OPENSTACK HIGH-AVAILABLE CONTROLLERS

(Liberty version on Ubuntu trusty release)

Version 1.2

Created date 01/04/2016

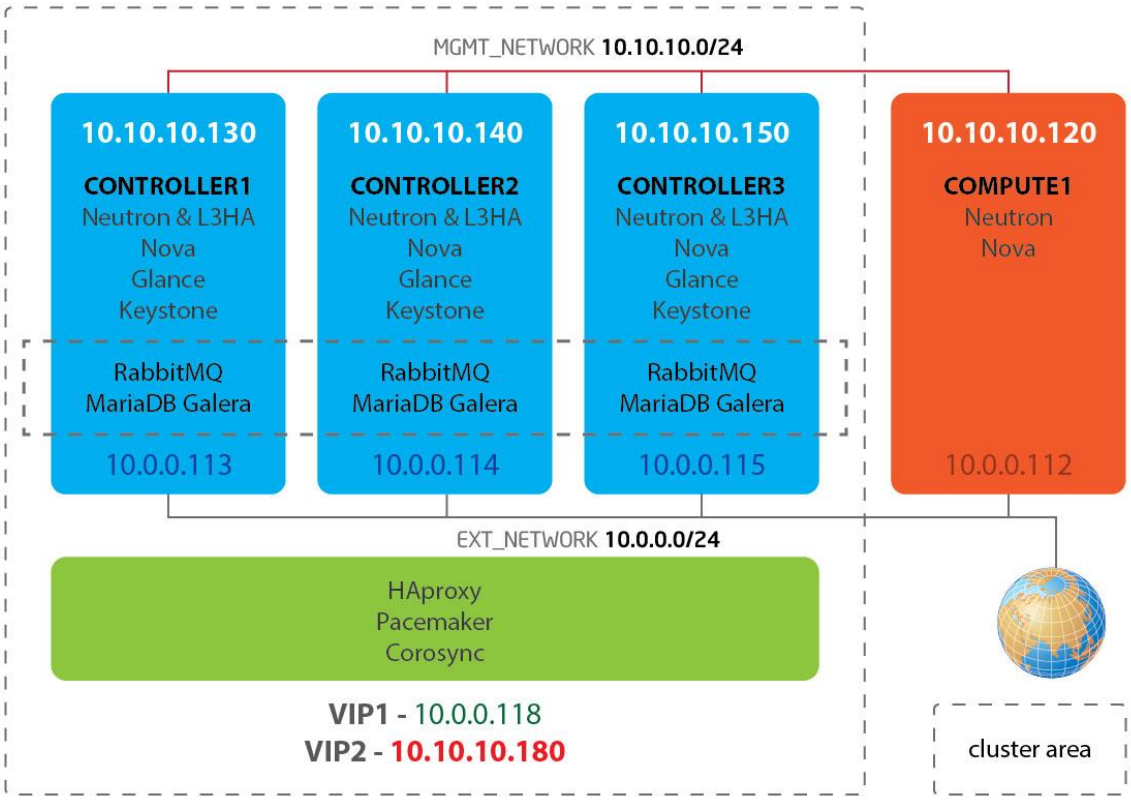
Created by toanchik

Contents

Mình xin chia sẻ document triển khai bài lab 1.2, so với bài lab 1.1 với vài thay đổi sau:

- Triển khai `Pacemaker/Corosync, HAproxy` trên 03 controller nodes thay thế cho 2 loadbalancing nodes LB_01, LB_02

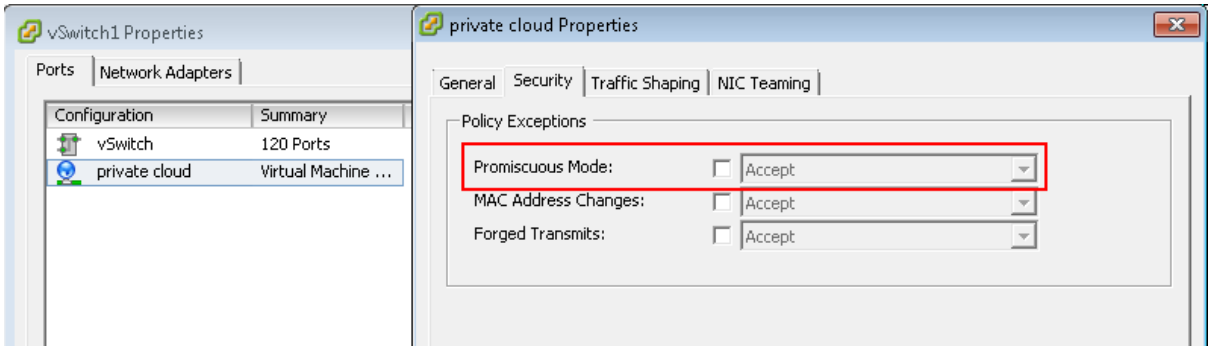
Mô hình được mô tả theo như hình minh họa



Môi trường lab như sau:

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

Nền tảng: VMware ESXi 5.5 (Promiscuous mode chuyển sang Accept)



5 máy ảo: 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 14.04.4 LTS, Trusty Tahr

Network: management network 10.10.10.0/24

External network 10.0.0.0/24

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 được xây dựng, kiểm nghiệm rất tỷ mẩn, 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 14.04.4 LTS, Trusty Tahr

3 controller nodes có hostname theo file `/etc/hostname` là: `controller1`, `controller2`, `controller3`

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/host` 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
```

II. Cài đặt, cấu hình Pacemaker & HAproxy trên CONTROLLER1, CONTROLLER2, CONTROLLER3

Sử dụng script `1_lb_script.sh` để cài đặt, cấu hình và chạy dịch vụ `corosync, pacemaker` và `haproxy`

Trên 3 nodes cấu hình `/etc/corosync/corosync.conf` như sau:

