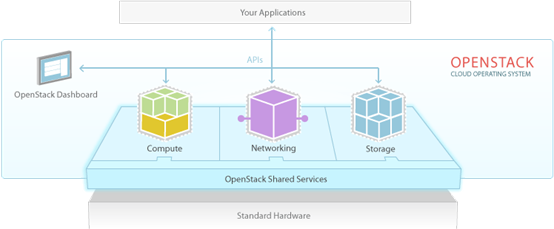
# **centos7安装部署openstack**

# 1、openstack介绍

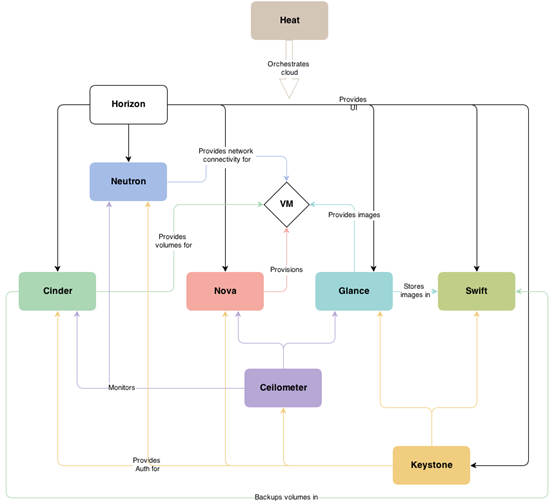
openstack是一个云平台管理的项目，它不是一个软件。也就是说我们可以使用openstack来管理我们一个数据中心大量资源池。它里面包含了很多子项目



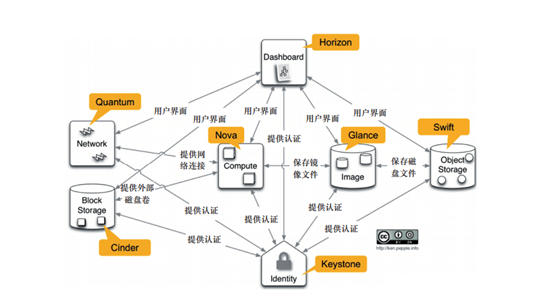
**openstack包含三大项:计算 网络 存储**

openstack主要目标是来简化资源的管理和分配，把计算 网络 存储。三大项虚拟成三大资源池，例如需要计算资源我这里可以提供，需要网络资源这里也可以提供以及存储资源的需求，对外提供api，通过api进行交互

**openstack架构**



服务名称 项目名称 描述 Dasgviard Horizon 基于Openstack API接口使用diango开发的Web管理 Compute Nova 通过虚拟化技术提供计算资源池 Networking Neutron 实现了虚拟机的网络资源管理。 Storage (存储) Object Storage Swift 对象存储，适用于“一次写入、多次读取” Block Storage Cinder 块存储，提供存储资源池 Share Services （共享服务） Identify Service Keystone 认证管理 Image Service Glance 提供虚拟镜像的注册和存储管理 Telemetry Ceilometer 提供监控和数据采集、计量服务 Higher-level Services (高层服务) Orchestration Heat 自动化部署的组件 Database Service Trove 提供数据库应用服务



**Openstack服务介绍**

MySQL：为各个服务提供数据存储

RabbitMq：为各个服务之间通信提供认证和服务注册

Keystone：为各个服务器之间通讯提供认证和服务注册

Glance：为虚拟机提供镜像管理

Nova：为虚拟机提供计算资源

Neutron：为虚拟机提供网络资源

# 2、部暑环境

## 2.1、主机信息

|  |  |  |  |
| --- | --- | --- | --- |
| 主机名 | IP地址 | 角色 | 版本 |
| controller | 192.168.52.15(vm8) 172.16.2.15(vm1) | **控制节点** | Centos7 |
| compute1 | 192.168.52.16(vm8) 172.16.2.16(vm1) | **计算节点** | Centos7 |
| compute2 | 192.168.52.17(vm8) 172.16.2.17(vm1) | **计算节点** | Centos7 |

各角色描述及需求：

控制器：

①控制节点运行身份认证服务，镜像服务，管理部分计算和网络服务和不同的网络代理。同样包括像SQL数据库，消息队列这样的支撑服务。

②控制节点需要最少两块网卡。

计算：

①计算节点运行操作实例的 :hypervisor计算部分。默认情况下使用 KVM 作为hypervisor。计算节点同样运行网络服务代理，用来连接实例到虚拟网络，通过:security groups 为实例提供防火墙服务。

②这个服务可以部署超过1个计算节点。每个节点要求最少两个网络接口。

## 2.2、系统配置

1.添加hosts:

cat >> /etc/hosts << EOF

192.168.52.15 controller

192.168.52.16 compute1

192.168.52.17 compute2

EOF

2.关闭 selinux:

sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/sysconfig/selinux

**setenforce 0**

3.关闭防火墙：

**systemctl stop firewalld**

**systemctl disable firewalld**

4.更改主机名:

**hostnamectl set-hostname controller #控制节点**

**hostnamectl set-hostname compute1 #计算节点**

**hostnamectl set-hostname compute2 #计算节点**

5.免密登录:

**ssh-keygen -t rsa**

**ssh-copy-id controller**

**ssh-copy-id compute1**

**ssh-copy-id compute2**

6.同步系统时间

**yum install chrony -y**

**vim /etc/chrony.conf**

**server ntp.aliyun.com iburst**

**systemctl enable chronyd**

**systemctl restart chronyd**

**chronyc sources**

7.修改网卡地址

**①修改VM8网卡配置**

**vi /etc/sysconfig/network-scripts/ifcfg-ens33**

**BOOTPROTO=none**

**IPV4\_ROUTE\_METRIC=90 #调由优先级，NAT网卡优先**

**ONBOOT=yes**

**IPADDR=192.168.52.15**

**NETMASK=255.255.255.0**

**GATEWAY=192.168.52.2**

**②修改VM1网卡配置**

**vi /etc/sysconfig/network-scripts/ifcfg-ens33**

**BOOTPROTO=none**

**ONBOOT=yes**

**IPADDR=172.16.2.15**

**NETMASK=255.255.255.0**

**systemctl restart network #重启网卡**

**③配置DNS**

**vim /etc/resolv.conf**

**nameserver 114.114.114.114**

**nameserver 233.5.5.5**

8.安装和openstack对应版本相关的仓库

**yum -y install net-tools bash-completion vim gcc gcc-c++ make pcre pcre-devel expat-devel cmake bzip2**

**yum -y install centos-release-openstack-train python-openstackclient openstack-selinux openstack-utils**

**yum list |grep openstack**

# 3、安装配置数据库服务MYSQL(控制节点controller)

## 3.1、安装mysql

yum install mariadb mariadb-server MySQL-python python2-PyMySQL -y

vim /etc/my.cnf.d/openstack.cnf

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

log-error=/var/log/mariadb/mariadb.log

pid-file=/var/run/mariadb/mariadb.pid

bind-address = 0.0.0.0

default-storage-engine = innodb

innodb\_file\_per\_table = on

max\_connections = 4096

collation-server = utf8\_general\_ci

character-set-server = utf8

## 3.2、启动mysql

systemctl enable mariadb

systemctl start mariadb

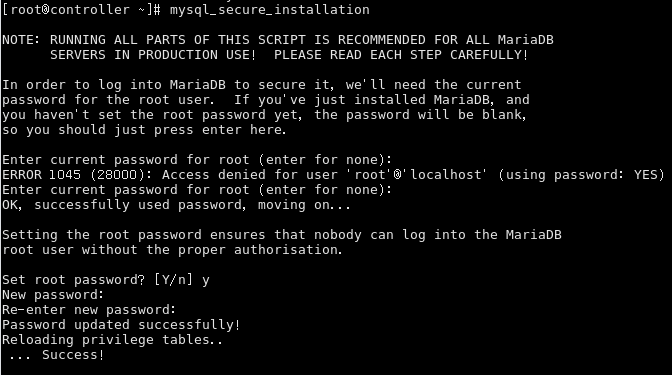
systemctl status mariadb

systemctl list-unit-files |grep mariadb

mysql\_secure\_installation #设置密码及初始化

Enter current password for root (enter for none): #回车

密码: abc123，一路 y 回车



systemctl restart mariadb

systemctl status mariadb

## 3.3、登陆mysql

mysql -uroot -p'abc123'

