CentOS6.5 + OpenStack + KVM文档

[CentOS6.5 + OpenStack + KVM文档 1](#_Toc418765065)

[1. Openstack概念图 3](#_Toc418765066)

[2. 本次试验架构图 4](#_Toc418765067)

[2.1 本次采用三个节点 4](#_Toc418765068)

[3. keystone认证服务 7](#_Toc418765069)

[3.1 定义认证用户，租户，角色 8](#_Toc418765070)

[4. glance镜像服务 10](#_Toc418765071)

[4.1 安装 10](#_Toc418765072)

[4.2 配置数据库 10](#_Toc418765073)

[4.3 创建数据库并授权 10](#_Toc418765074)

[4.4 生成数据库表文件 11](#_Toc418765075)

[4.5 配置glance主配置文件 11](#_Toc418765076)

[4.6 注册glance服务api 12](#_Toc418765077)

[4.7 启动glance服务 12](#_Toc418765078)

[4.8 验证glance服务 12](#_Toc418765079)

[5. Nova计算服务 14](#_Toc418765080)

[5.1 Nova控制节点 14](#_Toc418765081)

[5.2 Nova计算节点 16](#_Toc418765082)

[6. Network网络服务 18](#_Toc418765083)

[6.1 控制节点 18](#_Toc418765084)

[6.2 网络节点 22](#_Toc418765085)

[6.3 计算节点 26](#_Toc418765086)

[7. Dashboard Web服务 31](#_Toc418765087)

[7.1 软件包安装 31](#_Toc418765088)

[7.2 启动memcached服务 31](#_Toc418765089)

[7.3 配置支持控制节点 31](#_Toc418765090)

[7.4 重启httpd和memcached服务 31](#_Toc418765091)

[8. Heat服务 32](#_Toc418765092)

[8.1 软件包安装 32](#_Toc418765093)

[8.2 配置数据库文件 32](#_Toc418765094)

[8.3 创建数据库并赋予相应权限 32](#_Toc418765095)

[8.4 生成数据库表 32](#_Toc418765096)

[8.5 配置Heat配置文件 32](#_Toc418765097)

[8.6 注册heat服务 33](#_Toc418765098)

[8.7 创建heat账户 34](#_Toc418765099)

[8.8 启动heat服务并设置开机启动 34](#_Toc418765100)

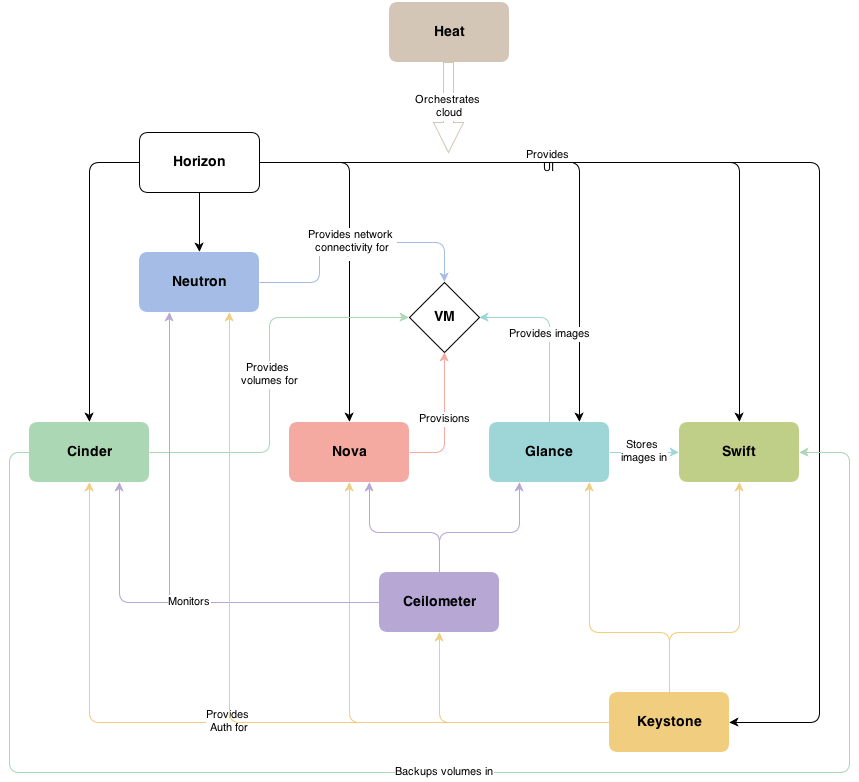
[8.9 验证 34](#_Toc418765101)

[9. 热迁移 35](#_Toc418765102)

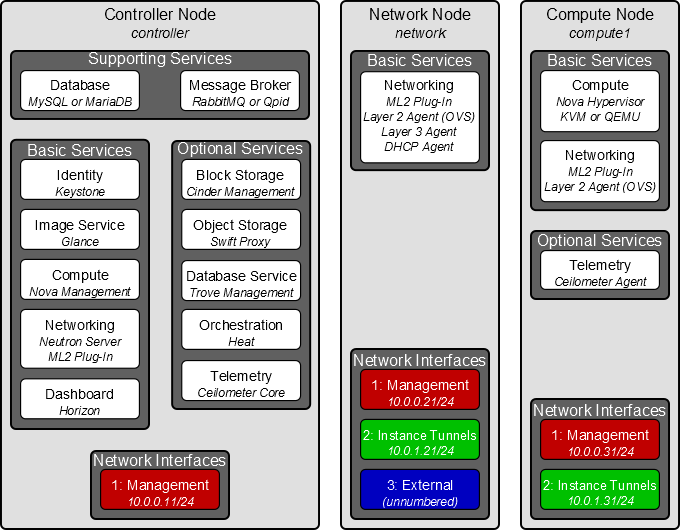
[9.1. Openstack热迁移计算节点 36](#_Toc418765103)

[Q&A 39](#_Toc418765104)

