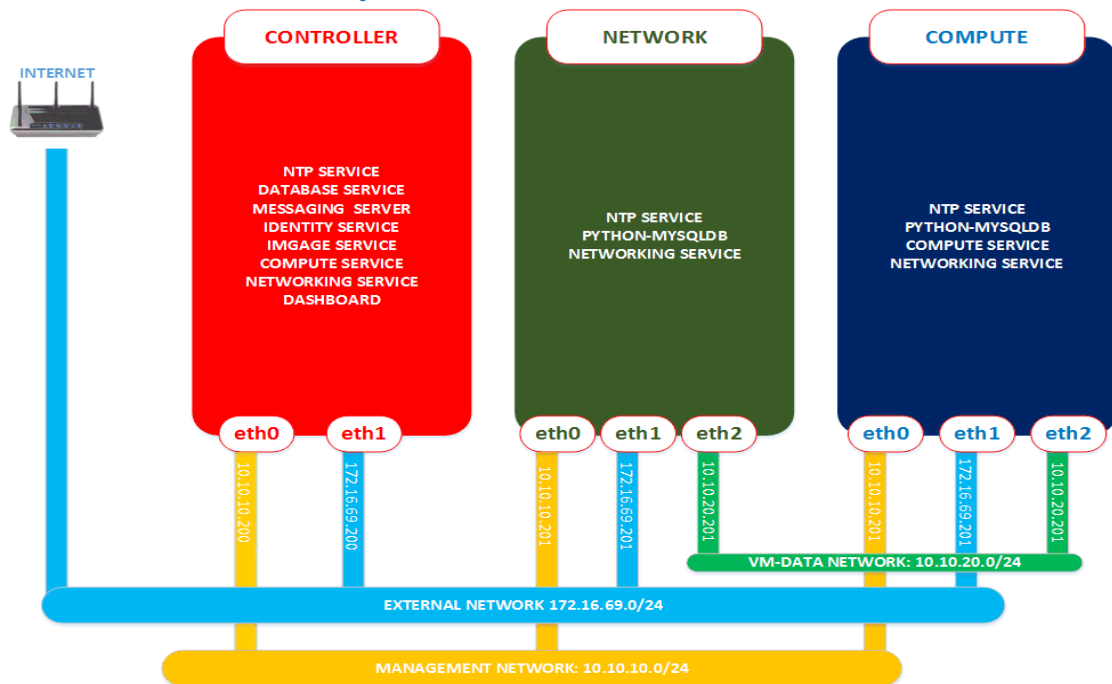
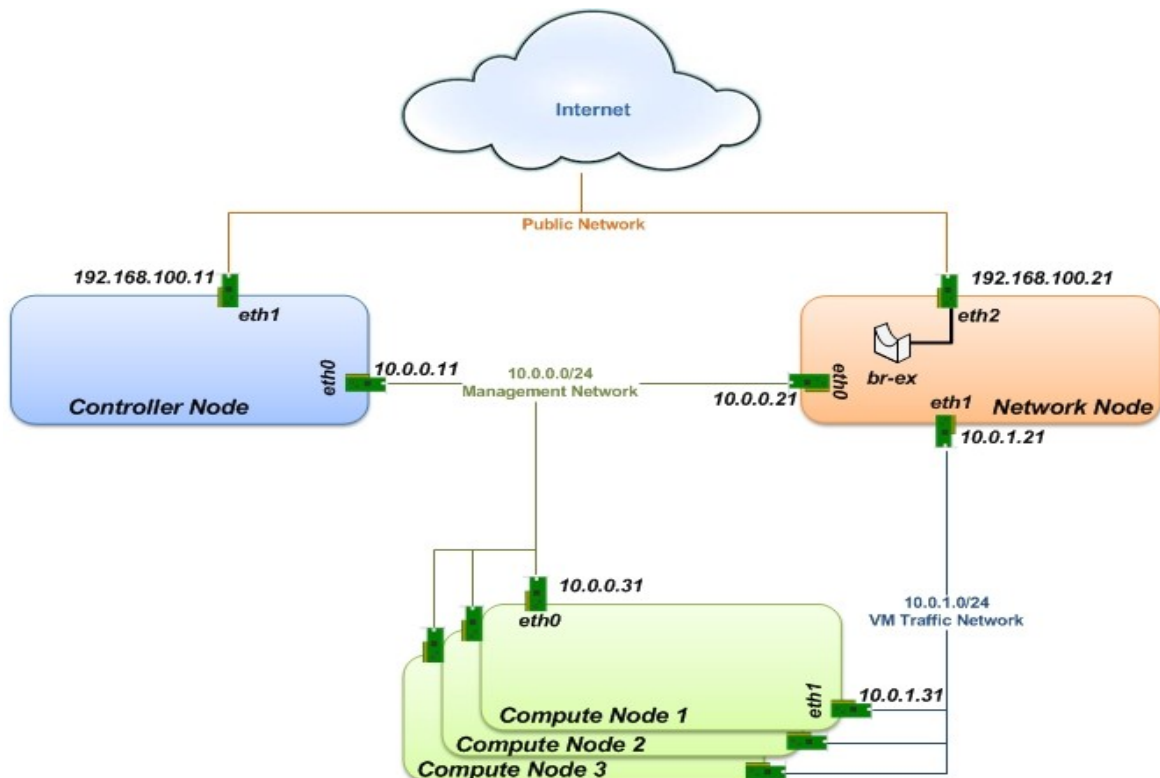


HƯỚNG DẪN CÀI ĐẶT OPENSTACK KILO - MULTI NODE

MÔ HÌNH CÀI ĐẶT



Mô hình cài đặt OPS (tham khảo VietOpenStack)