```
totem {
        version: 2
        token: 3000
        token_retransmits_before_loss_const: 10
        join: 60
        consensus: 3600
        vsftype: none
        max_messages: 20
        clear node high bit: yes
        secauth: off
        threads: 0

        # This specifies the mode of redundant ring, which may be
none, active, or passive.
        rrp_mode: active

        interface {
                # The following values need to be set based on your
environment
                ringnumber: 0
                bindnetaddr: 10.10.10.0
                mcastaddr: 226.94.1.1
                mcastport: 5405
        }
        interface {
                ringnumber: 1
                bindnetaddr: 10.0.0.0
                mcastaddr: 226.94.1.2
                mcastport: 5405
        }
}

amf {
        mode: disabled
}

quorum {
        # Quorum for the Pacemaker Cluster Resource Manager
        provider: corosync_votequorum
        expected_votes: 3
        wait_for_all: 1
        last_man_standing: 1
        last_man_standing_window: 10000
}

aisexec {
        user:   root
        group:  root
}
```

```
logging {
        fileline: off
        to_stderr: yes
        to_logfile: no
        to_syslog: yes
        syslog_facility: daemon
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
                tags: enter|leave|trace1|trace2|trace3|trace4|trace6
        }
}
```

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 10.0.0.118:80
  balance  source
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 10.0.0.113:80 check inter 2000 rise 2 fall 5
  server controller2 10.0.0.114:80 check inter 2000 rise 2 fall 5
  server controller3 10.0.0.115:80 check inter 2000 rise 2 fall 5

listen galera_cluster
  bind 10.10.10.180:3306
  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
```

```
  bind 10.10.10.180:9292
  balance  roundrobin
  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  roundrobin
  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  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 10.10.10.130:35358 check inter 2000 rise 2 fall 5
  server controller2 10.10.10.140:35358 check inter 2000 rise 2 fall 5
  server controller3 10.10.10.150:35358 check inter 2000 rise 2 fall 5

listen keystone_public *:5000
  balance  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 10.10.10.130:5001 check inter 2000 rise 2 fall 5
  server controller2 10.10.10.140:5001 check inter 2000 rise 2 fall 5
  server controller3 10.10.10.150:5001 check inter 2000 rise 2 fall 5

listen nova_ec2_api
  bind 10.10.10.180:8773
  balance  roundrobin
  option  tcpka
  option  tcplog
  server controller1 10.10.10.130:8773 check inter 2000 rise 2 fall 5
  server controller2 10.10.10.140:8773 check inter 2000 rise 2 fall 5
  server controller3 10.10.10.150:8773 check inter 2000 rise 2 fall 5

listen nova_compute_api
  bind 10.10.10.180:8774
  balance  roundrobin
  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
  bind 10.10.10.180:8775
  balance  roundrobin
  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
  bind 10.10.10.180:8776
  balance  roundrobin
  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
  bind 10.10.10.180:8777
  balance  roundrobin
  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
  bind 10.10.10.180:6080
  balance  roundrobin
  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
  bind 10.10.10.180:9696
  balance  roundrobin
  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
  bind 10.10.10.180:8080
  balance  roundrobin
  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
    bind 10.10.10.180:5672
    balance  roundrobin
    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
```

Kiểm tra trạng thái `corosync` từ `Controller1`

```
  # corosync-cfgtool -s
  Printing ring status.
  Local node ID 168430250
  RING ID 0
        id      = 10.10.10.130
        status  = ring 0 active with no faults
  RING ID 1
        id      = 10.0.0.113
        status  = ring 1 active with no faults
```

```
  # corosync-cmapctl  | grep member
  runtime.totem.pg.mrp.srp.members.168430240.config_version (u64) = 0
  runtime.totem.pg.mrp.srp.members.168430240.ip (str) = r(0)
ip(10.10.10.130) r(1) ip(10.0.0.113)
  runtime.totem.pg.mrp.srp.members.168430240.join_count (u32) = 1
  runtime.totem.pg.mrp.srp.members.168430240.status (str) = joined
  runtime.totem.pg.mrp.srp.members.168430250.config_version (u64) = 0
  runtime.totem.pg.mrp.srp.members.168430250.ip (str) = r(0)
ip(10.10.10.140) r(1) ip(10.0.0.114)
  runtime.totem.pg.mrp.srp.members.168430250.join_count (u32) = 1
  runtime.totem.pg.mrp.srp.members.168430250.status (str) = joined
  runtime.totem.pg.mrp.srp.members.168430260.config_version (u64) = 0
  runtime.totem.pg.mrp.srp.members.168430260.ip (str) = r(0)
ip(10.10.10.150) r(1) ip(10.0.0.115)
  runtime.totem.pg.mrp.srp.members.168430260.join_count (u32) = 1
  runtime.totem.pg.mrp.srp.members.168430260.status (str) = joined
```