## Openstack概念图



## 本次试验架构图



### 2.1 本次采用三个节点

控制节点（**虚机**）

管理网(Management)：192.168.5.228 ---- eth0

外部网(External)：111.204.241.184 ---- eth1

网络节点（**虚机**）

管理网(Management)：192.168.5.76 ---- eth0

外部网(External)：111.204.241.209 ---- eth1

通道网(Tunnels)：111.204.241.209 ---- eth1（可以使用另外网段）

计算节点（**实机**）

管理网(Management)：192.168.5.220 ---- eth0

通道网(Tunnels)：111.204.241.187 ---- eth1（可以使用另外网段）

外部网(External)：111.204.241.187 ---- eth1

#####**PS：管理网、外部网、通道网必须在同一网段，且在对应网卡**

对应系统最低需求：

* Controller Node: 1 processor, 2 GB memory, and 5 GB storage
* Network Node: 1 processor, 512 MB memory, and 5 GB storage
* Compute Node: 1 processor, 2 GB memory, and 10 GB storage

操作系统版本：CentOS6.5 x64

配置前的准备工作：

* # service NetworkManager stop
* # service network start
* # chkconfig NetworkManager off
* # chkconfig network on
* # service firewalld stop
* # service iptables start
* # chkconfig firewalld off
* # chkconfig iptables on

网络环境准备：

在对应的网卡上配置相应的IP地址

【PS：为了具有更好的扩展性，修改hosts文件。将各个节点的主机名和IP添加进去。在三个节点都需要操作同样的事情。便于以后直接使用域名】

# controller

192.168.5.228 controller

# network

192.168.5.76 network

# compute1

192.168.5.220 compute1

在三个节点都对应配置好之后，请相互之间ping测试一下。【必须操作】

安装时间同步ntp 服务。

#yum install ntp

然后到计划任务中添加以下几行

# Puppet Name: time sync

\*/5 1-23 \* \* \* (/usr/sbin/ntpdate time.soufun.com)

# Puppet Name: time sync 40

40 0 \* \* \* (/usr/sbin/ntpdate time.soufun.com)

# Puppet Name: time sync 30

30 0 \* \* \* (/usr/sbin/ntpdate time.soufun.com)

# Puppet Name: time sync 50

50 0 \* \* \* (/usr/sbin/ntpdate time.soufun.com)

数据库—MySQL

**控制节点：**

安装mysql数据库

# yum install mysql mysql-server MySQL-python

然后再/etc/my.cnf配置文件中，增加一下几个参数

bind-address = 0.0.0.0 ##推荐修改为控制节点的IP，192.168.5.228

default-storage-engine = innodb

innodb\_file\_per\_table

collation-server = utf8\_general\_ci

init-connect = 'SET NAMES utf8'

character-set-server = utf8

然后重启服务，并删除anonymous账户

# mysql\_install\_db ###生成数据库表

# mysql\_secure\_installation ###安全操作，基本上可以直接Yes到底

**其它节点**

# yum install MySQL-python

Openstack软件包准备并更新系统环境和kernel内核

【PS：我还是推荐把163源添加进去，到163源官网可以直接下载使用，或者从现有机器上拷贝直接可以使用】

# yum install yum-plugin-priorities

# yum install http://repos.fedorapeople.org/repos/openstack/openstack-icehouse/rdo-release-icehouse-3.noarch.rpm

# yum install http://dl.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

# yum install openstack-utils

# yum install openstack-selinux

# yum upgrade

# reboot

消息队列服务，在控制节点上安装，并作为消息队列服务的服务器

# yum install qpid-cpp-server

然后取消认证

/etc/qpidd.conf

auth=no

启动服务，并设置为开机启动

# service qpidd start

# chkconfig qpidd on

## keystone认证服务

认证服务是联系所有服务。任何操作都必须经过认证才能够操作。

1. 安装认证服务

# yum install openstack-keystone python-keystoneclient

认证服务中的认证信息都是存储在数据库中，所以必须配置数据库配置文件，如下：

# openstack-config --set /etc/keystone/keystone.conf database connection mysql://keystone:*KEYSTONE\_DBPASS*@*controller*/keystone

【KEYSTONE\_DBPASS为你自己定义的数据库连接密码

Controller为你的控制节点，可以直接使用IP地址】

然后登陆数据库，创建数据库，并授予相应权限。

$ mysql -u root -p

mysql> CREATE DATABASE keystone;

mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY '*KEYSTONE\_DBPASS*';

mysql> GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY '*KEYSTONE\_DBPASS*';

mysql> exit

根据配置文件，生成数据库表

# su -s /bin/sh -c "keystone-manage db\_sync" keystone

【操作完这一步之后，一定记得去数据库中看一下，是否有表生成，如果没有，接下来的操作会失败。关于这一步失败的原因在最后的QA中有分析】

生成对应的认证文件，并写入配置文件中

# ADMIN\_TOKEN=$(openssl rand -hex 10)

# echo $ADMIN\_TOKEN#推荐echo到一个文件中，因为下面会常用这个密码

# openstack-config --set /etc/keystone/keystone.conf DEFAULT admin\_token $ADMIN\_TOKEN

对认证文件进行授权操作

# keystone-manage pki\_setup --keystone-user keystone --keystone-group keystone

# chown -R keystone:keystone /etc/keystone/ssl

# chmod -R o-rwx /etc/keystone/ssl

设置开机启动

# service openstack-keystone start

# chkconfig openstack-keystone on

这一步，非必须操作，但是推荐。【这一步作用是用于清空日志，因为日志生成太快了，当然可以使用 cat /dev/null>/var/log/keystone/keystone-tokenflush.log】

# (crontab -l -u keystone 2>&1 | grep -q token\_flush) || \

echo '@hourly /usr/bin/keystone-manage token\_flush >/var/log/keystone/keystone-tokenflush.log 2>&1' >> /var/spool/cron/keystone

### 3.1 定义认证用户，租户，角色

$ export OS\_SERVICE\_TOKEN=*ADMIN\_TOKEN* ##ADMIN\_TOKEN就是上面随机生成的密码

$ export OS\_SERVICE\_ENDPOINT=http://controller:35357/v2.0

创建一个admin用户

$ keystone user-create --name=admin --pass=*ADMIN\_PASS* --email=*ADMIN\_EMAIL*

创建一个admin角色

$ keystone role-create --name=admin

创建一个admin租户

$ keystone tenant-create --name=admin --description="Admin Tenant"

将对应的用户加入到对应的角色和租户中

$ keystone user-role-add --user=admin --tenant=admin --role=admin

Link the admin user, \_member\_ role, and admin tenant:

$ keystone user-role-add --user=admin --role=\_member\_ --tenant=admin

创建一个demo用户

$ keystone user-create --name=demo --pass=*DEMO\_PASS* --email=*DEMO\_EMAIL*

$ keystone tenant-create --name=demo --description="Demo Tenant"

$ keystone user-role-add --user=demo --role=\_member\_ --tenant=demo

$ keystone tenant-create --name=service --description="Service Tenant"

创建一个对应的认证服务

$ keystone service-create --name=keystone --type=identity

--description="OpenStack Identity"

+-------------+----------------------------------+

| Property | Value |

+-------------+----------------------------------+

| description | OpenStack Identity |

| id | 15c11a23667e427e91bc31335b45f4bd |

| name | keystone |

| type | identity |

+-------------+----------------------------------+

创建对应API接口

$ keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ identity / {print $2}') \

--publicurl=http://*controller*:5000/v2.0 \

--internalurl=http://*controller*:5000/v2.0 \

--adminurl=http://*controller*:35357/v2.0

+-------------+-----------------------------------+

| Property | Value |

+-------------+-----------------------------------+

| adminurl | http://controller:35357/v2.0 |

| id | 11f9c625a3b94a3f8e66bf4e5de2679f |

| internalurl | http://controller:5000/v2.0 |

| publicurl | http://controller:5000/v2.0 |

| region | regionOne |

| service\_id | 15c11a23667e427e91bc31335b45f4bd |

+-------------+-----------------------------------+

为了便于以后操作，切换，请创建admin-openrc.sh，作用是切换到admin环境。使用操作：source admin-openrc.sh

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_TENANT\_NAME=admin

export OS\_AUTH\_URL=http://controller:35357/v2.0

创建demo环境，创建demo-openrc.sh,作用是切换到demo环境。使用操作：source demo-openrc.sh

export OS\_USERNAME=demo

export OS\_PASSWORD=DEMO\_PASS

export OS\_TENANT\_NAME=demo

export OS\_AUTH\_URL=http://controller:35357/v2.0

查看现有用户命令：

Keystone user-list

keystone user-role-list --user admin --tenant admin

## glance镜像服务

* glance-api. Accepts Image API calls for image discovery, retrieval, and storage.
* glance-registry. Stores, processes, and retrieves metadata about images. Metadata includes items such as size and type.
* Database. Stores image metadata. You can choose your database depending on your preference. Most deployments use MySQL or SQlite.
* Storage repository for image files. The Image Service supports a variety of repositories including normal file systems, Object Storage, RADOS block devices, HTTP, and Amazon S3. Some types of repositories support only read-only usage.

### 4.1 安装

# yum install openstack-glance python-glanceclient

### 4.2 配置数据库

# openstack-config --set /etc/glance/glance-api.conf database \

connection mysql://glance:*GLANCE\_DBPASS*@*controller*/glance

# openstack-config --set /etc/glance/glance-registry.conf database \

connection mysql://glance:*GLANCE\_DBPASS*@*controller*/glance

### 4.3 创建数据库并授权

$ mysql -u root -p

mysql> CREATE DATABASE glance;

mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY '*GLANCE\_DBPASS*';

mysql> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' \

IDENTIFIED BY '*GLANCE\_DBPASS*';

### 4.4 生成数据库表文件

# su -s /bin/sh -c "glance-manage db\_sync" glance

创建keystone用户

$ keystone user-create --name=glance --pass=*GLANCE\_PASS* \

--email=*glance@example.com*

$ keystone user-role-add --user=glance --tenant=service --role=admin

### 4.5 配置glance主配置文件

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken auth\_uri http://*controller*:5000

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken auth\_host *controller*

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken auth\_port 35357

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken auth\_protocol http

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken admin\_tenant\_name service

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken admin\_user glance

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken admin\_password *GLANCE\_PASS*

# openstack-config --set /etc/glance/glance-api.conf paste\_deploy \

flavor keystone

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_uri http://*controller*:5000

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_host *controller*

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_port 35357

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_protocol http

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken admin\_tenant\_name service

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken admin\_user glance

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken admin\_password *GLANCE\_PASS*

# openstack-config --set /etc/glance/glance-registry.conf paste\_deploy flavor keystone

### 4.6 注册glance服务api

$ keystone service-create --name=glance --type=image \

--description="OpenStack Image Service"

$ keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ image / {print $2}') \

--publicurl=http://*controller*:9292 \

--internalurl=http://*controller*:9292 \

--adminurl=http://*controller*:9292

### 4.7 启动glance服务

# service openstack-glance-api start

# service openstack-glance-registry start

# chkconfig openstack-glance-api on

# chkconfig openstack-glance-registry on

### 4.8 验证glance服务

#### 4.8.1 下载glance镜像

$ mkdir /tmp/images

$ cd /tmp/images/

$ wget http://cdn.download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86\_64-disk.img

#### 4.8.2 上传镜像到glance服务中存储

$ mkdir /tmp/images

$ cd /tmp/images/

$ wget http://cdn.download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86\_64-disk.img

可以通过以下命令查看文件信息

$ file cirros-0.3.2-x86\_64-disk.img

cirros-0.3.2-x86\_64-disk.img: **QEMU** QCOW Image (v2), 41126400 bytes

#### 4.8.3 通过命令，将img制作成glance镜像

$ source admin-openrc.sh

$ glance image-create --name "cirros-0.3.2-x86\_64" --disk-format qcow2 --container-format bare --is-public True --progress < cirros-0.3.2-x86\_64-disk.img

【扩展：如果镜像是在网上，可以通过url地址使用，可用户跨机房使用的情况】

$ glance image-create --name="cirros-0.3.2-x86\_64" --disk-format=qcow2 \

--container-format=bare --is-public=true \

--copy-from http://cdn.download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86\_64-disk.img

