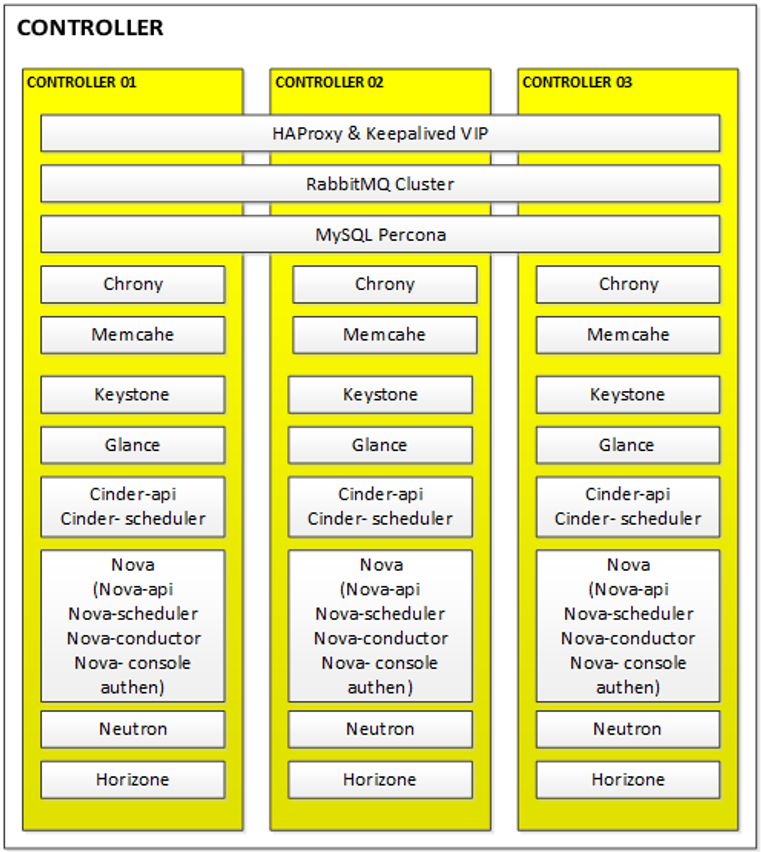
# **HƯỚNG DẪN CÀI ĐẶT OPENSTACK SỬ DỤNG SCRIPTS**

## Tổng quan

Tài liệu này hướng dẫn triển khai cài đặt Openstack sử dụng Docker trên Redhat 7.6.

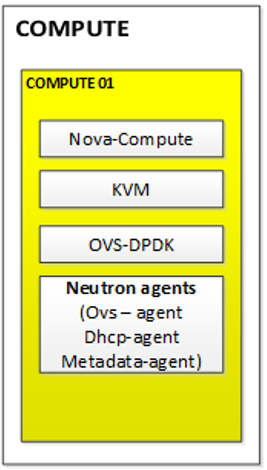
Phiên bản sử dụng: Openstack Queens

Tài liệu gồm có 3 phần chính tương ứng với quy trình chuẩn bị và sử dụng script cài đặt:

* Chuẩn bị các server: Mô hình hệ thống gồm các server, vai trò và đường mạng nối giữa các server với nhau.
* Sử dụng script cài đặt các service trên Controller Node

Hình 1: Các service cài đặt trên Controller Node.

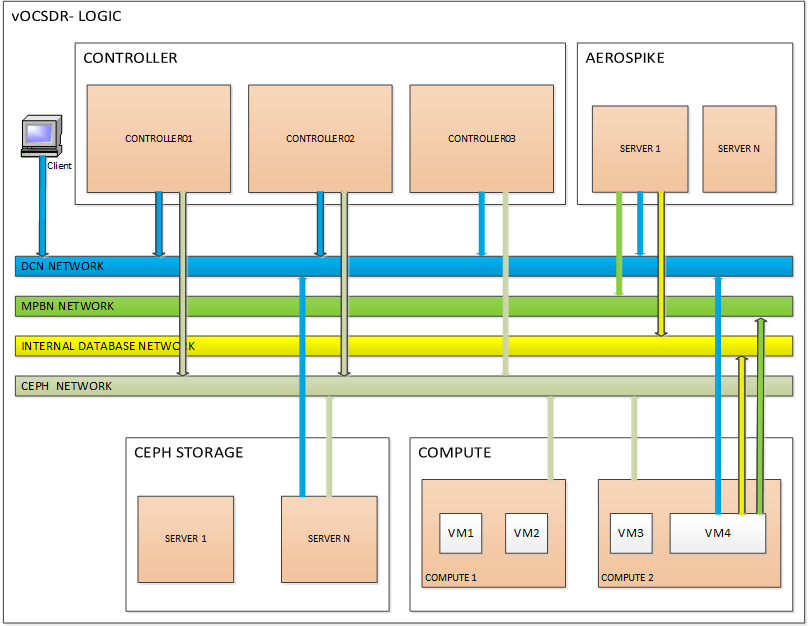
* + Sử dụng script back\_end.sh cài đặt các back\_end service cho Controller:
    - Network Time Protocol Chrony.service: Đồng nhất thời gian giữa các server trong hệ thống với nhau.
    - RabbitMQ: Một message queue được Openstack để phối hợp hoạt động và thông tin trạng thái các service, cho phép các service sử dụng RPC (Remote Proceduce Calls) để giao tiếp với nhau.
    - Memcached.service: Sử dụng để cache token cho cơ chế xác thực service của Identity service (Keystone)
  + Sử dụng script back\_end\_ha.sh cài đặt các back\_end service kết hợp với HaProxy:
    - Haproxy và Keepalived: Tạo một Virtual IP và phân đều các request tới VIP này tới các Controller Node
    - RabbitMQ Cluster: Nhóm các RabbitMQ trên các Controller Node thành một Cluster hoạt động active/active
    - Percona XtraDB Cluster: Tạo một Cluster DB trên các Controller Node hoạt động active/passive
  + Cài đặt các Openstack service sử dụng controller01.sh và controller02.sh:
    - Keystone: Cung cấp một Identity service
    - Glance: Cung cấp Image Registry service quản lý các Virtual Machine image và metadata liên hệ với các image đó.
    - Cinder: Cung cấp Block-Storage cho VM, làm nhiệm vụ tạo, xóa volume và snapshot của volume.
    - Nova: Cung cấp các Compute service.
    - Neutron: Cung cấp các Network service.
    - Horizon: Cung cấp giao diện web để quản lý openstack.
* Sử dụng script cài đặt các service trên Compute Node:



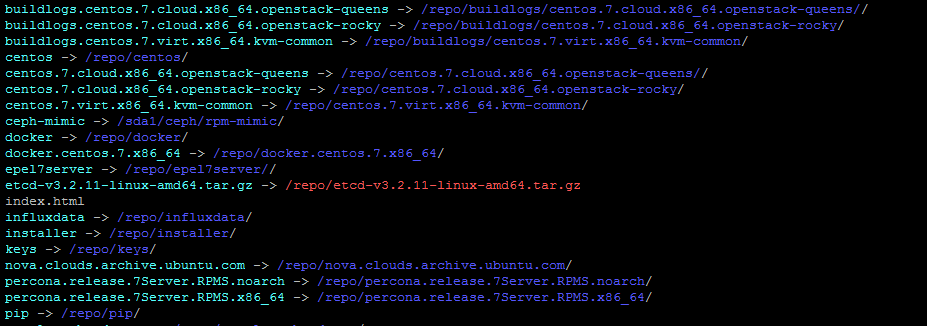
Hình 2: Các service cài đặt trên Compute Node