Trên node `Controller1` cấu hình VIP1, VIP2, và HAproxy gõ `crm`

```
  crm(live)# cib new conf-failover
  INFO: conf-failover shadow CIB created
  crm(conf-failover)# configure
  crm(conf-failover)configure# property stonith-enabled=false
  crm(conf-failover)configure# property no-quorum-policy=ignore
  crm(conf-failover)configure# primitive VIP1 ocf:heartbeat:IPaddr2
params ip=10.0.0.118 cidr_netmask=24 op monitor interval=30s
  crm(conf-failover)configure# primitive VIP2 ocf:heartbeat:IPaddr2
params ip=10.10.10.180 cidr_netmask=24 op monitor interval=30s
  crm(conf-failover)configure# verify
  crm(conf-failover)configure#
  crm(conf-failover)configure# group IP_VIP VIP1 VIP2
  crm(conf-failover)configure# primitive HAPROXY lsb:haproxy op monitor
interval=10s
  crm(conf-failover)configure# group PROXY HAPROXY
  crm(conf-failover)configure# colocation service-with-ip infinity:
PROXY IP_VIP
```

```
  crm(conf-failover)configure# order service-after-ip mandatory: IP_VIP
PROXY
  crm(conf-failover)configure# end
  There are changes pending. Do you want to commit them? Y
  crm(conf-failover)# quit
  bye
  #crm cib commit conf-failover
  INFO: committed 'conf-failover' shadow CIB to the cluster
```

Kiểm tra cấu hình như sau:

```
  # crm configure show
  node $id="168430240" controller1
  node $id="168430250" controller2
  node $id="168430260" controller3
  primitive HAPROXY lsb:haproxy \
          op monitor interval="10s"
  primitive VIP1 ocf:heartbeat:IPaddr2 \
          params ip="10.0.0.118" cidr_netmask="24" \
          op monitor interval="30s"
  primitive VIP2 ocf:heartbeat:IPaddr2 \
          params ip="10.10.10.180" cidr_netmask="24" \
          op monitor interval="30s"
  group IP_VIP VIP1 VIP2
  group PROXY HAPROXY
  colocation service-with-ip inf: PROXY IP_VIP
  order service-after-ip inf: IP_VIP PROXY
  property $id="cib-bootstrap-options" \
          dc-version="1.1.10-42f2063" \
          cluster-infrastructure="corosync" \
          stonith-enabled="false" \
          no-quorum-policy="ignore"
```

Kiểm tra trạng thái hoạt động của Pacemaker

```
  # crm_mon -1
  Last updated: Wed Mar 30 15:08:13 2016
  Last change: Wed Mar 30 14:51:46 2016 via crm_shadow on lb01
  Stack: corosync
  Current DC: lb01 (168430240) - partition with quorum
  Version: 1.1.10-42f2063
  3 Nodes configured
  3 Resources configured

  Online: [ controller1 controller2 controller3 ]

   Resource Group: IP_VIP
       VIP1       (ocf::heartbeat:IPaddr2):       Started controller2
       VIP2       (ocf::heartbeat:IPaddr2):       Started controller2
   Resource Group: PROXY
       HAPROXY    (lsb:haproxy):  Started controller2
```

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

1. *Mariadb Galera Cluster:*

Sừ dụng `script 2_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=10.10.10.130

# 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=10.10.10.140

# 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=10.10.10.150

# 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, joint 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 `3_keystone_01.sh` để cài đặt, cấu hình keystone,

Sau đó trên `controller2`, `controller3`, sử dụng script `4_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]
memcached_servers=controller1:11211,controller2:11211,controller3:11211
[oauth1]
[os_inherit]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
```

```
rabbit_ha_queues=true

[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 5001
Listen 35358
<VirtualHost *:5001>
    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 *:35358>
    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
export OS_IMAGE_API_VERSION=2
export OS_IMAGE_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
export OS_IMAGE_API_VERSION=2
export OS_IMAGE_API_VERSION=2
```

2. *Glance:*

Trên `controller1`, sử dụng script `5_glance_01.sh`

Sau đó trên controller2, controller3 sử dụng script 6_glance_02.sh

Trên controller2, controller3, file cấu hình /etc/glance/glance-api.conf sẽ tương tự trên controller1 như sau (khác nhau giá trị bind_host)

```
[DEFAULT]
notification_driver = noop
verbose = True
bind_host = 10.10.10.130

[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]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[oslo_policy]
[paste_deploy]
flavor = keystone

[store_type_location_strategy]
[task]
[taskflow_executor]
```

Trên controller2, controller3, file cấu hình /etc/glance/glance-registry.conf sẽ tương tự trên controller1 như sau (khác nhau giá trị bind_host)

```
[DEFAULT]
notification_driver = noop
verbose = True
```

```
bind_host = 10.10.10.130

[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]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[oslo_policy]

[paste_deploy]
flavor = keystone
```

3. *Nova:*

Trên `controller1`, chạy script sau `7_nova_01.sh`, sau đó chạy script `8_nova_02.sh` trên `controller2`, `controller3`

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

```
[DEFAULT]

rpc_backend = rabbit
auth_strategy = keystone
bind-host = 10.10.10.130
osapi_compute_listen = 10.10.10.130
metadata_listen = 10.10.10.130
novncproxy_host = 10.10.10.130

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 = 10.0.0.113

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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 = 10.0.0.113
vncserver_proxyclient_address = 10.0.0.113

[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
bind-host = 10.10.10.140
osapi_compute_listen = 10.10.10.140
metadata_listen = 10.10.10.140
novncproxy_host = 10.10.10.140

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 = 10.0.0.114

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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 = 10.0.0.114
vncserver_proxyclient_address = 10.0.0.114

[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
bind-host = 10.10.10.150
osapi_compute_listen = 10.10.10.150
metadata_listen = 10.10.10.150
novncproxy_host = 10.10.10.150

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 = 10.0.0.115

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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 = 10.0.0.115
vncserver_proxyclient_address = 10.0.0.115

[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 `9_neutron_01.sh`, sau đó chạy script `10_neutron_02.sh` trên `controller2` và `controller3`.