flush privileges;

show databases;

select user,host from mysql.user; #查看权限

# 安装配置消息队列服务rabbitmq（控制节点controller）

## 4.1、安装rabbitmq

yum install rabbitmq-server -y

systemctl enable rabbitmq-server

systemctl start rabbitmq-server

systemctl status rabbitmq-server

systemctl list-unit-files |grep rabbitmq-server

## 4.2、创建用户

rabbitmqctl add\_user openstack openstack

rabbitmqctl set\_permissions openstack ".\*" ".\*" ".\*"

rabbitmqctl set\_permissions -p "/" openstack ".\*" ".\*" ".\*"

rabbitmqctl set\_user\_tags openstack administrator

#查看支持的插件

rabbitmq-plugins list

# 启用web管理插件，需要重启服务使之生效

rabbitmq-plugins enable rabbitmq\_management

systemctl restart rabbitmq-server

systemctl status rabbitmq-server

rabbitmq-plugins list

rabbitmqctl list\_users #创建和赋角色完成后查看并确认



systemctl restart rabbitmq-server

ss -natp | grep 5672

访问RabbitMQ,访问地址：

http://192.168.52.15:15672,用户：openstack 密码: openstack

# 安装缓存数据库memcached（控制节点controller）

yum install memcached python-memcached -y

vi /etc/sysconfig/memcached

OPTIONS="-l 127.0.0.1,controller"

systemctl enable memcached

systemctl start memcached

systemctl status memcached

netstat -anltp |grep memcached

systemctl enable memcached

systemctl list-unit-files |grep memcached

# 安装etcd服务（控制节点controller）

yum install etcd -y

vi /etc/etcd/etcd.conf

#[Member]

ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"

ETCD\_LISTEN\_PEER\_URLS="http://192.168.52.15:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://192.168.52.15:2379"

ETCD\_NAME="controller"

#[Clustering]

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.52.15:2380"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.52.15:2379"

ETCD\_INITIAL\_CLUSTER="controller=http://192.168.52.15:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster-01"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

启动服务：

systemctl enable etcd

systemctl start etcd

systemctl status etcd

netstat -tnpl |grep etcd

systemctl list-unit-files |grep etcd

# 安装配置keystone（控制节点controller）

## 7.1、创建数据库实例和数据库用户

mysql -uroot -p'abc123'

create database keystone;

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost'IDENTIFIED BY 'keystone';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY 'keystone';

flush privileges;

show databases;

select user,host from mysql.user;

mysql -ukeystone -pkeystone #测试keystone能否进行登录

## 7.2、安装、配置keystone

yum install httpd mod\_wsgi -y

yum install openstack-keystone python-keystoneclient -y

yum install openstack-utils -y

openstack-config --set /etc/keystone/keystone.conf database connection mysql+pymysql://keystone:keystone@controller/keystone

openstack-config --set /etc/keystone/keystone.conf token provider fernet

# 查看生效的配置

egrep -v "^#|^$" /etc/keystone/keystone.conf

grep '^[a-z]' /etc/keystone/keystone.conf

#同步数据库

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

#同步完成进行连接测试

mysql -ukeystone -pkeystone -e "use keystone;show tables;"

mysql -h192.168.52.15 -ukeystone -pkeystone -e "use keystone;show tables;"|wc -l

#数据库初始化

keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

keystone-manage credential\_setup --keystone-user keystone --keystone-group keystone

**# 配置apache httpd 服务器**

sed -i "s/#ServerName www.example.com:80/ServerName controller/g" /etc/httpd/conf/httpd.conf

ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/

systemctl enable httpd

systemctl start httpd

systemctl status httpd

systemctl list-unit-files |grep httpd

netstat -anpt | grep http

#初始化引导keystone认证服务

keystone-manage bootstrap --bootstrap-password admin \

--bootstrap-admin-url http://controller:5000/v3/ \

--bootstrap-internal-url http://controller:5000/v3/ \

--bootstrap-public-url http://controller:5000/v3/ \

--bootstrap-region-id RegionOne

## 7.3、创建服务实体和API端点

### 7.3.1、临时配置管理员账户的相关变量进行管理

cat >> ~/.bashrc << EOF

export OS\_USERNAME=admin

export OS\_PASSWORD=admin

export OS\_PROJECT\_NAME=admin

export OS\_USER\_DOMAIN\_NAME=Default

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_AUTH\_URL=http://controller:5000/v3

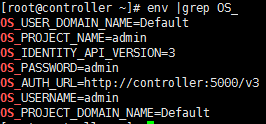
export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

EOF

source ~/.bashrc

env |grep OS\_  **#查看设置是否生效**



openstack token issue # 查看token

openstack user list # 查看user

### 7.3.2、创建域、项目、用户和角色

### （1）创建域

openstack domain create --description "Domain" example

### （2）创建服务项目

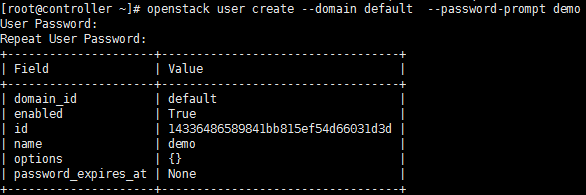
openstack project create --domain default --description "Service Project" service