#### 4.8.4 验证glance镜像

$ glance image-list

## Nova计算服务

### 5.1 Nova控制节点

#### 5.1.1 安装openstack nova软件包

# yum install openstack-nova-api openstack-nova-cert openstack-nova-conductor \

openstack-nova-console openstack-nova-novncproxy openstack-nova-scheduler \

python-novaclient

#### 5.1.2 配置数据库配置文件

# openstack-config --set /etc/nova/nova.conf \

database connection mysql://nova:*NOVA\_DBPASS*@*controller*/nova

#### 5.1.3 指定可以访问消息队列服务

# openstack-config --set /etc/nova/nova.conf \

DEFAULT rpc\_backend qpid

# openstack-config --set /etc/nova/nova.conf DEFAULT qpid\_hostname *controller*

#### 5.1.4 设置vnc配置

# openstack-config --set /etc/nova/nova.conf DEFAULT my\_ip 10.0.0.11

# openstack-config --set /etc/nova/nova.conf DEFAULT vncserver\_listen 10.0.0.11

# openstack-config --set /etc/nova/nova.conf DEFAULT vncserver\_proxyclient\_address 10.0.0.11

#### 5.1.5 创建数据库并授权

$ mysql -u root -p

mysql> CREATE DATABASE nova;

mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY '*NOVA\_DBPASS*';

mysql> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' \

IDENTIFIED BY '*NOVA\_DBPASS*';

#### 5.1.6 生成数据库表文件

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

#### 5.1.7 创建nova账户

$ keystone user-create --name=nova --pass=*NOVA\_PASS* --email=*nova@example.com*

$ keystone user-role-add --user=nova --tenant=service --role=admin

#### 5.1.8 配置nova配置文件

# openstack-config --set /etc/nova/nova.conf DEFAULT auth\_strategy keystone

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_uri http://*controller*:5000

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_host *controller*

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_protocol http

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_port 35357

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken admin\_user nova

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken admin\_tenant\_name service

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken admin\_password *NOVA\_PASS*

#### 5.1.9 注册nova服务

$ keystone service-create --name=nova --type=compute \

--description="OpenStack Compute"

$ keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ compute / {print $2}') \

--publicurl=http://*controller*:8774/v2/%\(tenant\_id\)s \

--internalurl=http://*controller*:8774/v2/%\(tenant\_id\)s \

--adminurl=http://*controller*:8774/v2/%\(tenant\_id\)s

#### 5.1.10 启动nova服务并设置开机启动

# service openstack-nova-api start

# service openstack-nova-cert start

# service openstack-nova-consoleauth start

# service openstack-nova-scheduler start

# service openstack-nova-conductor start

# service openstack-nova-novncproxy start

# chkconfig openstack-nova-api on

# chkconfig openstack-nova-cert on

# chkconfig openstack-nova-consoleauth on

# chkconfig openstack-nova-scheduler on

# chkconfig openstack-nova-conductor on

# chkconfig openstack-nova-novncproxy on

#### 5.1.11 验证nova服务

$ nova image-list

### 5.2 Nova计算节点

#### 5.2.1 安装nova软件

# yum install openstack-nova-compute

【如果多安装了其他服务，只要不启动服务，不会影响结果】

#### 5.2.2 编辑/etc/nova/nova.conf配置文件

# openstack-config --set /etc/nova/nova.conf database connection mysql://nova:*NOVA\_DBPASS*@controller/nova

# openstack-config --set /etc/nova/nova.conf DEFAULT auth\_strategy keystone

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_uri http://*controller*:5000

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_host *controller*

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_protocol http

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_port 35357

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken admin\_user nova

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken admin\_tenant\_name service

# openstack-config --set /etc/nova/nova.conf keystone\_authtoken admin\_password *NOVA\_PASS*

#### 5.2.3 配置消息队列

# openstack-config --set /etc/nova/nova.conf \

DEFAULT rpc\_backend qpid

# openstack-config --set /etc/nova/nova.conf DEFAULT qpid\_hostname *controller*

#### 5.2.4 配置提供远程console接口服务

# openstack-config --set /etc/nova/nova.conf DEFAULT my\_ip 10.0.0.31

# openstack-config --set /etc/nova/nova.conf DEFAULT vnc\_enabled True

# openstack-config --set /etc/nova/nova.conf DEFAULT vncserver\_listen 0.0.0.0

# openstack-config --set /etc/nova/nova.conf DEFAULT vncserver\_proxyclient\_address 10.0.0.31

# openstack-config --set /etc/nova/nova.conf \

DEFAULT novncproxy\_base\_url http://*controller*:6080/vnc\_auto.html

#### 5.2.5 指定hous运行镜像服务

# openstack-config --set /etc/nova/nova.conf DEFAULT glance\_host *controller*

##### 5.2.5.1 查看计算节点是否支持虚拟化技术

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

如果返回为0，表示不支持，修虚拟化技术修改为qemu,如果非0，表示支持虚拟化技术。则修改为kvm，当然，如果使用的是其他虚拟化机器，请对应修改。

# openstack-config --set /etc/nova/nova.conf libvirt virt\_type qemu

#### 5.2.6 启动相应服务

# service libvirtd start

# service messagebus start

# service openstack-nova-compute start

# chkconfig libvirtd on

# chkconfig messagebus on

# chkconfig openstack-nova-compute on

## 6. Network网络服务

### 6.1 控制节点

#### 6.1.1 配置网络数据库

# openstack-config --set /etc/neutron/neutron.conf database connection mysql://neutron:*NEUTRON\_DBPASS*@*controller*/neutron

#### 6.1.2 配置网络认证

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

auth\_strategy keystone

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_uri http://*controller*:5000

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_host *controller*

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_protocol http

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_port 35357

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

admin\_tenant\_name service

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

admin\_user neutron

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

admin\_password *NEUTRON\_PASS*

#### 6.1.3 配置网络使用消息队列

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

rpc\_backend neutron.openstack.common.rpc.impl\_qpid

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

qpid\_hostname *controller*

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

notify\_nova\_on\_port\_status\_changes True

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

notify\_nova\_on\_port\_data\_changes True

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

nova\_url http://*controller*:8774/v2

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

nova\_admin\_username nova

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

nova\_admin\_tenant\_id $(keystone tenant-list | awk '/ service / { print $2 }')

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

nova\_admin\_password *NOVA\_PASS*

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

nova\_admin\_auth\_url http://*controller*:35357/v2.0

#### 6.1.4 配置网络使用Modular Layer 2 (ML2)插件

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

core\_plugin ml2

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

service\_plugins router

##### 6.1.4.1 配置ml2配置文件

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

type\_drivers gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

tenant\_network\_types gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

mechanism\_drivers openvswitch

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_gre \

tunnel\_id\_ranges 1:1000

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup \

firewall\_driver neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup \

enable\_security\_group True

##### 6.1.4.2 在nova.conf配置，支持neutron

# openstack-config --set /etc/nova/nova.conf DEFAULT \

network\_api\_class nova.network.neutronv2.api.API

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_url http://*controller*:9696