Trên `controller2` và `controller3`, file cấu hình `/etc/neutron/neutron.conf` sẽ tương tự trên `controller1` như sau (khác giá trị `bind_host`)

```
[DEFAULT]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = True
rpc_backend = rabbit
bind_host = 10.10.10.130

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

l3_ha = True
allow_automatic_l3agent_failover = True
max_l3_agents_per_router = 3
min_l3_agents_per_router = 2
l3_ha_net_cidr = 169.254.192.0/18
dhcp_agents_per_network = 2

[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_hosts = controller1:5672,controller2:5672,controller3:5672
```

```
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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
use_namespaces = True
agent_mod = legacy

ha_confs_path = $state_path/ha_confs
ha_vrrp_auth_type = PASS
ha_vrrp_auth_password = cisco123
ha_vrrp_advert_int = 2

[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 `11_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 = 10.0.0.112

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 = 10.0.0.112
novncproxy_base_url = http://10.0.0.118:6080/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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true
```

```
  [qos]
```

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

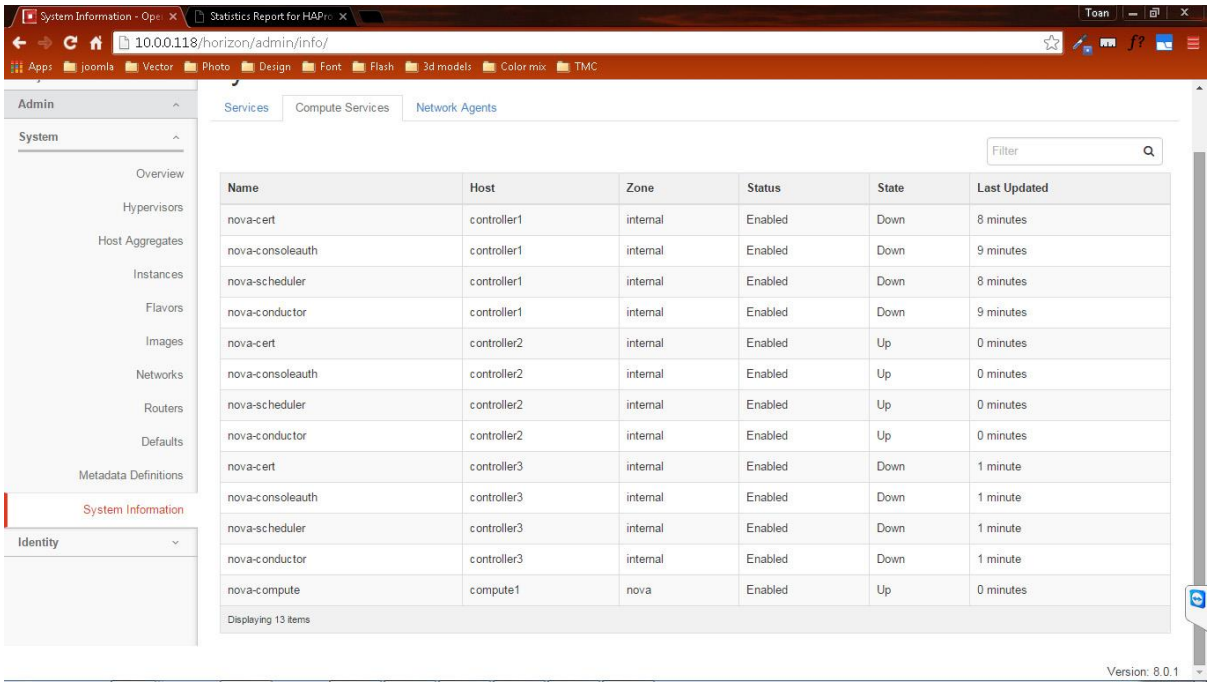
```
  [linux_bridge]
  physical_interface_mappings = public:eth1

  [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
```

