

**云计算技术与应用课程作业**

**OpenStack(Kilo)的安装和部署**

**学 院：计算机科学与技术学院**

**专 业： 软件工程**

**班 级： 专 硕**

**学 号： 2019021964**

**学生姓名： 孙 武**

**指导教师： 李秦伟**

2020 年 6 月 19 日

# 目 录

[1 安装前准备 1](#_Toc43762937)

[1.1 环境准备 1](#_Toc43762938)

[1.2 各节点架构 1](#_Toc43762939)

[1.3 组件部署 2](#_Toc43762940)

[1.4 配置主机映射 3](#_Toc43762941)

[1.5 系统基础组件配置 3](#_Toc43762942)

[1.5.1 安装基础组件及源 3](#_Toc43762943)

[1.5.2 关闭防火墙 4](#_Toc43762944)

[1.6 相关密码 4](#_Toc43762945)

[2 集群的安装 5](#_Toc43762946)

[2.1 controller节点安装 5](#_Toc43762947)

[2.1.1 配置时钟同步 5](#_Toc43762948)

[2.1.2 安装数据库 6](#_Toc43762949)

[2.1.3 安装rabbitmq 9](#_Toc43762950)

[2.1.4 验证基础组件 9](#_Toc43762951)

[2.1.5 安装keystone 10](#_Toc43762952)

[2.1.6 安装镜像服务glance 15](#_Toc43762953)

[2.2 Nova组件安装 18](#_Toc43762954)

[2.2.1 在controller节点安装 18](#_Toc43762955)

[2.2.2 compute-node-01安装 20](#_Toc43762956)

[2.3 Neutron组件安装 22](#_Toc43762957)

[2.3.1 在controller节点安装 22](#_Toc43762958)

[2.3.2 在network-node-01节点安装 26](#_Toc43762959)

[2.3.3 在compute-node-01节点安装 31](#_Toc43762960)

[2.3.4 controller节点安装Dashboard 35](#_Toc43762961)

[2.4 Cinder组件安装 36](#_Toc43762962)

[2.4.1 Controller节点安装 36](#_Toc43762963)

[2.4.2 在block-node-01节点上安装 38](#_Toc43762964)

[2.5 Swift组件安装 42](#_Toc43762965)

[2.5.1 在controller节点安装 42](#_Toc43762966)

[2.5.2 在object-node-01节点安装 43](#_Toc43762967)

[2.5.3 在object-node-02节点安装 47](#_Toc43762968)

[2.5.4 controller节点创建环（Ring） 47](#_Toc43762969)

# 安装前准备

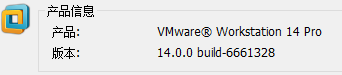
## 环境准备

本次实验在 Windows10 中进行，所有节点在 VMWare Pro 14 中进行创建，所有节点操作系统均为 centos 7。

实验环境：



Vmware 版本：



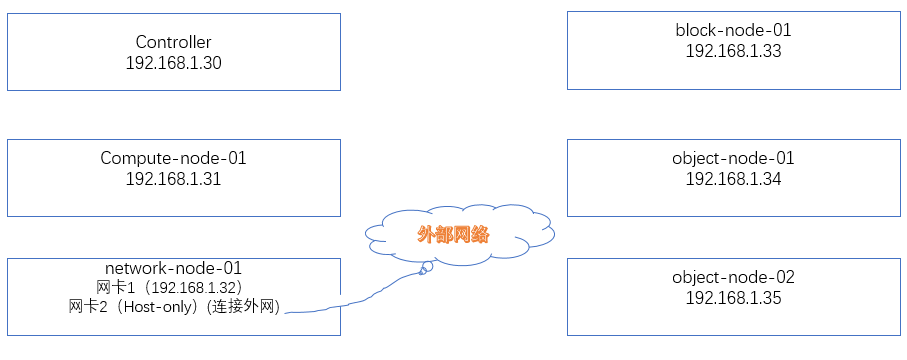
Centos版本：

[root@controller ~]# uname -a

Linux controller 3.10.0-1127.10.1.el7.x86\_64 #1 SMP Wed Jun 3 14:28:03 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux

## 各节点架构

网络架构如下图所示。

说明：

Controller-node：控制节点

Conpute-node：计算节点

Network-node：网络节点

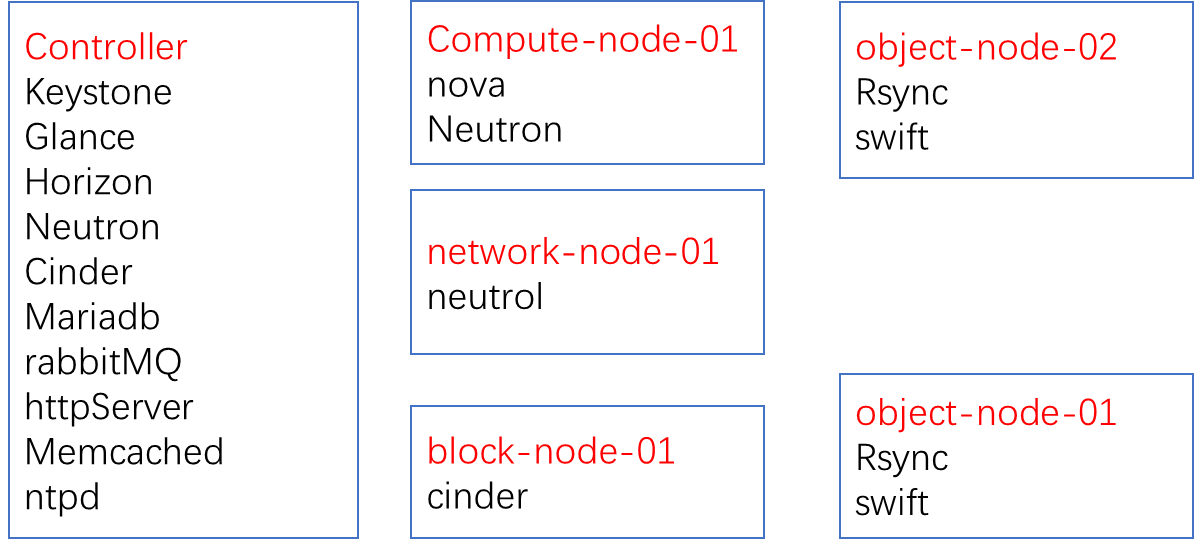
Block-node：块存储节点

Object-node：对象存储节点

节点后面的数节点的序号，如 object-node-01 表示对象存储的第一台机器。

## 组件部署

个节点需要安装的组件如下图所示。



组件说明：

Horizon：openstack web 管理界面

Nova：拟机的管理、cpu 内存等资源的分配

Neutron：网络管理、网络通讯

Glance： 虚拟机的镜像管理（镜像信息管理、镜像上传、删除、、编辑、查找等）

Keystone：服务组件提供身份认证服务

Cinder： 提供磁盘块存储

Swift：对象存储

## 配置主机映射

在各节点(每个节点) hosts 文件中添加以下域名映射。

vim /etc/hosts

192.168.1.30 controller

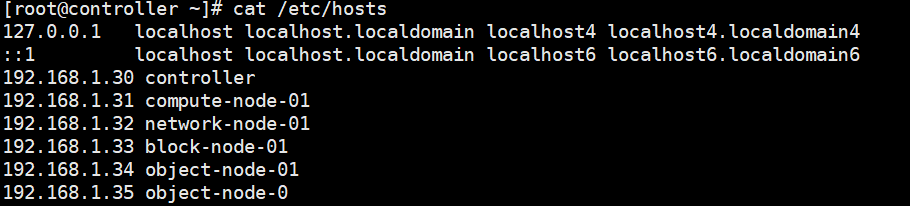
192.168.1.31 compute-node-01

192.168.1.32 network-node-01

192.168.1.33 block-node-01

192.168.1.34 object-node-01

192.168.1.35 object-node-0



## 系统基础组件配置

各节点均需安装这些基础组件。

### 安装基础组件及源

# 安装时间同步组件

yum install ntp –y

# yum repo 优先级插件

yum install yum-plugin-priorities –y

# 安装 OpenStack镜像仓库

yum install <http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm> -y yum install <http://rdo.fedorapeople.org/openstack-kilo/rdo-release-kilo.rpm> -y

# 更新

 yum upgrade

# 安装 OpenStack selinux

 yum install openstack-selinux -y

### 关闭防火墙

# 关闭防火墙

systemctl stop firewalld.service

# 禁止开机自动启动

systemctl disable firewalld.service

## 相关密码

本实现环境所涉及到组件密码，安装时根据自行需求，可以改变

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 用户名 | 密码 | 用途 |
| mysql | root | root. | mysql系统管理员 |
| rabbitMQ | openstack | sunwu123. | Openstack组件间MQ消息通讯 |
| mysql | keystone | keystone | keystone组件访问mysql数据库 |
| Keystone | admin | sunwu123. | Keystone管理员 |
| Keystone | demo | sunwu123. | 访问租户项目demo |
| mysql | glance | glance | Glance组件访问mysql数据库 |
| mysql | nova | nova | Nova组件访问mysql数据库 |
| mysql | neutron | neutron | neutron组件访问mysql数据库 |
| mysql | cinder | cinder | Cinder组件访问mysql数据库 |