### （3）创建平台demo项目

openstack project create --domain default --description "Demo Project" demo

### （4）创建demo用户

openstack user create --domain default --password-prompt demo



### （5）创建用户角色

openstack role create user

### （6）添加用户角色到demo项目和用户

openstack role add --project demo --user demo user

### （7）查看keystone实例相关信息

openstack endpoint list

openstack project list

openstack user list

openstack role list #查看openstack角色列表

## 7.4、验证令牌获得

### （1）重置OS\_TOKEN和OS\_URL 环境变量

unset OS\_AUTH\_URL OS\_PASSWORD

env |grep OS\_

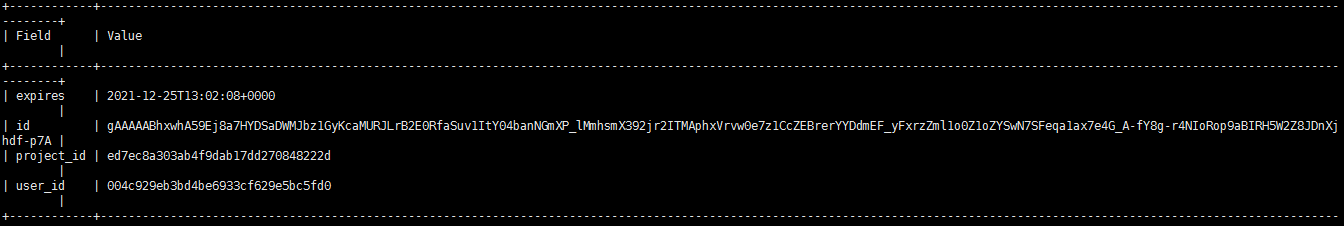
### （2）admin用户请求认证令牌

openstack --os-auth-url http://controller:5000/v3 \

--os-project-domain-name Default --os-user-domain-name Default \

--os-project-name admin --os-username admin token issue

#使用 admin用户，请求认证令牌,密码：admin



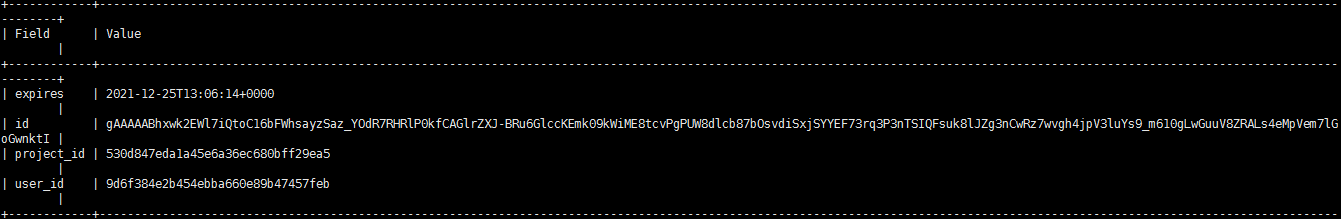
### （3）demo用户请求认证令牌

openstack --os-auth-url http://controller:5000/v3 \

--os-project-domain-name Default --os-user-domain-name Default \

--os-project-name demo --os-username demo token issue

#使用 demo 用户，请求认证令牌,密码：demo



## 7.5、创建 admin 和 demo项目和用户创建客户端环境变量脚本

### （1）创建环境变量脚本（admin和demo）

$ vi admin-openrc

# 文件内容

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_USER\_DOMAIN\_NAME=Default

export OS\_PROJECT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=admin

export OS\_AUTH\_URL=http://controller:5000/v3

export OS\_IDENTITY\_API\_VERSION=3

export OS\_IMAGE\_API\_VERSION=2

$ vi demo-openrc

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_USER\_DOMAIN\_NAME=Default

export OS\_PROJECT\_NAME=demo

export OS\_USERNAME=demo

export OS\_PASSWORD=demo

export OS\_AUTH\_URL=http://controller:5000/v3

export OS\_IDENTITY\_API\_VERSION=3

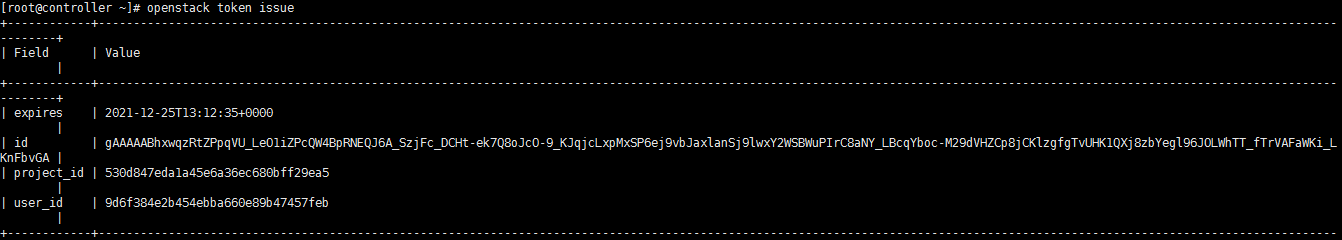
export OS\_IMAGE\_API\_VERSION=2

### （2）运行脚本

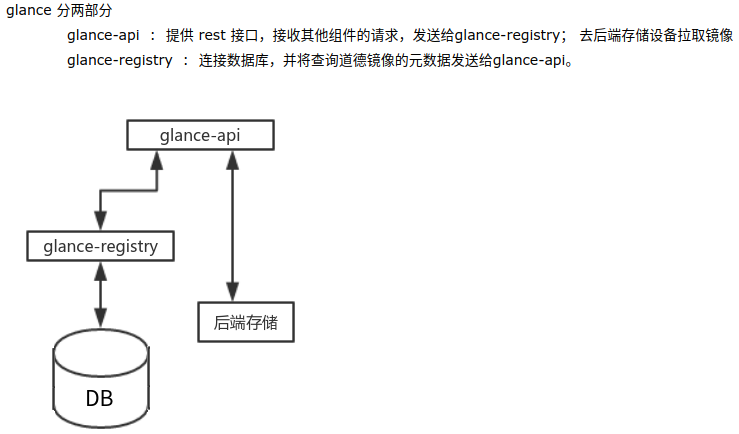
source admin-openrc

source demo-openrc

openstack token issue #请求认证令牌信息



# 8、添加镜像服务glance（控制节点controller）



## 8.1、安装和配置

### （1）创建数据库，并授权

mysql -uroot -p'abc123'

CREATE DATABASE glance;

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY 'glance';

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY 'glance';

flush privileges;

show databases;

select user,host from mysql.user;

mysql -uglance -pglance #测试连接

### （2）获取admin权限

source admin-openrc

### （3）创建glance用户

openstack user create --domain default --password=glance glance

openstack user list

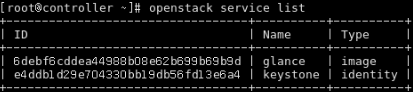
### （4）添加admin角色到glance用户和service项目上

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

### （5）创建glance服务实体

openstack service create --name glance --description "OpenStack Image" image

openstack service list



### （6）创建镜像服务的API端点

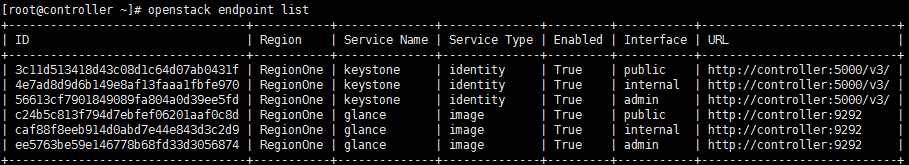
openstack endpoint create --region RegionOne image public http://controller:9292

openstack endpoint create --region RegionOne image internal http://controller:9292

openstack endpoint create --region RegionOne image admin http://controller:9292

查看API端点

openstack endpoint list



## 8.2、安装及配置组件

### （1）安装包

**python --version**

**yum install openstack-glance python-glance python-glanceclient -y**

### 配置glance-api.conf

**vim controller-node-glance-api.conf.sh**

**#!/bin/bash**

**#controller-node-glance-api.conf.sh**

**openstack-config --set /etc/glance/glance-api.conf database connection mysql+pymysql://glance:glance@controller/glance**

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

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

**openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken memcached\_servers controller:11211**

**openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken auth\_type password**

**openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken project\_domain\_name Default**

**openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken user\_domain\_name Default**

**openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken project\_name service**

**openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken username glance**

**openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken password glance**

**openstack-config --set /etc/glance/glance-api.conf paste\_deploy flavor keystone**

**openstack-config --set /etc/glance/glance-api.conf glance\_store stores file,http**

**openstack-config --set /etc/glance/glance-api.conf glance\_store default\_store file**

**openstack-config --set /etc/glance/glance-api.conf glance\_store filesystem\_store\_datadir /var/lib/glance/images/**

**echo "Result of Configuration"**

**grep '^[a-z]' /etc/glance/glance-api.conf**

**sh controller-node-glance-api.conf.sh**

### 配置glance-registry.conf

**vim controller-node-glance-registry.conf.sh**

**#!/bin/bash**

**#controller-node-glance-registry.conf.sh**

**openstack-config --set /etc/glance/glance-registry.conf database connection mysql+pymysql://glance:glance@controller/glance**

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

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

**openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken memcached\_servers controller:11211**

**openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_type password**

**openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken project\_domain\_name Default**

**openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken user\_domain\_name Default**

**openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken project\_name service**

**openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken username glance**

**openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken password glance**

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

**echo "Result of Configuration"**

**grep '^[a-z]' /etc/glance/glance-registry.conf**

**sh controller-node-glance-registry.conf.sh**

### 写入镜像服务数据库

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

**mysql -uglance -pglance -e "use glance;show tables;"**

### （5）完成安装

**systemctl start openstack-glance-api openstack-glance-registry**

**systemctl status openstack-glance-api openstack-glance-registry**

**systemctl enable openstack-glance-api openstack-glance-registry**

**systemctl list-unit-files |grep openstack-glance\***

**netstat -lnutp |grep 9191 #registry**

**netstat -lnutp |grep 9292 #api**

## 8.3、验证操作

### （1）获取admin权限

**source admin-openrc**

### （2）下载源镜像

wget http://download.cirros-cloud.net/0.3.5/cirros-0.3.5-x86\_64-disk.img

https://www.cnblogs.com/kevingrace/p/5821823.html #镜像制作教程

### （3）设置镜像为公共可见

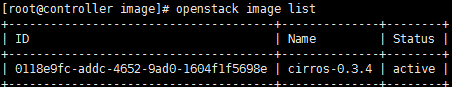
openstack image create "cirros-0.3.4" --file ./cirros-0.3.4-x86\_64-disk.img --disk-format qcow2 --container-format bare --public