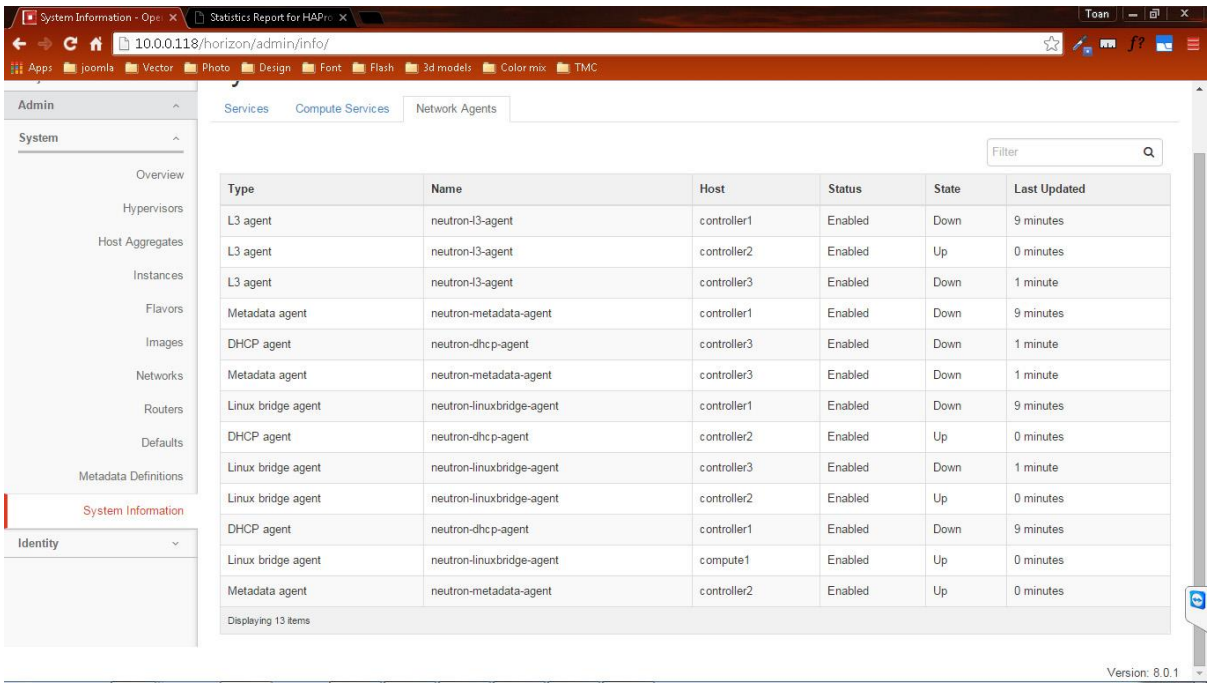
VI. Kết quả Failover test

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

- Fail card eth0
- Fail 1 trong 3 nodes `controller1`, `controller2`, `controller3`
- Fail 2 trong 3 nodes `controller1`, `controller2`, `controller3`

Hình ảnh từ kết quả test





```
Stack: corosync
Current DC: controller3 (168841589) - partition WITHOUT quorum
Version: 1.1.10-42f2063
3 Nodes configured
3 Resources configured


Node controller1 (168841587): OFFLINE
Node controller2 (168841588): OFFLINE
Node controller3 (168841589): online
        VIP2    (ocf::heartbeat:IPaddr2):       Started
        HAPROXY (lsb:haproxy):  Started
        VIP1    (ocf::heartbeat:IPaddr2):       Started
```

VII. Kết quả test với L3HA

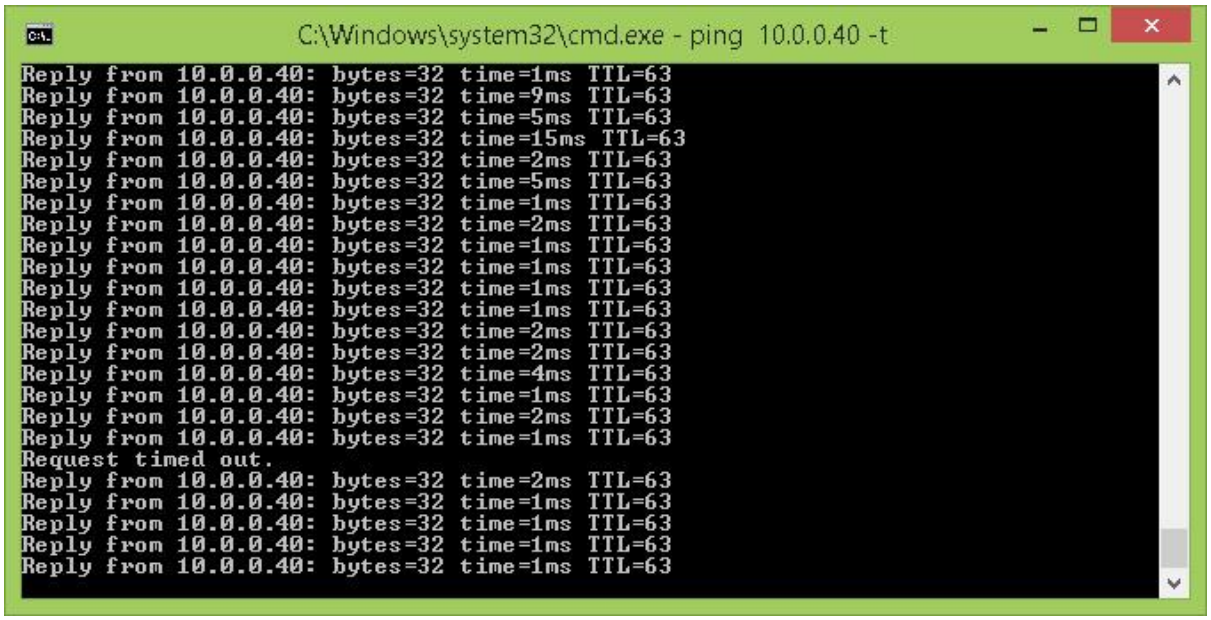
```
root@controller1:~# neutron router-list
+--------------------------------------+--------+------------------------------------------------------------------------------------------+-------------+------+
| id                                   | name   | external_gateway_info                                                                    | distributed | ha   |
+--------------------------------------+--------+------------------------------------------------------------------------------------------+-------------+------+
| 97370818-ca19-46ea-a772-28add7883ad7 | router | {"network_id": "9e5d77be-ceaa-4291-bebb-4f0d1f2306b5", "enable_snat": true, "external_fixed_ips": [{"subnet_id": "630a5817-49c2-428a-b86f-3d0ab4957e8f", "ip_address": "10.0.0.40"}]} | False | True |
+--------------------------------------+--------+------------------------------------------------------------------------------------------+-------------+------+
root@controller1:~# neutron router-show router
+-----------------------+------------------------------------------------------------------------------------------+
| Field                 | Value                                                                                    |
+-----------------------+------------------------------------------------------------------------------------------+
| admin_state_up        | True                                                                                     |
| distributed           | False                                                                                    |
| external_gateway_info | {"network_id": "9e5d77be-ceaa-4291-bebb-4f0d1f2306b5", "enable_snat": true, "external_fixed_ips": [{"subnet_id": "630a5817-49c2-428a-b86f-3d0ab4957e8f", "ip_address": "10.0.0.40"}]} |
| ha                    | True                                                                                     |
| id                    | 97370818-ca19-46ea-a772-28add7883ad7                                                     |
| name                  | router                                                                                   |
| routes                |                                                                                          |
| status                | ACTIVE                                                                                   |
| tenant_id             | f8c3dbb8302c4f2a944cce3a6d14e914                                                         |
+-----------------------+------------------------------------------------------------------------------------------+
root@controller1:~#
```

```
root@controller1:~# neutron router-port-list router
+--------------------------------------+-------------------------------------------------+-------------------+------------------------------------------------------------------------------------+
| id                                   | name                                            | mac_address       | fixed_ips                                                                          |
+--------------------------------------+-------------------------------------------------+-------------------+------------------------------------------------------------------------------------+
| 0f4a9efb-1314-4d33-a9cd-4b953ff54b19 |                                                 | fa:16:3e:2c:72:70 | {"subnet_id": "6a232f71-844d-40de-b689-26346c5930c9", "ip_address": "192.168.0.1"} |
| 4629b6b5-c7e0-479a-bb4e-b806ae83ef1e | HA port tenant f8c3dbb8302c4f2a944cce3a6d14e914 | fa:16:3e:50:72:68 | {"subnet_id": "72243ae1-04e8-4774-b2be-3bf361e55086", "ip_address": "169.254.192.2"} |
| 922ae87d-0956-4e01-af94-efc61cc1fda4 | HA port tenant f8c3dbb8302c4f2a944cce3a6d14e914 | fa:16:3e:d0:ce:2b | {"subnet_id": "72243ae1-04e8-4774-b2be-3bf361e55086", "ip_address": "169.254.192.3"} |
| b15eee5a-9d6a-4239-9a8d-ba25555d0506 | HA port tenant f8c3dbb8302c4f2a944cce3a6d14e914 | fa:16:3e:78:ac:03 | {"subnet_id": "72243ae1-04e8-4774-b2be-3bf361e55086", "ip_address": "169.254.192.1"} |
| eabc8f70-54d6-47a6-a088-b32985b21375 |                                                 | fa:16:3e:ba:c6:b4 | {"subnet_id": "630a5817-49c2-428a-b86f-3d0ab4957e8f", "ip_address": "10.0.0.40"}   |
+--------------------------------------+-------------------------------------------------+-------------------+------------------------------------------------------------------------------------+
```