# 集群的安装

## controller节点安装

本节介绍 controller 节点的安装。

### 配置时钟同步

#### 备份配置文件

cp /etc/ntp.conf /etc/ntp.conf.bak

#### 添加参数

vim /etc/ntp.conf

server controller iburst

restrict -4 default kod notrap nomodify

restrict -6 default kod notrap nomodify

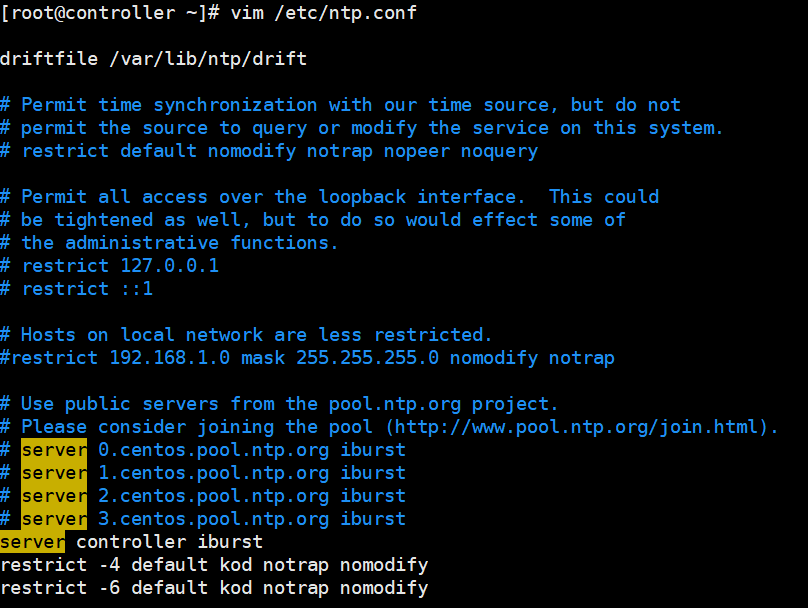
#### 注释以下参数

#restrict default nomodify notrap nopeer noquery

#restrict 127.0.0.1

#restrict ::1

上三步配置如下图



#### 开启时钟同步服务

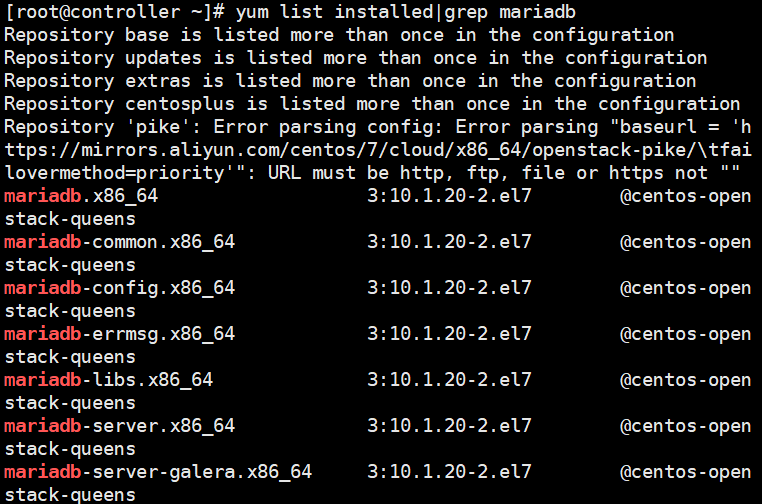
systemctl enable ntpd.service

systemctl start ntpd.service

### 安装数据库

#### 安装mariadb

yum install mariadb mariadb-server MySQL-python –y



#### 备份配置文件

cp /etc/my.cnf /etc/my.cnf.bak

#### 修改配置文件

vim /etc/my.cnf

[mysqld]

bind-address = controller

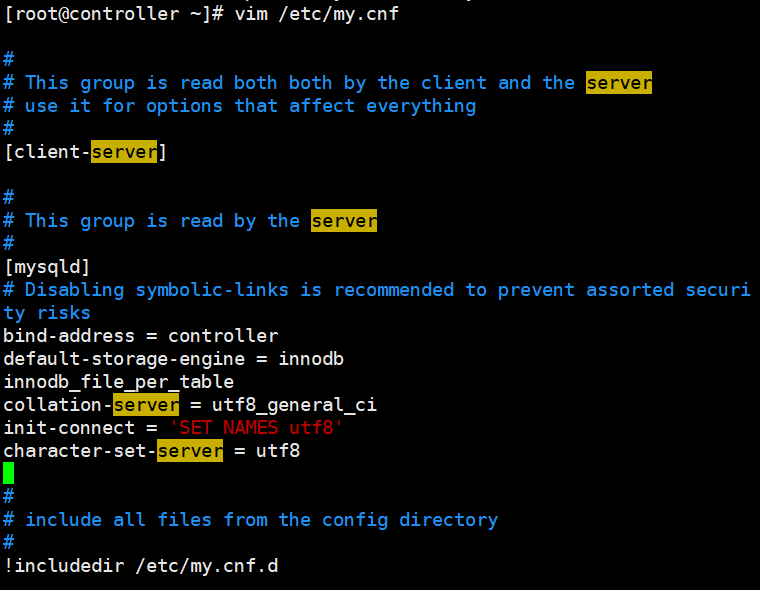
default-storage-engine = innodb

innodb\_file\_per\_table

collation-server = utf8\_general\_ci

init-connect = 'SET NAMES utf8'

character-set-server = utf8

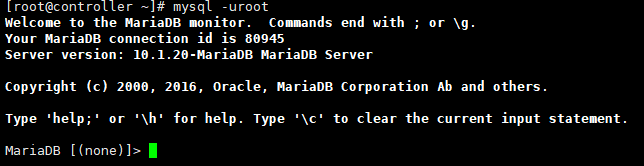


#### 启动数据库服务

systemctl enable mariadb.service

systemctl start mariadb.service

#### 验证



#### 配置数据库服务安全参数,设置root密码

mysqladmin -uroot password root

### 安装rabbitmq

#### 安装

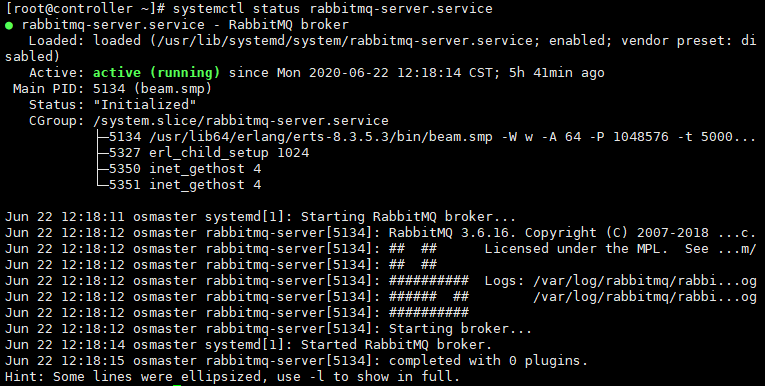
yum install rabbitmq-server -y

#### 启动服务

systemctl enable rabbitmq-server.service

systemctl start rabbitmq-server.service

#### 验证

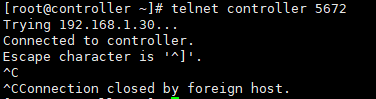


#### 添加用户，并允许远程访问

rabbitmqctl add\_user openstack sunwu123.