### （4）查看及确定镜像的上传并验证属性

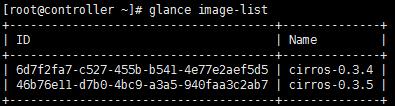
ls /var/lib/glance/images/

e40f4ae0-3bc8-4615-b7e5-5c58a694c915

openstack image list



glance image-list

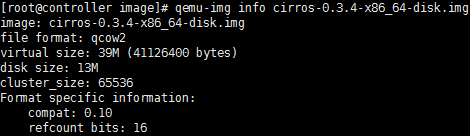


### （5）查看镜像格式

yum search qemu-img #查看qemu-img所需依赖包

yum install qemu-img.x86\_64 qemu-img-ev.x86\_64 -y

qemu-img info cirros-0.3.4-x86\_64-disk.img



# 9、placement服务组件（控制节点controller）

## 9.1、安装和配置

### （1）创建数据库，并授权

mysql -uroot -p'abc123'

CREATE DATABASE placement;

GRANT ALL PRIVILEGES ON placement.\* TO 'placement'@'localhost' IDENTIFIED BY 'placement';

GRANT ALL PRIVILEGES ON placement.\* TO 'placement'@'%' IDENTIFIED BY 'placement';

flush privileges;

show databases;

select user,host from mysql.user;

mysql -uplacement -pplacement #测试连接

### （2）创建服务凭据

source admin-openrc

# 域用户

openstack user create --domain default --password=placement placement

# 项目

openstack role add --project service --user placement admin

# 实体

openstack service create --name placement --description "Placement API" placement

### （3）创建placement项目的endpoint（API端口）

openstack endpoint create --region RegionOne placement public http://controller:8778

openstack endpoint create --region RegionOne placement internal http://controller:8778

openstack endpoint create --region RegionOne placement admin http://controller:8778

openstack endpoint list

## 9.2、placement相关软件安装与配置

yum install openstack-placement-api -y

vim placement.conf.sh

#!/bin/bash

#placement.conf.sh

openstack-config --set /etc/placement/placement.conf api auth\_strategy keystone

openstack-config --set /etc/placement/placement.conf keystone\_authtoken auth\_url http://controller:5000/v3

openstack-config --set /etc/placement/placement.conf keystone\_authtoken memcached\_servers controller:11211

openstack-config --set /etc/placement/placement.conf keystone\_authtoken auth\_type password

openstack-config --set /etc/placement/placement.conf keystone\_authtoken project\_domain\_name default

openstack-config --set /etc/placement/placement.conf keystone\_authtoken user\_domain\_name default

openstack-config --set /etc/placement/placement.conf keystone\_authtoken project\_name service

openstack-config --set /etc/placement/placement.conf keystone\_authtoken username placement

openstack-config --set /etc/placement/placement.conf keystone\_authtoken password placement

openstack-config --set /etc/placement/placement.conf placement\_database connection mysql+pymysql://placement:placement@controller/placement

echo "Result of Configuration"

grep '^[a-z]' /etc/placement/placement.conf

sh placement.conf.sh

vim /etc/httpd/conf.d/00-placement-api.conf #在末尾添加以下内容

<Directory /usr/bin>

<IfVersion >= 2.4>

Require all granted

</IfVersion>

<IfVersion < 2.4>

Order allow,deny

Allow from all

</IfVersion>

</Directory>

systemctl restart httpd

systemctl status httpd

## 9.3、同步placement数据库

（1）同步并初始化

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

#如有警告，再执行一遍

1. 同步完成进行连接测试

mysql -uplacement -pplacement -e "use placement;show tables;"

source admin-openrc

placement-status upgrade check

yum install -y python-osc-placement

openstack --os-placement-api-version 1.2 resource class list --sort-column name

openstack --os-placement-api-version 1.6 trait list --sort-column name

## 9.4、测试

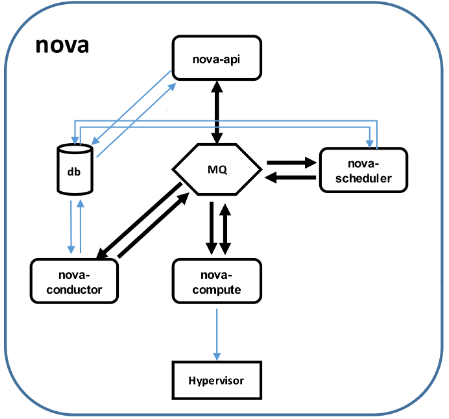
curl controller:8778

netstat -natp | grep 8778

检查placement状态：

placement-status upgrade check

# 10、配置计算服务nova（控制节点controller）



**nova-api（nova主服务）**

**nova-scheduler（nova调度服务）**

**nova-conductor（nova数据库服务，提供数据库访问）**

**nova-novncproxy（nova的vnc服务，提供实例的控制台）**

## 10.1、安装和配置控制节点（controller）

### （1）创建nova\_api和nova数据库

**mysql -uroot -p'abc123'**

CREATE DATABASE nova\_api;

CREATE DATABASE nova;

create database nova\_cell0;

### （2）授权数据库

GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'localhost' IDENTIFIED BY 'nova';

GRANT ALL PRIVILEGES ON nova\_api.\* TO 'nova'@'%' IDENTIFIED BY 'nova';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY 'nova';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY 'nova';

GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'localhost' IDENTIFIED BY 'nova';

GRANT ALL PRIVILEGES ON nova\_cell0.\* TO 'nova'@'%' IDENTIFIED BY 'nova';

flush privileges;

show databases;

select user,host from mysql.user;

### （3）测试登陆

mysql -u nova -pnova

### （4）创建nova用户

source admin-openrc

openstack user create --domain default --password=nova nova

openstack user list

### （5）给nova用户添加admin角色

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

### （6）创建nova服务实体

openstack service create --name nova --description "OpenStack Compute" compute

### （7）创建Compute服务API端点

openstack endpoint create --region RegionOne compute public http://controller:8774/v2.1

openstack endpoint create --region RegionOne compute internal http://controller:8774/v2.1

openstack endpoint create --region RegionOne compute admin http://controller:8774/v2.1

openstack endpoint list

## 10.2、安装nova及配置组件（controller）

### （1）安装软件包

yum install -y openstack-nova-api openstack-nova-conductor openstack-nova-novncproxy openstack-nova-scheduler

### （2）配置nova.conf

vim controller-node-nova.conf.sh

#!/bin/bash

#controller-node-nova.conf.sh

openstack-config --set /etc/nova/nova.conf DEFAULT enabled\_apis osapi\_compute,metadata

openstack-config --set /etc/nova/nova.conf DEFAULT my\_ip 172.16.2.15

openstack-config --set /etc/nova/nova.conf DEFAULT use\_neutron true

openstack-config --set /etc/nova/nova.conf DEFAULT firewall\_driver nova.virt.firewall.NoopFirewallDriver

openstack-config --set /etc/nova/nova.conf DEFAULT transport\_url rabbit://openstack:openstack@controller

openstack-config --set /etc/nova/nova.conf api\_database connection mysql+pymysql://nova:nova@controller/nova\_api

openstack-config --set /etc/nova/nova.conf database connection mysql+pymysql://nova:nova@controller/nova

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

openstack-config --set /etc/nova/nova.conf keystone\_authtoken www\_authenticate\_uri http://controller:5000/

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

openstack-config --set /etc/nova/nova.conf keystone\_authtoken memcached\_servers controller:11211

openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_type password

openstack-config --set /etc/nova/nova.conf keystone\_authtoken project\_domain\_name default

openstack-config --set /etc/nova/nova.conf keystone\_authtoken user\_domain\_name default

openstack-config --set /etc/nova/nova.conf keystone\_authtoken project\_name service

openstack-config --set /etc/nova/nova.conf keystone\_authtoken username nova

openstack-config --set /etc/nova/nova.conf keystone\_authtoken password nova

openstack-config --set /etc/nova/nova.conf vnc enabled true

openstack-config --set /etc/nova/nova.conf vnc server\_listen '$my\_ip'

openstack-config --set /etc/nova/nova.conf vnc server\_proxyclient\_address '$my\_ip'

openstack-config --set /etc/nova/nova.conf glance api\_servers http://controller:9292

openstack-config --set /etc/nova/nova.conf oslo\_concurrency lock\_path /var/lib/nova/tmp

openstack-config --set /etc/nova/nova.conf placement region\_name RegionOne

openstack-config --set /etc/nova/nova.conf placement project\_domain\_name Default

openstack-config --set /etc/nova/nova.conf placement project\_name service

openstack-config --set /etc/nova/nova.conf placement auth\_type password

openstack-config --set /etc/nova/nova.conf placement user\_domain\_name Default

openstack-config --set /etc/nova/nova.conf placement auth\_url http://controller:5000/v3

openstack-config --set /etc/nova/nova.conf placement username placement

openstack-config --set /etc/nova/nova.conf placement password placement

openstack-config --set /etc/nova/nova.conf scheduler discover\_hosts\_in\_cells\_interval 300

echo "Result of Configuration"

egrep -v "^#|^$" /etc/nova/nova.conf

sh controller-node-nova.conf.sh

### （3）同步compute数据库

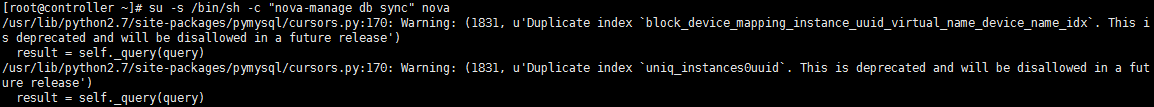
su -s /bin/sh -c "nova-manage api\_db sync" nova

mysql -unova -pnova -e "use nova\_api;show tables;"

