



# OPENSTACK HIGH-AVAILABLE CONTROLLERS

*(Liberty version on Ubuntu trusty release)*

Version 1.0

Created date 14/03/2016

Created by toanchik

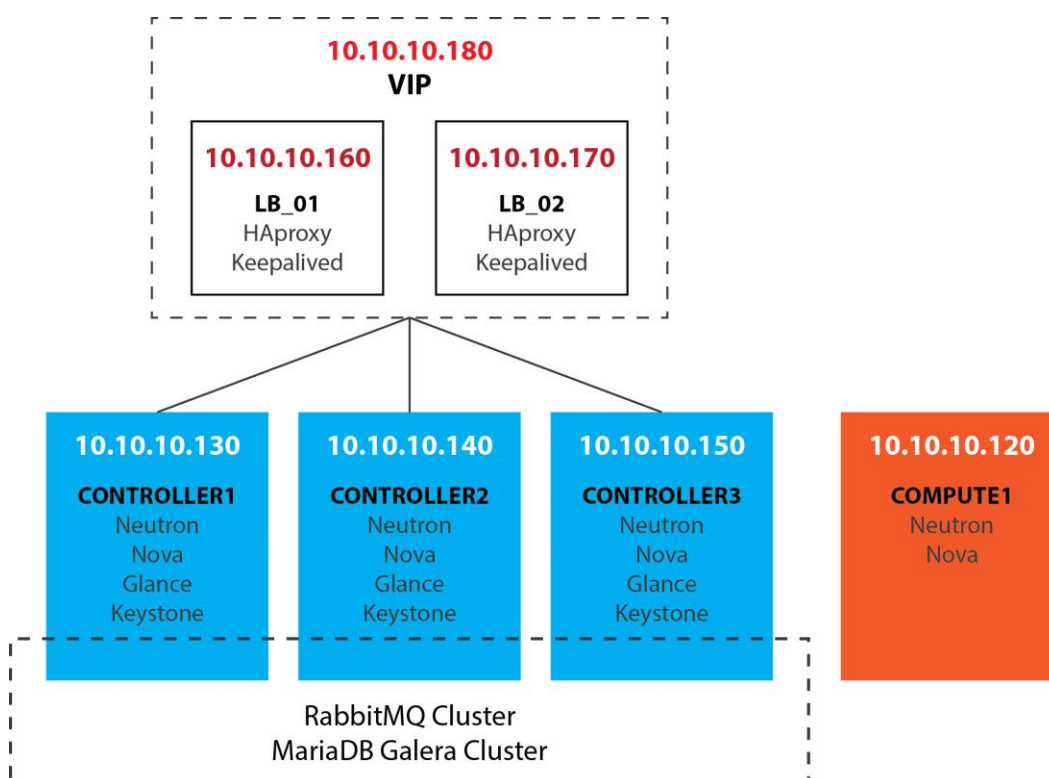
## Contents

I.	Công tác chuẩn bị.....	4
II.	Cài đặt, cấu hình HAproxy & Keepalived trên LB_01 và LB_02.....	4
III.	Cài đặt, cấu hình Mariadb Galera Cluster và RabbitMQ cluster trên CONTROLLER1, CONTROLLER2, CONTROLLER3: .....	8
1.	Mariadb Galera Cluster:.....	8
2.	RabbitMQ cluster .....	10
IV.	Cài đặt các controller services trên 3 nodes CONTROLLER1, CONTROLLER2, CONTROLLER3..	11
1.	Keystone.....	11
2.	Glance: .....	13
3.	Nova: .....	15
4.	Neutron.....	18
5.	Horizon.....	22
V.	Cài đặt nova-compute và neutron trên COMPUTE node:.....	22
VI.	Kết quả Failover test .....	24
Phụ lục 1:	File lb_script.sh.....	26
Phụ lục 2:	File config.cfg .....	29
Phụ lục 3:	File 0_prepare.sh.....	30
Phụ lục 4:	File 1_galera.sh.....	31
Phụ lục 5:	File 2_keystone_01.sh.....	32
Phụ lục 6:	File 3_keystone_02.sh.....	36
Phụ lục 7:	File 4_glance_01.sh .....	39
Phụ lục 8:	File 5_glance_02.sh .....	42
Phụ lục 9:	File 6_nova_01.sh.....	44
Phụ lục 10:	File 7_nova_02.sh.....	47
Phụ lục 11:	File 8_neutron_01.sh .....	49
Phụ lục 12:	File 9_neutron_02.sh .....	53
Phụ lục 13:	File 10_horizon.sh .....	57
Phụ lục 14:	File com_script.sh.....	58

Trên tinh thần “lá rách ít đùm lá rách nhiều”, mình xin chia sẻ anh em kinh nghiệm thử nghiệm HA các services của OpenStack controller. Bài viết chủ yếu hướng tới các đối tượng, là những anh em mới tập tành và muốn được tận mắt kiểm chứng HA controller thành hiện thực, ngoài ra cũng để các cao thủ đi trước giúp review, tối ưu, nâng cấp lab này cho chuẩn chỉnh hơn.

Vì đây là bài lab nhỏ, nhằm mục đích dựng được mô hình kiểm chứng HA các core services của OpenStack controller, vì vậy mình cũng chỉ sử dụng tối thiểu tài nguyên nói chung, đặc biệt là phần cứng, cũng như không đi vào giải thích (vì các tài liệu đã hướng dẫn quá kỹ và tốt rồi), đơn giản hóa, và cố gắng tự động hóa nhiều bước trong quá trình cài đặt, không tập trung vào performance.

Mô hình được mô tả theo như hình minh họa “HA\_controller.jpg”



#### Môi trường lab như sau:

Máy chủ vật lý IBM X3850M2 (16Gb RAM, 12 CPUs x 2,4GHz)

Nền tảng: VMware ESXi 5.5

5 máy ảo: 2 máy nhiệm vụ Load balancer VIP (2Gb RAM, 4 cores, HDD 50Gb - thin provisioning)

3 máy nhiệm vụ Controller (8Gb RAM, 8 cores, HDD 50Gb - thin provisioning)

1 máy nhiệm vụ Computer (ít nhất là để test việc tạo instance, network)

Hệ điều hành: Ubuntu 14.04 bản trusty

Network: 1 subnet 10.10.10.0/24 (kết nối Internet)

OpenStack bản Liberty

Tài liệu hỗ trợ:

Cài đặt cấu hình OpenStack

<http://docs.openstack.org/liberty/install-guide-ubuntu/>

Hướng dẫn OpenStack HA

<http://docs.openstack.org/ha-guide/>

<https://docs.mirantis.com/openstack/fuel/fuel-7.0/reference-architecture.html>

Tài liệu hướng dẫn HAproxy & keepalived

<http://dasunhegoda.com/how-to-setup-haproxy-with-keepalived/833/>

Tài liệu hướng dẫn được xây dựng, kiểm nghiệm rất kỹ lưỡng, cẩn thận của VietOpenStack

<https://github.com/vietstacker/openstack-liberty-multinode/>

Và một nguồn quan trọng nữa là từ kinh nghiệm, ý tưởng của anh em Cộng đồng VietOpenStack

<https://vietstack.slack.com/>

Liên quan đến cài đặt thì tài liệu các nguồn tài liệu đã có đầy đủ chi tiết, nên mình chủ yếu chỉ noted lại những gì liên quan tới phần cấu hình.

## I. Công tác chuẩn bị

Ở bước này, cùng thống nhất là những thứ sau đã sẵn sàng:

OS Ubuntu 14.04 đã sẵn sàng

Sử dụng subnet 10.10.10.0/24 trên các interface eth1

Chuẩn bị file `config.sh` (password, token, IP, Gateway...)

Chạy script `0_prepare.sh` trên 3 nodes `controller1`, `controller2`, `controller3` để update cài đặt, cấu hình Liberty repository, update Ubuntu, ntp servers, `/etc/hosts`

Cấu hình file `/etc/hosts` sẽ như sau:

```
127.0.0.1      localhost
10.10.10.180   controller
10.10.10.130   controller1
10.10.10.140   controller2
10.10.10.120   compute1
10.10.10.160   lb01
10.10.10.170   lb02
```

## II. Cài đặt, cấu hình HAproxy & Keepalived trên LB\_01 và LB\_02

Sử dụng script `lb_script.sh` để cài đặt, cấu hình và chạy 2 dịch vụ `haproxy` và `keepalived` trên 2 nodes này

Trên node LB\_01 file cấu hình `/etc/keepalived/keepalived.conf` sẽ như sau:

```
vrrp_script haproxy {
    script "killall -0 haproxy"
    interval 2
    weight 2
}
vrrp_instance 50 {
    virtual_router_id 50
    advert_int 1
    priority 101
```

```

        state MASTER
        interface eth1
        virtual_ipaddress {
            10.10.10.180/24
        }
        track_script {
            haproxy
        }
    }
}

```

Trên node LB\_01 file cấu hình `/etc/keepalived/keepalived.conf` sẽ như sau:

```

vrrp_script haproxy {
    script "killall -0 haproxy"
    interval 2
    weight 2
}
vrrp_instance 50 {
    virtual_router_id 50
    advert_int 1
    priority 100
    state MASTER
    interface eth1
    virtual_ipaddress {
        10.10.10.180/24
    }
    track_script {
        haproxy
    }
}

```

Trên cả hai nodes file cấu hình `/etc/haproxy/haproxy.cfg` sẽ như sau:

```

global
    chroot /var/lib/haproxy
    daemon
    group haproxy
    maxconn 4000
    pidfile /var/run/haproxy.pid
    stats socket /var/lib/haproxy/stats
    user haproxy

defaults
    log global
    mode tcp
    maxconn 4000
    option redispatch
    retries 3
    timeout http-request 10s
    timeout queue 1m
    timeout connect 10s
    timeout client 1m
    timeout server 1m
    timeout check 10s

listen dashboard *:80

```

```
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:80 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:80 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:80 check inter 2000 rise 2 fall 5

listen galera_cluster *:3306
balance source
mode tcp
option tcpka
server controller1 10.10.10.130:3306 check
server controller2 10.10.10.140:3306 backup check
server controller3 10.10.10.150:3306 backup check

listen glance_api *:9292
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:9292 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:9292 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:9292 check inter 2000 rise 2 fall 5

listen glance_registry *:9191
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:9191 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:9191 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:9191 check inter 2000 rise 2 fall 5

listen keystone_admin *:35357
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:35357 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:35357 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:35357 check inter 2000 rise 2 fall 5

listen keystone_public *:5000
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:5000 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:5000 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:5000 check inter 2000 rise 2 fall 5

listen nova_compute_api *:8774
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:8774 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8774 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8774 check inter 2000 rise 2 fall 5

listen nova_metadata_api *:8775
```

```
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:8775 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8775 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8775 check inter 2000 rise 2 fall 5

listen cinder_api *:8776
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:8776 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8776 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8776 check inter 2000 rise 2 fall 5

listen ceilometer_api *:8777
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:8777 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8777 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8777 check inter 2000 rise 2 fall 5

listen nova_vncproxy *:6080
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:6080 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:6080 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:6080 check inter 2000 rise 2 fall 5

listen neutron_api *:9696
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:9696 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:9696 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:9696 check inter 2000 rise 2 fall 5

listen swift_proxy *:8080
balance source
option tcplog
option tcpka
server controller1 10.10.10.130:8080 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8080 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8080 check inter 2000 rise 2 fall 5

listen rabbitmq *:5672
balance source
option clitcpka
timeout client 900m
server controller1 10.10.10.130:5672 check inter 1s
server controller2 10.10.10.140:5672 check inter 1s
server controller3 10.10.10.150:5672 check inter 1s

listen stats *:1936
mode http
stats enable
stats uri /stats
```

```
stats realm HAProxy\ Statistics
stats auth admin:654321
```

### III. Cài đặt, cấu hình Mariadb Galera Cluster và RabbitMQ cluster trên CONTROLLER1, CONTROLLER2, CONTROLLER3:

#### 1. *Mariadb Galera Cluster:*

Sử dụng script `1_galera.sh` để cài đặt cấu hình Mariadb Galera Cluster trên cả 3 nodes.