* + Sử dụng script backend.sh để cài đặt các backend service cho Compute Node:
    - Network Time Protocol Chrony.service: Đồng nhất thời gian giữa Compute Node và Controller Node tránh lỗi time out khi giao tiếp.
    - Docker: Cài đặt các service qua Docker Container
  + Sử dụng script compute.sh để cài đặt các openstack service cho Compute Node:
    - KVM: Làm nhiệm vụ của Hypervisor
    - Openstack Nova Compute: Tạo và quản lý các Virtual Machine.
    - Openvswitch-DPDK: Hỗ trợ các tính năng sau VLAN tagging & 802.1q trunking - Standard Spanning Tree Protocol (802.1D) - LACP - Port Mirroring (SPAN/RSPAN) - Tunneling Protocols – QoS. Mục đích chính là cung cấp lớp chuyển mạch cho môi trường ảo hóa phần cứng
    - Neutron OVS-layer2 agent: Sử dụng Openvswitch để cung cấp network segmentation và khả năng cách ly cho project network
    - Neutron DHCP agent: Phân bổ IP cho một VM mới được tạo
    - Neutron metadata agent: Quản lý các metadata cho VM, cung cấp khả năng SSH sử dụng private key.

## Chuẩn bị các server

* Chuẩn bị các nhóm server cơ bản
  + Server cài đặt Controller (Ít nhất 3 server)
  + Server cài đặt Compute (Ít nhất 2 server)
  + Server cài đặt CEPH Storage
  + Server cài đặt AEROSPIKE
* ****Chuẩn bị tối thiểu 4 dải mạng:

Hình 3: Minh họa các server và các dải mạng cần chuẩn bị

* + **DCN NETWORK:** Dải mạng Management phục vụ việc cài đặt, giám sát và quản lý các máy ảo.
  + **MPBN NETWORK:** Dải mạng phục vụ việc truyền dữ liệu (ví dụ bản tin tính cước) và kết nối với mạng MPBN bên hệ thống ngoài.
  + **INTERNAL DATABASE NETWORK:** Giao tiếp giữa VM và Aerospike.
  + **CEPH NETWORK:** Giao tiếp giữa các CONTROLLER, COMPUTE với CEPH server.
* Trên mỗi server cấu hình sẵn địa chỉ IP cho 4 card mạng của mỗi server theo thứ tự:
  + 1st interface – kết nối vào DCN NETWORK
  + 2nd interface – kết nối vào MPBN NETWORK
  + 3rd interface – kết nối vào INTERNAL NETWORK
  + 4th interface – kết nối vào CEPH NETWORK
* Ngoài ra cần tạo một máy ảo làm server Registry đóng vai trò repo chứa các gói .rpm cài Openstack và đồng thời làm nhiệm vụ Docker-Registry chứa các Docker\_Image.
  + Các repo cần cho quá trình cài Openstack
    - /repo/centos.7.cloud.x86\_64.openstack-queens
    - /repo/centos7.virt.x86\_64.kvm-common
    - /repo/percona.release.7Server.RPMS.x86\_64
    - /repo/epel7server/
    - /repo/docker.centos.7.x86\_64
  + Tạo link html cho các repo:
  + Các docker images lưu trên Docker-Registry:
    - localhost:4000/keystone:q
    - localhost:4000/glance:q
    - localhost:4000/nova:q
    - localhost:4000/neutron:q
    - localhost:4000/horizon:q
    - localhost:4000/cinder\_api:q
    - localhost:4000/cinder\_scheduler:q
    - localhost:4000/cinder\_volume:q
* Tải và giải nén Openstack doc-OCS:

Script gồm 2 thư mục chính:

* + **openstack-queens-dockerfiles:** Chứa các file build Docker Image.
  + **openstack-queens-deployment:**  Chứa các file cấu hình và các lệnh chạy cài đặt Openstack.

Chuyển vào thư mục openstack-queens-deployment

cd ./openstack-queens-deployment/

* + - **./docker:** Các file cấu hình
    - **./controller:** Các file cài đặt cho Controller Node
    - **./compute:** Các file cài đặt cho Compute Node

## Thiết lập repo và các biến môi trường cho Controller và Compute server.

### 3.1 Cài đặt yum repo:

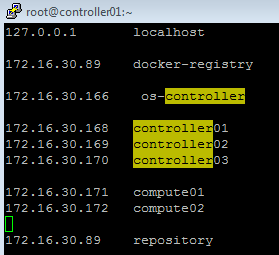
* Để kết nối các server tới repo trên REPO\_IP (ip của repo server) cần tạo các file .repo trong thư mục /etc/yum.repos.d
* Thao tác này thực hiện giống nhau trên server Controller và server Compute.
* Cài đặt repo sử dụng script bash make\_repo/make\_repo.sh**:**
  + Cài các file release.rpm
  + Ghi đè các file .repo vào /etc/yum.repos.d/
  + Cập nhật repo : yum repolist.

### 3.2 Thiết lập các biến môi trường:

* Các biến môi trường được sử dụng cho việc sinh các file cấu hình cho Openstack service:
* Để cài đặt cho các Controller Node di chuyển vào thư mục controller và sửa các file môi trường
* vim /etc/host
* cd ./controller
* vim env.list
* vim password/passwd.env
* Để cài đặt cho các Compute di chuyển vào thư mục compute và sửa các file môi trường
* vim /etc/host
* cd ./compute
* vim env.list
* vim password/passwd.env
* Thiết đặt các biến môi trường theo hướng dẫn bên dưới cho cả Controller Node và Compute Node, tùy biến theo IP và vai trò trong hệ thống.

#### 3.2.1 /etc/host:

* + Bổ xung vào /etc/host thông tin ip các server gồm ip docker-registry, ip os-controller, ip controller01,controller02,controller03 , ip compute01, compute02, ip repository



#### *3.2.2 ./env.list:*

File env.list chứa các thông sốs khởi tạo user admin trong openstack được dùng trong Keystone\_Bootstrap:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | KEYSTONE\_START | Biến mô tả lệnh chạy kèm khi khởi tạo container | BOOTSTRAP |
| 2 | CONTROLLER\_HOST | Định danh cho nhóm CONTROLLER | os-controller |
| 3 | ADMIN\_PASS | password cho openstack admin user | Vttek@123 |
| 4 | REGION\_NAME | Đặt tên cho region | North\_VN |
| 5 | CELL\_NAME | Tên cell được nova sử dụng | cell1 |

#### *3.2.3 ./pasword/passwd.env:*

Chứa các password và thông tin server.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | NTP\_NETWORK | Thiết lập cho cấu hình NTP SERVER /etc/chrony.conf để cho phép các NTP CLIENT trong dải mạng được quyền đồng bộ date time với NTP SERVER. | 172.16.30.0/24 |
| 2 | HOST | Thông tin hostname | controller01 |
| 3 | MY\_IP | Thông tin IP để các service http bind vào.  Đặt thông tin này theo IP trong dải PRIVATE\_NETWORK (đồng thời là dải IP sử dụng cho HaProxy) | 0.0.0.0  🡪 Đặt về ip 172.16.30.168 |