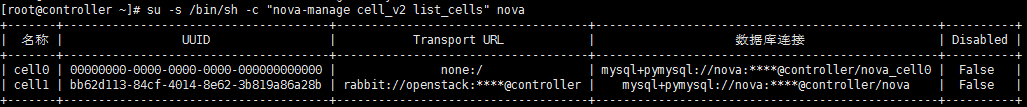
su -s /bin/sh -c "nova-manage cell\_v2 map\_cell0" nova

su -s /bin/sh -c "nova-manage cell\_v2 create\_cell --name=cell1 --verbose" nova

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



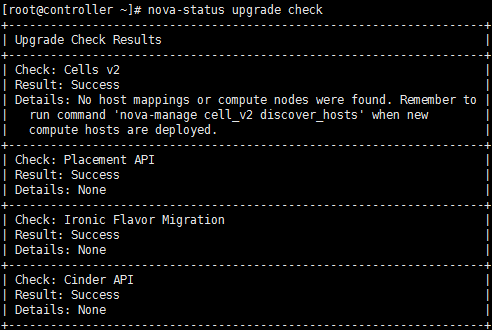
su -s /bin/sh -c "nova-manage cell\_v2 list\_cells" nova #验证nova cell0和cell1是否正确注册



mysql -unova -pnova -e "use nova\_api;show tables;"

mysql -uplacement -pplacement -e "use placement;show tables;"

nova-status upgrade check #检查部署是否正常



## 10.3、完成安装

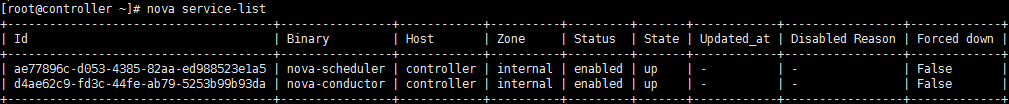
systemctl start openstack-nova-api openstack-nova-scheduler openstack-nova-conductor openstack-nova-novncproxy

systemctl status openstack-nova-api openstack-nova-scheduler openstack-nova-conductor openstack-nova-novncproxy

systemctl enable openstack-nova-api openstack-nova-scheduler openstack-nova-conductor openstack-nova-novncproxy

systemctl list-unit-files |grep openstack-nova\* |grep enabled

nova service-list



netstat -tnlup|egrep '8774|8775'

# 11、计算节点计算服务nova安装（compute）

### （1）安装包

yum install openstack-nova-compute -y

yum install python-openstackclient openstack-selinux -y

#用于快速配置

yum install openstack-utils -y

### （2）配置nova.conf

#compute2和compute1相同，只是ip地址不同

vim compute-node-nova.conf.sh

#!/bin/bash

#compute-node-nova.conf.sh

openstack-config --set /etc/nova/nova.conf DEFAULT my\_ip 172.16.2.16

openstack-config --set /etc/nova/nova.conf DEFAULT use\_neutron True

openstack-config --set /etc/nova/nova.conf DEFAULT firewall\_driver nova.virt.firewall.NoopFirewallDriver

openstack-config --set /etc/nova/nova.conf DEFAULT enabled\_apis osapi\_compute,metadata

openstack-config --set /etc/nova/nova.conf DEFAULT transport\_url rabbit://openstack:openstack@controller

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

openstack-config --set /etc/nova/nova.conf keystone\_authtoken www\_authenticate\_uri http://controller:5000/

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

openstack-config --set /etc/nova/nova.conf keystone\_authtoken memcached\_servers controller:11211

openstack-config --set /etc/nova/nova.conf keystone\_authtoken auth\_type password

openstack-config --set /etc/nova/nova.conf keystone\_authtoken project\_domain\_name default

openstack-config --set /etc/nova/nova.conf keystone\_authtoken user\_domain\_name default

openstack-config --set /etc/nova/nova.conf keystone\_authtoken project\_name service

openstack-config --set /etc/nova/nova.conf keystone\_authtoken username nova

openstack-config --set /etc/nova/nova.conf keystone\_authtoken password nova

openstack-config --set /etc/nova/nova.conf vnc enabled True

openstack-config --set /etc/nova/nova.conf vnc server\_listen 0.0.0.0

openstack-config --set /etc/nova/nova.conf vnc server\_proxyclient\_address '$my\_ip'

openstack-config --set /etc/nova/nova.conf vnc novncproxy\_base\_url http://controller:6080/vnc\_auto.html

openstack-config --set /etc/nova/nova.conf glance api\_servers http://controller:9292

openstack-config --set /etc/nova/nova.conf oslo\_concurrency lock\_path /var/lib/nova/tmp

openstack-config --set /etc/nova/nova.conf placement region\_name RegionOne

openstack-config --set /etc/nova/nova.conf placement project\_domain\_name Default

openstack-config --set /etc/nova/nova.conf placement project\_name service

openstack-config --set /etc/nova/nova.conf placement auth\_type password

openstack-config --set /etc/nova/nova.conf placement user\_domain\_name Default

openstack-config --set /etc/nova/nova.conf placement auth\_url http://controller:5000/v3

openstack-config --set /etc/nova/nova.conf placement username placement

openstack-config --set /etc/nova/nova.conf placement password placement

echo "Result of Configuration"

egrep -v "^#|^$" /etc/nova/nova.conf

sh compute-node-nova.conf.sh

openstack-config --set /etc/nova/nova.conf vnc novncproxy\_base\_url http://controller:6080/vnc\_auto.html

### （3）是否支持虚拟机的硬件加速

#如果这个命令返回``0``，则计算节点不支持硬件加速，必须配置 libvirt使用QEMU而不是使用KVM

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

4

### （4）不支持执行操作

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

egrep -v "^#|^$" /etc/nova/nova.conf|grep 'virt\_type'

### （5）完成安装、设置自启

systemctl start libvirtd openstack-nova-compute

systemctl status libvirtd openstack-nova-compute

systemctl enable libvirtd openstack-nova-compute

systemctl list-unit-files |grep libvirtd

systemctl list-unit-files |grep openstack-nova-compute

注意：

（1）如果nova-compute服务无法启动，请检查 /var/log/nova/nova-compute.log。

该错误消息可能表明控制器节点上的防火墙阻止访问端口5672。将控制节点防火墙配置为打开控制器节点上的端口5672，并重新启动计算节点上的服务。

1. 控制节点

systemctl stop firewalld

systemctl disable firewalld

systemctl status firewalld

systemctl restart rabbit\_server

systemctl restart rabbitmq-server

systemctl status rabbitmq-server

firewall-cmd --zone=public --add-port=5672/tcp --permanent

systemctl restart firewalld

firewall-cmd --zone=public --query-port=5672/tcp

# 12、controller控制端验证操作

### （1）获取admin权限

source admin-openrc

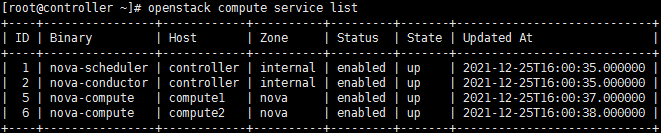
### （2）添加计算节点compute到cell数据库（controller ）

openstack compute service list --service nova-compute

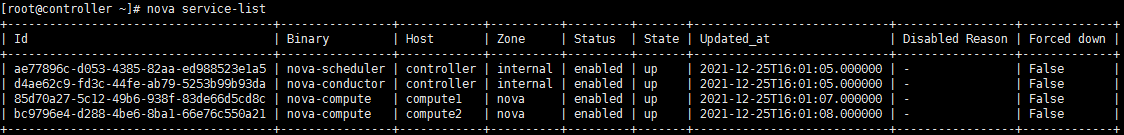
su -s /bin/sh -c "nova-manage cell\_v2 discover\_hosts --verbose" nova