Kiểm tra nội dung file /var/lib/neutron/ha_confs/97370818-ca19-46ea-a772-28add7883ad7/keepalived.conf

```
vrrp_instance VR_1 {
    state BACKUP
    interface ha-b15eee5a-9d
    virtual_router_id 1
    priority 50
    garp_master_delay 60
    nopreempt
    advert_int 2
    authentication {
        auth_type PASS
        auth_pass cisco123
    }
    track_interface {
        ha-b15eee5a-9d
    }
    virtual_ipaddress {
        169.254.0.1/24 dev ha-b15eee5a-9d
    }
    virtual_ipaddress_excluded {
        10.0.0.40/25 dev qg-eabc8f70-54
        10.0.0.41/32 dev qg-eabc8f70-54
        192.168.0.1/24 dev qr-0f4a9efb-13
        fe80::f816:3eff:fe2c:7270/64 dev qr-0f4a9efb-13 scope link
        fe80::f816:3eff:feba:c6b4/64 dev qg-eabc8f70-54 scope link
    }
    virtual_routes {
        0.0.0.0/0 via 10.0.0.1 dev qg-eabc8f70-54
    }
```

Kết quả ping tới route gateway khi controller hosts router bị down:



Phụ lục 1: File config.cfg

```
## Network Info

##################### KHAI BAO CAC BIEN CHO SCRIPT ######################
#Gateway for EXT network
GATEWAY_IP_EXT=10.0.0.1
NETMASK_ADD_EXT=255.255.255.0
NETMASK_ADD_MGNT=255.255.255.0

## Virtual IP for services (depends on your own network)
VIP_IP=10.10.10.180

# Set password
DEFAULT_PASS='654321'

RABBIT_PASS="$DEFAULT_PASS"
MYSQL_PASS="$DEFAULT_PASS"
TOKEN_PASS="$DEFAULT_PASS"
ADMIN_PASS="$DEFAULT_PASS"
SERVICE_PASSWORD="$DEFAULT_PASS"
METADATA_SECRET="$DEFAULT_PASS"

SERVICE_TENANT_NAME="service"
ADMIN_TENANT_NAME="admin"
DEMO_TENANT_NAME="demo"
INVIS_TENANT_NAME="invisible_to_admin"
ADMIN_USER_NAME="admin"
DEMO_USER_NAME="demo"

# Environment variable for OPS service
KEYSTONE_PASS="$DEFAULT_PASS"
GLANCE_PASS="$DEFAULT_PASS"
NOVA_PASS="$DEFAULT_PASS"
NEUTRON_PASS="$DEFAULT_PASS"
CINDER_PASS="$DEFAULT_PASS"
SWIFT_PASS="$DEFAULT_PASS"
HEAT_PASS="$DEFAULT_PASS"

# Environment variable for DB
KEYSTONE_DBPASS="$DEFAULT_PASS"
GLANCE_DBPASS="$DEFAULT_PASS"
NOVA_DBPASS="$DEFAULT_PASS"
NEUTRON_DBPASS="$DEFAULT_PASS"
CINDER_DBPASS="$DEFAULT_PASS"
HEAT_DBPASS="$DEFAULT_PASS"

# User declaration in Keystone
ADMIN_ROLE_NAME="admin"
MEMBER_ROLE_NAME="Member"
KEYSTONEADMIN_ROLE_NAME="KeystoneAdmin"
KEYSTONESERVICE_ROLE_NAME="KeystoneServiceAdmin"
```

