 3,
    “image”: 2,
    “volume”: 2,
}
OPENSTACK_KEYSTONE_URL = “http://%s:5000/v3” % OPENSTACK_HOST
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = “Default”
```

```
OPENSTACK_KEYSTONE_DEFAULT_ROLE = “user”
OPENSTACK_NEUTRON_NETWORK = {
    ‘enable_router’: False,
    ‘enable_quotas’: False,
    ‘enable_ipv6’: False,
    ‘enable_distributed_router’: False,
    ‘enable_ha_router’: False,
    ‘enable_lb’: False,
    ‘enable_firewall’: False,
    ‘enable_vpn’: False,
    ‘enable_fip_topology_check’: False,
}
DEFAULT_THEME = ‘default’
```

Save and close the file.

Reloading Apache:

```
# systemctl reload apache2
```

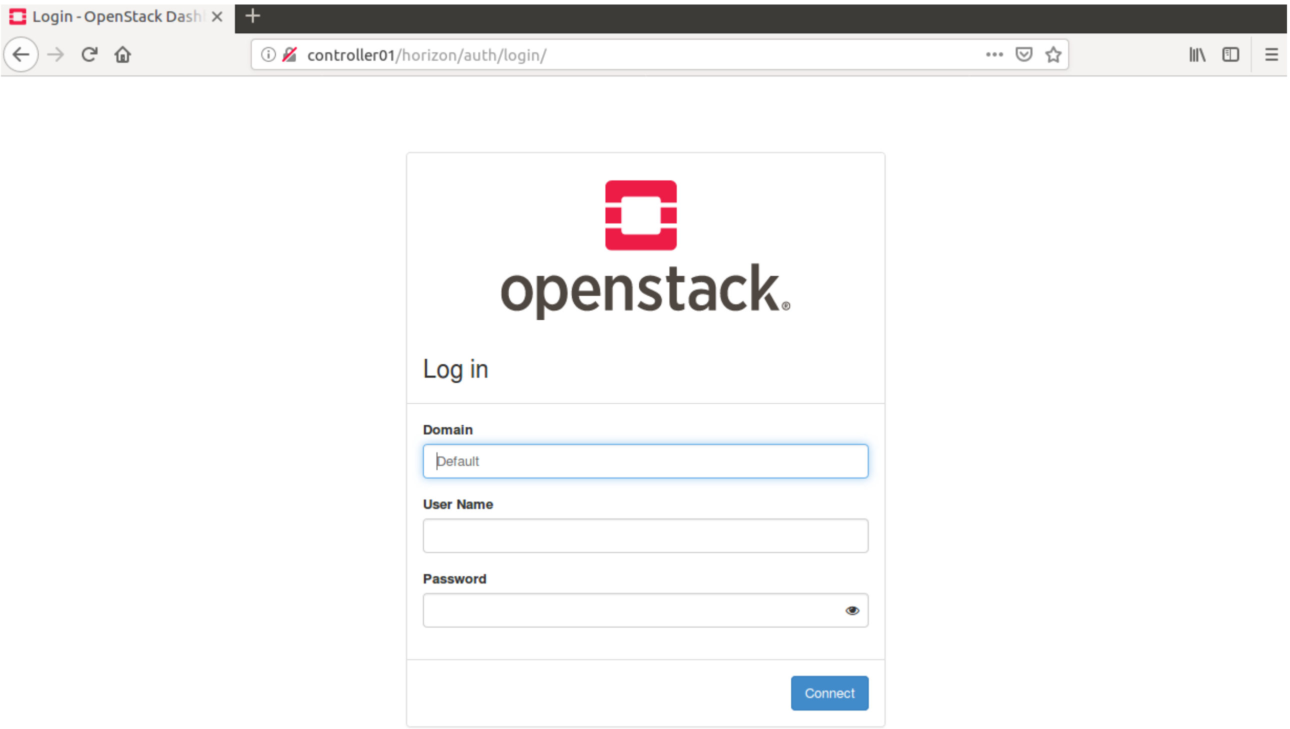
Testing connectivity to the dashboard:
In this installation, the domain is Default, the username is admin, and the password is openstack.

On the host outside virtual machines:
update the /etc/hosts file to add the IP address of the controller node:

```
# nano /etc/hosts
192.168.100.10 controller01.alvin.com controller01
```

Open the following URL in a web browser: http://controller01/horizon/

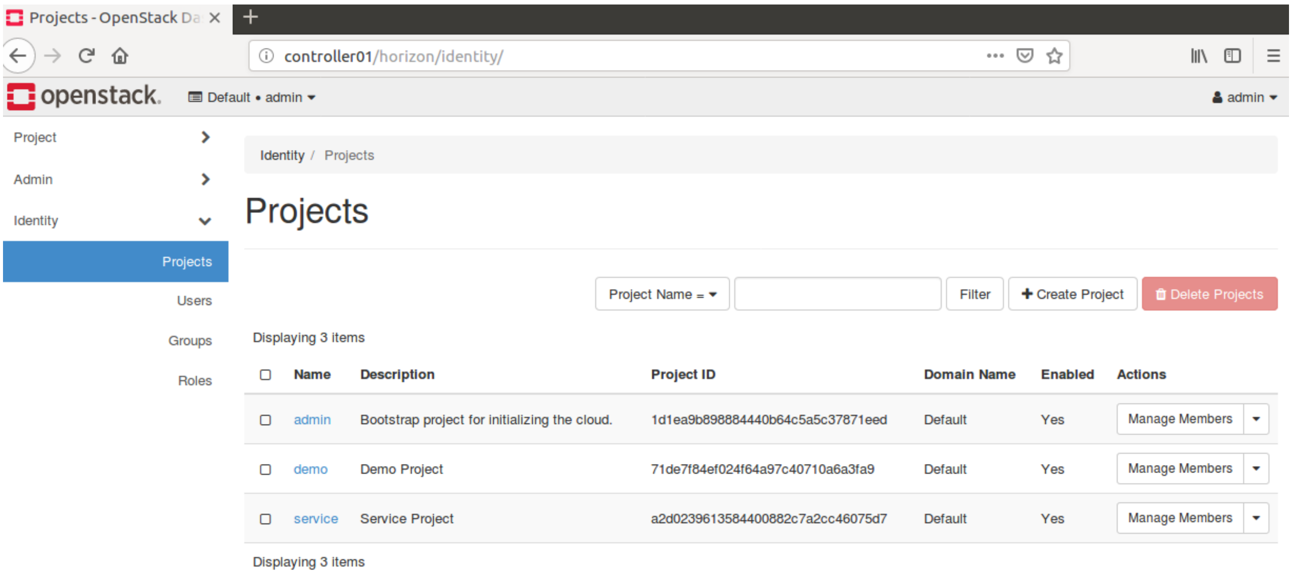
You should see this view:



Login - OpenStack Dash

controller01/horizon/auth/login/

Log in by the given user name and password as mentioned:
You should see this view:



Projects

Displaying 3 items

□	Name	Description	Project ID	Domain Name	Enabled	Actions
□	admin	Bootstrap project for initializing the cloud.	1d1ea9b898884440b64c5a5c37871eed	Default	Yes	Manage Members
□	demo	Demo Project	71de7f84ef024f64a97c40710a6a3fa9	Default	Yes	Manage Members
□	service	Service Project	a2d0239613584400882c7a2cc46075d7	Default	Yes	Manage Members

Displaying 3 items