Mô hình cấu hình mạng Multi-Node

I. THIẾT LẬP CẤU HÌNH MẠNG CÁC NODE

1. Controller Node

Sửa /etc/hostname như sau:

```
controller
```

Sửa /etc/hosts như sau:

```
#controller
10.0.0.11      controller

#network
10.0.0.21      network

#compute
10.0.0.31      compute
```

Sửa /etc/network/interfaces để cấu hình IP cho các card mạng

```
#The management network interface
auto eth0
iface eth0 inet static
    address    10.0.0.11
    netmask 255.255.255.0
    network 10.0.0.0

#The public network interface
auto eth1
iface eth1 inet static
    address    192.168.122.11
    netmask 255.255.255.0
    gateway 192.168.122.1
    network 192.168.122.0
    dns-nameservers 8.8.8.8 8.8.4.4
```

Cần phải restart network hoặc reboot để thay đổi có hiệu quả.

2. Network Node

Sửa /etc/hostname như sau:

```
network
```

Sửa /etc/hosts như sau:

```
#controller
10.0.0.11          controller

#network
10.0.0.21          network

#compute
10.0.0.31          compute
```

Sửa /etc/network/interfaces để cấu hình IP cho các card mạng

```
#The management network interface
auto eth0
iface eth0 inet static
    address    10.0.0.21
    netmask 255.255.255.0
    network 10.0.0.0

#The VM traffic network interface
auto eth1
iface eth1 inet static
    address    10.0.1.21
    netmask 255.255.255.0
    network 10.0.1.0

#The public network interface
auto eth2
iface eth2 inet static
    address    192.168.122.21
    netmask 255.255.255.0
    gateway 192.168.122.1
    network 192.168.122.0
    dns-nameservers 8.8.8.8 8.8.4.4
```

Cần phải restart network hoặc reboot để thay đổi có hiệu quả.

3. Compute Node

Sửa /etc/hostname như sau:

```
compute
```

Sửa /etc/hosts như sau:

```
#controller
10.0.0.11      controller

#network
10.0.0.21      network

#compute
10.0.0.31      compute
```

Sửa /etc/network/interfaces để cấu hình IP cho các card mạng

```
#The management network interface
auto eth0
iface eth0 inet static
    address    10.0.0.31
    netmask 255.255.255.0
    network 10.0.0.0

#The VM traffic network interface
auto eth1
iface eth1 inet static
    address    10.0.1.31
    netmask 255.255.255.0
    network 10.0.1.0
```

Chú ý: có thể bổ sung thêm cho Compute Node một card mạng kết nối Public Network để hỗ trợ ssh trong quá trình cài đặt

```
#The public network interface
auto eth2
iface eth2 inet static
    address    192.168.122.31
    netmask 255.255.255.0
    gateway 192.168.122.1
    network 192.168.122.0
    dns-nameservers 8.8.8.8 8.8.4.4
```

Cần phải restart network hoặc reboot để thay đổi có hiệu quả.

Kiểm tra việc cấu hình mạng, ta thực hiện ping tại mỗi node:

Tại controller node:

```
$ ping -c 4 8.8.8.8
$ ping -c 4 network
$ ping -c 4 compute
```

Tại network node:

```
$ ping 8.8.8.8
$ ping controller
$ ping compute
$ ping 10.0.1.31
```

Tại compute node:

```
$ ping 8.8.8.8
$ ping controller
$ ping network
$ ping 10.0.1.21
```

II. CÀI ĐẶT MÔI TRƯỜNG NỀN TẢNG

1. Network Time Protocol (NTP)

Cần phải cài đặt NTP trên mỗi node để đồng bộ thời gian giữa các node. Ta lấy controller node làm NTP server, các node còn lại sẽ đồng bộ theo controller node.

Cài đặt NTP dùng lệnh:

```
$ apt-get install ntp
```

Sửa file cấu hình /etc/ntp.conf như sau:

Trên controller node:

```
server 0.ubuntu .pool ntp.org iburst
server 1.ubuntu .pool ntp.org iburst
server 2.ubuntu .pool ntp.org iburst
server 3.ubuntu .pool ntp.org iburst

server ntp.ubuntu.com iburst
restrict -4 default kod notrap nomodify
restrict -6 default kod notrap nomodify
```

Khởi động lại NTP

```
service ntp restart
```

Trên các node khác:

Xóa hoặc comment tất cả các server khác, chỉ để lại 1 server duy nhất là controller

```
server controller iburst
```

Khởi động lại NTP

```
service ntp restart
```

Kiểm tra hoạt động của NTP

Trên controller node:

```
$ ntpq -c peers
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
*ntp-server1	192.0.2.11	2	u	169	1024	377	1.901	-0.611	5.483
+ntp-server2	192.0.2.12	2	u	887	1024	377	0.922	-0.246	2.864

```
$ ntpq -c assoc
```

ind	assid	status	conf	reach	auth	condition	last_event	cnt
1	20487	961a	yes	yes	none	sys.peer	sys_peer	1
2	20488	941a	yes	yes	none	candidate	sys_peer	1

Cột condition có ít nhất 1 hàng là sys.peer là ok!

Trên các node khác:

```
$ ntpq -c peers
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
*controller	192.0.2.21	3	u	47	64	37	0.308	-0.251	0.079

Cột remote là controller là ok!

```
$ ntpq -c assoc
```

ind	assid	status	conf	reach	auth	condition	last_event	cnt
1	21181	963a	yes	yes	none	sys.peer	sys_peer	3

Cột condition là sys.peer là ok!