* + **#rabbitmq**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | HUB | Thông tin HUB cluster | controller01 |
| 2 | SSH\_USER | Thông tin ssh\_user để copy file cấu hình chung /etc/rabitmq | root |
| 3 | RB\_OPENSTACK | password cho rabbitmq user openstack. | KFTA8eHMgiX4DDUcCKSsoTpZFGnCoBrA |
| 4 | typeset -A CONTROLLERS | thông tin ip các server trong controller cluster | (  [controller01]=172.16.30.168  [controller02]=172.16.30.169  [controller02]=172.16.30.170  ) |

* + **#Percona**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | SQL\_WSREP\_CLUSTER\_NAME | Tên cluster | pxc-cluster |
| 2 | SQL\_ROOT\_PASSWD | Password mặc định cho root user | Vttek@123 |
| 3 | SQL\_SST\_USER | User cho các Percona DB client join vào cluster | sstuser |
| 4 | SQL\_SST\_PASSWD | Password cho các Percona DB client join vào cluster | Vttek@123 |
| 5 | SQL\_MAX\_CONNECT\_ERRORS | max connection error | 10 |
| 6 | SQL\_MAX\_ALLOWED\_PACKET | max allowed packet | 10 |
| 7 | MY\_HOST\_NAME | Host name của member trong cluster | pxc-cluster-node-1 |

* + **#keystone**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | DEMO\_PASS | Password cho user demo | Vttek@123 |
| 2 | GLANCE\_PASS | Password cho user glance | Vttek@123\_glance |
| 3 | CINDER\_PASS | Password cho user cinder | Vttek@123\_cinder |
| 4 | NOVA\_PASS | Password cho user nova | Vttek@123\_nova |
| 5 | PLACEMENT\_PASS | Password cho user placement | Vttek@123\_placement |
| 6 | NEUTRON\_PASS | Password cho user neutron | Vttek@123\_neutron |
| 7 | METADATA\_SECRET | Password cho user metadata | Vttek@123\_metadata |

password cho các user trong openstack, mỗi user được sử dụng bởi service có tên tương ứng: glance, cinder, nova, neutron.

* + **#mysql**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | MQ\_GLANCE | Password để các service lấy quyền đọc, ghi dữ liệu vào Mysql.  Ví dụ để service nova-api, nova-conductor trên controller node và openstack-nova-compute trên compute node lấy quyền đọc, ghi vào database nova, nova-api, nova-placement. | Tl1bMfzOdSJYi9HZx2ic47rxbPKPXk4q |
| 2 | MQ\_CINDER | Tl1bMfzOdSJYi9HZx2ic47rxbPKPXk4x |
| 3 | MQ\_KEYSTONE | LybBmiQk2t0djlrNf2zLrR1vjih4LrZK |
| 4 | MQ\_NOVA | o45OyDJ9b5jvD4fJzUBcLQI9qH4GrYUh |
| 5 | MQ\_NEUTRON | 3fjWwEnE803MWeq4hJOE23GPQAgEDVc8 |

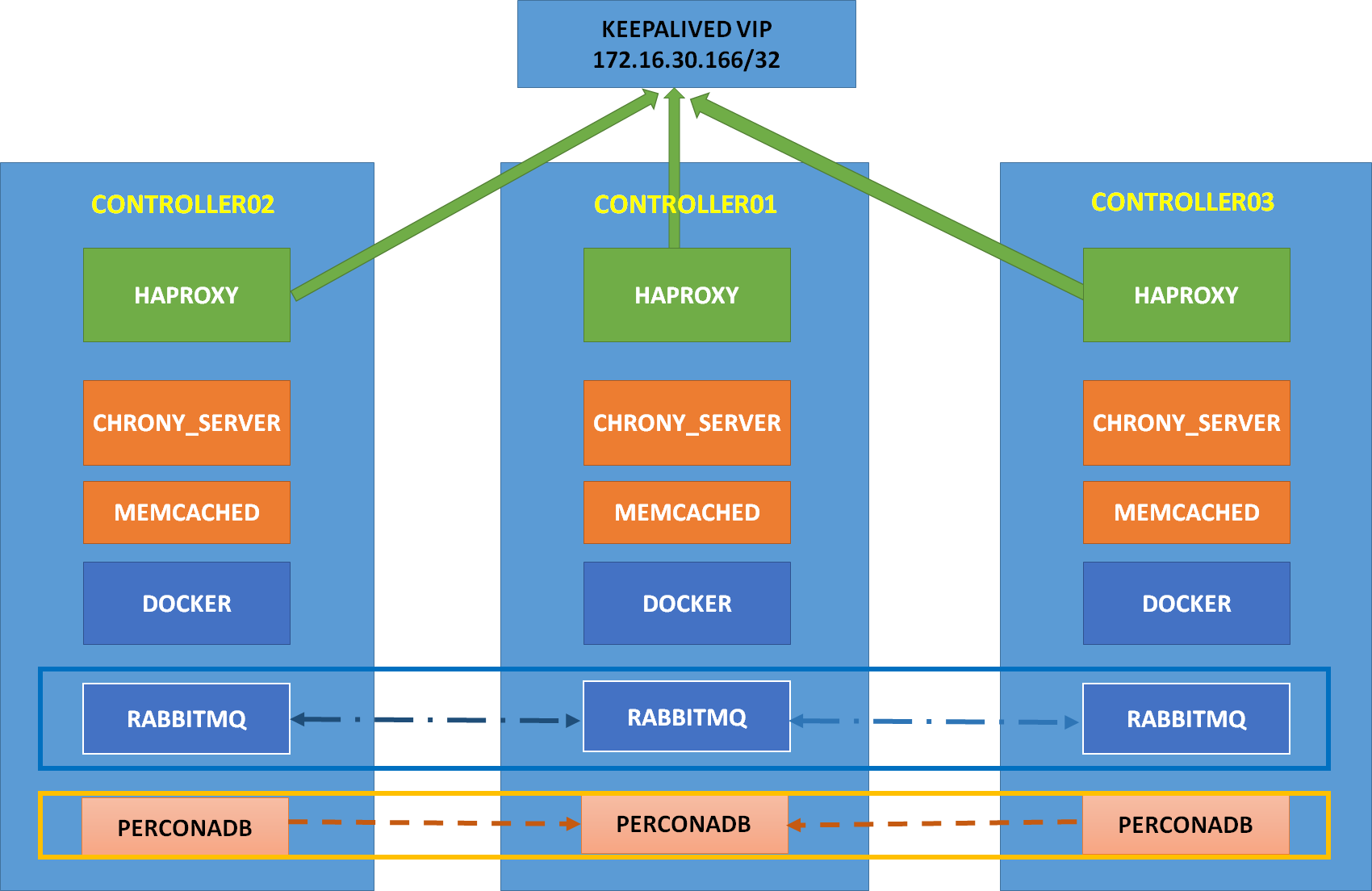
* + **#haproxy\_keepalived**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | VIP\_PUBLIC | IP chung cho cluster openstack controller, là ip của os-controller | 172.16.30.166 |
| 2 | HAPROXY\_STATS\_PORT | Sử dụng cho HAPROXY\_STATS để theo dõi haproxy. | 1994 |
| 3 | HAPROXY\_STATS\_USER | cloud |
| 4 | HAPROXY\_STATS\_PASS | Vttek@123\_cloud |
| 5 | VIP\_PRIVATE\_INF | Interface được VIP sử dụng | eth0 |
| 6 | VIP\_PRIVATE | IP chung cho cluster openstack controller, là ip của os-controller | 172.16.30.166 |
| 7 | typeset -A KEEPALIVED\_PRIORS | Độ ưu tiên cho các member trong Keepalived khi vote nhận VIP | **(**  [controller01]=3  [controller02]=2  [controller03]=1  ) |