### （3）查看compute结果（controller ）

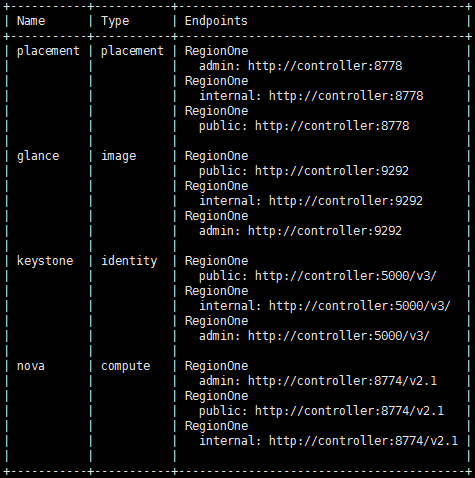
openstack compute service list #在控制节点上执行新节点的检查



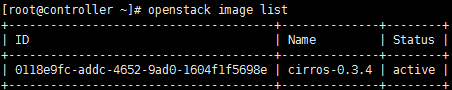
nova service-list #列出服务组件



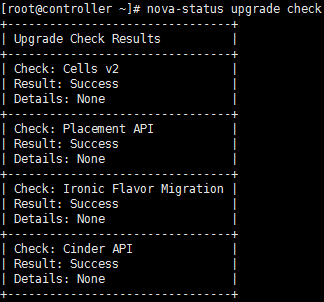
openstack catalog list #列出身份服务中的API端点以验证与身份服务的连接



openstack image list #列出镜像

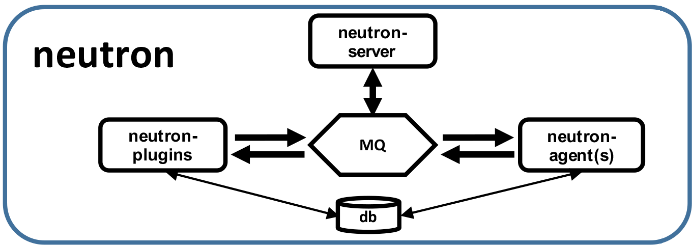


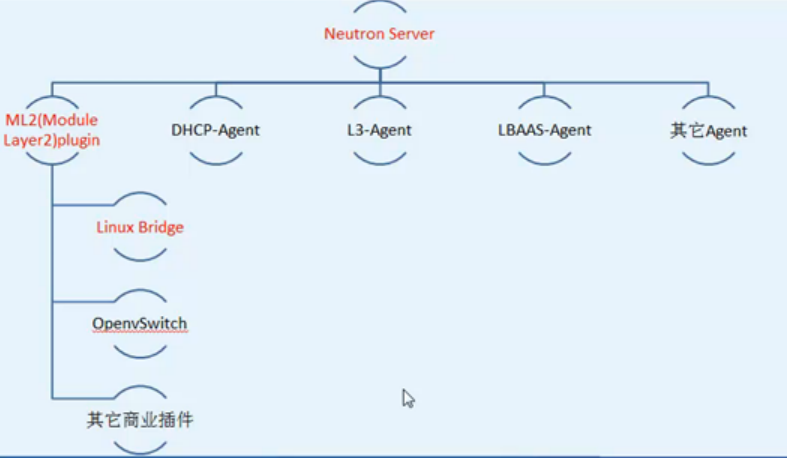
nova-status upgrade check



# 13、neutron组件部署(controller)

neutron组件:





## 13.1、安装配置控制节点(controller)

### （1）数据库操作（controller）

**mysql -uroot -p'abc123'**

CREATE DATABASE neutron;

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' IDENTIFIED BY 'neutron';

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' IDENTIFIED BY 'neutron';

flush privileges;

exit

### （2）创建neutron用户

source admin-openrc

openstack user create --domain default --password=neutron neutron

openstack user list

### （3）添加admin 角色到neutron用户

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

### （4）创建neutron服务实体

openstack service create --name neutron --description "OpenStack Networking" network

openstack service list

### （5）创建网络服务端点

openstack endpoint create --region RegionOne network public http://controller:9696

openstack endpoint create --region RegionOne network internal http://controller:9696

openstack endpoint create --region RegionOne network admin http://controller:9696

openstack endpoint list

### （6）配置网络部分（controller节点）

#### <1> 安装组件

yum -y install openstack-neutron openstack-neutron-ml2 openstack-neutron-linuxbridge ebtables conntrack-tools

#### <2> 配置服务组件

vim controller-node-neutron.conf.sh

#!bin/bash

#controller-node-neutron.conf.sh

openstack-config --set /etc/neutron/neutron.conf database connection mysql+pymysql://neutron:neutron@controller/neutron

openstack-config --set /etc/neutron/neutron.conf DEFAULT core\_plugin ml2

openstack-config --set /etc/neutron/neutron.conf DEFAULT service\_plugins router

openstack-config --set /etc/neutron/neutron.conf DEFAULT allow\_overlapping\_ips true

openstack-config --set /etc/neutron/neutron.conf DEFAULT transport\_url rabbit://openstack:openstack@controller

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

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 keystone\_authtoken www\_authenticate\_uri http://controller:5000

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

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken memcached\_servers controller:11211

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken auth\_type password

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken project\_domain\_name default

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken user\_domain\_name default

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken project\_name service

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

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

openstack-config --set /etc/neutron/neutron.conf nova auth\_url http://controller:5000

openstack-config --set /etc/neutron/neutron.conf nova auth\_type password

openstack-config --set /etc/neutron/neutron.conf nova project\_domain\_name default

openstack-config --set /etc/neutron/neutron.conf nova user\_domain\_name default

openstack-config --set /etc/neutron/neutron.conf nova region\_name RegionOne

openstack-config --set /etc/neutron/neutron.conf nova project\_name service

openstack-config --set /etc/neutron/neutron.conf nova username nova

openstack-config --set /etc/neutron/neutron.conf nova password nova

openstack-config --set /etc/neutron/neutron.conf oslo\_concurrency lock\_path /var/lib/neutron/tmp

echo "Result of Configuration"

egrep -v "^#|^$" /etc/neutron/neutron.conf

sh controller-node-neutron.conf.sh

#### <3> 配置Modular Layer 2 (ML2) 插件

vim controller-node-ml2\_conf.ini.sh

#!bin/bash

#controller-node-ml2\_conf.ini.sh

openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 type\_drivers flat,vlan,vxlan

openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 tenant\_network\_types vxlan

openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 mechanism\_drivers linuxbridge,l2population

openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 extension\_drivers port\_security

openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_flat flat\_networks provider

openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2\_type\_vxlan vni\_ranges 1:1000

openstack-config --set /etc/neutron/plugins/ml2/ml2\_conf.ini securitygroup enable\_ipset true

echo "Result of Configuration"

egrep -v "^#|^$" /etc/neutron/plugins/ml2/ml2\_conf.ini

sh controller-node-ml2\_conf.ini.sh

#### <4> 配置Linuxbridge代理

vim controller-node-linuxbridge\_agent.ini.sh

#!bin/bash

#controller-node-linuxbridge\_agent.ini.sh

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini linux\_bridge physical\_interface\_mappings provider:ens37 ###ens37网卡名称

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan enable\_vxlan true

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan local\_ip 172.16.2.15 ##控制节点IP地址

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan l2\_population true

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup enable\_security\_group true