Phụ lục 2: File 0_prepare.sh

```
#!/bin/bash -ex

case $(hostname) in
 controller1) CON_MGMT_IP=10.10.10.130
              CON_EXT_IP=10.0.0.113
        ;;
 controller2) CON_MGMT_IP=10.10.10.140
              CON_EXT_IP=10.0.0.114
        ;;
 controller3) CON_MGMT_IP=10.10.10.150
              CON_EXT_IP=10.0.0.115
        ;;
esac

ifaces=/etc/network/interfaces
test -f $ifaces.orig || cp $ifaces $ifaces.orig
rm $ifaces
touch $ifaces
cat << EOF >> $ifaces
#Assign IP for Controller node

# LOOPBACK NET
auto lo
iface lo inet loopback

# MGNT NETWORK
auto eth0
iface eth0 inet static
address $CON_MGNT_IP
netmask $NETMASK_ADD_MGNT

# EXT NETWORK
auto eth1
iface eth1 inet static
address $CON_EXT_IP
netmask $NETMASK_ADD_EXT
gateway $GATEWAY_IP_EXT
dns-nameservers 8.8.8.8
EOF

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
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 1_lb_script.sh

```
#!/bin/bash -ex

Source config.cfg

apt-get install pacemaker haproxy -y

echo "###### Configure corosync #####"
sleep 3

sed -i s/START=no/START=yes/ /etc/default/corosync

corosync_config=/etc/corosync/corosync.conf
test -f $corosync_config.orig \
    || cp $corosync_config $corosync_config.orig
rm $corosync_config
touch $corosync_config

cat << EOF > $corosync_config
totem {
        version: 2
        token: 3000
        token_retransmits_before_loss_const: 10
        join: 60
```

```
        consensus: 3600
        vsftype: none
        max_messages: 20
        clear_node_high_bit: yes
        secauth: off
        threads: 0

        # This specifies the mode of redundant ring, which may be none,
active, or passive.
        rrp_mode: active

        interface {
                # The following values need to be set based on your
environment
                ringnumber: 0
                bindnetaddr: 10.10.10.0
                mcastaddr: 226.94.1.1
                mcastport: 5405
        }
        interface {
                ringnumber: 1
                bindnetaddr: 10.0.0.0
                mcastaddr: 226.94.1.2
                mcastport: 5405
        }
 }

 amf {
        mode: disabled
 }

 quorum {
        # Quorum for the Pacemaker Cluster Resource Manager
        provider: corosync_votequorum
        expected_votes: 3
        wait_for_all: 1
        last_man_standing: 1
        last_man_standing_window: 10000
 }

 aisexec {
        user:   root
        group:  root
 }

 logging {
        fileline: off
        to_stderr: yes
        to_logfile: no
        to_syslog: yes
        syslog_facility: daemon
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
                tags: enter|leave|trace1|trace2|trace3|trace4|trace6
        }
 }
 EOF
```

```
update-rc.d pacemaker start 20 2 3 4 5 . stop 00 0 1 6 .

cat << EOF > /etc/corosync/uidgid.d/pacemaker
uidgid {
  uid: hacluster
  gid: haclient
}
EOF

service corosync start

service pacemaker start

echo "###### Configure Haproxy #####"
sleep 3

MGMT_IP1=`grep controller1 /etc/hosts | awk '{print $1}'`
MGMT_IP2=`grep controller2 /etc/hosts | awk '{print $1}'`
MGMT_IP3=`grep controller3 /etc/hosts | awk '{print $1}'`

################################################

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 10.0.0.113:80 check inter 2000 rise 2 fall 5
  server controller2 10.0.0.114:80 check inter 2000 rise 2 fall 5
  server controller3 10.0.0.115:80 check inter 2000 rise 2 fall 5

listen galera_cluster
```

```
  bind $VIP_IP:3306
  balance  source
  mode   tcp
  option tcpka
  server controller1 $MGMT_IP1:3306 check
  server controller2 $MGMT_IP2:3306 backup check
  server controller3 $MGMT_IP3:3306 backup check

listen glance_api
  bind $VIP_IP:9292
  balance  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 $MGMT_IP1:9292 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:9292 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:9292 check inter 2000 rise 2 fall 5

listen glance_registry *:9191
  balance  roundrobin
  option  tcpka
  option  tcplog
  server controller1 $MGMT_IP1:9191 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:9191 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:9191 check inter 2000 rise 2 fall 5

listen keystone_admin *:35357
  balance  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 $MGMT_IP1:35358 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:35358 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:35358 check inter 2000 rise 2 fall 5

listen keystone_public *:5000
  balance  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 $MGMT_IP1:5001 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:5001 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:5001 check inter 2000 rise 2 fall 5

listen nova_ec2_api
  bind $VIP_IP:8773
  balance  roundrobin
  option  tcpka
  option  tcplog
  server controller1 $MGMT_IP1:8773 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:8773 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:8773 check inter 2000 rise 2 fall 5

listen nova_compute_api
  bind $VIP_IP:8774
  balance  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 $MGMT_IP1:8774 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:8774 check inter 2000 rise 2 fall 5
```