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_auth\_strategy keystone

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_tenant\_name service

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_username neutron

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_password *NEUTRON\_PASS*

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_auth\_url http://*controller*:35357/v2.0

# openstack-config --set /etc/nova/nova.conf DEFAULT \

linuxnet\_interface\_driver nova.network.linux\_net.LinuxOVSInterfaceDriver

# openstack-config --set /etc/nova/nova.conf DEFAULT \

firewall\_driver nova.virt.firewall.NoopFirewallDriver

# openstack-config --set /etc/nova/nova.conf DEFAULT \

security\_group\_api neutron

#### 6.1.5 将ml2插件加入

# ln -s plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

#### 6.1.6 启动相应服务并设置开机启动

# service openstack-nova-api restart

# service openstack-nova-scheduler restart

# service openstack-nova-conductor restart

# service neutron-server start

# chkconfig neutron-server on

### 6.2 网络节点

#### 6.2.1 基本环境设置

编辑sysctl.conf，修改为如下，如果没有，则添加

net.ipv4.ip\_forward=1

net.ipv4.conf.all.rp\_filter=0

net.ipv4.conf.default.rp\_filter=0

net.bridge.bridge-nf-call-arptables=1

net.bridge.bridge-nf-call-iptables=1

net.bridge.bridge-nf-call-ip6tables=1

然后sysctl -p让配置的环境变量生效

#### 6.2.2 安装网络插件服务

# yum install openstack-neutron openstack-neutron-ml2 \

openstack-neutron-openvswitch

#### 6.2.3 配置网络

##### 6.2.3.1 配置认证

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

auth\_strategy keystone

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_uri http://*controller*:5000

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_host *controller*

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_protocol http

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

auth\_port 35357

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

admin\_tenant\_name service

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

admin\_user neutron

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken \

admin\_password *NEUTRON\_PASS*

##### 6.2.3.2 配置消息块

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

rpc\_backend neutron.openstack.common.rpc.impl\_qpid

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

qpid\_hostname *controller*

##### 6.2.3.3 配置支持ml2和router

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

core\_plugin ml2

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

service\_plugins router

#### 6.3 配置L3客户端

# openstack-config --set /etc/neutron/l3\_agent.ini DEFAULT \

interface\_driver neutron.agent.linux.interface.OVSInterfaceDriver

# openstack-config --set /etc/neutron/l3\_agent.ini DEFAULT \

use\_namespaces True

#### 6.4 配置DHCP客户端

# openstack-config --set /etc/neutron/dhcp\_agent.ini DEFAULT \

interface\_driver neutron.agent.linux.interface.OVSInterfaceDriver

# openstack-config --set /etc/neutron/dhcp\_agent.ini DEFAULT \

dhcp\_driver neutron.agent.linux.dhcp.Dnsmasq

# openstack-config --set /etc/neutron/dhcp\_agent.ini DEFAULT \

use\_namespaces True

# openstack-config --set /etc/neutron/dhcp\_agent.ini DEFAULT \

dnsmasq\_config\_file /etc/neutron/dnsmasq-neutron.conf

新建文件/etc/neutron/dnsmasq-neutron.conf **必须对应**

dhcp-option-force=26,1454

配置好之后，查看是否有dnsmasq服务，如果有，就杀死，没有，就OK

#### 6.5 配置metadata客户端

# openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT \

auth\_url http://*controller*:5000/v2.0

# openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT \

auth\_region regionOne

# openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT \

admin\_tenant\_name service

# openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT \

admin\_user neutron

# openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT \

admin\_password *NEUTRON\_PASS*

# openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT \

nova\_metadata\_ip *controller*

# openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT \

metadata\_proxy\_shared\_secret *METADATA\_SECRET*

**此条在控制节点上操作，目的是让计算节点支持metadata，下面的METADATA\_SECRET可以任意指定,我使用的是soufun**

# openstack-config --set /etc/nova/nova.conf DEFAULT \

service\_neutron\_metadata\_proxy true

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_metadata\_proxy\_shared\_secret *METADATA\_SECRET*

然后再控制节点上重启一下服务

# service openstack-nova-api restart

#### 6.6 配置L2配置文件

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

type\_drivers gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

tenant\_network\_types gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

mechanism\_drivers openvswitch

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_gre \

tunnel\_id\_ranges 1:1000

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ovs \

local\_ip *INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS*

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ovs \

tunnel\_type gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ovs \

enable\_tunneling True

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup \

firewall\_driver neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup \

enable\_security\_group True

#### 6.7 启动Openvswitch服务

# service openvswitch start

# chkconfig openvswitch on

# ovs-vsctl add-br br-int

# ovs-vsctl add-br br-ex

# ovs-vsctl add-port br-ex *INTERFACE\_NAME*

【PS：官网推荐关掉gro，可以不用关，直接忽略

# ethtool -K *INTERFACE\_NAME* gro off】

# ln -s plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini 创建软连接

修改启动服务，这个主要是openstack的bug造成的

# cp /etc/init.d/neutron-openvswitch-agent /etc/init.d/neutron-openvswitch-agent.orig

# sed -i 's,plugins/openvswitch/ovs\_neutron\_plugin.ini,plugin.ini,g' /etc/init.d/neutron-openvswitch-agent

#### 6.8 启动openstack服务

# service neutron-openvswitch-agent start

# service neutron-l3-agent start

# service neutron-dhcp-agent start

# service neutron-metadata-agent start

# chkconfig neutron-openvswitch-agent on

# chkconfig neutron-l3-agent on

# chkconfig neutron-dhcp-agent on

# chkconfig neutron-metadata-agent on

### 6.3 计算节点

#### 6.3.1 基本环境设置

编辑sysctl.conf，修改为如下，如果没有，则添加

net.ipv4.ip\_forward=1

net.ipv4.conf.all.rp\_filter=0

net.ipv4.conf.default.rp\_filter=0

net.bridge.bridge-nf-call-arptables=1

net.bridge.bridge-nf-call-iptables=1

net.bridge.bridge-nf-call-ip6tables=1

然后sysctl -p让配置的环境变量生效

#### 6.3.2 安装网络组件

# yum install openstack-neutron-ml2 openstack-neutron-openvswitch

#### 6.3.3 配置网络公共组件

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

auth\_strategy keystone

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken auth\_uri http://*controller*:5000