2. OpenStack packages

Thực hiện trên cả 3 node

Install the Ubuntu Cloud archive keyring and repository:

```
$ 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
```

Cập nhật hệ thống:

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

3. MySQL

Phần lớn các dịch vụ của OpenStack sử dụng SQL để lưu trữ thông tin. Cơ sở dữ liệu được cài đặt trên controller node. Ở đây, ta sử dụng MariaDB.

Cài đặt thư viện Python MySQL và MariaDB

```
$ apt-get install mariadb-server python-mysqldb
```

Tạo và sửa file `/etc/mysql/conf.d/mysqld_openstack.cnf` như sau:

```
[mysqld]
bind-address = 10.0.0.11
default-storage-engine = innodb
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
```

bind-address đặt là Management IP của controller node để các node khác có thể truy cập vào database thông qua Management Network

Khởi động lại mysql:

```
$ service mysql restart
```

Để bảo mật cho cơ sở dữ liệu:

```
$ mysql_secure_installation
```

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.
```

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...
```

```
Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.
```

```
Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

4. Message queue

OpenStack sử dụng message queue (hàng đợi thông điệp) để điều phối hoạt động và thông tin giữa các dịch vụ. Message queue chủ yếu chạy trên controller node. OpenStack hỗ trợ một số dịch vụ message queue như RabbitMQ, Qpid hay ZeroMQ. Ở đây, ta sử dụng RabbitMQ.

Cài đặt RabbitMQ:


```
$ apt-get install rabbitmq-server
```

Thêm người dùng cho RabbitMQ:

```
$ rabbitmqctl add_user RABBIT_USER RABBIT_PASS
```

RABBIT_USER và RABBIT_PASS có thể đặt tùy ý nhưng cần ghi nhớ để cấu hình các bước tiếp theo. Ở đây, khuyến khích sử dụng cặp RABBIT_USER là openstack và RABBIT_PASS openstack để dễ ghi nhớ và thống nhất trong việc cài đặt.

Khi đó, câu lệnh trên như sau:

```
$ rabbitmqctl add_user openstack openstack
```

Cấp quyền cho người dùng “openstack” (đã thêm ở phía trên):

```
$ rabbitmqctl set_permissions openstack ".*" ".*" ".*"
```

Hoàn tất quá trình chuẩn bị cơ bản để chính thức cài đặt OpenStack!!!

5. Thiết lập thông số để tránh lỗi trong quá trình cài đặt và vận hành OpenStack

Trong phiên bản Ubuntu 14.04 có lỗi gói ngôn ngữ “unset LC_ALL” dẫn đến lỗi “unsupported locale setting” khi sử dụng các dịch vụ của OpenStack. Do đó, cần kiểm tra trước khi cài đặt.

Dùng lệnh:

```
$ locale
```

Nếu thấy dòng LC_ALL = en_us.UTF-8 thì ok, không cần sửa. Còn nếu thấy LC_ALL = (unset) thì cần thực hiện:

Trong file /etc/default/locale thêm dòng:

```
LC_ALL = en_us.UTF-8
```

Khởi động lại hệ thống để thay đổi có hiệu quả.

III. CÀI ĐẶT CÁC DỊCH VỤ OPENSTACK

Nếu không có chú thích gì thì mặc định là cài đặt dịch vụ trên controller node. Các dịch vụ được cài đặt cho mỗi node vui lòng xem lại mô hình cài đặt ở đầu bài viết.

1. Keystone – Identity Service

Keystone là dịch vụ để chứng thực cho tất cả các dịch vụ khác của Openstack.

Cài đặt Keystone

Tạo database cho 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';

GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'10.0.0.11' IDENTIFIED BY
'KEYSTONE_DBPASS';

exit;
```

10.0.0.11 là Management IP của controller node.

KEYSTONE_DBPASS có thể đặt tùy ý nhưng cần ghi nhớ để cấu hình cho các bước tiếp theo. Ở đây đặt là KEYSTONE_DBPASS cho dễ ghi nhớ.

Ngăn keystone tự động bắt đầu sau khi cài đặt:

```
echo "manual" > /etc/init/keystone.override
```

Cài đặt và cấu hình

Cài đặt các thành phần:

```
apt-get install keystone python-openstackclient apache2 libapache2-mod-wsgi
memcached python-memcache
```

Cấu hình:

Sửa file `/etc/keystone/keystone.conf` theo nội dung sau:

Lưu ý: Cách trình bày

[DEFAULT]

...

verbose = True

*được hiểu là trong phần [DEFAULT] sửa hoặc thêm dòng
verbose = True*

```
[DEFAULT]
...
admin_token = ADMIN_TOKEN
verbose = True

[database]
...
connection = mysql://keystone:KEYSTONE_DBPASS@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
```

ADMIN_TOKEN là một chuỗi số hay ký tự tùy ý để, cần ghi nhớ để khai báo xác thực tạm quyền Admin khi sử dụng keystone. Ở đây đặt là ADMIN_TOKEN

KEYSTONE_DBPASS là pass đã khai báo khi tạo database

Trong phần [database] cần xóa bỏ những dòng cấu hình kết nối tới loại database khác như sqlite.