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

### 验证基础组件



### 安装keystone

#### 创建数据库

mysql -uroot -proot

create database keystone;

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

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

#### 安装keystone组件

1.安装

yum install  -y openstack-keystone httpd mod\_wsgi python-openstackclient memcached python-memcached

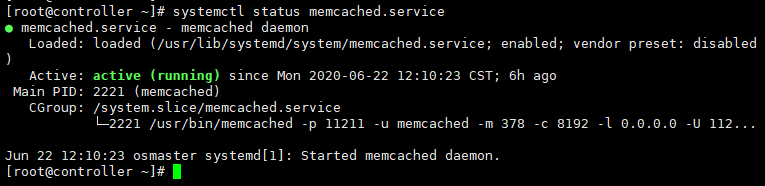
2.启动memcached

systemctl enable memcached.service

systemctl start memcached.service

3. 验证

systemctl status memcached.service



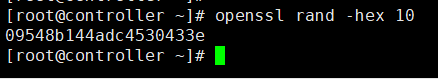
#### 配置keystone.conf

1.备份配置文件

cp /etc/keystone/keystone.conf /etc/keystone/keystone.conf.bak

2.生成token

openssl rand -hex 10



记录值 09548b144adc4530433e，之后要用

3.修改配置文件

vim /etc/keystone/keystone.conf

[DEFAULT]  
# 这个值是上面产生的

admin\_token = 09548b144adc4530433e

verbose = True

[database]

connection = mysql://keystone:[keystone@localhost](mailto:keystone@localhost)/keystone

[memcache]

servers = localhost:11211

[revoke]

driver = keystone.contrib.revoke.backends.sql.Revoke

[token]

provider = keystone.token.providers.uuid.Provider

driver = keystone.token.persistence.backends.memcache.Token

#### 初始化keystone数据库

1.执行创建表脚本

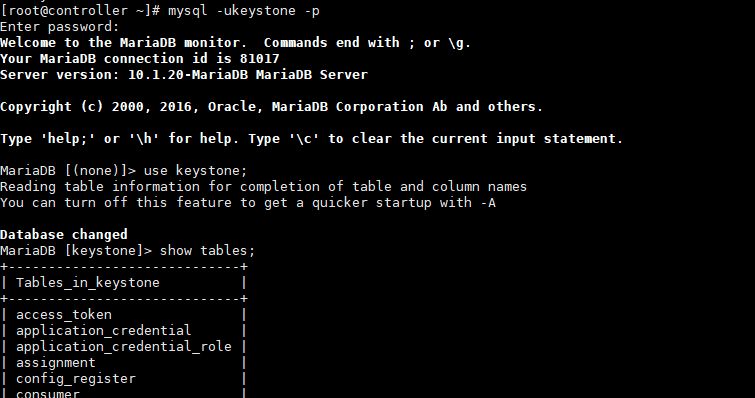
su -s /bin/sh -c "keystone-manage db\_sync" keystone  
keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

2.检查表是否均已创建完成

mysql -ukeystone -pkeystone

use keystone;

show tables;



#### 配置keystone的http服务

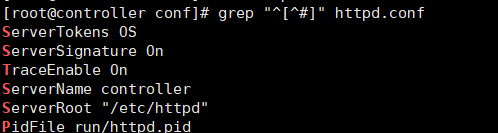
1.备份配置文件httpd.conf

cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.bak

2.修改配置文件httpd.conf

vim /httpd/conf/httpd.conf  
ServerName controller

如下图



3.启动httpServer

systemctl enable httpd.service

systemctl start httpd.service

systemctl status httpd.service

4.检查http服务是否正常启动

Netstat –ntlp|grep httpd

可能的错误

error while loading shared libraries: /lib64/libaprutil-1.so.0: file too short

yum -y install apr apr-util

openstack service create --name keystone --description "OpenStack Identity" identity

Internal Server Error (HTTP 500)

cat /var/log/keystone/keystone.log

#### 注册keystone api服务

1.配置系统环境变量(ADMIN\_TOKEN为2.1.5.3步骤通过openssl命令生成的值)

export OS\_TOKEN=ADMIN\_TOKEN

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

2.创建keystone服务信息

openstack service create --name keystone --description "OpenStack Identity" identity

3.创建keystone服务实例

openstack endpoint create \

--publicurl <http://controller:5000/v2.0> \

--internalurl <http://controller:5000/v2.0> \

--adminurl <http://controller:35357/v2.0> \

--region RegionOne \

identity

#### 创建项目、用户、角色等信息

1．创建项目 **admin**

openstack project create --description "Admin Project" admin

2. 为项目**admin**创建用户：**admin**

openstack user create --password-prompt admin

3. 创建角色 admin

openstack role create admin

4. 将角色admin授权给用户admin

openstack role add --project admin --user admin admin

5.创建项目demo及相关用户信息

openstack project create --description "Demo Project" demo

openstack user create --password-prompt demo

openstack role create user

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

6.创建项目：service

openstack project create --description "Service Project" service

2.5.9 验证keystone服务

1.删除系统配置参数

unset OS\_TOKEN

unset OS\_URL

2.备份并修改keystone-dist-paste.ini配置文件

cp /usr/share/keystone/keystone-dist-paste.ini /usr/share/keystone/keystone-dist-paste.ini.bak

删除以下配置参数中的admin\_token\_auth字符串, 三个都要删除

3.为项目admin配置环境变量脚本

vi /root/admin-openrc.sh

export OS\_PROJECT\_DOMAIN\_ID=default

export OS\_USER\_DOMAIN\_ID=default

export OS\_PROJECT\_NAME=admin

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=sunwu123.

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

4.为项目demo配置环境变量脚本

vi /root/demo-openrc.sh

export OS\_PROJECT\_DOMAIN\_ID=default

export OS\_USER\_DOMAIN\_ID=default

export OS\_PROJECT\_NAME=demo

export OS\_TENANT\_NAME=demo

export OS\_USERNAME=demo

export OS\_PASSWORD=sunwu123.

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

5.使用admin用户查看项目列表

Source admin-openrc.sh

Openstack project list

6.使用demo用户获取token

Source demo-openrc.sh

Openstack token issue

### 安装镜像服务glance

#### 创建数据库

CREATE DATABASE glance;

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

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

#### 创建用户：glance，并授权

source /root/admin-openrc.sh

openstack user create --password-prompt glance

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

#### 创建服务及服务实例

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

openstack endpoint create \

--publicurl [http://controller:9292](http://controller:9292/) \

--internalurl [http://controller:9292](http://controller:9292/) \

--adminurl [http://controller:9292](http://controller:9292/) \

--region RegionOne \

image

#### 安装glance

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

#### 配置glance api组件

1.备份配置文件

cp /etc/glance/glance-api.conf /etc/glance/glance-api.conf.bak

2.修改配置文件

[DEFAULT]

verbose = True

notification\_driver = noop

[database]

connection = mysql://glance:[glance@](mailto:glance@c)localhost/glance

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = glance

password = sunwu123.