## Cài đặt cho các Controller Node:

### 4.1. Cài đặt các backend service

Bước này sử dụng script ***controller/back\_end.sh*** và script ***controller/back\_end\_ha.sh***

****

Hình 4: Các backend service cài đặt trên Controller Node

* Yêu cầu cài đặt:
  + Các service **docker**, **memcached**, **chrony\_server** chạy độc lập trên từng máy controller
  + Các service **rabbitmq**, **perconadb** chạy theo cụm cluster,

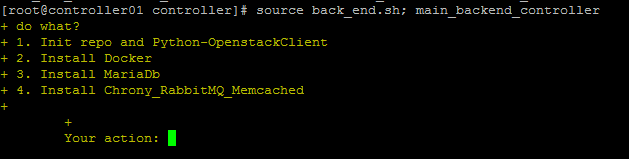
RabbitMQ cluster hoạt động **active/active**

PerconaDB cluster hoạt động theo cơ chế **replicate, active/passive**

* + Các service Haproxy trên 3 controller sử dụng chung 1 cấu hình, cùng phục vụ cho 1 VIP chung và thực hiện phân phối tới 3 controller.
* Lần lượt cài đặt các backend service: **docker, chrony, rabbitmq, memcache, percona, haproxy, keepalived** theo 2 bước:

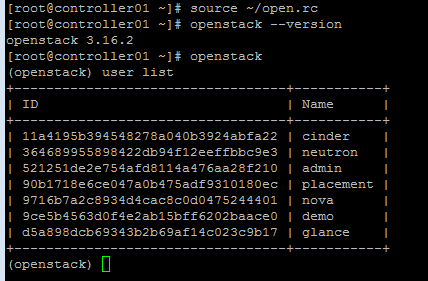
#### 4.1.1 Cài đặt back\_end service độc lập trên từng máy controller

Sử dụng file back\_end.sh:

source back\_end.sh; main\_backend\_controller

* + Cài đặt **python-openstackclient**: python-openstackclient để quản lý openstack dưới dạng commandline.

Kiểm tra python-openstackclient đã cài đúng phiên bản:

openstack --version

# Kết qủa trả về cho bản Queen là 3.16.2, có thể tra cứu version tương ứng tại <https://releases.openstack.org/queens/index.html>

đảm bảo version của các ứng dụng nằm trong khoảng từ earliest tới recent.

* + Cài đặt **Docker**, copy thư mục ../docker/ vào /usr/share/docker/. Đây là thư mục chứa các file cấu hình cho các openstack service chạy trong docker-container.

# Tại bước cài đặt Docker có phần tạo file /etc/docker/daemon.json để cấu hình cho phép Docker pull image từ các nguồn private docker-registry.

* “insecure-registries”: [“REPO\_IP\_1:DOCKER\_REGISTRY\_PORT”, REPO\_IP\_2:DOCKER\_REGISTRY\_PORT]

Mảng các Docker registry được phép truy cập để pull image.

* “graph”: “path/to/docker\_storage”

Đặt vị trí lưu Docker Image và Docker Container, nếu không set sẽ lưu ở /var/lib/docker/

* “debug”: false

Đặt Docker chạy ở debug mode hoặc không

* “log-level”: “debug | info | warn | error | fatal”
  + Cài đặt **Chrony** thực hiện NTP, lựa chọn 1 để cài chronyserver, lựa chọn 2 để cài chrony client.

# Kiểm tra /etc/chrony.conf

* server NTP\_SERVER iburst

NTP\_SERVER set theo Ip server

* allow NTP\_NETWORK

Cho phép các server khác trong mạng lấy thông tin date, time của server.

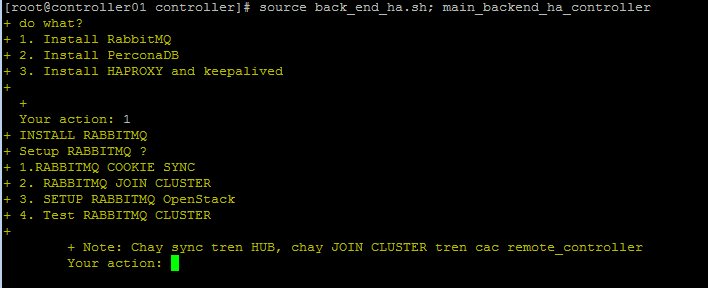
* + Cài đặt **RabbitMQ**. Sau khi cài đặt sẽ tự tắt rabbitmq-server để bước sau tạo cluster
  + Cài đặt **Memcached** **Server**. Memcached được cài đặt độc lập trên các CONTROLLER, không cần cài đặt cluster.

# Kiểm tra /etc/memcached: OPTIONS= “-l 127.0.0.1,::1,MY\_IP”

* + Nhập ‘0’ để cài toàn bộ.

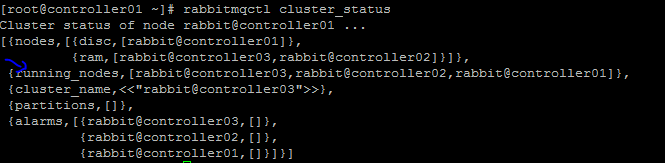
#### Cài đặt back end service theo cụm cluster controller

**Sử dụng file back\_end\_ha.sh**

source back\_end\_ha.sh; main\_backend\_ha\_controller

Lần lượt cài đặt **rabbitmq cluster, percona cluster db**, **haproxy, keepalived**

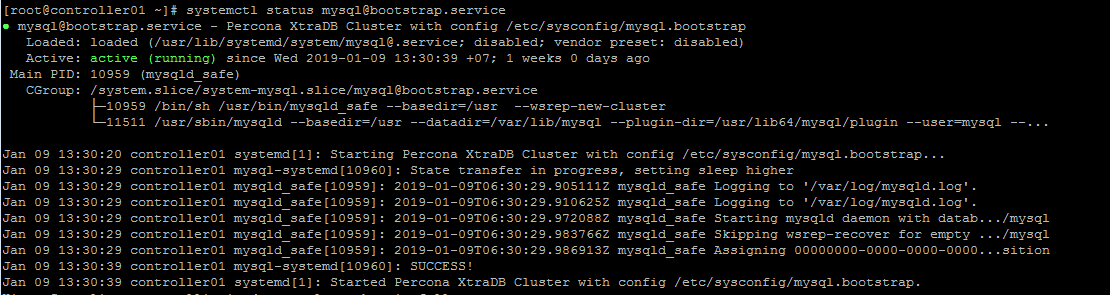
* + **Setup rabbitmq cluster.**
    - Đầu tiên trên node CONTROLLER01 chạy RABBITMQ COOKIE **SYNC** để copy file /var/lib/rabbitmq/.erlang.cookie sang CONTROLLER02 và CONTROLLER03
    - Trên các node còn lại chạy RABBITMQ JOIN CLUSTER để join vào cluster do CONTROLLER01 tạo
    - Test cluster bằng : rabbitmqctl cluster\_status

Kết quả thành công khi trả về running\_nodes là mảng chứa 3 controller hostname:

* + - **SETUP RABBITMQ Openstack:** Tạo user rabbitmq openstack chung cho cả cluster.
  + **Setup PerconaDb Cluster**
    - Trong file cài đặt sử dụng biến môi trường ${HUB} để khi cài đặt sẽ có một CONTROLLER tạo cluster, 2 CONTROLLER còn lại join vào.
    - Install Percona-XtraDB-Cluster-57
    - Tạo file cấu hình wsrep.cnf
    - Start service : start [mysql@bootstrap.service](mailto:mysql@bootstrap.service)

**#** Để kiểm tra Percona DB Cluster đã start thành công:

systemctl stop mysql

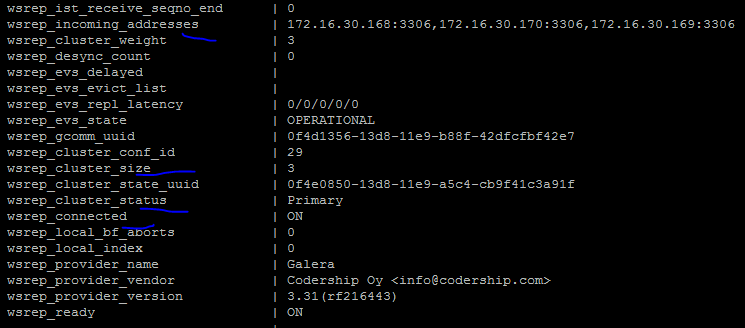
systemctl status mysql@bootstrap.service

* + - Trên các node còn lại tự join vào bằng cách start mysql.

**#** Kiểm tra cluster:

$mysql

# Danh sách addresses, cluster\_size, cluster status

$mysql[controller01] > show status like ‘wsrep\_%’;

* + **Setup HAPROXY và KEEPALIVED**

source back\_end\_ha.sh;main\_backend\_ha\_controller

* + - VIP trong file password/passwd.env sẽ được dùng làm VIP cho keepalived
    - Sau đó HAPROXY sẽ cấu hình để bind vào VIP được tạo và phân phối request qua VIP tới các controller node.

#### 4.1.3 Kiểm tra các backend service đã cài

* + Sau khi cài đặt kiểm tra Keepalived:

ip addr show

tail –f /var/log/messages

* + Kiểm tra HAPROXY:

systemctl status haproxy

* Kết thúc cài đặt có thể kiểm tra các service đã cài:

systemctl status memcached || netstat –nap | grep 11211

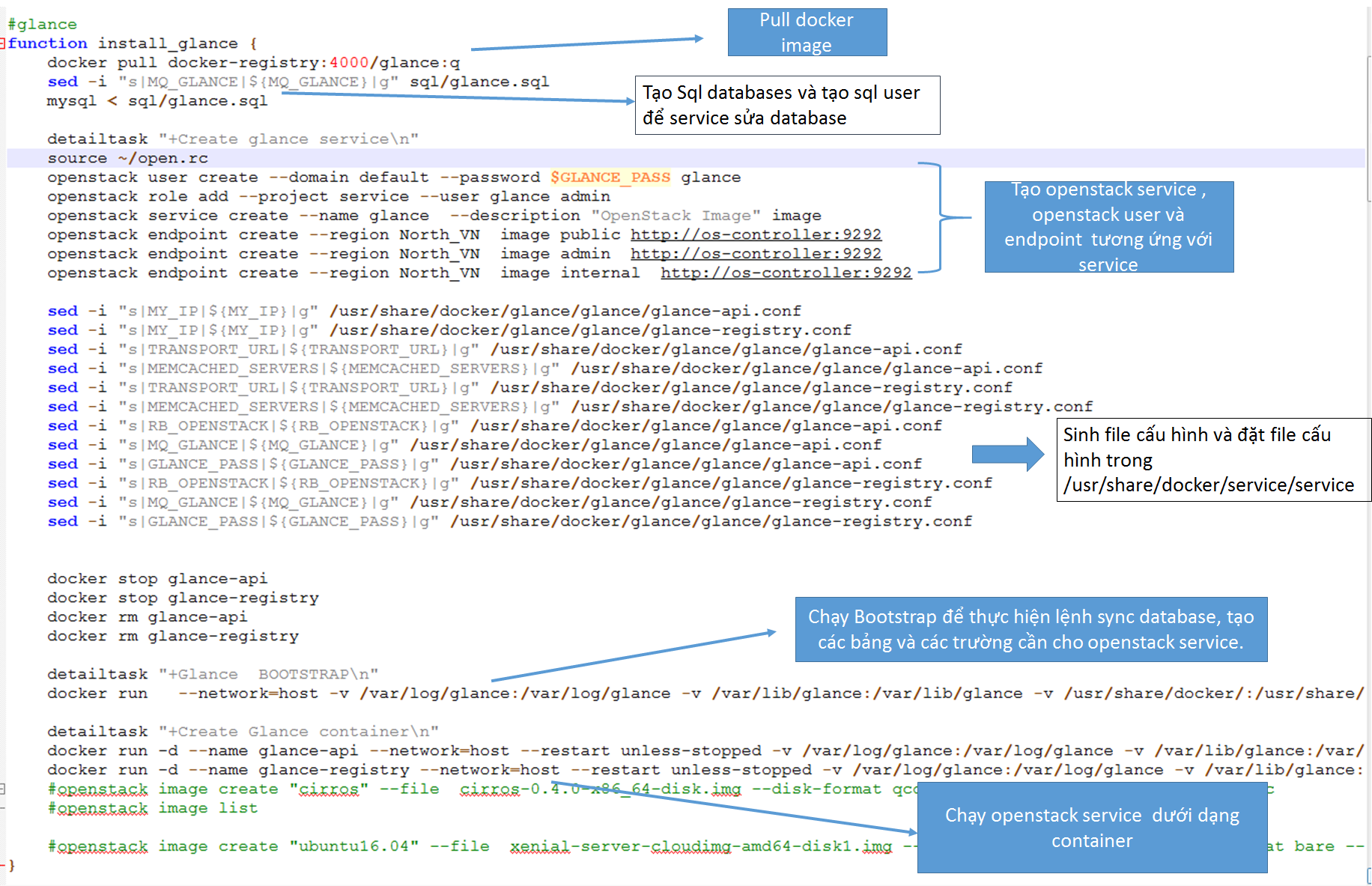
systemctl status rabbitmq-server (đảm bảo đã stop) || netstat –nap | grep 5672

systemctl status docker

systemctl status chronyd || netstat –nap | grep 123

### 4.2. Cài đặt các openstack service thông qua các Docker container

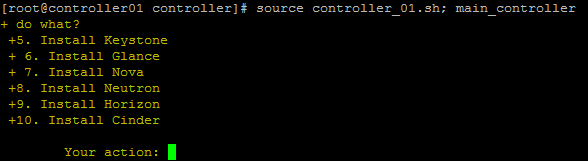
#### 4.2.1 Hướng dẫn sử dụng Docker cài đặt Openstack service