Đồng bộ với cơ sở dữ liệu:

```
$ su -s /bin/sh -c "keystone-manage db_sync" keystone
```

Khởi động lại Keystone

```
$ service keystone restart
```

Sử dụng cron để loại bỏ những token quá hạn theo chu kỳ cho keystone

```
(crontab -l -u keystone 2>&1 | grep -q token_flush) || \
echo '@hourly /usr/bin/keystone-manage token_flush >/var/log/keystone/key-
stone-tokenflush.log 2>&1' \
>> /var/spool/cron/crontabs/keystone
```

Cấu hình Apache HTTP Server

Sửa file `/etc/apache2/apache2.conf` để cấu hình `ServerName`:

```
ServerName controller
```

Tạo file `/etc/apache2/sites-available/wsgi-keystone.conf` với nội dung:

```
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>
```

Kích hoạt Identity Service virtual host

```
ln -s /etc/apache2/sites-available/wsgi-keystone.conf /etc/apache2/sites-enabled
```

Tạo thư mục cho các thành phần WSGI

```
mkdir -p /var/www/cgi-bin/keystone
```

Sao chép các thành phần WSGI từ kho dữ liệu về thư mục vừa tạo

```
curl http://git.openstack.org/cgit/openstack/keystone/plain/httpd/keystone.py?h=stable/kilo \
| tee /var/www/cgi-bin/keystone/main /var/www/cgi-bin/keystone/admin
```

Cấp quyền cho thư mục:

```
chown -R keystone:keystone /var/www/cgi-bin/keystone
chmod 755 /var/www/cgi-bin/keystone/*
```

Để hoàn tất quá trình cài đặt, khởi động lại Apache Server:

```
service apache2 restart
```

Tạo Service và API endpoint cho Keystone

Khai báo token xác thực:

```
export OS_TOKEN=ADMIN_TOKEN
```

Sử dụng ADMIN_TOKEN đã thiết lập phía trên để khai báo xác thực Admin

Khai báo địa chỉ Endpoint:

```
export OS_URL=http://controller:35357/v2.0
```

Tạo service Identity

```
openstack service create \
--name keystone --description "OpenStack Identity" identity
```

Field	Value
description	OpenStack Identity
enabled	True
id	4ddaae90388b4ebc9d252ec2252d8d10
name	keystone
type	identity

Tạo Identity service API Endpoint:

```
openstack endpoint create \
  --publicurl http://controller:5000/v2.0 \
  --internalurl http://controller:5000/v2.0 \
  --adminurl http://controller:35357/v2.0 \
  --region RegionOne \
  identity
```

Field	Value
adminurl	http://controller:35357/v2.0
id	4a9ffc04b8eb4848a49625a3df0170e5
internalurl	http://controller:5000/v2.0
publicurl	http://controller:5000/v2.0
region	RegionOne
service_id	4ddaae90388b4ebc9d252ec2252d8d10
service_name	keystone
service_type	identity

Tạo các project (tenant), user, role

Tạo project Admin, user Admin và role Admin để quản trị các hoạt động Admin

Tạo project Admin

```
openstack project create --description "Admin Project" admin
```

Field	Value
description	Admin Project
enabled	True
id	cf12a15c5ea84b019aec3dc45580896b
name	admin

Tạo user Admin

```
openstack user create --password-prompt admin
```

User Password:

Repeat User Password:

Field	Value
email	None
enabled	True
id	4d411f2291f34941b30eef9bd797505a
name	admin
username	admin

Tạo role Admin

```
$ openstack role create admin
```

Field	Value
id	cd2cb9a39e874ea69e5d4b896eb16128
name	admin

Thêm role Admin cho project và user Admin

```
$ openstack role add --project admin --user admin admin
```

Field	Value
id	cd2cb9a39e874ea69e5d4b896eb16128
name	admin

Tạo service project để chứa các user tương ứng với dịch vụ của OpenStack

```
$ openstack project create --description "Service Project" service
```

Field	Value
description	Service Project
enabled	True
id	55cbd79c0c014c8a95534ebd16213ca1
name	service

Để kiểm tra kết quả cài đặt Keystone, thực hiện một số thao tác sau:

Loại bỏ biết môi trường xác thực Admin đã khai báo trước đó

```
$ unset OS_TOKEN OS_URL
```

Thử gửi một yêu cầu cấp token xác thực trong vai trò Admin user

```
$ openstack --os-auth-url http://controller:35357 \
  --os-project-name admin --os-username admin --os-auth-type password \
  token issue
```

Password:

Field	Value
expires	2015-03-24T18:55:01Z
id	ff5ed908984c4a4190f584d826d75fed
project_id	cf12a15c5ea84b019aec3dc45580896b
user_id	4d411f2291f34941b30eef9bd797505a

Tạo đoạn script khai báo biến môi trường cho việc chứng thực user Admin

Tạo file `admin-openrc.sh` với nội dung như sau:

```
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=ADMIN_PASS
export OS_AUTH_URL=http://controller:35357/v3
```

Khi đó, ta chỉ cần gọi:

```
$ source admin-openrc.sh
```

Và tiến hành truy cập các dịch vụ trong vai trò Admin.

Một số lệnh của Keystone như :

Yêu cầu token xác thực :

```
$ openstack token issue
```


Field	Value
expires	2015-03-25T01:45:49.950092Z
id	cd4110152ac24bdeaa82e1443c910c36
project_id	cf12a15c5ea84b019aec3dc45580896b
user_id	4d411f2291f34941b30eef9bd797505a

Xem danh sách user:

```
$ openstack user list
```

Xem danh sách project:

```
$ openstack project list
```

2. Glance – Image Service

Cài đặt Glance

Tạo cơ sở dữ liệu cho 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';
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'10.0.0.11' \
  IDENTIFIED BY 'GLANCE_DBPASS';

exit;
```

Đăng ký Glance với Keystone:

Load script admin-openrc.sh

```
$ source admin-openrc.sh
```

Tạo user Glance

```
$ openstack user create --password-prompt glance
User Password:
Repeat User Password:
```

Field	Value
email	None
enabled	True
id	1dc206e084334db2bee88363745da014
name	glance
username	glance

Thêm user Glance vào project Service với role Admin