[paste\_deploy]

flavor = keystone

[glance\_store]

default\_store = file

filesystem\_store\_datadir = /var/lib/glance/images/

#### 配置glance registry组件

1.备份配置文件

cp /etc/glance/glance-registry.conf /etc/glance/glance-registry.conf.bak

2.修改配置文件

[DEFAULT]

verbose = True

notification\_driver = noop

[database]

connection = mysql://glance:[glance@](mailto:glance@c)localhost/glance

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = glance

password = sunwu123.

[paste\_deploy]

flavor = keystone

#### 初始化glance数据库

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

#### 启动glance服务

systemctl enable openstack-glance-api.service openstack-glance-registry.service

systemctl start openstack-glance-api.service openstack-glance-registry.service

#### 验证glance服务

echo "export OS\_IMAGE\_API\_VERSION=2" | tee -a admin-openrc.sh \

demo-openrc.sh

glance image-create --name "cirros-0.3.4-x86\_64" \

--file /root/cirros-0.3.4-x86\_64-disk.img \

--disk-format qcow2 --container-format bare --visibility public –-progress

openstack image list

## Nova组件安装

### 在controller节点安装

#### 创建数据库

CREATE DATABASE nova;

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

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

#### 创建用户：nova，并授权用户权限

source admin-openrc.sh

openstack user create --password-prompt nova

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

#### 创建服务及服务实例

openstack service create --name nova \

--description "OpenStack Compute" compute

openstack endpoint create --publicurl [http://controller:8774/v2/%](http://controller:8774/v2/%25)\(tenant\_id\)s \

--internalurl [http://controller:8774/v2/%](http://controller:8774/v2/%25)\(tenant\_id\)s \

--adminurl [http://controller:8774/v2/%](http://controller:8774/v2/%25)\(tenant\_id\)s \

--region RegionOne \

compute

#### 安装nova组件

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

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

python-novaclient -y

#### 配置nova.conf

1.备份配置文件

cp /etc/nova/nova.conf /etc/nova/nova.conf.bak

2.修改配置文件(ip地址为controller节点所属IP，可以根据情况自行修改)

[DEFAULT]

verbose = True

rpc\_backend = rabbit

auth\_strategy = keystone

my\_ip = 192.168.1.30

vncserver\_listen = 192.168.1.30

vncserver\_proxyclient\_address = 192.168.1.30

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = sunwu123.

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = nova

password = sunwu123.

[glance]

host = controller

[oslo\_concurrency]

lock\_path = /var/lib/nova/tmp

[database]

connection = mysql://nova:nova[@](mailto:NOVA_DBPASS@controller)localhost/nova

#### 初始化nova数据库

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

#### 启动nova

systemctl enable openstack-nova-api.service openstack-nova-cert.service \

openstack-nova-consoleauth.service openstack-nova-scheduler.service \

openstack-nova-conductor.service openstack-nova-novncproxy.service

systemctl start openstack-nova-api.service openstack-nova-cert.service \

openstack-nova-consoleauth.service openstack-nova-scheduler.service \

openstack-nova-conductor.service openstack-nova-novncproxy.service

### compute-node-01安装

#### 安装nova组件

yum install openstack-nova-compute sysfsutils -y

#### 配置nova

1.备份配置文件

cp /etc/nova/nova.conf /etc/nova/nova.conf.bak

2.修改配置文件

[DEFAULT]

verbose = True

rpc\_backend = rabbit

auth\_strategy = keystone

my\_ip = 192.168.1.31

vnc\_enabled = True

vncserver\_listen = 0.0.0.0

vncserver\_proxyclient\_address = 192.168.1.31

novncproxy\_base\_url = <http://controller:6080/vnc_auto.html>

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = sunwu123.

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = nova

password = sunwu123.

[glance]

host = controller

[oslo\_concurrency]

lock\_path = /var/lib/nova/tmp

[libvirt]

virt\_type = qemu

#### 启动nova

systemctl enable libvirtd.service openstack-nova-compute.service

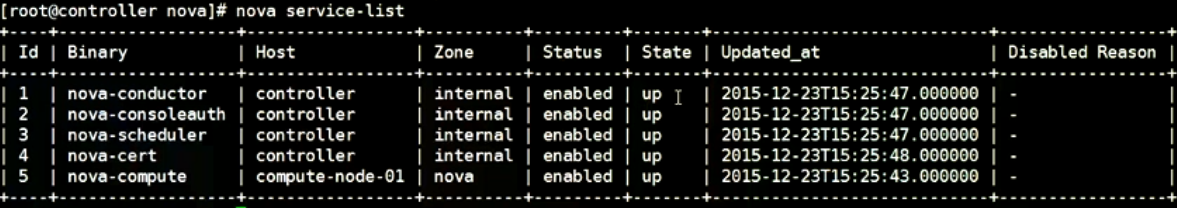
systemctl start libvirtd.service openstack-nova-compute.service

#### 验证

以下操作均在controller节点上操作.

source /root/admin-openrc.sh

nova service-list



## Neutron组件安装

### 在controller节点安装

#### 创建数据库

create database neutron;

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

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

#### 创建用户：neutron，并授权用户权限

source admin-openrc.sh

openstack user create --password-prompt neutron

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

#### 注册服务及服务实例

openstack service create --name neutron \

--description "OpenStack Networking" network

openstack endpoint create \

--publicurl [http://controller:9696](http://controller:9696/) \

--adminurl [http://controller:9696](http://controller:9696/) \

--internalurl [http://controller:9696](http://controller:9696/) \

--region RegionOne \

network

#### 安装neutron组件

yum install openstack-neutron openstack-neutron-ml2 \

python-neutronclientwhich -y

#### 配置neutron.conf

1.备份

cp /etc/neutron/neutron.conf /etc/neutron/neutron.conf.bak

2.修改配置文件

[DEFAULT]

verbose = True

rpc\_backend = rabbit

auth\_strategy = keystone

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

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

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

nova\_url = <http://controller:8774/v2>

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = sunwu123.