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup firewall\_driver neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver

echo "Result of Configuration"

egrep -v '(^$|^#)' /etc/neutron/plugins/ml2/linuxbridge\_agent.ini

sh controller-node-linuxbridge\_agent.ini.sh

modprobe br\_netfilter

echo net.bridge.bridge-nf-call-iptables = 1 >> /etc/sysctl.conf

echo net.bridge.bridge-nf-call-ip6tables = 1 >> /etc/sysctl.conf

sysctl -p

ls /proc/sys/net/bridge

#### <5> 配置DHCP代理

vim controller-node-dhcp\_agent.ini.sh

#!/bin/bash

#controller-node-dhcp\_agent.ini.sh

openstack-config --set /etc/neutron/dhcp\_agent.ini DEFAULT interface\_driver linuxbridge

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 enable\_isolated\_metadata True

echo "Result of Configuration"

egrep -v '(^$|^#)' /etc/neutron/dhcp\_agent.ini

sh controller-node-dhcp\_agent.ini.sh

#### <6>配置l3\_agent.ini

openstack-config --set /etc/neutron/l3\_agent.ini DEFAULT interface\_driver linuxbridge

#### <7>配置元数据代理

cp -a /etc/neutron/metadata\_agent.ini{,.bak}

grep -Ev '^$|#' /etc/neutron/metadata\_agent.ini.bak > /etc/neutron/metadata\_agent.ini

openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT nova\_metadata\_host controller

openstack-config --set /etc/neutron/metadata\_agent.ini DEFAULT metadata\_proxy\_shared\_secret neutron

egrep -v '(^$|^#)' /etc/neutron/metadata\_agent.ini

### （7）为计算节点配置网络服务

vim controller-node-neutron-nova.conf.sh

#!bin/bash

#controller-node-neutron-nova.conf.sh

openstack-config --set /etc/nova/nova.conf neutron url http://controller:9696

openstack-config --set /etc/nova/nova.conf neutron auth\_url http://controller:5000

openstack-config --set /etc/nova/nova.conf neutron auth\_type password

openstack-config --set /etc/nova/nova.conf neutron project\_domain\_name default

openstack-config --set /etc/nova/nova.conf neutron user\_domain\_name default

openstack-config --set /etc/nova/nova.conf neutron region\_name RegionOne

openstack-config --set /etc/nova/nova.conf neutron project\_name service

openstack-config --set /etc/nova/nova.conf neutron username neutron

openstack-config --set /etc/nova/nova.conf neutron password neutron

openstack-config --set /etc/nova/nova.conf neutron service\_metadata\_proxy true

openstack-config --set /etc/nova/nova.conf neutron metadata\_proxy\_shared\_secret neutron

echo "Result of Configuration"

egrep -v '(^$|^#)' /etc/nova/nova.conf

sh controller-node-neutron-nova.conf.sh

### （8）完成安装

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

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 #同步数据库

systemctl restart openstack-nova-api #重启计算API服务

systemctl enable neutron-server neutron-linuxbridge-agent neutron-dhcp-agent neutron-metadata-agent.service neutron-l3-agent

systemctl start neutron-server neutron-linuxbridge-agent neutron-dhcp-agent neutron-metadata-agent neutron-l3-agent

systemctl status neutron-server neutron-linuxbridge-agent neutron-dhcp-agent neutron-metadata-agent neutron-l3-agent

netstat -anutp |grep 9696

## 13.2、安装和配置计算节点(compute)

### （1）安装组件

yum -y install openstack-neutron-linuxbridge ebtables ipset conntrack-tools

### （2）配置通用组件

vim compute-node-neutron.conf.sh

#!bin/bash

#compute-node-neutron.conf.sh

openstack-config --set /etc/neutron/neutron.conf DEFAULT transport\_url rabbit://openstack:openstack@controller

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

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken www\_authenticate\_uri http://controller:5000

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

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken memcached\_servers controller:11211

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken auth\_type password

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken project\_domain\_name default

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken user\_domain\_name default

openstack-config --set /etc/neutron/neutron.conf keystone\_authtoken project\_name service

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

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

openstack-config --set /etc/neutron/neutron.conf oslo\_concurrency lock\_path /var/lib/neutron/tmp

echo "Result of Configuration"

egrep -v '(^$|^#)' /etc/neutron/neutron.conf

sh compute-node-neutron.conf.sh

### （3）配置网络选项（公共网络）

vim compute-node-linuxbridge\_agent.ini.sh

#!bin/bash

#compute-node-linuxbridge\_agent.ini.sh

#map the provider virtual network to the provider physical network interface，the name of the underlying provider physical network interface

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini linux\_bridge physical\_interface\_mappings provider:ens37

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan enable\_vxlan true

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan local\_ip 172.16.2.16

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini vxlan l2\_population true

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup enable\_security\_group true

openstack-config --set /etc/neutron/plugins/ml2/linuxbridge\_agent.ini securitygroup firewall\_driver neutron.agent.linux.iptables\_firewall.IptablesFirewallDriver

echo "Result of Configuration"

egrep -v '(^$|^#)' /etc/neutron/plugins/ml2/linuxbridge\_agent.ini

sh compute-node-linuxbridge\_agent.ini.sh

modprobe br\_netfilter

echo net.bridge.bridge-nf-call-iptables = 1 >> /etc/sysctl.conf

echo net.bridge.bridge-nf-call-ip6tables = 1 >> /etc/sysctl.conf

sysctl -p

ls /proc/sys/net/bridge

### （4）为计算节点配置网络服务

vim compute-node-neutron-nova.conf.sh

#!bin/bash

#compute-node-neutron-nova.conf.sh

openstack-config --set /etc/nova/nova.conf neutron url http://controller:9696

openstack-config --set /etc/nova/nova.conf neutron auth\_url http://controller:5000

openstack-config --set /etc/nova/nova.conf neutron auth\_type password

openstack-config --set /etc/nova/nova.conf neutron project\_domain\_name default

openstack-config --set /etc/nova/nova.conf neutron user\_domain\_name default

openstack-config --set /etc/nova/nova.conf neutron region\_name RegionOne

openstack-config --set /etc/nova/nova.conf neutron project\_name service

openstack-config --set /etc/nova/nova.conf neutron username neutron

openstack-config --set /etc/nova/nova.conf neutron password neutron

echo "Result of Configuration"