Trên node `controller1`, file cấu hình `/etc/mysql/my.cnf` sẽ như sau:

```
[mysqld]
datadir=/var/lib/mysql
user=mysql
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
innodb_flush_log_at_trx_commit=0
innodb_buffer_pool_size=122M
query_cache_type=0
query_cache_size=0
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_provider_options="pc.recovery=TRUE;gcache.size=300M"

# Galera Cluster Configuration
wsrep_cluster_name="Galera_cluster"
wsrep_cluster_address="gcomm://10.10.10.130,10.10.10.140,10.10.10.150"

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="10.10.10.130"
wsrep_node_name="controller1"
```

Trên node `controller2`, file cấu hình `/etc/mysql/my.cnf` sẽ như sau:

```
[mysqld]
datadir=/var/lib/mysql
user=mysql
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
innodb_flush_log_at_trx_commit=0
innodb_buffer_pool_size=122M
query_cache_type=0
query_cache_size=0
bind-address=0.0.0.0

# Galera Provider Configuration
```



```

wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_provider_options="pc.recovery=TRUE;gcache.size=300M"

# Galera Cluster Configuration
wsrep_cluster_name="Galera_cluster"
wsrep_cluster_address="gcomm://10.10.10.130,10.10.10.140,10.10.10.150"

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="10.10.10.140"
wsrep_node_name="controller2"

```

Trên node `controller3`, file cấu hình `/etc/mysql/my.cnf` sẽ như sau:

```

[mysqld]
datadir=/var/lib/mysql
user=mysql
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
innodb_flush_log_at_trx_commit=0
innodb_buffer_pool_size=122M
query_cache_type=0
query_cache_size=0
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_provider_options="pc.recovery=TRUE;gcache.size=300M"

# Galera Cluster Configuration
wsrep_cluster_name="Galera_cluster"
wsrep_cluster_address="gcomm://10.10.10.130,10.10.10.140,10.10.10.150"

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="10.10.10.150"
wsrep_node_name="controller3"

```

Trên `controller1` khởi tạo galera cluster với lệnh sau

```
service mysql start --wsrep-new-cluster
```

Có thể vào database và kiểm tra số lượng node theo lệnh sau:

```
SHOW STATUS LIKE 'wsrep_cluster_size';
```

Sau đó lần lượt start dịch vụ MariaDB & kiểm tra trên `controller2` và `controller3`

## 2. RabbitMQ cluster

Cài đặt gói rabbitmq trên 3 nodes: `controller1`, `controller2`, `controller3`

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

Stop dịch vụ rabbitmq-server trên cả 3 nodes:

```
service rabbitmq-server stop
```

Copy cookie file từ `controller1` sang `controller2` và `controller3`

```
scp /var/lib/rabbitmq/.erlang.cookie
root@controller2:/var/lib/rabbitmq/.erlang.cookie
scp /var/lib/rabbitmq/.erlang.cookie root@
controller3:/var/lib/rabbitmq/.erlang.cookie
```

Điều chỉnh lại quyền và user:

```
chown rabbitmq:rabbitmq /var/lib/rabbitmq/.erlang.cookie
chmod 400 /var/lib/rabbitmq/.erlang.cookie
```

Khởi động lại service rabbitmq trên cả 3 nodes và kiểm tra trạng thái

```
rabbitmqctl cluster_status
```

Trên 2 nodes `controller2`, `controller3` lần lượt stop dịch vụ, join cluster và khởi động lại.

```
rabbitmqctl stop_app
rabbitmqctl join_cluster --ram rabbit@controller1
rabbitmqctl start_app
```

Kiểm tra trạng thái

```
rabbitmqctl cluster_status
```

Kết quả phải như sau

```
Cluster status of node rabbit@controller1 ...
[{nodes,[{disc,[rabbit@controller1]},
          {ram,[rabbit@controller2,rabbit@controller3]}]},
 {running_nodes,[rabbit@controller2,rabbit@controller3,rabbit@controller1]},
 {cluster_name,<<"rabbit@controller1">>},
 {partitions,[]}]
```

Tạo và gán quyền cho user `openstack` trên `controller1`

```
rabbitmqctl add_user openstack 654321
rabbitmqctl set_permissions openstack ".*" ".*" ".*"
```

Kiểm tra trên `controller2` hoặc `controller3`

```
rabbitmqctl list_users
rabbitmqctl list_permissions
```

Trên `controller1` chạy câu lệnh sau:

```
rabbitmqctl set_policy ha-all '^(?!amq\.)\.*' '{"ha-mode": "all"}'
```

## IV. Cài đặt các controller services trên 3 nodes CONTROLLER1, CONTROLLER2, CONTROLLER3

### 1. Keystone

Trên `controller1`, sử dụng script `2_keystone_01.sh` để cài đặt, cấu hình keystone,

Sau đó trên `controller2`, `controller3`, sử dụng script `3_keyston_02.sh` để cài đặt cấu hình keystone.

Trên cả 3 node, file cấu hình `/etc/keystone/keystone.conf` sẽ như sau:

```
[DEFAULT]
log_dir = /var/log/keystone

admin_token = 654321
bind_host = 10.10.10.180
public_bind_host = 10.10.10.180
admin_bind_host = 10.10.10.180

[assignment]
[auth]
[cache]
[catalog]
driver = keystone.catalog.backends.sql.Catalog
[cors]
[cors.subdomain]
[credential]
[database]
connection = mysql+pymysql://keystone:654321@10.10.10.180/keystone

[domain_config]
[endpoint_filter]
[endpoint_policy]
[eventlet_server]
[eventlet_server_ssl]
[federation]
[fernet_tokens]
[identity]
driver = keystone.identity.backends.sql.Identity
[identity_mapping]
[kvs]
[ldap]
[matchmaker_redis]
[matchmaker_ring]
[memcache]
servers = localhost:11211

[oauth1]
[os_inherit]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_middleware]
[oslo_policy]
[paste_deploy]
```

```

[policy]
[resource]
[revoke]
driver = sql

[role]
[saml]
[signing]
[ssl]
[token]
provider = uuid
driver = sql

[tokenless_auth]
[trust]
[extra_headers]
Distribution = Ubuntu

```

Trên `controller1`, dòng cuối của file `/etc/apache2/apache2.conf` sẽ là

```
ServerName 10.10.10.130
```

Trên `controller2`, dòng cuối của file `/etc/apache2/apache2.conf` sẽ là

```
ServerName 10.10.10.140
```

Trên `controller3`, dòng cuối của file `/etc/apache2/apache2.conf` sẽ là

```
ServerName 10.10.10.150
```

Trên 3 node, file cấu hình `/etc/apache2/sites-enabled/wsgi-keystone.conf` như sau

```

Listen 5000
Listen 35357
<VirtualHost *:5000>
    WSGIDaemonProcess keystone-public processes=5 threads=1
    user=keystone group=keystone display-name=%{GROUP}
    WSGIProcessGroup keystone-public
    WSGIScriptAlias / /usr/bin/keystone-wsgi-public
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
    <IfVersion >= 2.4>
        ErrorLogFormat "%{cu}t %M"
    </IfVersion>
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined

    <Directory /usr/bin>
        <IfVersion >= 2.4>
            Require all granted
        </IfVersion>
        <IfVersion < 2.4>
            Order allow,deny
            Allow from all
        </IfVersion>
    </Directory>
</VirtualHost>

```

```

<VirtualHost *:35357>
    WSGIDaemonProcess keystone-admin processes=5 threads=1
    user=keystone group=keystone display-name=%{GROUP}
    WSGIProcessGroup keystone-admin
    WSGIScriptAlias / /usr/bin/keystone-wsgi-admin
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
    <IfVersion >= 2.4>
        ErrorLogFormat "%{cu}t %M"
    </IfVersion>
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined

    <Directory /usr/bin>
        <IfVersion >= 2.4>
            Require all granted
        </IfVersion>
        <IfVersion < 2.4>
            Order allow,deny
            Allow from all
        </IfVersion>
    </Directory>
</VirtualHost>

```

Trên cả 3 nodes file `admin_openrc.sh` 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=654321
export OS_AUTH_URL=http://10.10.10.180:35357/v3
export OS_VOLUME_API_VERSION=2

```

Trên cả 3 nodes file `demo_openrc.sh` như sau:

```

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=654321
export OS_AUTH_URL=http://10.10.10.180:35357/v3
export OS_VOLUME_API_VERSION=2

```

## 2. Glance:

Trên `controller1`, sử dụng script `4_glance_01.sh`

Sau đó trên `controller2`, `controller3` sử dụng script `5_glance_02.sh`

Trên cả 3 nodes, file cấu hình `/etc/glance/glance-api.conf` sẽ như sau

```

[DEFAULT]
notification_driver = noop
verbose = True

[database]

```

```

connection = mysql+pymysql://glance:654321@10.10.10.180/glance
backend = sqlalchemy

[glance_store]
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
[image_format]
[keystone_authtoken]
auth_uri = http://10.10.10.180:5000
auth_url = http://10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = 654321

[matchmaker_redis]
[matchmaker_ring]
[oslo_concurrency]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_policy]
[paste_deploy]
flavor = keystone