[database]

connection = mysql://neutron:[neutron@](mailto:neutron@c)localhost/neutron

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = neutron

password = sunwu123.

[nova]

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

region\_name = RegionOne

project\_name = service

username = nova

password = [sunwu123.](file:///C:\Users\Administrator\AppData\Local\Youdao\YNote\editor\web\wwwwww)

#### 配置ml2插件

1.备份

cp /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugins/ml2/ml2\_conf.ini.bak

2.配置

[ml2]

type\_drivers = flat,vlan,gre,vxlan

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

[ml2\_type\_gre]

tunnel\_id\_ranges = 1:1000

[securitygroup]

enable\_security\_group = True

enable\_ipset = True

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

3.建立link文件

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

#### 配置nova.conf

1.备份

cp /etc/nova/nova.conf /etc/nova/nova.conf.bak1

2.配置

[DEFAULT]

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

security\_group\_api = neutron

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

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

[neutron]

url = [http://controller:9696](http://controller:9696/)

auth\_strategy = keystone

admin\_auth\_url = <http://controller:35357/v2.0>

admin\_tenant\_name = service

admin\_username = neutron

admin\_password = [sunwu123.](file:///C:\Users\Administrator\AppData\Local\Youdao\YNote\editor\web\wwwwww)

#### 初始化数据库

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

#### 重启nova服务

systemctl restart openstack-nova-api.service openstack-nova-scheduler.service \

openstack-nova-conductor.service

#### 启动neutron服务

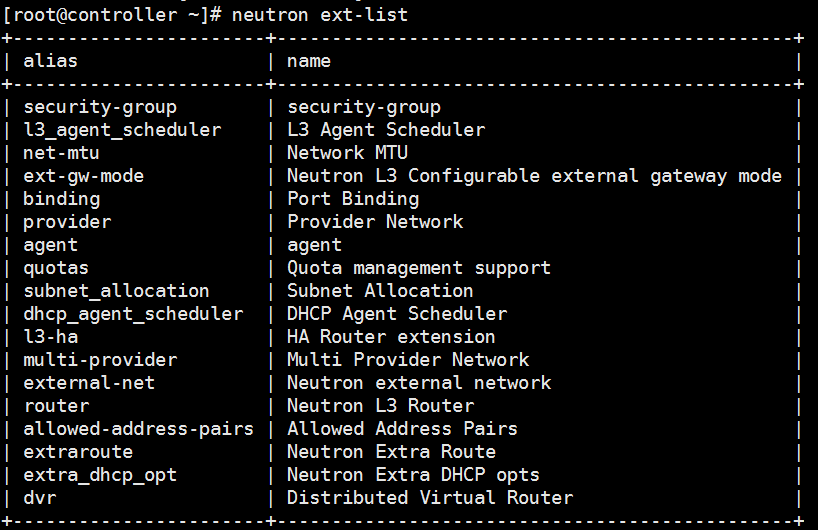
systemctl enable neutron-server.service

systemctl start neutron-server.service

#### 验证

source /root/admin-openrc.sh

neutron ext-list



### 在network-node-01节点安装

#### 启用IP转发功能

1.备份配置文件

cp /etc/sysctl.conf /etc/sysctl.conf.bak

2.修改配置

net.ipv4.ip\_forward=1

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

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

3.启用配置

sysctl -p

#### 安装neutron组件

yum install openstack-neutron openstack-neutron-ml2 \

openstack-neutron-openvswitch -y

#### 配置neutron.conf

1.备份

cp /etc/neutron/neutron.conf /etc/neutron/neutron.conf.bak

2.配置

[DEFAULT]

verbose = True

rpc\_backend = rabbit

auth\_strategy = keystone

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = sunwu123.

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = neutron

password = [sunwu123.](file:///C:\Users\Administrator\AppData\Local\Youdao\YNote\editor\web\wwwwww)

#### 配置ML2插件

配置【OVS】段中的IP为network-node-01节点的桥接网卡的地址，br-ex为虚拟网桥

1.备份

cp /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugins/ml2/ml2\_conf.ini.bak

2.配置

[ml2]

type\_drivers = flat,vlan,gre,vxlan

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

[ml2\_type\_flat]

flat\_networks = external

[ml2\_type\_gre]

tunnel\_id\_ranges = 1:1000

[securitygroup]

enable\_security\_group = True

enable\_ipset = True

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

[ovs]

local\_ip = 192.168.1.32

bridge\_mappings = external:br-ex

[agent]

tunnel\_types = gre

3.创建链接文件

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

#### 配置ML3插件

1.备份

cp /etc/neutron/l3\_agent.ini /etc/neutron/l3\_agent.ini.bak

2.配置

[DEFAULT]

verbose = True

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

router\_delete\_namespaces = True

#### 配置DHCP插件

1.备份

cp /etc/neutron/dhcp\_agent.ini /etc/neutron/dhcp\_agent.ini.bak

2.配置

verbose = True

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

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

dhcp\_delete\_namespaces = True

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

3.创建dnsmasq-neutron.conf。并添加以下参数：

dhcp-option-force=26,1454

#### 配置元数据插件

1.备份

cp /etc/neutron/metadata\_agent.ini /etc/neutron/metadata\_agent.ini.bak

2.配置

# 删除[DEFAULT]段中的默认配置

# 在[DEFAULT]段中增加以下内容

verbose = True

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_region = RegionOne

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = neutron

password = sunwu123.