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken auth\_host *controller*

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken auth\_protocol http

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken auth\_port 35357

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken admin\_tenant\_name service

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken admin\_user neutron

# openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken admin\_password *NEUTRON\_PASS*

#### 6.3.4 配置使用消息块

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

rpc\_backend neutron.openstack.common.rpc.impl\_qpid

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

qpid\_hostname *controller*

#### 6.3.5 配置ml2和router

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

core\_plugin ml2

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

service\_plugins router

#### 6.3.6 配置m2配置文件

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

type\_drivers gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

tenant\_network\_types gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 \

mechanism\_drivers openvswitch

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_gre tunnel\_id\_ranges 1:1000

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ovs \

local\_ip *INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS*

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ovs \

tunnel\_type gre

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ovs \

enable\_tunneling True

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup firewall\_driver neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

# openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup enable\_security\_group True

#### 6.3.7 配置OVS服务

# service openvswitch start

# chkconfig openvswitch on

# ovs-vsctl add-br br-int

#### 6.3.8 配置nova使用neutron

# openstack-config --set /etc/nova/nova.conf DEFAULT \

network\_api\_class nova.network.neutronv2.api.API

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_url http://*controller*:9696

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_auth\_strategy keystone

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_tenant\_name service

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_username neutron

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_password *NEUTRON\_PASS*

# openstack-config --set /etc/nova/nova.conf DEFAULT \

neutron\_admin\_auth\_url http://*controller*:35357/v2.0

# openstack-config --set /etc/nova/nova.conf DEFAULT \

linuxnet\_interface\_driver nova.network.linux\_net.LinuxOVSInterfaceDriver

# openstack-config --set /etc/nova/nova.conf DEFAULT \

firewall\_driver nova.virt.firewall.NoopFirewallDriver

# openstack-config --set /etc/nova/nova.conf DEFAULT \

security\_group\_api neutron

创建软连接

# ln -s plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

# cp /etc/init.d/neutron-openvswitch-agent /etc/init.d/neutron-openvswitch-agent.orig

# sed -i 's,plugins/openvswitch/ovs\_neutron\_plugin.ini,plugin.ini,g' /etc/init.d/neutron-openvswitch-agent

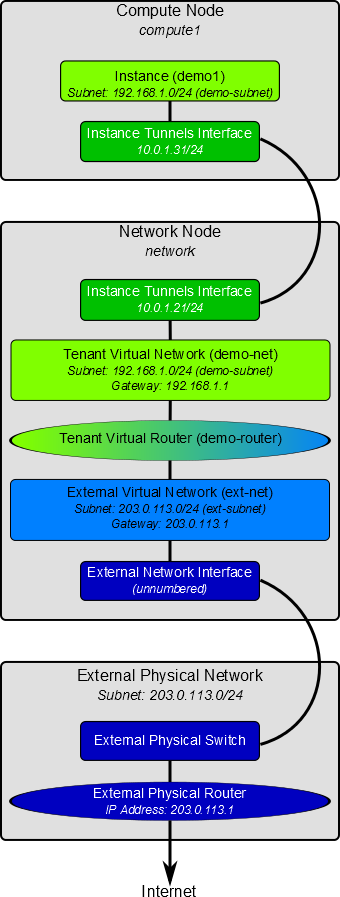
# service openstack-nova-compute restart

# service neutron-openvswitch-agent start

# chkconfig neutron-openvswitch-agent on

到此为止，网络配置全部结束。

#### 6.3.9 贴出实例到外网网络流程图



## 7. Dashboard Web服务

多说一句，Dashboard是基于python下的框架Django。所以如果想要使用这个功能，python版本必须支持，推荐使用Python2.6，因为Openstack Icehouse官方使用的python版本也是Python2.6。

### 7.1 软件包安装

# yum install memcached python-memcached mod\_wsgi openstack-dashboard

### 7.2 启动memcached服务

注意，必须启动，如果配置了下面，但是memcached没有启动，肯定会报错。

CACHES = {

'default': {

'BACKEND' : 'django.core.cache.backends.memcached.MemcachedCache',

'LOCATION' : '127.0.0.1:11211'

}

}

### 7.3 配置支持控制节点

OPENSTACK\_HOST = "controller"

# setsebool -P httpd\_can\_network\_connect on

### 7.4 重启httpd和memcached服务

# service httpd start

# service memcached start

# chkconfig httpd on

# chkconfig memcached on

接下来，就可以通过以下URL地址访问了

 http://controller/dashboard

## 8. Heat服务

### 8.1 软件包安装

# yum install openstack-heat-api openstack-heat-engine \

openstack-heat-api-cfn

### 8.2 配置数据库文件

# yum install openstack-heat-api openstack-heat-engine \

openstack-heat-api-cfn

### 8.3 创建数据库并赋予相应权限

$ mysql -u root -p

mysql> CREATE DATABASE heat;

mysql> GRANT ALL PRIVILEGES ON heat.\* TO 'heat'@'localhost' IDENTIFIED BY '*HEAT\_DBPASS*';

mysql> GRANT ALL PRIVILEGES ON heat.\* TO 'heat'@'%' \

IDENTIFIED BY '*HEAT\_DBPASS*';

### 8.4 生成数据库表

# su -s /bin/sh -c "heat-manage db\_sync" heat

### 8.5 配置Heat配置文件

# openstack-config --set /etc/heat/heat.conf DEFAULT qpid\_hostname *controller*

$ keystone user-create --name=heat --pass=*HEAT\_PASS* \

--email=*heat@example.com*

$ keystone user-role-add --user=heat --tenant=service --role=admin

# openstack-config --set /etc/heat/heat.conf keystone\_authtoken \

auth\_uri http://*controller*:5000/v2.0

# openstack-config --set /etc/heat/heat.conf keystone\_authtoken \

auth\_port 35357

# openstack-config --set /etc/heat/heat.conf keystone\_authtoken \

auth\_protocol http

# openstack-config --set /etc/heat/heat.conf keystone\_authtoken \

admin\_tenant\_name service

# openstack-config --set /etc/heat/heat.conf keystone\_authtoken \

admin\_user heat

# openstack-config --set /etc/heat/heat.conf keystone\_authtoken \

admin\_password*HEAT\_PASS*

# openstack-config --set /etc/heat/heat.conf ec2authtoken \

auth\_uri http://*controller*:5000/v2.0

### 8.6 注册heat服务

$ keystone service-create --name=heat --type=orchestration \

--description="Orchestration"

$ keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ orchestration / {print $2}') \

--publicurl=http://*controller*:8004/v1/%\(tenant\_id\)s \

--internalurl=http://*controller*:8004/v1/%\(tenant\_id\)s \

--adminurl=http://*controller*:8004/v1/%\(tenant\_id\)s

$ keystone service-create --name=heat-cfn --type=cloudformation \

--description="Orchestration CloudFormation"

$ keystone endpoint-create \

--service-id=$(keystone service-list | awk '/ cloudformation / {print $2}') \

--publicurl=http://*controller*:8000/v1 \

--internalurl=http://*controller*:8000/v1 \

--adminurl=http://*controller*:8000/v1

### 8.7 创建heat账户

$ keystone role-create --name heat\_stack\_user

# openstack-config --set /etc/heat/heat.conf \

DEFAULT heat\_metadata\_server\_url http://*10.0.0.11*:8000

# openstack-config --set /etc/heat/heat.conf \

DEFAULT heat\_waitcondition\_server\_url http://*10.0.0.11*:8000/v1/waitcondition

### 8.8 启动heat服务并设置开机启动

# service openstack-heat-api start

# service openstack-heat-api-cfn start

# service openstack-heat-engine start

# chkconfig openstack-heat-api on

# chkconfig openstack-heat-api-cfn on

# chkconfig openstack-heat-engine on

### 8.9 验证

Heat需要事先创建模板文件，官方格式为yaml格式。如下：

heat\_template\_version: 2013-05-23

description: Test Template

parameters:

ImageID:

type: string

description: Image use to boot a server

NetID:

type: string

description: Network ID for the server

resources:

server1:

type: OS::Nova::Server

properties:

name: "Test server"

image: { get\_param: ImageID }

flavor: "m1.tiny"

networks:

- network: { get\_param: NetID }

outputs:

server1\_private\_ip:

description: IP address of the server in the private network

value: { get\_attr: [ server1, first\_address ] }

如上所示，为一个创建实例的yaml模块文件

$ NET\_ID=$(nova net-list | awk '/ demo-net / { print $2 }')

$ heat stack-create -f test-stack.yml \

-P "ImageID=cirros-0.3.2-x86\_64;NetID=$NET\_ID" testStack

然后通过指定网段和相应的镜像文件以及模板文件既可以创建相应的实例

## 9. 热迁移

热迁移配置很简单，只需要了解了，就OK了

和上一样，我们采用的是 两个计算节点+一个控制节点（热迁移不涉及的，没有列出）

采用热迁移【共享存储：即将所有的计算节点都挂载在NFS共享磁盘上】

为什么采用共享存储热迁移？

采用共享存储热迁移技术，迁移的时候，只需要将内存迁移到另外一台KVM机器上即可。系统文件不需要从一台机器拷贝到另外一台机器。避免因为网络，磁盘IO等各种其他因素的干扰。

计算节点1：compute1: 111.204.241.187 - 192.168.5.220

计算节点2：compute2:111.204.241.240 - 192.168.5.209

我将NFS服务器放在了计算节点1上。【生成环境，需要单独放到一台NFS服务器上】

【**PS:nfs配置，在此不再阐述，google一下，需要注意，挂载的时需要加上nolock 参数**】

并注意权限设置问题

### 9.1. Openstack热迁移计算节点

#### 9.1.1 配置如下：

首先需要将实例存储位置都指向NFS上的相同位置，这里采用的是/data1

[root@jyw-w-kvmtest2 ~]# df -Th

Filesystem Type Size Used Avail Use% Mounted on

…..

192.168.5.220:/ nfs4 276G 2.9G 259G 2% /data1

[root@jyw-w-kvmtest2 ~]# ls /data1/nfs/lib/nova/ ##需要将/var/lib/nova下的全部拷贝到这里

buckets images instances keys networks tmp

Vim /etc/nova/nova.conf [两台计算节点都需要，如果有多台，多台都需要]

state\_path=/data1/nfs/lib/nova ##必须，否则不成功

instances\_path=/data1/nfs/lib/nova/instances ###必须，否则不成功

vncserver\_listen = 0.0.0.0 ###必须，否则不成功

###########

在各个主机直接，需要配置DNS服务，当然可以直接修改hosts文件，让个机器之间，可以通过主机名互通

例如：

192.168.5.220 jyw-o-kvmtest.light.soufun.com jyw-o-kvmtest

192.168.5.209 jyw-w-kvmtest2.light.soufun.com jyw-w-kvmtest2

#### 9.1.2 /etc/export

/data1 192.168.4.0/22(rw,sync,fsid=0,no\_all\_squash,no\_root\_squash) 111.204.241.128/25(rw,sync,fsid=0,no\_all\_squash,no\_root\_squash)

针对/data1/nfs下面的文件，我设置权限为777,上线的话，可以将权限设置小一点。

至少需要执行权限，倒不如设置777，重启nfs服务

#### 9.1.3. 修改/etc/nova/nova.conf

修改/etc/nova/nova.conf配置文件，让他支持热迁移，默认是不支持，需要额外配置

| **Configuration option = Default value** | **Description** |
| --- | --- |
| **[DEFAULT]** | |
| live\_migration\_retry\_count = *30* | (IntOpt) Number of 1 second retries needed in live\_migration |
| **[libvirt]** | |
| live\_migration\_bandwidth = *0* | (IntOpt) Maximum bandwidth to be used during migration, in Mbps |
| live\_migration\_flag = *VIR\_MIGRATE\_UNDEFINE\_SOURCE, VIR\_MIGRATE\_PEER2PEER, VIR\_MIGRATE\_LIVE, VIR\_MIGRATE\_TUNNELLED* | (StrOpt) Migration flags to be set for live migration |
| live\_migration\_uri = *qemu+tcp://%s/system* | (StrOpt) Migration target URI (any included "%s" is replaced with the migration target hostname) |