[store_type_location_strategy]
[task]
[taskflow_executor]

```

Trên cả 3 nodes, file cấu hình `/etc/glance/glance-registry.conf` sẽ như sau

```

[DEFAULT]
notification_driver = noop
verbose = True

[database]
connection = mysql+pymysql://glance:654321@10.10.10.180/glance
backend = sqlalchemy

[glance_store]

[keystone_authtoken]
auth_uri = http:// 10.10.10.180:5000
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = 654321

[matchmaker_redis]
[matchmaker_ring]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]

```

```
[oslo_policy]

[paste_deploy]
flavor = keystone
```

### 3. Nova:

Trên `controller1`, chạy script sau `6_nova_01.sh`, sau đó chạy script `7_nova_02.sh` trên `controller2`, `controller3`

Trên `controller1`, file cấu hình `/etc/nova/nova.conf` như sau

```
[DEFAULT]

rpc_backend = rabbit
auth_strategy = keystone

dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

my_ip = 10.10.10.130

network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

enabled_apis=osapi_compute,metadata
verbose = True

enable_instance_password = True

[database]
connection = mysql+pymysql://nova:654321@10.10.10.180/nova

[oslo_messaging_rabbit]
rabbit_host = 10.10.10.180
rabbit_userid = openstack
rabbit_password = 654321

[keystone_authtoken]
auth_uri = http:// 10.10.10.180:5000
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = nova
password = 654321
```

```

[vnc]
vncserver_listen = 10.10.10.130
vncserver_proxyclient_address = 10.10.10.130

[glance]
host = 10.10.10.180

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

[neutron]
url = http:// 10.10.10.180:9696
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = 654321

service_metadata_proxy = True
metadata_proxy_shared_secret = 654321

```

Trên `controller2`, file cấu hình `/etc/nova/nova.conf` như sau

```

[DEFAULT]

rpc_backend = rabbit
auth_strategy = keystone

dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

my_ip = 10.10.10.140

network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

enabled_apis=osapi_compute,metadata
verbose = True

enable_instance_password = True

[database]
connection = mysql+pymysql://nova:654321@10.10.10.180/nova

```



```

[oslo_messaging_rabbit]
rabbit_host = 10.10.10.180
rabbit_userid = openstack
rabbit_password = 654321

[keystone_authtoken]
auth_uri = http:// 10.10.10.180:5000
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = nova
password = 654321

[vnc]
vncserver_listen = 10.10.10.140
vncserver_proxyclient_address = 10.10.10.140

[glance]
host = 10.10.10.180

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

[neutron]
url = http:// 10.10.10.180:9696
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = 654321

service_metadata_proxy = True
metadata_proxy_shared_secret = 654321

```

Trên `controller3`, file cấu hình `/etc/nova/nova.conf` như sau

```

[DEFAULT]

rpc_backend = rabbit
auth_strategy = keystone

dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

my_ip = 10.10.10.150

```

```

network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

enabled_apis=osapi_compute,metadata
verbose = True

enable_instance_password = True

[database]
connection = mysql+pymysql://nova:654321@10.10.10.180/nova

[oslo_messaging_rabbit]
rabbit_host = 10.10.10.180
rabbit_userid = openstack
rabbit_password = 654321

[keystone_auth]
auth_uri = http:// 10.10.10.180:5000
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = nova
password = 654321

[vnc]
vncserver_listen = 10.10.10.150
vncserver_proxyclient_address = 10.10.10.150

[glance]
host = 10.10.10.180

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

[neutron]
url = http:// 10.10.10.180:9696
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = 654321

service_metadata_proxy = True
metadata_proxy_shared_secret = 654321

```

#### 4. Neutron

Trên `controller1`, chạy xong script `8_neutron_01.sh`, sau đó chạy script `9_neutron_02.sh` trên `controller2` và `controller3`.

Trên cả 3 node, file cấu hình `/etc/neutron/neutron.conf` sẽ như sau

```

[DEFAULT]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True
rpc_backend = rabbit

auth_strategy = keystone

notify_nova_on_port_status_changes = True
notify_nova_on_port_data_changes = True
nova_url = http:// 10.10.10.180:8774/v2

verbose = True

[matchmaker_redis]
[matchmaker_ring]
[quotas]
[agent]
root_helper = sudo /usr/bin/neutron-rootwrap
/etc/neutron/rootwrap.conf

[keystone_authtoken]
auth_uri = http:// 10.10.10.180:5000
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = neutron
password = 654321

[database]
connection = mysql+pymysql://neutron:654321@10.10.10.180/neutron

[nova]
auth_url = http:// 10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = nova
password = 654321

[oslo_concurrency]
lock_path = $state_path/lock
[oslo_policy]
[oslo_messaging_amqp]
[oslo_messaging_qpid]

[oslo_messaging_rabbit]
rabbit_host = 10.10.10.180
rabbit_userid = openstack
rabbit_password = 654321

[qos]

```

Trên cả 3 node, file `/etc/neutron/l3_agent.ini` sẽ như sau:

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

[AGENT]
```

Trên cả 3 nodes, file `/etc/neutron/dhcp_agent.ini` sẽ như sau:

```
[DEFAULT]
interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = True

verbose = True
dnsmasq_config_file = /etc/neutron/dnsmasq-neutron.conf

[AGENT]
```

Trên 3 nodes, file `/etc/neutron/dnsmasq-neutron.conf` sẽ như sau

```
dhcp-option-force=26,1450
```

Trên 3 nodes, file `/etc/neutron/metadata_agent.ini` sẽ như sau:

```
[DEFAULT]
verbose = True

auth_uri = http:// 10.10.10.180:5000
auth_url = http:// 10.10.10.180:35357
auth_region = regionOne
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = neutron
password = 654321

nova_metadata_ip = 10.10.10.180

metadata_proxy_shared_secret = 654321
```

Trên cả 3 nodes, file `/etc/neutron/plugins/ml2/ml2_conf.ini` sẽ như sau:

```
[ml2]
tenant_network_types = vxlan
type_drivers = flat,vlan,vxlan
mechanism_drivers = linuxbridge,l2population
extension_drivers = port_security

[ml2_type_flat]
flat_networks = external

[ml2_type_vlan]
```

```
[ml2_type_gre]
[ml2_type_vxlan]
vni_ranges = 1:1000

[ml2_type_geneve]
[securitygroup]
enable_ipset = True
```

Trên **controller1**, 14 dòng cuối file

`/etc/neutron/plugins/ml2/linuxbridge_agent.ini` sẽ như sau:

```
[linux_bridge]
physical_interface_mappings = external:eth1

[vxlan]
enable_vxlan = True
local_ip = 10.10.10.130
l2_population = True

[agent]
prevent_arp_spoofing = True

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

Trên **controller2**, 14 dòng cuối file

`/etc/neutron/plugins/ml2/linuxbridge_agent.ini` sẽ như sau:

```
[linux_bridge]
physical_interface_mappings = external:eth1

[vxlan]
enable_vxlan = True
local_ip = 10.10.10.140
l2_population = True

[agent]
prevent_arp_spoofing = True

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

Trên **controller3**, 14 dòng cuối file

`/etc/neutron/plugins/ml2/linuxbridge_agent.ini` sẽ như sau:

```
[linux_bridge]
physical_interface_mappings = external:eth1
```

```
[vxlan]
enable_vxlan = True
local_ip = 10.10.10.150
l2_population = True

[agent]
prevent_arp_spoofing = True

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

## 5. Horizon

Chạy script `10_horizon.sh` trên 3 nodes

## V. Cài đặt nova-compute và neutron trên COMPUTE node:

Chạy script `com_script.sh` trên node `compute1`

File cấu hình `/etc/nova/nova.conf` như sau:

```
[DEFAULT]
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
verbose=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

rpc_backend = rabbit
auth_strategy = keystone
my_ip = 10.10.10.120

network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

verbose = True

enable_instance_password = True

[oslo_messaging_rabbit]
rabbit_host = 10.10.10.180
rabbit_userid = openstack
rabbit_password = 654321
```

```

[keystone_authtoken]
auth_uri = http://10.10.10.180:5000
auth_url = http://10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = nova
password = 654321

[vnc]
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = 10.10.10.120
novncproxy_base_url = http://10.10.10.180:6080/vnc_auto.html

[glance]
host = 10.10.10.180

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

[neutron]
url = http://10.10.10.180:9696
auth_url = http://10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = 654321

[libvirt]
inject_key = True
inject_partition = -1
inject_password = True

```

File cấu hình `/etc/neutron/neutron.conf` như sau:

```

[DEFAULT]
core_plugin = ml2

rpc_backend = rabbit
auth_strategy = keystone
verbose = True

[matchmaker_redis]
[matchmaker_ring]
[quotas]
[agent]
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone_authtoken]
auth_uri = http://10.10.10.180:5000
auth_url = http://10.10.10.180:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default

```

```

project_name = service
username = neutron
password = 654321

[database]

[nova]
[oslo_concurrency]
lock_path = $state_path/lock
[oslo_policy]
[oslo_messaging_amqp]
[oslo_messaging_qpid]

[oslo_messaging_rabbit]
rabbit_host = 10.10.10.180
rabbit_userid = openstack
rabbit_password = 654321

[qos]

```

14 dòng cuối file `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` như sau:

```

[linux_bridge]
physical_interface_mappings = public:eth1

[vxlan]
enable_vxlan = True
local_ip = 10.10.10.120
l2_population = True

[agent]
prevent_arp_spoofing = True

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

```

## VI. Kết quả Failover test

Hệ thống vẫn hoạt động bình thường sau một số kịch bản failover test cơ bản sau:

- Fail card eth1
- Fail 1 trong 2 nodes LB01, LB02
- Fail 1 trong 3 nodes controller1, controller2, controller3
- Fail 2 trong 3 nodes controller1, controller2, controller3
- Fail 1 node trong cặp LB01, LB02, và 2 nodes trong 3 nodes controller1, controller2, controller3



Hình ảnh từ kết quả test

System Information - Ops x Statistics Report for HAP... x

10.10.10.180/horizon/admin/info/

Apps joomla Vector Photo Design Font Flash 3d models Color mix TMC

Admin

System

Overview

Hypervisors

Host Aggregates

Instances

Flavors

Images

Networks

Routers

Defaults

Metadata Definitions

System Information

Identity

Services

Compute Services

Network Agents

Filter

Name	Host	Zone	Status	State	Last Updated
nova-cert	controller1	internal	Enabled	Down	8 minutes
nova-consoleauth	controller1	internal	Enabled	Down	9 minutes
nova-scheduler	controller1	internal	Enabled	Down	8 minutes
nova-conductor	controller1	internal	Enabled	Down	9 minutes
nova-cert	controller2	internal	Enabled	Up	0 minutes
nova-consoleauth	controller2	internal	Enabled	Up	0 minutes
nova-scheduler	controller2	internal	Enabled	Up	0 minutes
nova-conductor	controller2	internal	Enabled	Up	0 minutes
nova-cert	controller3	internal	Enabled	Down	1 minute
nova-consoleauth	controller3	internal	Enabled	Down	1 minute
nova-scheduler	controller3	internal	Enabled	Down	1 minute
nova-conductor	controller3	internal	Enabled	Down	1 minute
nova-compute	compute1	nova	Enabled	Up	0 minutes

Displaying 13 items

Version: 8.0.1

System Information - Ops x Statistics Report for HAP... x

10.10.10.180/horizon/admin/instances/

Apps joomla Vector Photo Design Font Flash 3d models Color mix TMC

Admin

System

Overview

Hypervisors

Host Aggregates

Instances

Flavors

Images

Networks

Routers

Defaults

Metadata Definitions

System Information

Identity

Services

Compute Services

Network Agents

Filter

Type	Name	Host	Status	State	Last Updated
L3 agent	neutron-l3-agent	controller1	Enabled	Down	9 minutes
L3 agent	neutron-l3-agent	controller2	Enabled	Up	0 minutes
L3 agent	neutron-l3-agent	controller3	Enabled	Down	1 minute
Metadata agent	neutron-metadata-agent	controller1	Enabled	Down	9 minutes
DHCP agent	neutron-dhcp-agent	controller3	Enabled	Down	1 minute
Metadata agent	neutron-metadata-agent	controller3	Enabled	Down	1 minute
Linux bridge agent	neutron-linuxbridge-agent	controller1	Enabled	Down	9 minutes
DHCP agent	neutron-dhcp-agent	controller2	Enabled	Up	0 minutes
Linux bridge agent	neutron-linuxbridge-agent	controller3	Enabled	Down	1 minute
Linux bridge agent	neutron-linuxbridge-agent	controller2	Enabled	Up	0 minutes
DHCP agent	neutron-dhcp-agent	controller1	Enabled	Down	9 minutes
Linux bridge agent	neutron-linuxbridge-agent	compute1	Enabled	Up	0 minutes
Metadata agent	neutron-metadata-agent	controller2	Enabled	Up	0 minutes

Displaying 13 items

Version: 8.0.1

## Phụ lục 1: File lb\_script.sh

```
#!/bin/bash -ex

if [ $(hostname) = "lb01" ]; then export priority="101"; else export
priority="100"; fi

cat << EOF > /etc/hosts
127.0.0.1 localhost
10.10.10.160 lb01
10.10.10.170 lb02
EOF

apt-get install haproxy keepalived -y

echo "##### Configure keepalived #####"
sleep 3

cat << EOF > /etc/keepalived/keepalived.conf
vrrp_script haproxy {
    script "killall -0 haproxy"
    interval 2
    weight 2
}
vrrp_instance 50 {
    virtual_router_id 50
    advert_int 1
    priority $priority
    state MASTER
    interface eth1
    virtual_ipaddress {
        10.10.10.180/24
    }
    track_script {
        haproxy
    }
}
EOF

service keepalived restart

echo "##### Configure haproxy #####"
sleep 3

haproxyfile=/etc/haproxy/haproxy.cfg
test -f $haproxyfile.orig \
    || cp $haproxyfile $haproxyfile.orig
rm $haproxyfile
touch $haproxyfile

cat << EOF > $haproxyfile
global
    chroot /var/lib/haproxy
    daemon
    group haproxy
    maxconn 4000
    pidfile /var/run/haproxy.pid
    stats socket /var/lib/haproxy/stats
    user haproxy
```

```
defaults
log global
mode tcp
maxconn 4000
option redispatch
retries 3
timeout http-request 10s
timeout queue 1m
timeout connect 10s
timeout client 1m
timeout server 1m
timeout check 10s

listen dashboard *:80
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:80 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:80 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:80 check inter 2000 rise 2 fall 5

listen galera_cluster *:3306
balance source
mode tcp
option tcpka
server controller1 10.10.10.130:3306 check
server controller2 10.10.10.140:3306 backup check
server controller3 10.10.10.150:3306 backup check

listen glance_api *:9292
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:9292 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:9292 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:9292 check inter 2000 rise 2 fall 5

listen glance_registry *:9191
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:9191 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:9191 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:9191 check inter 2000 rise 2 fall 5

listen keystone_admin *:35357
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:35357 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:35357 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:35357 check inter 2000 rise 2 fall 5

listen keystone_public *:5000
balance source
option tcpka
option httpchk
```

```
option tcplog
server controller1 10.10.10.130:5000 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:5000 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:5000 check inter 2000 rise 2 fall 5

listen nova_compute_api *:8774
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:8774 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8774 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8774 check inter 2000 rise 2 fall 5

listen nova_metadata_api *:8775
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:8775 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8775 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8775 check inter 2000 rise 2 fall 5

listen cinder_api *:8776
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:8776 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8776 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8776 check inter 2000 rise 2 fall 5

listen ceilometer_api *:8777
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:8777 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:8777 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8777 check inter 2000 rise 2 fall 5

listen nova_vncproxy *:6080
balance source
option tcpka
option tcplog
server controller1 10.10.10.130:6080 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:6080 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:6080 check inter 2000 rise 2 fall 5

listen neutron_api *:9696
balance source
option tcpka
option httpchk
option tcplog
server controller1 10.10.10.130:9696 check inter 2000 rise 2 fall 5
server controller2 10.10.10.140:9696 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:9696 check inter 2000 rise 2 fall 5

listen swift_proxy *:8080
balance source
option tcplog
option tcpka
server controller1 10.10.10.130:8080 check inter 2000 rise 2 fall 5
```

```

server controller2 10.10.10.140:8080 check inter 2000 rise 2 fall 5
server controller3 10.10.10.150:8080 check inter 2000 rise 2 fall 5

listen rabbitmq *:5672
    balance source
    option clitcpka
    timeout client 900m
    server controller1 10.10.10.130:5672 check inter 1s
    server controller2 10.10.10.140:5672 check inter 1s
    server controller3 10.10.10.150:5672 check inter 1s

listen stats *:1936
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth admin:654321

EOF

service haproxy restart

```

## Phụ lục 2: File config.cfg

```

#!/bin/bash -ex

echo "### Configure hosts file ###"
cat << EOF > /etc/hosts
127.0.0.1 localhost
10.10.10.180 controller
10.10.10.130 controller1
10.10.10.140 controller2
10.10.10.150 controller3
10.10.10.120 compute1
10.10.10.170 lb01
10.10.10.180 lb02
EOF

echo "#### Update for Ubuntu ####"

apt-get install software-properties-common -y
add-apt-repository cloud-archive:liberty -y

sleep 3
echo "##### update for Ubuntu #####"
apt-get update -y && apt-get upgrade -y && apt-get dist-upgrade -y

echo "Install python client"
apt-get -y install python-openstackclient
sleep 5

echo "Install and config NTP"
sleep 3
apt-get install ntp -y
cp /etc/ntp.conf /etc/ntp.conf.bka
rm /etc/ntp.conf
cat /etc/ntp.conf.bka | grep -v ^# | grep -v ^$ >> /etc/ntp.conf

```

```

## Config NTP in LIBERTY
sed -i 's/server ntp.ubuntu.com/ \
server 0.vn.pool.ntp.org iburst \
server 1.asia.pool.ntp.org iburst \
server 2.asia.pool.ntp.org iburst/g' /etc/ntp.conf

sed -i 's/restrict -4 default kod notrap nomodify nopeer noquery/ \
#restrict -4 default kod notrap nomodify nopeer noquery/g' /etc/ntp.conf

sed -i 's/restrict -6 default kod notrap nomodify nopeer noquery/ \
restrict -4 default kod notrap nomodify \
restrict -6 default kod notrap nomodify/g' /etc/ntp.conf
sleep 3

echo "Reboot Server"

#sleep 5
init 6

```

### Phụ lục 3: File 0\_prepare.sh

```

#!/bin/bash -ex

echo "### Configure hosts file ###"
cat << EOF > /etc/hosts
127.0.0.1 localhost
10.10.10.180 controller
10.10.10.130 controller1
10.10.10.140 controller2
10.10.10.150 controller3
10.10.10.120 compute1
10.10.10.170 lb01
10.10.10.180 lb02
EOF

echo "#### Update for Ubuntu ####"

apt-get install software-properties-common -y
add-apt-repository cloud-archive:liberty -y

sleep 3
echo "##### update for Ubuntu #####"
apt-get update -y && apt-get upgrade -y && apt-get dist-upgrade -y

echo "Install python client"
apt-get -y install python-openstackclient
sleep 5

echo "Install and config NTP"
sleep 3
apt-get install ntp -y
cp /etc/ntp.conf /etc/ntp.conf.bka
rm /etc/ntp.conf
cat /etc/ntp.conf.bka | grep -v ^# | grep -v ^$ >> /etc/ntp.conf

## Config NTP in LIBERTY

```

```

sed -i 's/server ntp.ubuntu.com/ \
server 0.vn.pool.ntp.org iburst \
server 1.asia.pool.ntp.org iburst \
server 2.asia.pool.ntp.org iburst/g' /etc/ntp.conf

sed -i 's/restrict -4 default kod notrap nomodify nopeer noquery/ \
#restrict -4 default kod notrap nomodify nopeer noquery/g' /etc/ntp.conf

sed -i 's/restrict -6 default kod notrap nomodify nopeer noquery/ \
restrict -4 default kod notrap nomodify \
restrict -6 default kod notrap nomodify/g' /etc/ntp.conf
sleep 3

echo "Reboot Server"

#sleep 5
init 6

```

## Phụ lục 4: File 1\_galera.sh

```

#!/bin/bash -ex
#
LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`
GAL_IP1=`grep controller1 /etc/hosts | awk '{print $1}'`
GAL_IP2=`grep controller2 /etc/hosts | awk '{print $1}'`
GAL_IP3=`grep controller3 /etc/hosts | awk '{print $1}'`

#####
echo "Install and Config MariaDB"
sleep 3

echo "Enabling the repository"
apt-key adv --recv-keys --keyserver \
    keyserver.ubuntu.com 0xcbb082a1bb943db

touch /etc/apt/sources.list.d/galera.list
cat << EOF > /etc/apt/sources.list.d/galera.list
deb http://mirror.jmu.edu/pub/mariadb/repo/10.0/ubuntu trusty main
EOF

apt-get update
sleep 3

apt-get -y install galera-3 mariadb-galera-server rsync

ln -s /etc/apparmor.d/usr /etc/apparmor.d/disable/.sbin.mysqlld
service apparmor restart
sleep 5

echo "##### Configuring MYSQL #####"
sleep 3

mysqlcfg=/etc/mysql/my.cnf
test -f $mysqlcfg.orig || cp $mysqlcfg $mysqlcfg.orig
rm $mysqlcfg
touch $mysqlcfg

```