```
$ openstack role add --project service --user glance admin
```

Field	Value
id	cd2cb9a39e874ea69e5d4b896eb16128
name	admin

Tạo image service Glance

```
$ openstack service create --name glance \
  --description "OpenStack Image service" image
```

Field	Value
description	OpenStack Image service
enabled	True
id	178124d6081c441b80d79972614149c6
name	glance
type	image

Tạo Image Service API Endpoint:

```
$ openstack endpoint create \
  --publicurl http://controller:9292 \
  --internalurl http://controller:9292 \
  --adminurl http://controller:9292 \
  --region RegionOne \
  image
```

Field	Value
adminurl	http://controller:9292
id	805b1dbc90ab47479111102bc6423313
internalurl	http://controller:9292
publicurl	http://controller:9292
region	RegionOne
service_id	178124d6081c441b80d79972614149c6
service_name	glance
service_type	image

Cài đặt và cấu hình Glance:

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

Sửa file `/etc/glance/glance-api.conf` theo nội dung sau đây:

Lưu ý: Cách trình bày dưới đây đã đề cập ở phần Keystone, vui lòng xem lại để tránh sai sót

```
[DEFAULT]
...
notification_driver = noop
verbose = True
rpc_backend = rabbit

[database]
...
connection = mysql://glance:GLANCE_DBPASS@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 = GLANCE_PASS

[paste_deploy]
...
flavor = keystone

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

[oslo_messaging_rabbit]
...
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = openstack
```

*Lưu ý: **GLANCE_PASS** là password khi tạo user glance khi đăng ký với Keystone phía trên. Khuyến nghị nên ghi lại các password đã thiết lập cẩn thận trong quá trình cài đặt với những hệ thống lớn hoặc đặt thống nhất 1 password chung (ví dụ openstack) cho lab cá nhân. **rabbit_userid** và **rabbit_password** đã thiết lập khi cài đặt RabbitMQ.*

Sửa file `/etc/glance/glance-registry.conf` theo nội dung sau đây:

```
[DEFAULT]
...
notification_driver = noop
verbose = True
rpc_backend = rabbit

[database]
...
connection = mysql://glance:GLANCE_DBPASS@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 = GLANCE_PASS

[paste_deploy]
...
flavor = keystone

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

[oslo_messaging_rabbit]
...
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = openstack
```

Đồng bộ với cơ sở dữ liệu:

```
# su -s /bin/sh -c "glance-manage db_sync" glance
```

Thêm biến môi trường `OS_IMAGE_API_VERSION` vào script `admin-openrc.sh`

```
$ echo "export OS_IMAGE_API_VERSION=2" | tee -a admin-openrc.sh demo-openrc.sh
```

Khởi động lại các dịch vụ glance để hoàn tất cài đặt

```
$ service glance-registry restart
$ service glance-api restart
```

Đăng ký Image tới Glance

Load script admin-openrc.sh

```
$ source admin-openrc.sh
```

Tạo thư mục tạm chứa file ảnh download về

```
$ mkdir /tmp/images
```

Download file ảnh về

```
$ wget -P /tmp/images http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img
```

Upload file ảnh đã tải về lên Image Service

```
$ 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 --visibility public --progress
```

```
[=====>] 100%
```

Property	Value
checksum	133eae9fb1c98f45894a4e60d8736619
container_format	bare
created_at	2015-03-26T16:52:10Z
disk_format	qcow2
id	38047887-61a7-41ea-9b49-27987d5e8bb9
min_disk	0
min_ram	0
name	cirros-0.3.4-x86_64
owner	ae7a98326b9c455588edd2656d723b9d
protected	False
size	13200896
status	active
tags	[]
updated_at	2015-03-26T16:52:10Z
virtual_size	None
visibility	public

Xem danh sách các file ảnh

```
$ glance image-list
```

ID	Name
38047887-61a7-41ea-9b49-27987d5e8bb9	cirros-0.3.4-x86_64

3. Nova – Compute Service

Cài đặt Nova

Trên controller node

Tạo cơ sở dữ liệu cho 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';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'10.0.0.11' \
    IDENTIFIED BY 'NOVA_DBPASS';

exit;
```

Đăng ký Nova với Keystone

Tạo user Nova:

```
$ openstack user create --password-prompt nova
User Password:
Repeat User Password:
```

Field	Value
email	None
enabled	True
id	8e0b71d732db4bfba04943a96230c8c0
name	nova
username	nova

Thêm user Nova vào project Service với role Admin:

```
$ openstack role add --project service --user nova admin
```

Field	Value
id	cd2cb9a39e874ea69e5d4b896eb16128
name	admin

Tạo compute service Nova

```
$ openstack service create --name nova \
  --description "OpenStack Compute" compute
```

Field	Value
description	OpenStack Compute
enabled	True
id	060d59eac51b4594815603d75a00aba2
name	nova
type	compute

Tạo Compute Service API Endpoint:

```
$ 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
```

Field	Value
adminurl	http://controller:8774/v2/%(tenant_id)s
id	4e885d4ad43f4c4fbf2287734bc58d6b
internalurl	http://controller:8774/v2/%(tenant_id)s
publicurl	http://controller:8774/v2/%(tenant_id)s
region	RegionOne
service_id	060d59eac51b4594815603d75a00aba2
service_name	nova
service_type	compute

Cài đặt và cấu hình Nova

Cài đặt các thành phần:

```
# apt-get install nova-api nova-cert nova-conductor nova-consoleauth \
nova-novncproxy nova-scheduler python-novaclient
```