nova\_metadata\_ip = controller

metadata\_proxy\_shared\_secret = [sunwu123.](file:///C:\Users\Administrator\AppData\Local\Youdao\YNote\editor\web\wwwwww)

#### 配置nova.conf(在controller节点配置)

1.备份

cp /etc/nova/nova.conf /etc/nova/nova.conf.bak

2.在[neutron]节点中增加以下内容

service\_metadata\_proxy = True

metadata\_proxy\_shared\_secret = [sunwu123.](file:///C:\Users\Administrator\AppData\Local\Youdao\YNote\editor\web\wwwwww)

2.重启nova服务

systemctl restart openstack-nova-api.service openstack-nova-scheduler.service \

openstack-nova-conductor.service

#### 启动服务

systemctl enable openvswitch.service

systemctl start openvswitch.service

#### 创建虚拟路由器

1.创建虚拟路由器

ovs-vsctl add-br br-ex

2.添加网卡到虚拟路由器

ovs-vsctl add-port br-ex enp0s8

3.禁止GRO

ethtool -K enp0s9 gro off

#### 启动服务

cp /usr/lib/systemd/system/neutron-openvswitch-agent.service \

/usr/lib/systemd/system/neutron-openvswitch-agent.service.orig

sed -i 's,plugins/openvswitch/ovs\_neutron\_plugin.ini,plugin.ini,g' \

/usr/lib/systemd/system/neutron-openvswitch-agent.service

systemctl enable neutron-openvswitch-agent.service neutron-l3-agent.service \

neutron-dhcp-agent.service neutron-metadata-agent.service neutron-ovs-cleanup.service

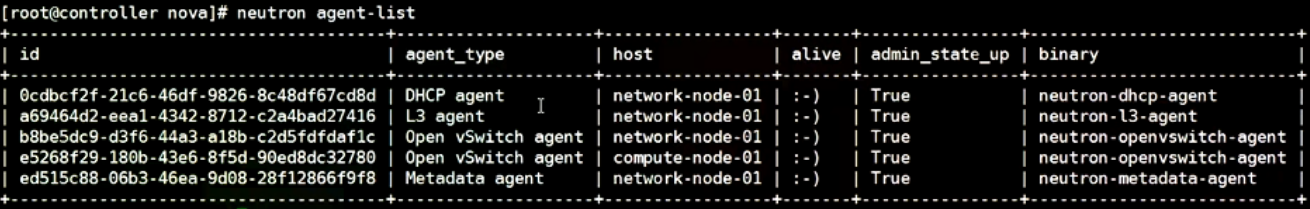
systemctl start neutron-openvswitch-agent.service neutron-l3-agent.service \

neutron-dhcp-agent.service neutron-metadata-agent.service

#### 验证

source /root/admin-openrc.sh

neutron agent-list



### 在compute-node-01节点安装

#### 配置IP转发

1.备份

cp /etc/sysctl.conf /etc/sysctl.conf.bak

2.配置

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

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

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

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

3.生效配置

sysctl -p

modprobe bridge

#### 安装

yum install openstack-neutron openstack-neutron-ml2 \

openstack-neutron-openvswitch -y

#### 配置neutron

1.备份

cp /etc/neutron/neutron.conf /etc/neutron/neutron.conf.bak

2.配置（需要删除[keystone\_authtoken]中已存在的配置）

[DEFAULT]

verbose = True

rpc\_backend = rabbit

auth\_strategy = keystone

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = sunwu123.

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = neutron

password = [sunwu123.](file:///C:\Users\Administrator\AppData\Local\Youdao\YNote\editor\web\wwwwww)

#### 配置ml2插件

1.备份

cp /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugins/ml2/ml2\_conf.ini.bak

2.配置

[ml2]

type\_drivers = flat,vlan,gre,vxlan

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

[ml2\_type\_gre]

tunnel\_id\_ranges = 1:1000

[securitygroup]

enable\_security\_group = True

enable\_ipset = True

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

[ovs]

local\_ip = 192.168.1.31

[agent]

tunnel\_types = gre

3.创建link文件

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

#### 配置nova.conf

1.备份

cp /etc/nova/nova.conf /etc/nova/nova.conf.bak1

2.配置

[DEFAULT]

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

security\_group\_api = neutron

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

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

[neutron]

url = [http://controller:9696](http://controller:9696/)

auth\_strategy = keystone

admin\_auth\_url = <http://controller:35357/v2.0>

admin\_tenant\_name = service

admin\_username = neutron

admin\_password = [sunwu123.](file:///C:\Users\Administrator\AppData\Local\Youdao\YNote\editor\web\wwwwww)