```

cat << EOF > $mysqlcfg
[mysqld]
datadir=/var/lib/mysql
user=mysql
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
innodb_flush_log_at_trx_commit=0
innodb_buffer_pool_size=122M
query_cache_type=0
query_cache_size=0
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_provider_options="pc.recovery=TRUE;gcache.size=300M"

# Galera Cluster Configuration
wsrep_cluster_name="Galera_cluster"
wsrep_cluster_address="gcomm://$GAL_IP1,$GAL_IP2,$GAL_IP3"

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="$LOCAL_IP"
wsrep_node_name="$(hostname)"

EOF

service mysql stop
sleep 3

```

## Phụ lục 5: File 2\_keystone\_01.sh

```

#!/bin/bash -ex
#
source config.cfg

LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`

echo "Create Database for Keystone"

cat << EOF | mysql -uroot -p$MYSQL_PASS
CREATE DATABASE keystone;
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY
'$KEYSTONE_DBPASS';
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY
'$KEYSTONE_DBPASS';
FLUSH PRIVILEGES;
EOF

echo "##### Install keystone #####"

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

```



```

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

#/* Back-up file keystone.conf
filekeystone=/etc/keystone/keystone.conf
test -f $filekeystone.orig || cp $filekeystone $filekeystone.orig

#Config file /etc/keystone/keystone.conf
cat << EOF > $filekeystone

[DEFAULT]
log_dir = /var/log/keystone

admin_token = $TOKEN_PASS
bind_host = $VIP_IP
public_bind_host = $VIP_IP
admin_bind_host = $VIP_IP

[assignment]
[auth]
[cache]
[catalog]
driver = keystone.catalog.backends.sql.Catalog
[cors]
[cors.subdomain]
[credential]
[database]
connection = mysql+pymysql://keystone:$KEYSTONE_DBPASS@$VIP_IP/keystone

[domain_config]
[endpoint_filter]
[endpoint_policy]
[eventlet_server]
[eventlet_server_ssl]
[federation]
[fernet_tokens]
[identity]
driver = keystone.identity.backends.sql.Identity
[identity_mapping]
[kvs]
[ldap]
[matchmaker_redis]
[matchmaker_ring]
[memcache]
servers = localhost:11211

[oauth1]
[os_inherit]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_middleware]
[oslo_policy]
[paste_deploy]
[policy]
[resource]
[revoke]
driver = sql

[role]

```

```

[saml]
[signing]
[ssl]
[token]
provider = uuid
driver = sql

[tokenless_auth]
[trust]
[extra_headers]
Distribution = Ubuntu

EOF

#
su -s /bin/sh -c "keystone-manage db_sync" keystone

echo "ServerName $LOCAL_IP" >> /etc/apache2/apache2.conf

cat << EOF > /etc/apache2/sites-available/wsgi-keystone.conf
Listen 5000
Listen 35357

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

    <Directory /usr/bin>
        <IfVersion >= 2.4>
            Require all granted
        </IfVersion>
        <IfVersion < 2.4>
            Order allow,deny
            Allow from all
        </IfVersion>
    </Directory>
</VirtualHost>

<VirtualHost *:35357>
    WSGIDaemonProcess keystone-admin processes=5 threads=1 user=keystone
group=keystone display-name=%{GROUP}
    WSGIProcessGroup keystone-admin
    WSGIScriptAlias / /usr/bin/keystone-wsgi-admin
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
    <IfVersion >= 2.4>
        ErrorLogFormat "%{cu}t %M"
    </IfVersion>
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined

    <Directory /usr/bin>

```

```

        <IfVersion >= 2.4>
            Require all granted
        </IfVersion>
        <IfVersion < 2.4>
            Order allow,deny
            Allow from all
        </IfVersion>
    </Directory>
</VirtualHost>

EOF

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

service apache2 restart

rm -f /var/lib/keystone/keystone.db

export OS_TOKEN="$TOKEN_PASS"
export OS_URL=http://$VIP_IP:35357/v2.0

### Identity service
openstack service create --name keystone --description \
    "OpenStack Identity" identity
### Create the Identity service API endpoint
openstack endpoint create \
--publicurl http://$VIP_IP:5000/v2.0 \
--internalurl http://$VIP_IP:5000/v2.0 \
--adminurl http://$VIP_IP:35357/v2.0 \
--region RegionOne \
identity

#### To create tenants, users, and roles ADMIN
openstack project create --description "Admin Project" admin
openstack user create --password $ADMIN_PASS admin
openstack role create admin
openstack role add --project admin --user admin admin

#### To create tenants, users, and roles SERVICE
openstack project create --description "Service Project" service

#### To create tenants, users, and roles DEMO
openstack project create --description "Demo Project" demo
openstack user create --password $ADMIN_PASS demo

### Create the user role
openstack role create user
openstack role add --project demo --user demo user

#####

unset OS_TOKEN OS_URL

# Tao bien moi truong
cd
echo "export OS_PROJECT_DOMAIN_ID=default" > admin-openrc.sh
echo "export OS_USER_DOMAIN_ID=default" >> admin-openrc.sh
echo "export OS_PROJECT_NAME=admin" >> admin-openrc.sh
echo "export OS_TENANT_NAME=admin" >> admin-openrc.sh
echo "export OS_USERNAME=admin" >> admin-openrc.sh

```

```

echo "export OS_PASSWORD=$ADMIN_PASS" >> admin-openrc.sh
echo "export OS_AUTH_URL=http://$VIP_IP:35357/v3" >> admin-openrc.sh
echo "export OS_VOLUME_API_VERSION=2" >> admin-openrc.sh

sleep 5
echo "##### Execute environment script #####"
chmod +x admin-openrc.sh
cat admin-openrc.sh >> /etc/profile
cp admin-openrc.sh /root/admin-openrc.sh
source admin-openrc.sh

echo "export OS_PROJECT_DOMAIN_ID=default" > demo-openrc.sh
echo "export OS_USER_DOMAIN_ID=default" >> demo-openrc.sh
echo "export OS_PROJECT_NAME=demo" >> demo-openrc.sh
echo "export OS_TENANT_NAME=demo" >> demo-openrc.sh
echo "export OS_USERNAME=demo" >> demo-openrc.sh
echo "export OS_PASSWORD=$ADMIN_PASS" >> demo-openrc.sh
echo "export OS_AUTH_URL=http://$VIP_IP:35357/v3" >> demo-openrc.sh
echo "export OS_VOLUME_API_VERSION=2" >> demo-openrc.sh

chmod +x demo-openrc.sh
cp demo-openrc.sh /root/demo-openrc.sh

echo "#### Verify operation ####"
openstack --os-auth-url http://controller:35357/v3 \
  --os-project-domain-id default --os-user-domain-id default \
  --os-project-name admin --os-username admin --os-auth-type password \
  token issue

openstack --os-auth-url http://controller:5000/v3 \
  --os-project-domain-id default --os-user-domain-id default \
  --os-project-name demo --os-username demo --os-auth-type password \
  token issue

```

## Phụ lục 6: File 3\_keystone\_02.sh

```

#!/bin/bash -ex
#
source config.cfg

LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`

echo "##### Install keystone #####"

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

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

#/* Back-up file keystone.conf
filekeystone=/etc/keystone/keystone.conf
test -f $filekeystone.orig || cp $filekeystone $filekeystone.orig

#Config file /etc/keystone/keystone.conf
cat << EOF > $filekeystone

```

```
[DEFAULT]
log_dir = /var/log/keystone

admin_token = $TOKEN_PASS
bind_host = $VIP_IP
public_bind_host = $VIP_IP
admin_bind_host = $VIP_IP

[assignment]
[auth]
[cache]
[catalog]
driver = keystone.catalog.backends.sql.Catalog
[cors]
[cors.subdomain]
[credential]
[database]
connection = mysql+pymysql://keystone:$KEYSTONE_DBPASS@$VIP_IP/keystone

[domain_config]
[endpoint_filter]
[endpoint_policy]
[eventlet_server]
[eventlet_server_ssl]
[federation]
[fernet_tokens]
[identity]
driver = keystone.identity.backends.sql.Identity
[identity_mapping]
[kvs]
[ldap]
[matchmaker_redis]
[matchmaker_ring]
[memcache]
servers = localhost:11211

[oauth1]
[os_inherit]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_middleware]
[oslo_policy]
[paste_deploy]
[policy]
[resource]
[revoke]
driver = sql

[role]
[saml]
[signing]
[ssl]
[token]
provider = uuid
driver = sql

[tokenless_auth]
[trust]
[extra_headers]
Distribution = Ubuntu
```

EOF

```
echo "ServerName $LOCAL_IP" >> /etc/apache2/apache2.conf
```

```
cat << EOF > /etc/apache2/sites-available/wsgi-keystone.conf
```

```
Listen 5000
```

```
Listen 35357
```

```
<VirtualHost *:5000>
```

```
    WSGIDaemonProcess keystone-public processes=5 threads=1 user=keystone  
    group=keystone display-name=%{GROUP}
```

```
    WSGIProcessGroup keystone-public
```

```
    WSGIScriptAlias / /usr/bin/keystone-wsgi-public
```

```
    WSGIApplicationGroup %{GLOBAL}
```

```
    WSGIPassAuthorization On
```

```
    <IfVersion >= 2.4>
```

```
        ErrorLogFormat "%{cu}t %M"
```

```
    </IfVersion>
```

```
    ErrorLog /var/log/apache2/keystone.log
```

```
    CustomLog /var/log/apache2/keystone_access.log combined
```

```
    <Directory /usr/bin>
```

```
        <IfVersion >= 2.4>
```

```
            Require all granted
```

```
        </IfVersion>
```

```
        <IfVersion < 2.4>
```

```
            Order allow,deny
```

```
            Allow from all
```

```
        </IfVersion>
```

```
    </Directory>
```

```
</VirtualHost>
```

```
<VirtualHost *:35357>
```

```
    WSGIDaemonProcess keystone-admin processes=5 threads=1 user=keystone  
    group=keystone display-name=%{GROUP}
```