Sửa file `/etc/nova/nova.conf` theo nội dung sau:

```
[DEFAULT]
...
verbose = True
rpc_backend = rabbit
auth_strategy = keystone
my_ip = 10.0.0.11
vncserver_listen = 10.0.0.11
vncserver_proxyclient_address = 10.0.0.11

[database]
...
connection = mysql://nova:NOVA_DBPASS@controller/nova

[glance]
...
host = controller

[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 = NOVA_PASS

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

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

Lưu ý: NOVA_PASS là password khi tạo user nova khi đăng ký với Keystone phía trên. rabbit_userid và rabbit_password đã thiết lập khi cài đặt RabbitMQ. 10.0.0.11 là Management IP của Controller Node đã cấu hình từ đầu.

Đồng bộ với cơ sở dữ liệu:

```
# su -s /bin/sh -c "nova-manage db sync" nova
```

Khởi động lại các dịch vụ để hoàn tất cài đặt:


```
# service nova-api restart
# service nova-cert restart
# service nova-consoleauth restart
# service nova-scheduler restart
# service nova-conductor restart
# service nova-novncproxy restart
```

Compute Node

Cài đặt các thành phần:

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

Sửa file `/etc/nova/nova.conf` theo nội dung sau:

```
[DEFAULT]
...
verbose = True
rpc_backend = rabbit
auth_strategy = keystone
my_ip = 10.0.0.31
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = 10.0.0.31
novncproxy_base_url = http://192.168.122.11:6080/vnc_auto.html

[database]
...
connection = mysql://nova:NOVA_DBPASS@controller/nova

[glance]
...
host = controller

[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 = NOVA_PASS

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

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

Lưu ý: NOVA_PASS là password khi tạo user nova khi đăng ký với Keystone phía trên. rabbit_userid và rabbit_password đã thiết lập khi

cài đặt RabbitMQ. 10.0.0.13 là Management IP của Compute Node đã cấu hình từ đầu.

Để kiểm tra xem phần cứng hệ thống có hỗ trợ máy ảo hay không, thực hiện lệnh :

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

Nếu kết quả lớn hơn 0 có nghĩa phần cứng hệ thống có hỗ trợ. Còn nếu kết quả bằng 0, cần cấu hình sử dụng ảo hóa qemu thay vì kvm như mặc định trong file `/etc/nova/nova-compute.conf` như sau:

```
[libvirt]
...
virt_type = qemu
```

Khởi động lại các dịch vụ để hoàn tất cài đặt:

```
# service nova-compute restart
```

Kiểm tra hoạt động của Nova

Trên Controller Node, thực hiện các thao tác sau:

```
$ source admin-openrc.sh
```

Xem danh sách các thành phần dịch vụ

```
$ nova service-list
```

Id	Binary	Host	Zone	Status	State	Updated_at	Disabled Reason
1	nova-conductor	controller	internal	enabled	up	2014-09-16T23:54:02.000000	-
2	nova-consoleauth	controller	internal	enabled	up	2014-09-16T23:54:04.000000	-
3	nova-scheduler	controller	internal	enabled	up	2014-09-16T23:54:07.000000	-
4	nova-cert	controller	internal	enabled	up	2014-09-16T23:54:00.000000	-
5	nova-compute	compute1	nova	enabled	up	2014-09-16T23:54:06.000000	-

Xem danh sách các image của Image Service

```
$ nova image-list
```

ID	Name	Status	Server
38047887-61a7-41ea-9b49-27987d5e8bb9	cirros-0.3.4-x86_64	ACTIVE	

-----+

4. Neutron - Network Service

Neutron là ...

Các thành phần của Neutron có mặt trên cả 3 node. Trên controller node cài neutron-server và neutron-plugin-ml2; trên network node cài neutron-plugin-ml2, neutron-plugin-openvswitch-agent, neutron-l3-agent, neutron-dhcp-agent và neutron-metadata-agent; trên compute node cài neutron-plugin-ml2 và neutron-plugin-openvswitch-agent.

Cài đặt neutron

Cài đặt Neutron trên controller node

Tạo cơ sở dữ liệu cho 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';
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'10.0.0.11' \
    IDENTIFIED BY 'NEUTRON_DBPASS';

exit;
```

Đăng ký neutron với keystone

Load admin-openrc.sh

```
$ source admin-openrc.sh
```

Tạo user Neutron

```
$ openstack user create --password-prompt neutron
User Password:
Repeat User Password:
```

```
+-----+-----+
| Field  | Value |
+-----+-----+
| email  | None  |
```

enabled	True
id	ab67f043d9304017aaa73d692eeb4945
name	neutron
username	neutron

Thêm user Neutron vào project Service với role Admin

```
$ openstack role add --project service --user neutron admin
```

Field	Value
id	cd2cb9a39e874ea69e5d4b896eb16128
name	admin

Tạo service Neutron

```
$ openstack service create --name neutron \
  --description "OpenStack Networking" network
```

Field	Value
description	OpenStack Networking
enabled	True
id	f71529314dab4a4d8eca427e701d209e
name	neutron
type	network

Tạo Network service API endpoint:

```
$ openstack endpoint create \
  --publicurl http://controller:9696 \
  --adminurl http://controller:9696 \
  --internalurl http://controller:9696 \
  --region RegionOne \
  network
```

Field	Value
adminurl	http://controller:9696
id	04a7d3c1de784099aaba83a8a74100b3
internalurl	http://controller:9696
publicurl	http://controller:9696
region	RegionOne
service_id	f71529314dab4a4d8eca427e701d209e
service_name	neutron
service_type	network

Cài đặt và cấu hình

Cài đặt các thành phần