egrep -v '(^$|^#)' /etc/nova/nova.conf

sh compute-node-neutron-nova.conf.sh

### （5）完成安装

systemctl restart openstack-nova-compute

systemctl status openstack-nova-compute

**#启动Linux桥接代理并配置它开机自启动**

systemctl restart neutron-linuxbridge-agent

systemctl status neutron-linuxbridge-agent

systemctl enable neutron-linuxbridge-agent

systemctl list-unit-files |grep neutron\* |grep enabled

## 13.3、验证操作(controller)

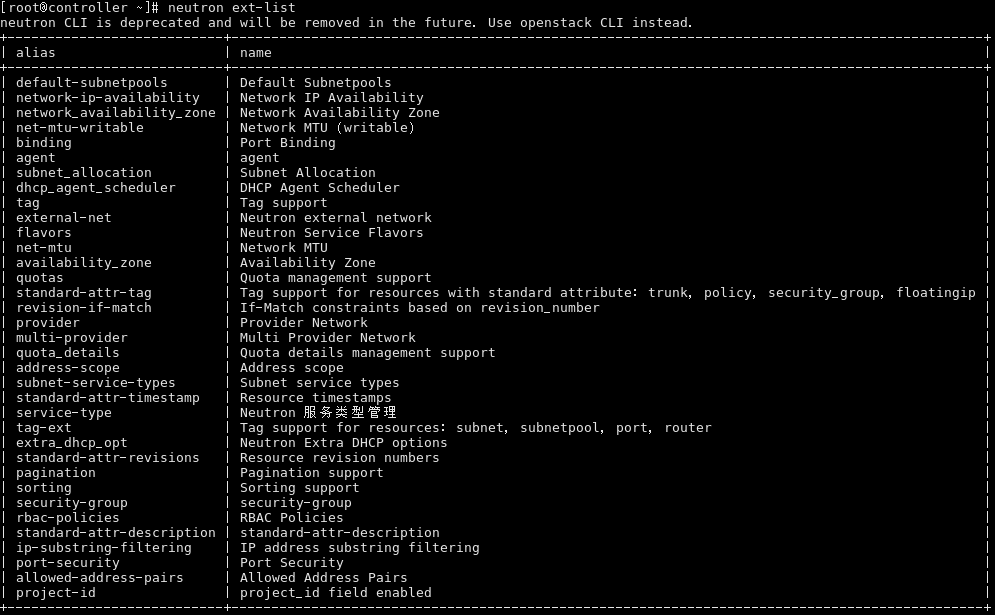
### （1）获取admin权限

source admin-openrc

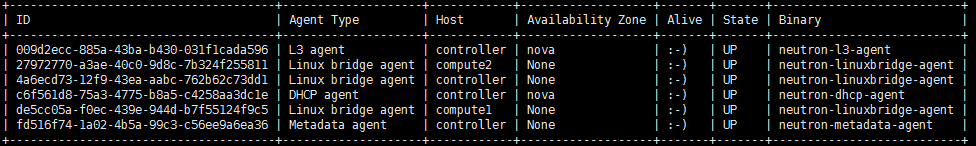
### （2）列出加载的扩展

openstack extension list --network

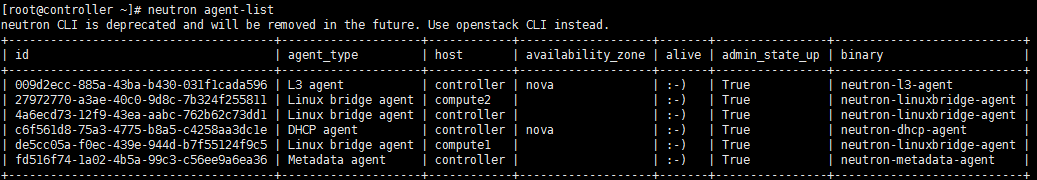
neutron ext-list



openstack network agent list



neutron agent-list



# 14、在控制节点安装dashboard(controller)

## 14.1、安装配置

### （1）安装包

yum install openstack-dashboard -y

### （2）配置local\_settings

vi /etc/openstack-dashboard/local\_settings

ALLOWED\_HOSTS = ['\*', ]

SESSION\_ENGINE = 'django.contrib.sessions.backends.file'

OPENSTACK\_API\_VERSIONS = {

"identity": 3,

"image": 2,

"volume": 2,

}

OPENSTACK\_HOST = "controller"

OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"

OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True

OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = "default"

CACHES = {

'default': {

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

'LOCATION': 'controller:11211',

}

}

OPENSTACK\_NEUTRON\_NETWORK = {

'enable\_auto\_allocated\_network': False,

'enable\_distributed\_router': False,

'enable\_fip\_topology\_check': False,

'enable\_ha\_router': False,

'enable\_lb': False,

'enable\_firewall': False,

'enable\_vpn': False,

'enable\_ipv6': True,

'enable\_quotas': True,

'enable\_rbac\_policy': True,

'enable\_router': True,

'default\_dns\_nameservers': [],

'supported\_provider\_types': ['\*'],

'segmentation\_id\_range': {},

'extra\_provider\_types': {},

'supported\_vnic\_types': ['\*'],

'physical\_networks': [],

}

TIME\_ZONE = "Asia/Shanghai"

### 完成安装

vim /etc/httpd/conf.d/openstack-dashboard.conf

WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias /dashboard /usr/share/openstack-dashboard/openstack\_dashboard/wsgi/django.wsgi

Alias /dashboard/static /usr/share/openstack-dashboard/static

编辑以下文件，找到WEBROOT = ‘/’ 修改为WEBROOT = ‘/dashboard’ (官方未提及坑点之一)

vim /usr/share/openstack-dashboard/openstack\_dashboard/defaults.py

vim /usr/share/openstack-dashboard/openstack\_dashboard/test/settings.py

vim /usr/share/openstack-dashboard/static/dashboard/js/1453ede06e9f.js

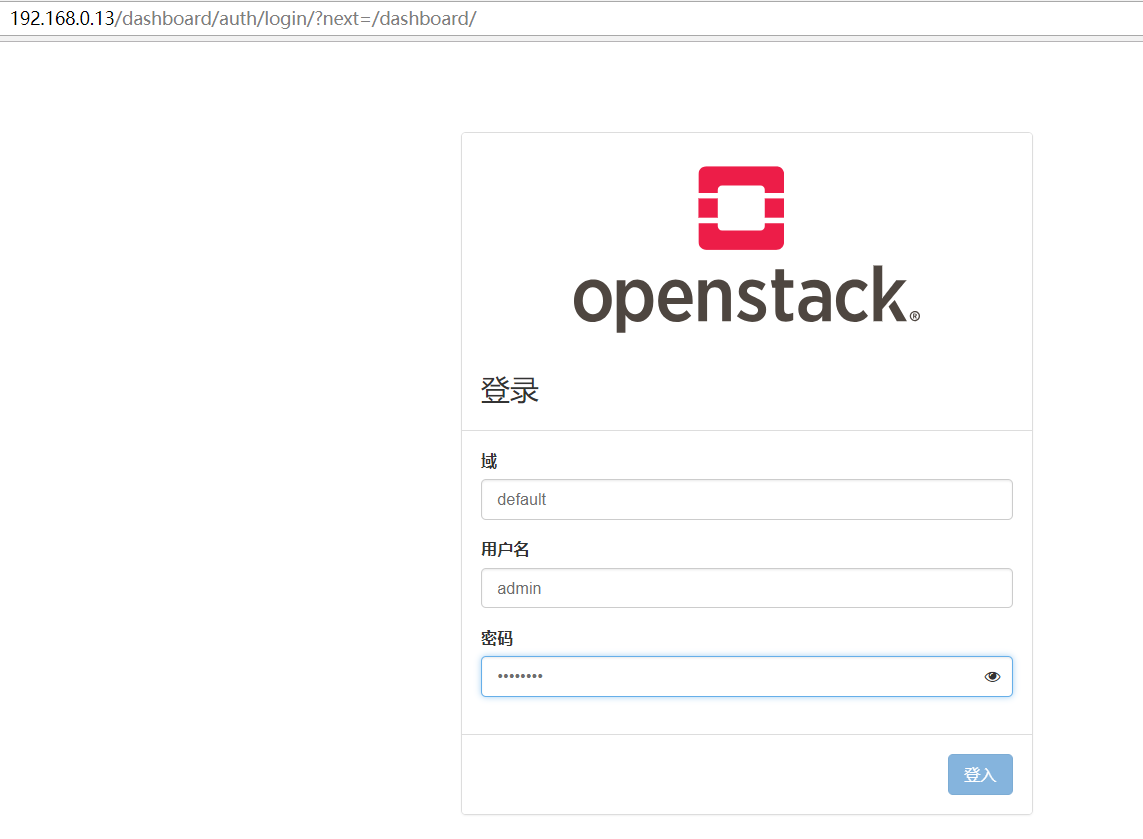
systemctl restart httpd memcached

systemctl status httpd memcached

## 14.2、访问web页面

http://192.168.52.15/dashboard

域：default 用户名：admin admin 用户名：demo demo





**报错1:**

如下图处理



$ vi /etc/openstack-dashboard/local\_settings

#SESSION\_ENGINE = 'django.contrib.sessions.backends.cache' #注释

SESSION\_ENGINE = 'django.contrib.sessions.backends.file' #添加

$ systemctl restart httpd memcached

$ systemctl status httpd memcached

# 15、创建 OpenStack云主机 (controller)

## 15.1、创建虚拟网络

source admin-openrc

openstack network create --share --external --provider-physical-network provider --provider-network-type flat public-net

openstack network list 或 neutron net-list

openstack subnet create --network public-net \

--allocation-pool start=172.16.2.220,end=172.16.2.230 \

--dns-nameserver 223.5.5.5 --gateway 172.16.2.1 \

--subnet-range 172.16.2.0/24 public-subnet1

#查看子网

openstack subnet list

## 15.2、创建网络选项二：Self-service networks

openstack network create selfservicenet

openstack subnet create --network selfservicenet \

--dns-nameserver 223.5.5.5 --gateway 172.16.1.1 \

--subnet-range 172.16.1.0/24 selfservicenet-subnet1

openstack subnet list

## 15.3、创建一个nano规格的实例

source admin-openrc

openstack flavor create --id 0 --vcpus 1 --ram 256 --disk 0 1U256M0G

openstack flavor create --id 1 --vcpus 1 --ram 1024 --disk 0 1U1GM0G

openstack flavor create --id 2 --vcpus 1 --ram 64 --disk 1 m1.nano

openstack flavor list