```
    WSGIProcessGroup keystone-admin
```

```
    WSGIScriptAlias / /usr/bin/keystone-wsgi-admin
```

```
    WSGIApplicationGroup %{GLOBAL}
```

```
    WSGIPassAuthorization On
```

```
    <IfVersion >= 2.4>
```

```
        ErrorLogFormat "%{cu}t %M"
```

```
    </IfVersion>
```

```
    ErrorLog /var/log/apache2/keystone.log
```

```
    CustomLog /var/log/apache2/keystone_access.log combined
```

```
    <Directory /usr/bin>
```

```
        <IfVersion >= 2.4>
```

```
            Require all granted
```

```
        </IfVersion>
```

```
        <IfVersion < 2.4>
```

```
            Order allow,deny
```

```
            Allow from all
```

```
        </IfVersion>
```

```
    </Directory>
```

```
</VirtualHost>
```

EOF

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

```

service apache2 restart

rm -f /var/lib/keystone/keystone.db

# Tao bien moi truong
cd
echo "export OS_PROJECT_DOMAIN_ID=default" > admin-openrc.sh
echo "export OS_USER_DOMAIN_ID=default" >> admin-openrc.sh
echo "export OS_PROJECT_NAME=admin" >> admin-openrc.sh
echo "export OS_TENANT_NAME=admin" >> admin-openrc.sh
echo "export OS_USERNAME=admin" >> admin-openrc.sh
echo "export OS_PASSWORD=$ADMIN_PASS" >> admin-openrc.sh
echo "export OS_AUTH_URL=http://$VIP_IP:35357/v3" >> admin-openrc.sh
echo "export OS_VOLUME_API_VERSION=2" >> admin-openrc.sh

sleep 5
echo "##### Execute environment script #####"
chmod +x admin-openrc.sh
cat admin-openrc.sh >> /etc/profile
cp admin-openrc.sh /root/admin-openrc.sh
source admin-openrc.sh

echo "export OS_PROJECT_DOMAIN_ID=default" > demo-openrc.sh
echo "export OS_USER_DOMAIN_ID=default" >> demo-openrc.sh
echo "export OS_PROJECT_NAME=demo" >> demo-openrc.sh
echo "export OS_TENANT_NAME=demo" >> demo-openrc.sh
echo "export OS_USERNAME=demo" >> demo-openrc.sh
echo "export OS_PASSWORD=$ADMIN_PASS" >> demo-openrc.sh
echo "export OS_AUTH_URL=http://$VIP_IP:35357/v3" >> demo-openrc.sh
echo "export OS_VOLUME_API_VERSION=2" >> demo-openrc.sh

chmod +x demo-openrc.sh
cp demo-openrc.sh /root/demo-openrc.sh

echo "#### Verify operation ####"
openstack --os-auth-url http://controller:35357/v3 \
  --os-project-domain-id default --os-user-domain-id default \
  --os-project-name admin --os-username admin --os-auth-type password \
  token issue

openstack --os-auth-url http://controller:5000/v3 \
  --os-project-domain-id default --os-user-domain-id default \
  --os-project-name demo --os-username demo --os-auth-type password \
  token issue

```

## Phụ lục 7: File 4\_glance\_01.sh

```

#!/bin/bash -ex
#
source config.cfg

echo "Create the database for GLANCE"
cat << EOF | mysql -uroot -p$MYSQL_PASS
CREATE DATABASE glance;
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY
'$GLANCE_DBPASS';

```

```

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

sleep 5
echo " Create user, endpoint for GLANCE"

openstack user create --password $ADMIN_PASS glance
openstack role add --project service --user glance admin
openstack service create --name glance --description \
    "OpenStack Image service" image

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

echo "##### Install GLANCE #####"
apt-get -y install glance python-glanceclient
sleep 10
echo "##### Configuring GLANCE API #####"
sleep 5
#!/* Back-up file glance-api.conf
fileglanceapicontrol=/etc/glance/glance-api.conf
test -f $fileglanceapicontrol.orig \
    || cp $fileglanceapicontrol $fileglanceapicontrol.orig
rm $fileglanceapicontrol
touch $fileglanceapicontrol

#Configuring glance config file /etc/glance/glance-api.conf

cat << EOF > $fileglanceapicontrol
[DEFAULT]
notification_driver = noop
verbose = True

[database]
connection = mysql+pymysql://glance:$GLANCE_DBPASS@$VIP_IP/glance
backend = sqlalchemy

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

[image_format]
[keystone_authtoken]
auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = $GLANCE_PASS

[matchmaker_redis]
[matchmaker_ring]
[oslo_concurrency]

```



```

[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_policy]
[paste_deploy]
flavor = keystone

[store_type_location_strategy]
[task]
[taskflow_executor]

EOF

#
sleep 10
echo "##### Configuring GLANCE REGISTER #####"
#/* Backup file file glance-registry.conf
fileglanceregcontrol=/etc/glance/glance-registry.conf
test -f $fileglanceregcontrol.orig \
    || cp $fileglanceregcontrol $fileglanceregcontrol.orig
rm $fileglanceregcontrol
touch $fileglanceregcontrol
#Config file /etc/glance/glance-registry.conf

cat << EOF > $fileglanceregcontrol

[DEFAULT]
notification_driver = noop
verbose = True

[database]
connection = mysql+pymysql://glance:$GLANCE_DBPASS@$VIP_IP/glance
backend = sqlalchemy

[glance_store]

[keystone_authtoken]
auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = $GLANCE_PASS

[matchmaker_redis]
[matchmaker_ring]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_policy]

[paste_deploy]
flavor = keystone

EOF

sleep 7
echo "##### Remove Glance default DB #####"
rm /var/lib/glance/glance.sqlite

```

```

chown glance:glance $fileglanceapicontrol
chown glance:glance $fileglanceregcontrol

sleep 7
echo "##### Syncing DB for Glance #####"
su -s /bin/sh -c "glance-manage db_sync" glance

sleep 5
echo "##### Restarting GLANCE service ... #####"
service glance-registry restart
service glance-api restart
sleep 3
service glance-registry restart
service glance-api restart

#
echo "export OS_IMAGE_API_VERSION=2" \
    | tee -a ~/admin-openrc.sh ~/demo-openrc.sh

echo "Remove glance.sqlite "
rm -f /var/lib/glance/glance.sqlite

sleep 3
echo "##### Registering Cirros IMAGE for GLANCE ... #####"
mkdir images
cd images/
wget http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img

glance image-create --name "cirros" \
--file cirros-0.3.4-x86_64-disk.img \
--disk-format qcow2 --container-format bare \
--visibility public --progress
cd /root/
# rm -r /tmp/images

sleep 5
echo "##### Testing Glance #####"
glance image-list

```

## Phụ lục 8: File 5\_glance\_02.sh

```

#!/bin/bash -ex
#
source config.cfg

echo "##### Install GLANCE #####"
apt-get -y install glance python-glanceclient
sleep 10
echo "##### Configuring GLANCE API #####"
sleep 5
#/* Back-up file glance-api.conf
fileglanceapicontrol=/etc/glance/glance-api.conf
test -f $fileglanceapicontrol.orig \
    || cp $fileglanceapicontrol $fileglanceapicontrol.orig
rm $fileglanceapicontrol
touch $fileglanceapicontrol

```

```

#Configuring glance config file /etc/glance/glance-api.conf

cat << EOF > $fileglanceapicontrol
[DEFAULT]
notification_driver = noop
verbose = True

[database]
connection = mysql+pymysql://glance:$GLANCE_DBPASS@$VIP_IP/glance
backend = sqlalchemy

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

[image_format]
[keystone_authtoken]
auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = $GLANCE_PASS

[matchmaker_redis]
[matchmaker_ring]
[oslo_concurrency]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_policy]
[paste_deploy]
flavor = keystone

[store_type_location_strategy]
[task]
[taskflow_executor]

EOF

#
sleep 10
echo "##### Configuring GLANCE REGISTER #####"
#/* Backup file file glance-registry.conf
fileglanceregcontrol=/etc/glance/glance-registry.conf
test -f $fileglanceregcontrol.orig \
    || cp $fileglanceregcontrol $fileglanceregcontrol.orig
rm $fileglanceregcontrol
touch $fileglanceregcontrol
#Config file /etc/glance/glance-registry.conf

cat << EOF > $fileglanceregcontrol

[DEFAULT]
notification_driver = noop
verbose = True

[database]

```

```

connection = mysql+pymysql://glance:$GLANCE_DBPASS@$VIP_IP/glance
backend = sqlalchemy

[glance_store]

[keystone_authtoken]
auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = $GLANCE_PASS

[matchmaker_redis]
[matchmaker_ring]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
[oslo_policy]

[paste_deploy]
flavor = keystone

EOF

sleep 7
echo "##### Remove Glance default DB #####"
rm /var/lib/glance/glance.sqlite

chown glance:glance $fileglanceapicontrol
chown glance:glance $fileglanceregcontrol

echo "##### Restarting GLANCE service ... #####"
service glance-registry restart
service glance-api restart
sleep 3
service glance-registry restart
service glance-api restart

#
echo "export OS_IMAGE_API_VERSION=2" \
    | tee -a ~/admin-openrc.sh ~/demo-openrc.sh

echo "Remove glance.sqlite "
rm -f /var/lib/glance/glance.sqlite

```

## Phụ lục 9: File 6\_nova\_01.sh

```

#!/bin/bash -ex
#
source config.cfg

LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`

```

```

echo "Create DB for NOVA "
cat << EOF | mysql -uroot -p$MYSQL_PASS
CREATE DATABASE nova;
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY
'$NOVA_DBPASS';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY '$NOVA_DBPASS';
FLUSH PRIVILEGES;
EOF

echo "Create user, endpoint for NOVA"

openstack user create --password $ADMIN_PASS nova
openstack role add --project service --user nova admin
openstack service create --name nova --description "OpenStack Compute"
compute

openstack endpoint create \
--publicurl http://$VIP_IP:8774/v2/%(tenant_id)s \
--internalurl http://$VIP_IP:8774/v2/%(tenant_id)s \
--adminurl http://$VIP_IP:8774/v2/%(tenant_id)s \
--region RegionOne \
compute

echo "##### Install NOVA in $VIP_IP #####"
sleep 5
apt-get -y install nova-api nova-cert nova-conductor nova-consoleauth \
nova-novncproxy nova-scheduler python-novaclient

# Cai tu dong libguestfs-tools
apt-get -y install libguestfs-tools sysfsutils guestfsd python-guestfs

##### Backup configurations for NOVA #####
sleep 7

#
controlnova=/etc/nova/nova.conf
test -f $controlnova.orig || cp $controlnova $controlnova.orig
rm $controlnova
touch $controlnova
cat << EOF >> $controlnova
[DEFAULT]

rpc_backend = rabbit
auth_strategy = keystone

dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

my_ip = $LOCAL_IP

network_api_class = nova.network.neutronv2.api.API

```

```

security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

enabled_apis=osapi_compute,metadata
verbose = True

enable_instance_password = True

[database]
connection = mysql+pymysql://nova:$NOVA_DBPASS@$VIP_IP/nova

[oslo_messaging_rabbit]
rabbit_host = $VIP_IP
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS

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

[vnc]
vncserver_listen = $my_ip
vncserver_proxyclient_address = $my_ip

[glance]
host = $VIP_IP

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

[neutron]
url = http://$VIP_IP:9696
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = $NEUTRON_PASS

service_metadata_proxy = True
metadata_proxy_shared_secret = $METADATA_SECRET

EOF

echo "##### Remove Nova default db #####"
sleep 7
rm /var/lib/nova/nova.sqlite

echo "##### Syncing Nova DB #####"
sleep 7

```

```

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

# echo 'kvm_intel' >> /etc/modules

echo "##### Restarting NOVA ... #####"
sleep 7
service nova-api restart
service nova-cert restart
service nova-consoleauth restart
service nova-scheduler restart
service nova-conductor restart
service nova-novncproxy restart

sleep 7
echo "##### Restarting NOVA ... #####"
service nova-api restart
service nova-cert restart
service nova-consoleauth restart
service nova-scheduler restart
service nova-conductor restart
service nova-novncproxy restart

echo "##### Testing NOVA service #####"
nova-manage service list

```

## Phụ lục 10: File 7\_nova\_02.sh

```

#!/bin/bash -ex
#
source config.cfg

LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`

echo "##### Install NOVA in $VIP_IP #####"
sleep 5
apt-get -y install nova-api nova-cert nova-conductor nova-consoleauth \
    nova-novncproxy nova-scheduler python-novaclient

# Cai tu dong libguestfs-tools
apt-get -y install libguestfs-tools sysfsutils guestfsd python-guestfs

##### Backup configurations for NOVA #####
sleep 7

#
controlnova=/etc/nova/nova.conf
test -f $controlnova.orig || cp $controlnova $controlnova.orig
rm $controlnova
touch $controlnova
cat << EOF >> $controlnova
[DEFAULT]

rpc_backend = rabbit
auth_strategy = keystone

dhcpbridge_flagfile=/etc/nova/nova.conf

```

```

dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

my_ip = $LOCAL_IP

network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

enabled_apis=osapi_compute,metadata
verbose = True

enable_instance_password = True

[database]
connection = mysql+pymysql://nova:$NOVA_DBPASS@$VIP_IP/nova

[oslo_messaging_rabbit]
rabbit_host = $VIP_IP
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS

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

[vnc]
vncserver_listen = $my_ip
vncserver_proxyclient_address = $my_ip

[glance]
host = $VIP_IP

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

[neutron]
url = http://$VIP_IP:9696
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = $NEUTRON_PASS

```



```

service_metadata_proxy = True
metadata_proxy_shared_secret = $METADATA_SECRET

EOF

echo "##### Remove Nova default db #####"
sleep 7
rm /var/lib/nova/nova.sqlite

# echo 'kvm_intel' >> /etc/modules

echo "##### Restarting NOVA ... #####"
sleep 7
service nova-api restart
service nova-cert restart
service nova-consoleauth restart
service nova-scheduler restart
service nova-conductor restart
service nova-novncproxy restart

sleep 7
echo "##### Restarting NOVA ... #####"
service nova-api restart
service nova-cert restart
service nova-consoleauth restart
service nova-scheduler restart
service nova-conductor restart
service nova-novncproxy restart

echo "##### Testing NOVA service #####"
nova-manage service list

```

## Phụ lục 11: File 8\_neutron\_01.sh

```

#!/bin/bash -ex
#

source config.cfg

LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`

echo "Create DB for NEUTRON "
cat << EOF | mysql -uroot -p$MYSQL_PASS
CREATE DATABASE neutron;
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' IDENTIFIED BY
'$NEUTRON_DBPASS';
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY
'$NEUTRON_DBPASS';
FLUSH PRIVILEGES;
EOF

echo "Create user, endpoint for NEUTRON"
openstack user create --password $ADMIN_PASS neutron
openstack role add --project service --user neutron admin
openstack service create --name neutron --description \

```

```

"OpenStack Networking" network

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

echo "##### Install NEUTRON in $VIP_IP or NETWORK node #####"
sleep 5
apt-get -y install neutron-server neutron-plugin-ml2 \
neutron-plugin-linuxbridge-agent neutron-l3-agent neutron-dhcp-agent \
neutron-metadata-agent python-neutronclient

##### Backup configuration NEUTRON.CONF in $VIP_IP#####
echo "##### Config NEUTRON in $VIP_IP/NETWORK node #####"
sleep 7

#
controlneutron=/etc/neutron/neutron.conf
test -f $controlneutron.orig || cp $controlneutron $controlneutron.orig
rm $controlneutron
touch $controlneutron
cat << EOF >> $controlneutron
[DEFAULT]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True
rpc_backend = rabbit

auth_strategy = keystone

notify_nova_on_port_status_changes = True
notify_nova_on_port_data_changes = True
nova_url = http://$VIP_IP:8774/v2

verbose = True

[matchmaker_redis]
[matchmaker_ring]
[quotas]
[agent]
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone_authtoken]
auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = neutron
password = $NEUTRON_PASS

[database]
connection = mysql+pymysql://neutron:$NEUTRON_DBPASS@$VIP_IP/neutron

[nova]
auth_url = http://$VIP_IP:35357
auth_plugin = password

```

```

project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = nova
password = $NOVA_PASS

[oslo_concurrency]
lock_path = \${state_path}/lock
[oslo_policy]
[oslo_messaging_amqp]
[oslo_messaging_qpid]

[oslo_messaging_rabbit]
rabbit_host = $VIP_IP
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS

[qos]

EOF

##### Backup configuration of ML2 in $VIP_IP#####
echo "##### Configuring ML2 in $VIP_IP/NETWORK node #####"
sleep 7

controlML2=/etc/neutron/plugins/ml2/ml2_conf.ini
test -f $controlML2.orig || cp $controlML2 $controlML2.orig
rm $controlML2
touch $controlML2

cat << EOF >> $controlML2
[ml2]
tenant_network_types = vxlan
type_drivers = flat,vlan,vxlan
mechanism_drivers = linuxbridge,l2population
extension_drivers = port_security

[ml2_type_flat]
flat_networks = external

[ml2_type_vlan]

[ml2_type_gre]
[ml2_type_vxlan]
vni_ranges = 1:1000

[ml2_type_geneve]
[securitygroup]
enable_ipset = True

EOF

echo "##### Configuring Linux Bbridge AGENT #####"
sleep 7

linuxbridgefile=/etc/neutron/plugins/ml2/linuxbridge_agent.ini

test -f $linuxbridgefile.orig || cp $linuxbridgefile $linuxbridgefile.orig

cat << EOF >> $linuxbridgefile

```

```

[linux_bridge]
physical_interface_mappings = external:eth1

[vxlan]
enable_vxlan = True
local_ip = $LOCAL_IP
l2_population = True

[agent]
prevent_arp_spoofing = True

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

EOF

echo "##### Configuring L3 AGENT #####"
sleep 7
netl3agent=/etc/neutron/l3_agent.ini

test -f $netl3agent.orig || cp $netl3agent $netl3agent.orig
rm $netl3agent
touch $netl3agent

cat << EOF >> $netl3agent
[DEFAULT]
interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver
external_network_bridge =
verbose = True

[AGENT]

EOF

echo "##### Configuring DHCP AGENT #####"
sleep 7
#
netdhcp=/etc/neutron/dhcp_agent.ini

test -f $netdhcp.orig || cp $netdhcp $netdhcp.orig
rm $netdhcp
touch $netdhcp

cat << EOF >> $netdhcp
[DEFAULT]
interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = True

verbose = True
dnsmasq_config_file = /etc/neutron/dnsmasq-neutron.conf

[AGENT]

EOF

echo "Fix lo MTU"
sleep 3
echo "dhcp-option-force=26,1450" > /etc/neutron/dnsmasq-neutron.conf

```

```

killall dnsmasq

echo "##### Configuring METADATA AGENT #####"
sleep 7
netmetadata=/etc/neutron/metadata_agent.ini

test -f $netmetadata.orig || cp $netmetadata $netmetadata.orig
rm $netmetadata
touch $netmetadata

cat << EOF >> $netmetadata
[DEFAULT]
verbose = True

auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP: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 = $VIP_IP

metadata_proxy_shared_secret = $METADATA_SECRET

EOF
#

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

echo "##### Restarting NOVA service #####"
sleep 7
service nova-api restart
service nova-scheduler restart
service nova-conductor restart

echo "##### Restarting NEUTRON service #####"
sleep 7
service neutron-server restart
service neutron-plugin-linuxbridge-agent restart
service neutron-dhcp-agent restart
service neutron-metadata-agent restart
service neutron-l3-agent restart

rm -f /var/lib/neutron/neutron.sqlite

echo "##### Verify operation #####"
neutron agent-list

```

## Phụ lục 12: File 9\_neutron\_02.sh

```

#!/bin/bash -ex
#

source config.cfg

LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`

echo "##### Install NEUTRON in $VIP_IP or NETWORK node #####"
sleep 5
apt-get -y install neutron-server neutron-plugin-ml2 \
neutron-plugin-linuxbridge-agent neutron-l3-agent neutron-dhcp-agent \
neutron-metadata-agent python-neutronclient

##### Backup configuration NEUTRON.CONF in $VIP_IP#####
echo "##### Config NEUTRON in $VIP_IP/NETWORK node #####"
sleep 7

#
controlneutron=/etc/neutron/neutron.conf
test -f $controlneutron.orig || cp $controlneutron $controlneutron.orig
rm $controlneutron
touch $controlneutron
cat << EOF >> $controlneutron
[DEFAULT]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True
rpc_backend = rabbit

auth_strategy = keystone

notify_nova_on_port_status_changes = True
notify_nova_on_port_data_changes = True
nova_url = http://$VIP_IP:8774/v2

verbose = True

[matchmaker_redis]
[matchmaker_ring]
[quotas]
[agent]
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone_authtoken]
auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = neutron
password = $NEUTRON_PASS

[database]
connection = mysql+pymysql://neutron:$NEUTRON_DBPASS@$VIP_IP/neutron

[nova]
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default

```

```

user_domain_id = default
region_name = RegionOne
project_name = service
username = nova
password = $NOVA_PASS

[oslo_concurrency]
lock_path = \${state_path}/lock
[oslo_policy]
[oslo_messaging_amqp]
[oslo_messaging_qpid]