```
# apt-get install neutron-server neutron-plugin-ml2 python-neutronclient
```

Sửa file cấu hình `/etc/neutron/neutron.conf` theo nội dung sau:

```
[DEFAULT]
...
verbose = True
rpc_backend = rabbit
auth_strategy = keystone
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True
notify_nova_on_port_status_changes = True
notify_nova_on_port_data_changes = True
nova_url = http://controller:8774/v2

[database]
...
connection = mysql://neutron:NEUTRON_DBPASS@controller/neutron

[keystone_auth]
...
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 = neutron
password = NEUTRON_PASS

[nova]
...
auth_url = http://controller:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
openstackproject_name = service
username = nova
password = NOVA_PASS

[oslo_messaging_rabbit]
...
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = openstack
```

Sửa file cấu hình /etc/neutron/plugins/ml2/ml2_conf.ini theo nội dung sau:

```
[ml2]
...
type_drivers = flat,vlan,gre,vxlan
tenant_network_types = gre
mechanism_drivers = openvswitch

[ml2_type_gre]
...
tunnel_id_ranges = 1:1000

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

Sửa file cấu hình /etc/nova/nova.conf theo nội dung sau:

```
[DEFAULT]
...
network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSIfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

[neutron]
...
url = http://controller:9696
auth_strategy = keystone
admin_auth_url = http://controller:35357/v2.0
admin_tenant_name = service
admin_username = neutron
admin_password = NEUTRON_PASS
```

Đồng bộ cơ sở dữ liệu:

```
# su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf \
--config-file /etc/neutron/plugins/ml2/ml2_conf.ini upgrade head" neutron
```

Khởi động lại các dịch vụ:

```
# service nova-api restart
# service neutron-server restart
```

Kiểm tra hoạt động của neutron trên controller node

Xem danh sách các thành phần mở rộng ext

```
$ neutron ext-list
```

alias	name
security-group	security-group
l3_agent_scheduler	L3 Agent Scheduler
ext-gw-mode	Neutron L3 Configurable external gateway mode
binding	Port Binding
provider	Provider Network
agent	agent
quotas	Quota management support
dhcp_agent_scheduler	DHCP Agent Scheduler
l3-ha	HA Router extension
multi-provider	Multi Provider Network
external-net	Neutron external network
router	Neutron L3 Router
allowed-address-pairs	Allowed Address Pairs
extraroute	Neutron Extra Route
extra_dhcp_opt	Neutron Extra DHCP opts
dvr	Distributed Virtual Router

Cài đặt Neutron trên network node

Cài đặt và cấu hình

Sửa file `/etc/sysctl.conf` theo nội dung sau:

```
net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0
```

Thực hiện thay đổi

```
# sysctl -p
```

Cài đặt các thành phần:

```
# apt-get install neutron-plugin-ml2 neutron-plugin-openvswitch-agent \
  neutron-l3-agent neutron-dhcp-agent neutron-metadata-agent
```

Cấu hình neutron tổng quan

Sửa file `/etc/neutron/neutron.conf` theo nội dung sau:

Xóa bỏ tất cả các cấu hình database vì network node không truy cập trực tiếp tới database

```
[DEFAULT]
...
verbose = True
rpc_backend = rabbit
auth_strategy = keystone
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True

[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 = neutron
password = NEUTRON_PASS

[oslo_messaging_rabbit]
...
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = openstack
```

Cấu hình Modular Layer 2 (ML2) plug-in

Sửa file `/etc/neutron/plugins/ml2/ml2_conf.ini` theo nội dung sau:

```
[ml2]
...
type_drivers = flat,vlan,gre,vxlan
tenant_network_types = gre
mechanism_drivers = openvswitch

[ml2_type_flat]
...
flat_networks = external

[ml2_type_gre]
...
tunnel_id_ranges = 1:1000

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

[ovs]
...
```



```
local_ip = 10.0.1.21
bridge_mappings = external:br-ex

[agent]
...
tunnel_types = gre
```

Lưu ý: local_ip đặt là VM traffic IP của network node

Cấu hình Layer 3 (L3) agent

Sửa file /etc/neutron/l3_agent.ini theo nội dung sau:

```
[DEFAULT]
...
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
external_network_bridge =
router_delete_namespaces = True
verbose = True
```

Cấu hình DHCP agent

Sửa file /etc/neutron/dhcp_agent.ini theo nội dung sau:

```
[DEFAULT]
...
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
dhcp_delete_namespaces = True
verbose = True
dnsmasq_config_file = /etc/neutron/dnsmasq-neutron.conf
```

Tạo file /etc/neutron/dnsmasq-neutron.conf với nội dung sau:

```
dhcp-option-force=26,1454
```

Hủy các tiến trình dnsmasq đang tồn tại

```
# pkill dnsmasq
```

Cấu hình Metadata agent

Sửa file /etc/neutron/metadata_agent.ini theo nội dung sau:

```
[DEFAULT]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
auth_region = RegionOne
auth_plugin = password
project_domain_id = default
```

```
user_domain_id = default
project_name = service
username = neutron
password = NEUTRON_PASS
nova_metadata_ip = controller
metadata_proxy_shared_secret = METADATA_SECRET
verbose = True
```

Lưu ý: METADATA_SECRET có thể đặt tùy ý nhưng cần ghi nhớ để cấu hình bước tiếp theo.