列出：

live\_migration\_retry\_count = *30*

live\_migration\_bandwidth = *0*

live\_migration\_flag = *VIR\_MIGRATE\_UNDEFINE\_SOURCE, VIR\_MIGRATE\_PEER2PEER, VIR\_MIGRATE\_LIVE, VIR\_MIGRATE\_TUNNELLED*

live\_migration\_uri = *qemu+tcp://%s/system*

live\_migration\_flag=VIR\_MIGRATE\_UNDEFINE\_SOURCE,VIR\_MIGRATE\_PEER2PEER,VIR\_MIGRATE\_LIVE, VIR\_MIGRATE\_TUNNELLED ###需要修改这条的默认

#### 9.1.4. 修改libvirtd

cat /etc/libvirt/libvirtd.conf |egrep -v "^#|^$"

listen\_tls = 0

listen\_tcp = 1

auth\_tcp = "none"

vim /etc/sysconfig/libvirtd ##开启监听libvirtd，需要放开16509默认端口，可以修改

LIBVIRTD\_ARGS="--listen"

#### 9.1.5 块迁移

##### 首先必须配置无密码访问

将nova账户设置为可登陆，修改/etc/passwd nova /bin/bash

然后设置nova账户公私钥，ssh-keygen

然后传到所有的计算节点，因为在openstack中，各个机器迁移的时候，是随机的，更具各个机器的负载情况，然后nova调度器进行调度。

ssh-copy-cp –I .ssh/id\_rsa.pub nova@$ip

##### 修改一下的配置文件

/etc/libvirt/libvirtd.conf

listen\_tls = 0

listen\_tcp = 1

auth\_tcp = "none"

$ nova live-migration --block\_migrate med server-01

/etc/nova/nova.conf中下面配置，也可以使用系统默认的

block\_migration\_flag=VIR\_MIGRATE\_UNDEFINE\_SOURCE,VIR\_MIGRATE\_PEER2PEER,VIR\_MIGRATE\_NON\_SHARED\_INC

##### 修改数据库，设置多一些数据库连接数

当控制节点中提示如下错误时，是数据库连接数做了限制，解决方案在/etc/my.cnf文件里面添加如下信息：

##connection###########

max\_connections = 1700

max\_connect\_errors = 10000

back\_log = 1000

interactive\_timeout = 150

wait\_timeout = 150

## Q&A

1. 在# su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf \

--config-file /etc/neutron/plugin.ini upgrade head" neutron时，在日志中找到ImportError: Class ml2 cannot be found ，解决办法如下：

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

core\_plugin neutron.plugins.ml2.plugin.Ml2Plugin

# openstack-config --set /etc/neutron/neutron.conf DEFAULT \

service\_plugins neutron.services.l3\_router.l3\_router\_plugin.L3RouterPlugin

2. 在配置好了所有的服务之后，通过$ip/dashboard访问的时候，提示没有权限，需要修改一下文件

/etc/openstack-dashboard/local\_settings 中添加本机的IP地址即可，如果不用考虑安全问题，直接使用”\*”,这种方式不推荐，不够安全

ALLOWED\_HOSTS = ['localhost', 'my-desktop', '111.204.241.184']

3. 如果使用虚拟机做环境，必须将网络模式修改为混杂模式，即允许不是自己的数据包通过。

4. OpenStack中虚拟化技术在哪里配置

主要两个文件：nova.conf nova-compute.conf。

其中的虚拟技术应该一致，比如都为kvm，或者都为qemu。

5. noVNC不能连接到实例，提示“Connection Refuesd”，什么原因  
可能是控制节点在收到vnc请求的时候，无法解析计算节点的主机名，从而无法和计算节点上的实例建立连接。  
另外可能是，当前浏览器不支持或者不能访问，将计算节点的ip和主机名的对应关系加入到控制节点的/etc/hosts文件中

6. 如果创建实例之后，不能够获取分配的IP地址，请查看是否已经启动了对应的DNSMASQ服务进程。dnsmasq服务是监控dhcp服务进程状态的。可以通过查看计算节点的/var/log/message的信息找到相应的错误

7. su -s /bin/sh -c "keystone-manage db\_sync" keystone 这里，如果在数据库中没有生成表，则是keystone.conf文件中的mysql配置错误

8. 在日志中找到显示binding-failed，表示使用的虚拟化技术错误，或者计算节点没有启动，导致服务创建实例。错误： 创建实例 "dhcp-cirros" 失败: 请稍后再试 [错误: No valid host was found. ].

9. /usr/lib64/python2.6/site-packages/Crypto/Util/number.py:57: PowmInsecureWarning: Not using mpz\_powm\_sec. You should rebuild using libgmp >= 5 to avoid timing attack vulnerability.

\_warn("Not using mpz\_powm\_sec. You should rebuild using libgmp >= 5 to avoid timing attack vulnerability.", PowmInsecureWarning，此提示随不是错误，但是感觉就是不爽，其实就是版本问题

10. # cinder create --display-name myVolume 1

ERROR: Unauthorized (HTTP 401)

顾名思义，肯定是认证问题，去查看cinder的认证配置是否正确

11. Failed to schedule\_create\_volume: No valid host was found

###权限不够，使用admin权限即可

12. ERROR: Policy doesn't allow volume\_extension:services to be performed. (HTTP 403)

###权限不够，使用admin权限即可

13. NAT表记录满，导致漫天的流量

/var/log/message日志信息如下：Apr 20 03:27:44 jyw-o-kvmtest kernel: \_\_ratelimit: 3712 callbacks suppressed

Apr 20 03:27:44 jyw-o-kvmtest kernel: nf\_conntrack: table full, dropping packet.

解决办法：

sysctl –w net.netfilter.nf\_conntrack\_max = 2097152

<http://www.cnblogs.com/higkoo/articles/iptables_tunning_for_conntrack.html>

14 dnsmasq **版本不支持，下载dnsmasp2.68**

**15. Command: ['sudo', 'neutron-rootwrap', '/etc/neutron/rootwrap.conf', 'kill', '-9', '19885']**

[**https://wiki.openstack.org/wiki/Rootwrap**](https://wiki.openstack.org/wiki/Rootwrap)

参考文档：

<http://docs.openstack.org/icehouse/install-guide/install/yum/content/install_dashboard.html>

<http://www.openstack.cn/>

<https://ask.openstack.org/zh/questions/>

<https://wiki.openstack.org/wiki/Main_Page>

<http://docs.openstack.org/cli-reference/content/>