[oslo_messaging_rabbit]
rabbit_host = $VIP_IP
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS

[qos]

EOF

##### Backup configuration of ML2 in $VIP_IP#####
echo "##### Configuring ML2 in $VIP_IP/NETWORK node #####"
sleep 7

controlML2=/etc/neutron/plugins/ml2/ml2_conf.ini
test -f $controlML2.orig || cp $controlML2 $controlML2.orig
rm $controlML2
touch $controlML2

cat << EOF >> $controlML2
[ml2]
tenant_network_types = vxlan
type_drivers = flat,vlan,vxlan
mechanism_drivers = linuxbridge,l2population
extension_drivers = port_security

[ml2_type_flat]
flat_networks = external

[ml2_type_vlan]

[ml2_type_gre]
[ml2_type_vxlan]
vni_ranges = 1:1000

[ml2_type_geneve]
[securitygroup]
enable_ipset = True

EOF

echo "##### Configuring Linux Bbridge AGENT #####"
sleep 7

linuxbridgefile=/etc/neutron/plugins/ml2/linuxbridge_agent.ini

test -f $linuxbridgefile.orig || cp $linuxbridgefile $linuxbridgefile.orig

cat << EOF >> $linuxbridgefile
[linux_bridge]

```

```

physical_interface_mappings = external:eth1

[vxlan]
enable_vxlan = True
local_ip = $LOCAL_IP
l2_population = True

[agent]
prevent_arp_spoofing = True

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

EOF

echo "##### Configuring L3 AGENT #####"
sleep 7
netl3agent=/etc/neutron/l3_agent.ini

test -f $netl3agent.orig || cp $netl3agent $netl3agent.orig
rm $netl3agent
touch $netl3agent

cat << EOF >> $netl3agent
[DEFAULT]
interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver
external_network_bridge =
verbose = True

[AGENT]

EOF

echo "##### Configuring DHCP AGENT #####"
sleep 7
#
netdhcp=/etc/neutron/dhcp_agent.ini

test -f $netdhcp.orig || cp $netdhcp $netdhcp.orig
rm $netdhcp
touch $netdhcp

cat << EOF >> $netdhcp
[DEFAULT]
interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = True

verbose = True
dnsmasq_config_file = /etc/neutron/dnsmasq-neutron.conf

[AGENT]

EOF

echo "Fix lo MTU"
sleep 3
echo "dhcp-option-force=26,1450" > /etc/neutron/dnsmasq-neutron.conf
killall dnsmasq

```



```

echo "##### Configuring METADATA AGENT #####"
sleep 7
netmetadata=/etc/neutron/metadata_agent.ini

test -f $netmetadata.orig || cp $netmetadata $netmetadata.orig
rm $netmetadata
touch $netmetadata

cat << EOF >> $netmetadata
[DEFAULT]
verbose = True

auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP: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 = $VIP_IP

metadata_proxy_shared_secret = $METADATA_SECRET

EOF
#

echo "##### Restarting NOVA service #####"
sleep 7
service nova-api restart
service nova-scheduler restart
service nova-conductor restart

echo "##### Restarting NEUTRON service #####"
sleep 7
service neutron-server restart
service neutron-plugin-linuxbridge-agent restart
service neutron-dhcp-agent restart
service neutron-metadata-agent restart
service neutron-l3-agent restart

rm -f /var/lib/neutron/neutron.sqlite

echo "##### Verify operation #####"
neutron agent-list

```

## Phụ lục 13: File 10\_horizon.sh

```

#!/bin/bash -ex

source config.cfg

#####
echo "##### START INSTALLING OPS DASHBOARD #####"

```

```
#####
sleep 5

echo "##### Installing Dashboard package #####"
apt-get -y install openstack-dashboard
apt-get -y remove --auto-remove openstack-dashboard-ubuntu-theme

# echo "##### Fix bug in apache2 #####"
# sleep 5
# Fix bug apache in ubuntu 14.04
# echo "ServerName localhost" > /etc/apache2/conf-
available/servername.conf
# sudo a2enconf servername

echo "##### Creating redirect page #####"

filehtml=/var/www/html/index.html
test -f $filehtml.orig || cp $filehtml $filehtml.orig
rm $filehtml
touch $filehtml
cat << EOF >> $filehtml
<html>
<head>
<META HTTP-EQUIV="Refresh" Content="0.5; URL=http://$VIP_IP/horizon">
</head>
<body>
<center> <h1>Dang chuyen den Dashboard cua OpenStack</h1> </center>
</body>
</html>
EOF
# Allowing insert password in dashboard ( only apply in image )
sed -i "s/'can_set_password': False/'can_set_password': True/g" \
    /etc/openstack-dashboard/local_settings.py

## /* Restarting apache2 and memcached
service apache2 restart
service memcached restart
echo "##### Finish setting up Horizon #####"

echo "##### LOGIN INFORMATION IN HORIZON #####"
echo "URL: http://$VIP_IP/horizon"
echo "User: admin or demo"
echo "Password:" $ADMIN_PASS
```

## Phụ lục 14: File com\_script.sh

```
#!/bin/bash -ex

source config.cfg

LOCAL_IP=`ifconfig eth1 | grep 'inet addr' | cut -d: -f2 | awk '{print $1}'`

echo "### Configure hosts file ###"
cat << EOF > /etc/hosts
127.0.0.1 localhost
```

```

10.10.10.180 controller
10.10.10.130 controller1
10.10.10.140 controller2
10.10.10.150 controller3
10.10.10.120 compute1
10.10.10.170 lb01
10.10.10.180 lb02
EOF

echo "#### Update for Ubuntu ####"

apt-get install software-properties-common -y
add-apt-repository cloud-archive:liberty -y

sleep 3
echo "##### update for Ubuntu #####"
apt-get update -y && apt-get upgrade -y && apt-get dist-upgrade -y

echo "Install python client"
apt-get -y install python-openstackclient
sleep 5

echo "Install and config NTP"
sleep 3
apt-get install ntp -y
cp /etc/ntp.conf /etc/ntp.conf.bka
rm /etc/ntp.conf
cat /etc/ntp.conf.bka | grep -v ^# | grep -v ^$ >> /etc/ntp.conf

## Config NTP in LIBERTY
sed -i 's/server ntp.ubuntu.com/ \
server 0.vn.pool.ntp.org iburst \
server 1.asia.pool.ntp.org iburst \
server 2.asia.pool.ntp.org iburst/g' /etc/ntp.conf

sed -i 's/restrict -4 default kod notrap nomodify nopeer noquery/ \
#restrict -4 default kod notrap nomodify nopeer noquery/g' /etc/ntp.conf

sed -i 's/restrict -6 default kod notrap nomodify nopeer noquery/ \
restrict -4 default kod notrap nomodify \
restrict -6 default kod notrap nomodify/g' /etc/ntp.conf
sleep 3

echo "##### Installl package for NOVA"
apt-get -y install nova-compute
echo "libguestfs-tools libguestfs/update-appliance boolean true" \
| debconf-set-selections

apt-get -y install libguestfs-tools sysfsutils guestfsd python-guestfs

#fix loi chen pass tren hypervisor la KVM
update-guestfs-appliance
chmod 0644 /boot/vmlinuz*
usermod -a -G kvm root

echo "##### Configuring in nova.conf ...#####"
sleep 5
#####
#/* Sao luu truoc khi sua file nova.conf
filenova=/etc/nova/nova.conf
test -f $filenova.orig || cp $filenova $filenova.orig

```

```

#Chen noi dung file /etc/nova/nova.conf vao
cat << EOF > $filenova
[DEFAULT]
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
force_dhcp_release=True
libvirt_use_virtio_for_bridges=True
verbose=True
ec2_private_dns_show_ip=True
api_paste_config=/etc/nova/api-paste.ini
enabled_apis=ec2,osapi_compute,metadata

rpc_backend = rabbit
auth_strategy = keystone
my_ip = $LOCAL_IP

network_api_class = nova.network.neutronv2.api.API
security_group_api = neutron
linuxnet_interface_driver =
nova.network.linux_net.NeutronLinuxBridgeInterfaceDriver
firewall_driver = nova.virt.firewall.NoopFirewallDriver

verbose = True

enable_instance_password = True

[oslo_messaging_rabbit]
rabbit_host = $VIP_IP
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS

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

[vnc]
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://$VIP_IP:6080/vnc_auto.html

[glance]
host = $VIP_IP

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

[neutron]
url = http://$VIP_IP:9696
auth_url = http://$VIP_IP:35357
auth_plugin = password

```

```
project_domain_id = default
user_domain_id = default
region_name = RegionOne
project_name = service
username = neutron
password = $NEUTRON_PASS

[libvirt]
inject_key = True
inject_partition = -1
inject_password = True

EOF

echo "##### Restart nova-compute #####"
sleep 5
service nova-compute restart

# Remove default nova db
rm /var/lib/nova/nova.sqlite

echo "##### Install linuxbridge-agent (neutron) on COMPUTE NODE #####"
sleep 10

apt-get -y install neutron-plugin-linuxbridge-agent

echo "Config file neutron.conf"
controlneutron=/etc/neutron/neutron.conf
test -f $controlneutron.orig || cp $controlneutron $controlneutron.orig
rm $controlneutron
touch $controlneutron
cat << EOF >> $controlneutron
[DEFAULT]
core_plugin = ml2

rpc_backend = rabbit
auth_strategy = keystone
verbose = True

[matchmaker_redis]
[matchmaker_ring]
[quotas]
[agent]
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone_authtoken]
auth_uri = http://$VIP_IP:5000
auth_url = http://$VIP_IP:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = neutron
password = $KEYSTONE_PASS

[database]

[nova]
[oslo_concurrency]
lock_path = \state_path/lock
[oslo_policy]
```

```
[oslo_messaging_amqp]
[oslo_messaging_qpid]

[oslo_messaging_rabbit]
rabbit_host = $VIP_IP
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS

[qos]
EOF

echo "##### Configuring Linux Bbridge AGENT #####"
sleep 7

linuxbridgefile=/etc/neutron/plugins/ml2/linuxbridge_agent.ini

test -f $linuxbridgefile.orig || cp $linuxbridgefile $linuxbridgefile.orig

cat << EOF >> $linuxbridgefile
[linux_bridge]
physical_interface_mappings = public:eth1

[vxlan]
enable_vxlan = True
local_ip = $LOCAL_IP
l2_population = True

[agent]
prevent_arp_spoofing = True

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

EOF

echo "Reset service nova-compute,linuxbridge-agent"
sleep 5
service nova-compute restart
service neutron-plugin-linuxbridge-agent restart
```