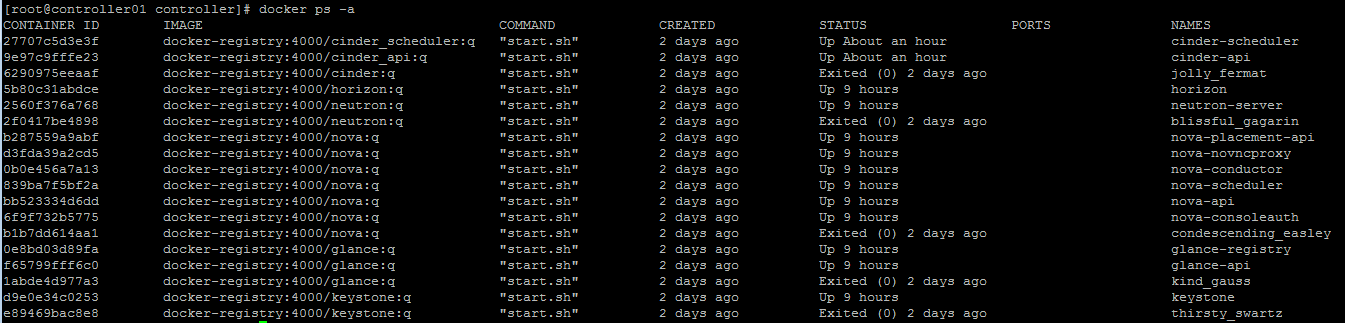
* + Các service được cài đặt qua docker:
    - Keystone
    - Glance-api, Glance-registry
    - Cinder-api, Cinder-scheduler
    - Nova-api, Nova-consoleauth, Nova-scheduler, Nova-conductor, Nova-novncproxy, Nova-placement-api
    - Neutron-server
    - Horizon
    - Neutron-ovs-agent
    - Neutron-dhcp-agent
    - Neutron-metadata-agent
    - Cinder-volume
  + Thứ tự thực hiện cài đặt các openstack service sử dụng docker là tương tự qua các bước:
    - Pull docker image từ docker-registry
    - Tạo Sql database và Sql user tương ứng với service
    - Tạo Openstack service, openstack user và openstack endpoints tương ứng với service
    - Sinh file cấu hình trong thư mục /usr/share/docker/service\_name/service\_name
    - Thực hiện Bootstrap tạo các bảng và các trường cần cho openstack service.
    - Chạy openstack service dưới dạng container
  + Ví dụ cài đặt openstack-glance:

Hình 5: Tập các lệnh cài đặt service Openstack-glance qua Docker.

* + Với một cụm 3 controller vì chỉ cần tạo openstack service, sql 1 lần trên 1 controller, các controller còn lại chỉ cần dựng docker container và bỏ qua các bước trên nên chia thành 2 file cài:
    - controller\_01.sh để cài cho controller đầu tiên
    - controller\_02.sh để cài cho các controller còn lại.

#### 4.2.2 Cài đặt các openstack service sử dụng file controller.sh

* Cài đặt bằng lệnh source controller\_01.sh; main\_controller
* Sau khi cài đặt xong kiểm tra trạng thái các container bằng lệnh docker ps –a, nếu container hoạt động bình thường sẽ có status là **Up**. Các container có tên lạ và có status Exited là container dùng cho bootstrap.



* Để kiểm tra các service hoạt động bình thường sử dụng telnet:
  + keystone: telnet os-controller 5000
  + glance-api: telnet os-controller 9292
  + glance-registry: telnet os-controller 9191
  + nova-api: telnet os-controller 8774 8775
  + nova-novncproxy: telnet os-controller 6080
  + nova-placement-api: telnet os-controller 8778
  + neutron-server: telnet os-controller 9696
  + horizon: telnet os-controller 80
  + cinder-api:telnet os-controller 8776
* Sau khi cài đặt xong trên controller01 thì cài đặt trên các controller node còn lại sử dụng file source controller\_02.sh; main\_controller

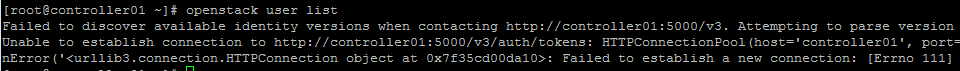
#### 4.2.3 Kiểm tra hệ thống sau khi cài đặt 3 Controller Node

* Sau khi cài đặt đủ 3 node controller kiểm tra hệ thống openstack bằng các lệnh:

source ~/open.rc

openstack user list

Danh sách các user tạo ra trong quá trình cài đặt các service, ứng với mỗi service là một user cùng tên:



Nếu kết quả trả về không tìm ra **identity version** thì kiểm tra lại keystone container với tail –f /var/log/apache2/keystone.log

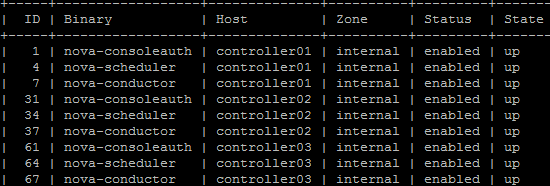
openstack service list # Danh sách service đang dùng

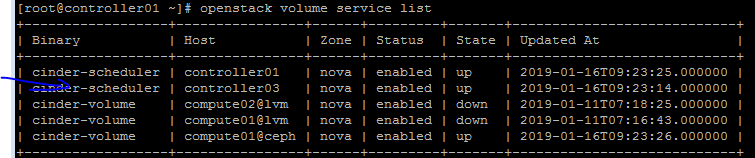
Kết quả trả về phải là danh sách service và type:

|  |  |  |
| --- | --- | --- |
| STT | Name | Type |
| 1 | keystone | identity |
| 2 | glance | image |
| 3 | nova | compute |
| 4 | placement | placement |
| 5 | neutron | network |
| 6 | cinderv2 | volumev2 |
| 7 | cinderv3 | volumev3 |

openstack compute service list

#Kiểm tra compute service và ha cluster đã hoạt động



openstack volume service list

#Kết quả trên cho thấy lỗi ở cinder-api và cinder-scheduler tại node controller02.

openstack endpoint list

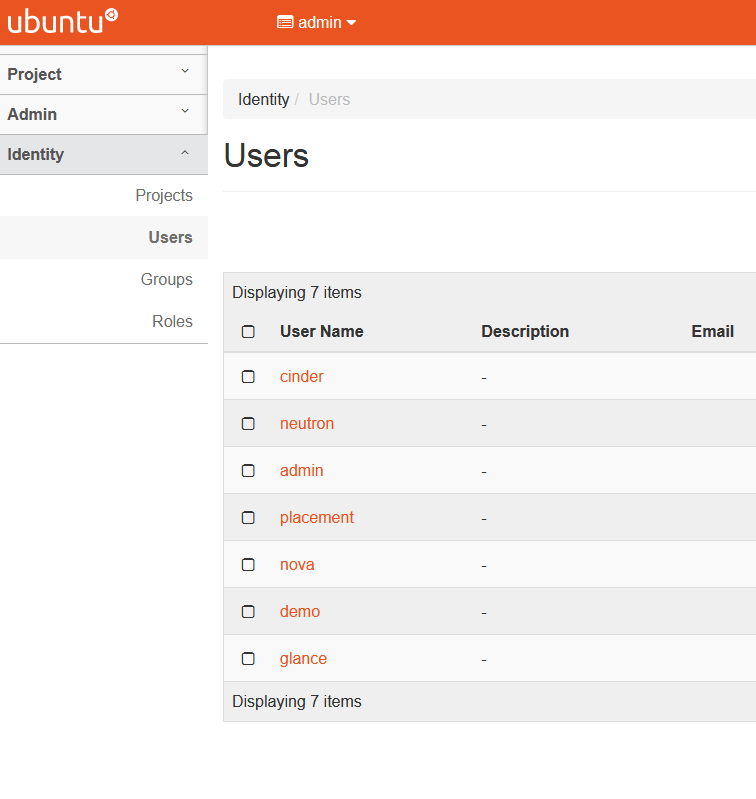
Liệt kê danh sách endpoint và URL tương ứng của các service