```
  server controller3 $MGMT_IP3:8774 check inter 2000 rise 2 fall 5

listen nova_metadata_api
  bind $VIP_IP:8775
  balance  roundrobin
  option  tcpka
  option  tcplog
  server controller1 $MGMT_IP1:8775 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:8775 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:8775 check inter 2000 rise 2 fall 5

listen cinder_api
  bind $VIP_IP:8776
  balance  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 $MGMT_IP1:8776 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:8776 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:8776 check inter 2000 rise 2 fall 5

listen ceilometer_api
  bind $VIP_IP:8777
  balance  roundrobin
  option  tcpka
  option  tcplog
  server controller1 $MGMT_IP1:8777 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:8777 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:8777 check inter 2000 rise 2 fall 5

listen nova_vncproxy
  bind $VIP_IP:6080
  balance  roundrobin
  option  tcpka
  option  tcplog
  server controller1 $MGMT_IP1:6080 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:6080 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:6080 check inter 2000 rise 2 fall 5

listen neutron_api
  bind $VIP_IP:9696
  balance  roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 $MGMT_IP1:9696 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:9696 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:9696 check inter 2000 rise 2 fall 5

listen swift_proxy
  bind $VIP_IP:8080
  balance  roundrobin
  option  tcplog
  option  tcpka
  server controller1 $MGMT_IP1:8080 check inter 2000 rise 2 fall 5
  server controller2 $MGMT_IP2:8080 check inter 2000 rise 2 fall 5
  server controller3 $MGMT_IP3:8080 check inter 2000 rise 2 fall 5
listen rabbitmq
    bind $VIP_IP:5672
    balance  roundrobin
    option clitcpka
```

```
    timeout client 900m
    server controller1 $MGMT_IP1:5672 check inter 1s
    server controller2 $MGMT_IP2:5672 check inter 1s
    server controller3 $MGMT_IP3:5672 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 4: File 2_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 0xcbcb082a1bb943db

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.mysqld
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=$LOCAL_IP

# 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 3_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]
memcached_servers = controller1:11211,controller2:11211,controller3:11211

[oauth1]
[os_inherit]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = 654321
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 5001
Listen 35358

<VirtualHost *:5001>
    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 *:35358>
    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
echo "export OS_IMAGE_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
echo "export OS_IMAGE_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 4_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]
memcached_servers = controller1:11211,controller2:11211,controller3:11211

[oauth1]
[os_inherit]
[oslo_messaging_amqp]
[oslo_messaging_qpid]
[oslo_messaging_rabbit]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 5_glance_01.sh

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

case $(hostname) in
 controller1) CON_MGMT_IP=10.10.10.130
        ;;
 controller2) CON MGMT IP=10.10.10.140
        ;;
 controller3) CON_MGMT_IP=10.10.10.150
        ;;
esac

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
bind_host = $CON_MGMT_IP
```

```
[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]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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
```

```
bind_host = $CON_MGMT_IP

[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]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 6_glance_02.sh

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

case $(hostname) in
 controller1) CON_MGMT_IP=10.10.10.130
        ;;
 controller2) CON_MGMT_IP=10.10.10.140
        ;;
 controller3) CON_MGMT_IP=10.10.10.150
        ;;
Esac

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
bind_host = $CON_MGMT_IP

[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]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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
bind_host = $CON_MGMT_IP
```

```
[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]
rabbit_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 7_nova_01.sh

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

 case $(hostname) in
  controller1) CON_MGMT_IP=10.10.10.130
         ;;
  controller2) CON MGMT IP=10.10.10.140
         ;;
  controller3) CON_MGMT_IP=10.10.10.150
         ;;
 Esac

 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
 bind-host = $CON_MGMT_IP
 osapi_compute_listen = $CON_MGMT_IP
 metadata_listen = $CON_MGMT_IP
 novncproxy_host = $CON_MGMT_IP

 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_hosts = controller1:5672,controller2:5672,controller3:5672
 rabbit_userid = openstack
 rabbit_password = $RABBIT_PASS
 rabbit_retry_interval=1
 rabbit_retry_backoff=2
 rabbit_max_retries=0
 rabbit_durable_queues=true
 rabbit_ha_queues=true

 [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 8_nova_02.sh

```
#!/bin/bash -ex
```

```
 #
 source config.cfg

 case $(hostname) in
  controller1) CON_MGMT_IP=10.10.10.130
         ;;
  controller2) CON_MGMT_IP=10.10.10.140
         ;;
  controller3) CON_MGMT_IP=10.10.10.150
         ;;
 Esac

 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
 bind-host = $CON_MGMT_IP
 osapi_compute_listen = $CON_MGMT_IP
 metadata_listen = $CON_MGMT_IP
 novncproxy_host = $CON_MGMT_IP

 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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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 9_neutron_01.sh

```
#!/bin/bash -ex
#

source config.cfg

LOCAL_IP=`ifconfig eth0 | 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
bind-host = $LOCAL_IP

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

l3_ha = True
allow_automatic_l3agent_failover = True
max_l3_agents_per_router = 3
min_l3_agents_per_router = 2
l3_ha_net_cidr = 169.254.192.0/18
dhcp_agents_per_network = 2

[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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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
 use_namespaces = True
 agent_mod = legacy

 ha_confs_path = $state_path/ha_confs
 ha_vrrp_auth_type = PASS
 ha_vrrp_auth_password = cisco123
 ha_vrrp_advert_int = 2

 [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 loi 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 10_neutron_02.sh

```
#!/bin/bash -ex
#

source config.cfg

LOCAL_IP=`ifconfig eth0 | 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
bind-host = $LOCAL_IP

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

l3_ha = True
allow_automatic_l3agent_failover = True
max_l3_agents_per_router = 3
min_l3_agents_per_router = 2
l3_ha_net_cidr = 169.254.192.0/18
dhcp_agents_per_network = 2

[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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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
use_namespaces = True
agent_mod = legacy

ha_confs_path = $state_path/ha_confs
ha_vrrp_auth_type = PASS
ha_vrrp_auth_password = cisco123
ha_vrrp_advert_int = 2

[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 loi 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 11_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
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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true
```

```
[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_hosts = controller1:5672,controller2:5672,controller3:5672
rabbit_userid = openstack
rabbit_password = $RABBIT_PASS
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_durable_queues=true
rabbit_ha_queues=true

[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
```