Sửa file /etc/nova/nova.conf của controller node theo nội dung sau:

```
[neutron]
...
service_metadata_proxy = True
metadata_proxy_shared_secret = METADATA_SECRET
```

Trên controller node, khởi động lại dịch vụ nova-api

```
# service nova-api restart
```

Cấu hình dịch vụ Open vSwitch (OVS)

Ta sửa lại interfaces của network node như sau:

```
#The management network interface
auto eth0
iface eth0 inet static
    address 10.0.0.21
    netmask 255.255.255.0
    network 10.0.0.0

#The VM traffic network interface
auto eth1
iface eth1 inet static
    address 10.0.1.21
    netmask 255.255.255.0
    network 10.0.1.0

#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

#OVS external bridge
auto br-ex
iface br-ex inet static
```

```
address 192.168.122.21
netmask 255.255.255.0
network 192.168.122.0
gateway 192.168.122.1
dns-nameservers 8.8.8.8 8.8.4.4
```

Khởi động lại dịch vụ OVS:

```
# service openvswitch-switch restart
```

Thêm internal bridge cho OVS

```
# ovs-vsctl add-br br-int
```

Thêm external bridge cho OVS

```
# ovs-vsctl add-br br-ex
```

Thêm cổng cho external bridge kết nối với interface external network vật lý (card mạng kết nối tới public network)

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

Khởi động lại các dịch vụ:

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

Kiểm tra hoạt động của neutron, thực hiện thao tác sau:

Trên controller node, xem danh sách các agent:

```
$ neutron agent-list
```

id	agent_type	host	alive	admin_state_up	binary
30275801-e17a-41e4-8f53-9db63544f689	Metadata agent	network	:-)	True	neutron-metadata-agent
4bd8c50e-7bad-4f3b-955d-67658a491a15	Open vSwitch agent	network	:-)	True	neutron-openvswitch-agent
756e5bba-b70f-4715-b80e-e37f59803d20	L3 agent	network	:-)	True	neutron-l3-agent
9c45473c-6d6d-4f94-8df1-ebd0b6838d5f	DHCP agent	network	:-)	True	neutron-dhcp-agent

Các agent alive hết là thành công!

Cài đặt Neutron trên compute node

Cài đặt và cấu hình

Sửa file `/etc/sysctl.conf` theo nội dung sau:

```
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
```

Thực thi thay đổi

```
# sysctl -p
```

Cài đặt các thành phần:

```
# apt-get install neutron-plugin-ml2 neutron-plugin-openvswitch-agent
```

Cấu hình Neutron tổng quát

Sửa file `/etc/neutron/neutron.conf` theo nội dung sau:

Xóa bỏ tất cả cấu hình database vì compute node không truy cập trực tiếp tới database

```
[DEFAULT]
...
verbose = True
rpc_backend = rabbit
[oslo_messaging_rabbit]
...
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = openstack
auth_strategy = keystone
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True

[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 = neutron
password = NEUTRON_PASS
```

Cấu hình Modular Layer 2 (ML2) plug-in

Sửa file `/etc/neutron/plugins/ml2/ml2_conf.ini` theo nội dung sau:

```
[ml2]
...
type_drivers = flat,vlan,gre,vxlan
tenant_network_types = gre
mechanism_drivers = openvswitch

[ml2_type_gre]
...
tunnel_id_ranges = 1:1000

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

[ovs]
...
local_ip = 10.0.1.31

[agent]
...
tunnel_types = gre
```

Lưu ý: local_ip đặt là VM traffic IP của compute node

Khởi động lại OVS:

```
# service openvswitch-switch restart
```

Cấu hình Compute để sử dụng Neutron

Sửa file /etc/nova/nova.conf theo nội dung:

```
[DEFAULT]
...
network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

[neutron]
...
url = http://controller:9696
auth_strategy = keystone
admin_auth_url = http://controller:35357/v2.0
admin_tenant_name = service
admin_username = neutron
admin_password = NEUTRON_PASS
```

Khởi động lại nova-compute và OVS:

```
# service nova-compute restart
# service neutron-plugin-openvswitch-agent restart
```

Kiểm tra hoạt động của neutron, thực hiện thao tác:

Trên controller node, xem danh sách các agent:

```
$ neutron agent-list
```

id	agent_type	host	alive	admin_state_up	binary
30275801-e17a-41e4-8f53-9db63544f689	Metadata agent	network	(-)	True	neutron-metadata-agent
4bd8c50e-7bad-4f3b-955d-67658a491a15	Open vSwitch agent	network	(-)	True	neutron-openvswitch-agent
756e5bba-b70f-4715-b80e-e37f59803d20	L3 agent	network	(-)	True	neutron-l3-agent
9c45473c-6d6d-4f94-8df1-ebd0b6838d5f	DHCP agent	network	(-)	True	neutron-dhcp-agent
a5a49051-05eb-4b4f-bfc7-d36235fe9131	Open vSwitch agent	compute1	(-)	True	neutron-openvswitch-agent

Các agent alive hết là thành công!

5. Horizon – Dashboard

Horizon là ...

Horizon cài đặt trên controller node.

Cài đặt và cấu hình

Cài đặt các thành phần:

```
# apt-get install openstack-dashboard
```

Cấu hình dashboard:

Sửa file `/etc/openstack-dashboard/local_settings.py` theo nội dung sau:

```
OPENSTACK_HOST = "controller"
ALLOWED_HOSTS = '*'
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211',
    }
}
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
TIME_ZONE = "Asia/Ho_Chi_Minh"
```

Khởi động lại apache2:

```
# service apache2 reload
```

Kiểm tra hoạt động của Horizon, thực hiện:

Từ một máy tính thuộc cùng Public Network với controller node (hoặc thông qua Internet nếu controller node có global IP tĩnh có trình duyệt Web), truy cập địa chỉ <http://192.168.122.11/horizon> (192.168.122.11 là Public IP của controller node) và sử dụng tài khoản user Admin (đã tạo ban đầu với keystone) để đăng nhập.