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Service Name | Service Type | URL |
| 1 | keystone | identity | http://os-controller:5000/v3/ |
| 2 | glance | image | http://os-controller:9292 |
| 3 | nova | compute | <http://os-controller:8774/v2.1> |
| 4 | placement | placement | http://os-controller:8778 |
| 5 | neutron | network | http://os-controller:9696 |
| 6 | cinderv2 | volumev2 | http://os-controller:8776/v2/%(project\_id)s |
| 7 | cinderv3 | volumev3 | http://os-controller:8776/v3/%(project\_id)s |

* Đăng nhập vào Horizon trên trang [**http://os-controller/horizon/auth/login**](http://os-controller/horizon/auth/login)

**user: admin**

**password: Vttek@123**



Hình 6: Trang quản lý user trong Horizon web page.

## Cài đặt cho các Compute Node:C:\Users\lamtv10\Desktop\index.png

Hình 7: Tập các service cài đặt trên Controller Node, Compute Node, Ceph node.

Yêu cầu cài đặt cho các compute node gồm:

* Python-openstack Client
* Chrony
* Openstack-nova-compute
* Openvswitch-dpdk
* Neutron agent: ovs-agent, dhcp-agent, metadata-agent.
* Cinder-volume.

### 5.1 Cài đặt back\_end service:

cd Queen\_DOCKER\_ANDOPENSTACK/compute

source backend.sh; main\_backend\_compute

* + **init\_repo:** Cài đặt python-openstackclient đồng thời tắt Selinux để tránh lỗi ovs-agent không thể kết nối tới openvswich database.

Kiểm tra selinux bằng lệnh sestatus kết quả:

SELinux status: disabled

* + **install\_docker:** Các compute node sử dụng Docker để cài đặt các network agent và cinder-volume.
  + **install\_chrony:** Tại bước cài đặt chrony, chọn nhánh 2 để chọn cài đặt chrony client. Chrony sau đó sẽ được cấu hình để đồng bộ thời gian với os-controller.

Kiểm tra chrony đã hoạt động bằng lệnh date và so sánh với kết quả trả về trên Controller Node cài đặt chrony-server.

\*\* Trong trường hợp lỗi không thể khởi động chronyd.service kiểm tra /etc/hosts xem đã set đúng ip cho os-controller và kiểm tra cấu hình /etc/chrony.conf trên os-controller ở phần bind ip và allow network.

### 5.2 Cài đặt openstack service trên Compute Node:

* Các service cài bên trong docker:
  + neutron-plugin-openvswitch-agent
  + neutron-dhcp-agent
  + neutron-metadata-agent
  + cinder-volume
* Các service cài bên ngoài docker:
  + libvirtd.service
  + openstack-nova-compute
  + openvswitch.service

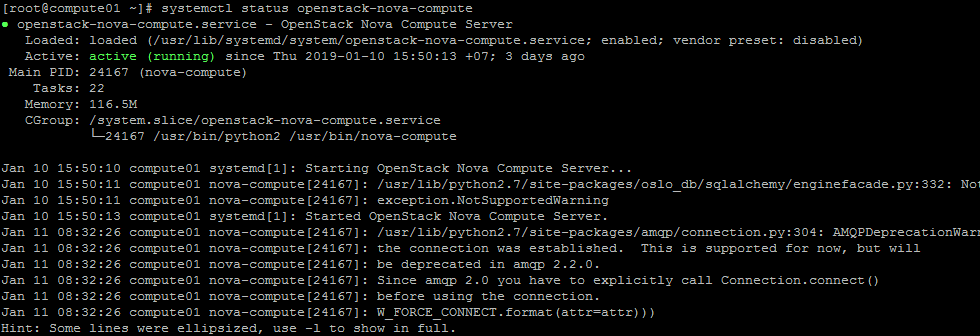
#### 5.2.1 Bổ xung các biến môi trường trong passwod/passwd.env

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên biến** | **Giải thích** | **Ví dụ** |
| 1 | HOST | Hostname của COMPUTE NODE | compute01 |
| 2 | MY\_IP | Sử dụng ip của server hiện tại dùng cho metadata-agent, ovs-agent | 172.16.30.171 |
| 3 | OS\_CONTROLLER | Địa chỉ VIP mà Keepalived tạo cho 3 controller. | 172.16.30.166 |
| 4 | FLAT\_INTERFACE | Interface trong dải DATA\_NETWORK, dùng để tạo flat\_net cho các Virtual Machine, mặc định là eth1. | eth1 |

#### 5.2.2 Cài đặt openstack-nova-compute

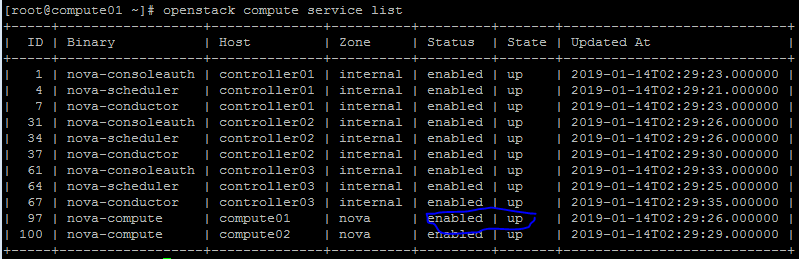
source compute.sh; main\_compute

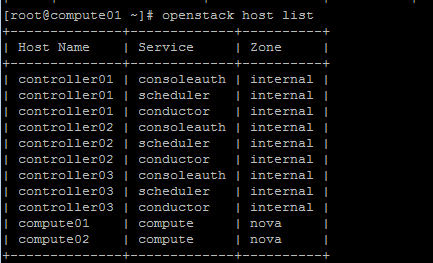
* Cài đặt openstack-nova-compute
* Copy file cấu hình tại ./nova.conf tới /etc/nova/
* Khởi động lại openstack-nova-compute.service
* Sau khi cài đặt nova-compute kiểm tra bằng các lệnh:

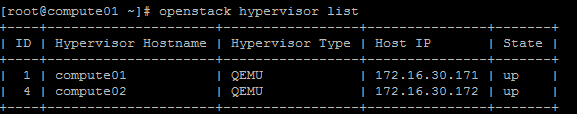
systemctl status openstack-nova-compute

* + Kiểm tra sử dụng python-openstackclient:

source ~/open.rc

****openstack compute service list

openstack host list

 openstack hypervisor list

#### Cài đặt neutron network agent

* **Bước 1: Cài đặt Openvswitch**
  + - Cài đặt openvswitch.service

yum install -y openvswitch

* + - Tạo các bridge br-flat, br-vlan

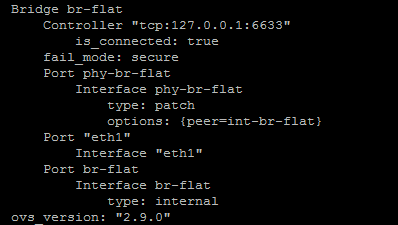
ovs-vsctl add-br br-flat

ovs-vsctl add-br br-vlan

* + - Kết nối các bridge này với các physical network interface, ví dụ kết nối br-flat tới eht1:

ovs-vsctl add-port br-flat eth1

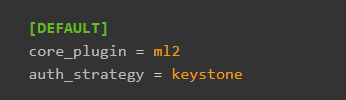
* Kiểm tra openvswitch

 ovs-vsctl show:

* **Bước 2: Dựng các container chạy network-agent**
  + - Cài đặt các network-agent gồm neutron-plugin-openvswitch-agent, neutron-dhcp-agent, neutron-metadata-agent thông qua docker container trong đó:
    - **neutron-plugin-openvswitch-agent:**

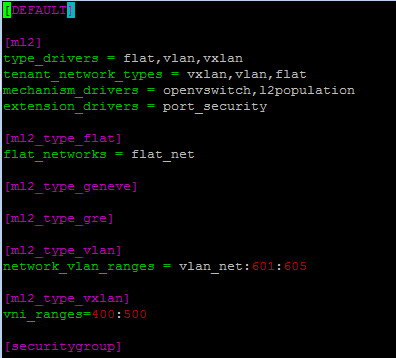
Chạy trong docker container ovs-agent

Sử dụng file cấu hình chung cho các network-agent /usr/share/docker/neutron/neutron.conf

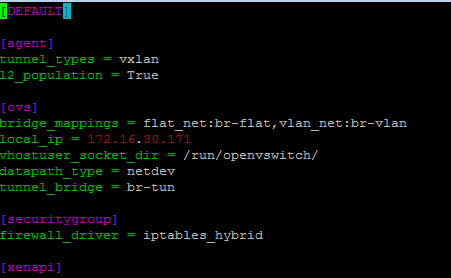


Sử dụng file cấu hình riêng của openvswitch-agent:

/usr/share/docker/neutron/neutron/plugins/ml2/ml2\_conf.ini



/usr/share/docker/neutron/neutron/plugins/ml2/openvswitch\_agent.ini

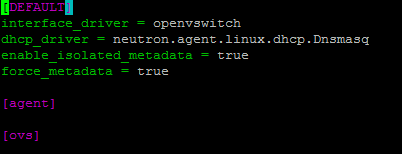


Trong đó bridge\_mappings map bridge trên openvswitch với alias của network.

* + - **neutron-dhcp-agent**

Chạy trong docker-container dhcp-agent

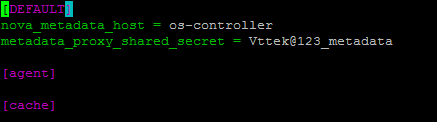
Sử dụng file cấu hình /usr/share/docker/neutron/neutron/dhcp\_agent.ini



* + - **neutron-metadata-agent**

Chạy trong docker-container metadata-agent

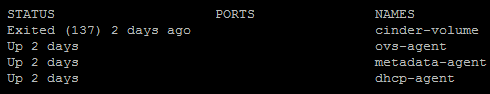
Sử dụng file cấu hình /usr/share/docker/neutron/neutron/metadata\_agent.ini



Trong đó metadata\_proxy\_share\_secret trùng với cấu hình trong file neutron.conf của CONTROLLER.

#### 5.2.4 Kiểm tra sau khi cài đặt

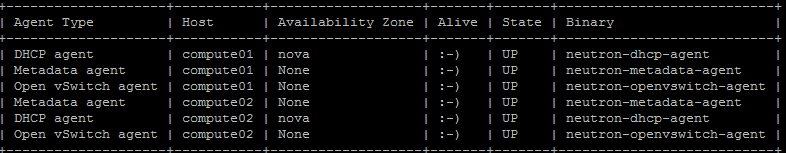
* + Kiểm tra các container:

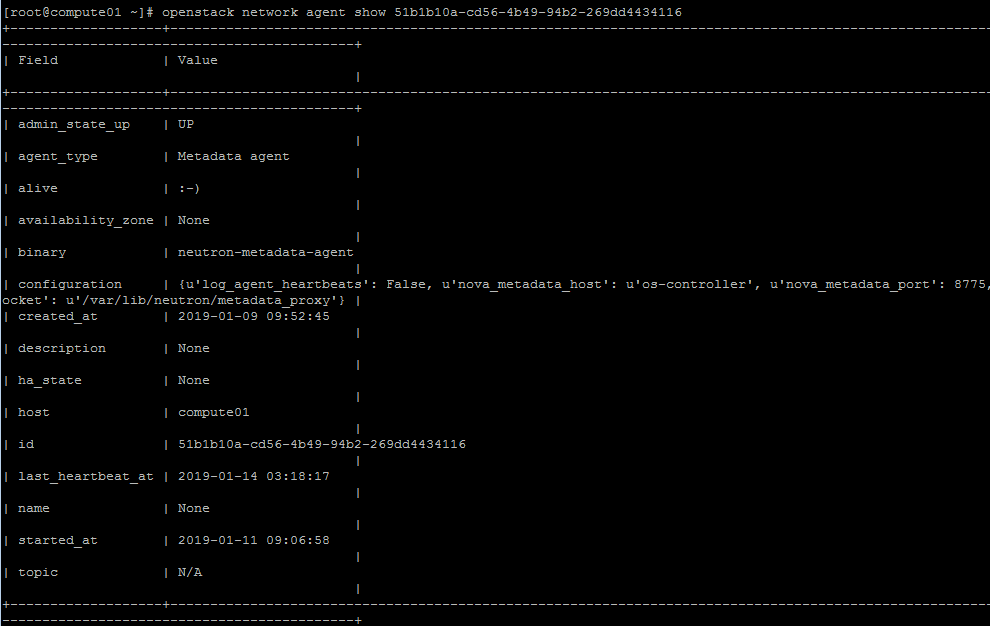


* + Kiểm tra bằng python-openstackclient:

source ~/open.rc

openstack network agent list



openstack network agent show

## Cài đặt cinder và ceph

Tham khảo tài liệu CEPH VTTEK-CEPH\_DEPLOYMENT-GUIDE.docx

## Phụ lục

Danh mục hình ảnh

[Hình 1: Các service cài đặt trên Controller Node. 1](#_Toc535421561)

[Hình 2: Các service cài đặt trên Compute Node 3](#_Toc535421562)

[Hình 3: Minh họa các server và các dải mạng cần chuẩn bị 4](file:///C:\Users\lamtv10\Downloads\VTTEK-OPENSTACK_QUEENS-PLATFORM_DEPLOYMENT-GUIDE_thuyettv1_review.docx#_Toc535421563)

[Hình 4: Các backend service cài đặt trên Controller Node 12](#_Toc535421564)

[Hình 5: Tập các lệnh cài đặt service Openstack-glance qua Docker. 19](file:///C:\Users\lamtv10\Downloads\VTTEK-OPENSTACK_QUEENS-PLATFORM_DEPLOYMENT-GUIDE_thuyettv1_review.docx#_Toc535421565)

[Hình 6: Trang quản lý user trong Horizon web page. 23](#_Toc535421566)

[Hình 7: Tập các service cài đặt trên Controller Node, Compute Node, Ceph node. 24](file:///C:\Users\lamtv10\Downloads\VTTEK-OPENSTACK_QUEENS-PLATFORM_DEPLOYMENT-GUIDE_thuyettv1_review.docx#_Toc535421567)