#### 启动

cp /usr/lib/systemd/system/neutron-openvswitch-agent.service \

/usr/lib/systemd/system/neutron-openvswitch-agent.service.orig

sed -i 's,plugins/openvswitch/ovs\_neutron\_plugin.ini,plugin.ini,g' \

/usr/lib/systemd/system/neutron-openvswitch-agent.service

systemctl enable openvswitch.service

systemctl start openvswitch.service

systemctl restart openstack-nova-compute.service

systemctl enable neutron-openvswitch-agent.service

systemctl start neutron-openvswitch-agent.service

#### 验证

neutron agent-list

1.创建扩展网络

neutron net-create ext-net --router:external \

--provider:physical\_network external --provider:network\_type flat

2.创建扩展网络子网

neutron subnet-create ext-net 192.168.57.0/24 --name ext-subnet \

--allocation-pool start=192.168.57.101,end=192.168.57.200 \

--disable-dhcp --gateway 192.168.57.1

3.创建租户网络

source /root/demo-openrc.sh

neutron net-create demo-net

 neutron subnet-create demo-net 192.168.2.0/24 \

--name demo-subnet --dns-nameserver 192.168.1.1 --gateway 192.168.2.1

4.创建路由器

neutron router-create demo-router

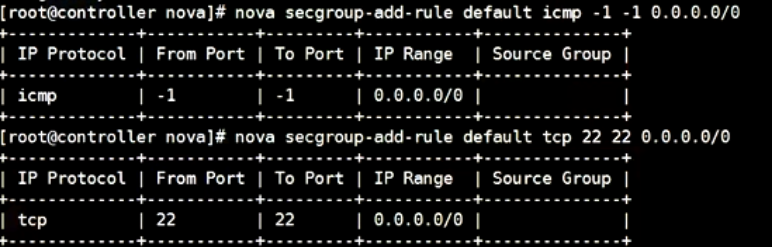
neutron router-interface-add demo-router demo-subnet

neutron router-gateway-set demo-router ext-net

5.设置安全组规则

nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0

nova secgroup-add-rule default tcp 22 22 0.0.0.0/0



### controller节点安装Dashboard

#### 安装

 yum install openstack-dashboard -y

#### 配置

1.备份

cp /etc/openstack-dashboard/local\_settings /etc/openstack-dashboard/local\_settings.bak

2.配置

import sys

reload(sys)

sys.setdefaultencoding('utf8')

OPENSTACK\_HOST = "controller"

ALLOWED\_HOSTS = ['\*', ]

CACHES = {

    'default': {

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

    'LOCATION': '127.0.0.1:11211',

    }

}

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "user"

TIME\_ZONE = "Asia/Shanghai"

#### 系统权限配置

setsebool -P httpd\_can\_network\_connect on

chown -R apache:apache /usr/share/openstack-dashboard/static

systemctl restart httpd.service memcached.service

## Cinder组件安装

### Controller节点安装

#### 创建数据库

CREATE DATABASE cinder;

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

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

#### 创建账号及服务

source /root/admin-openrc.sh

openstack user create --password-prompt cinder

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

openstack service create --name cinder \

--description "OpenStack Block Storage" volume

openstack service create --name cinderv2 \

--description "OpenStack Block Storage" volumev2

openstack endpoint create \

--publicurl [http://controller:8776/v2/%](http://controller:8776/v2/%25)\(tenant\_id\)s \

--internalurl [http://controller:8776/v2/%](http://controller:8776/v2/%25)\(tenant\_id\)s \

--adminurl [http://controller:8776/v2/%](http://controller:8776/v2/%25)\(tenant\_id\)s \

--region RegionOne \

volume

openstack endpoint create \

--publicurl [http://controller:8776/v2/%](http://controller:8776/v2/%25)\(tenant\_id\)s \

--internalurl [http://controller:8776/v2/%](http://controller:8776/v2/%25)\(tenant\_id\)s \

--adminurl [http://controller:8776/v2/%](http://controller:8776/v2/%25)\(tenant\_id\)s \

--region RegionOne \

volumev2

#### 安装Cinder组件

yum install openstack-cinder python-cinderclient python-oslo-db -y

#### 配置

cp /usr/share/cinder/cinder-dist.conf /etc/cinder/cinder.conf

chown -R cinder:cinder /etc/cinder/cinder.conf

vi /etc/cinder/cinder.conf

[DEFAULT]

verbose = True

rpc\_backend = rabbit

auth\_strategy = keystone

my\_ip = 192.168.1.30

[database]

connection = mysql://cinder:[cinder@](mailto:cinder@c)localhost/cinder

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = sunwu123.

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = cinder

password = sunwu123.

[oslo\_concurrency]

lock\_path = /var/lock/cinder

#### 数据初始化

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

#### 启动服务

systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service

systemctl start openstack-cinder-api.service openstack-cinder-scheduler.service

### 在block-node-01节点上安装

#### 安装支持包

1.安装

yum install qemu lvm2 –y

2.启动

systemctl enable lvm2-lvmetad.service

systemctl start lvm2-lvmetad.service

#### 创建逻辑卷

fdisk /dev/sdb

#### 更新内核中的分区表

partprobe

#### 创建物理卷及卷组

pvcreate /dev/sdb1

vgcreate cinder-volumes /dev/sdb1

#### 修改lvm服务配置

1.备份

cp /etc/lvm/lvm.conf /etc/lvm/lvm.conf.bak

2.修改

devices {

filter = [ "a/sda/", "a/sdb/", "r/.\*/"]

3.重启lvm服务

systemctl restart lvm2-lvmetad.service

7.2.4安装cinder组件

yum install openstack-cinder targetcli python-oslo-db python-oslo-log MySQL-python -y

#### 配置cinder

1.备份

cp /etc/cinder/cinder.conf /etc/cinder/cinder.conf.bak

2.配置cinder.conf

[DEFAULT]

rpc\_backend = rabbit

auth\_strategy = keystone

my\_ip = 192.168.1.33

enabled\_backends = lvm

glance\_host = controller

verbose = True

[oslo\_messaging\_rabbit]

rabbit\_host = controller

rabbit\_userid = openstack

rabbit\_password = sunwu123.

[database]

connection = mysql://cinder:[cinder@controller](mailto:cinder@controller)/cinder

[keystone\_authtoken]

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = cinder

password = sunwu123.

[lvm]

volume\_driver = cinder.volume.drivers.lvm.LVMVolumeDriver

volume\_group = cinder-volumes

iscsi\_protocol = iscsi

iscsi\_helper = lioadm

[oslo\_concurrency]

lock\_path = /var/lock/cinder

3.创建文件夹

mkdir -p /var/lock/cinder

chown cinder:cinder /var/lock/cinder

#### 启动cinder服务

systemctl enable openstack-cinder-volume.service target.service

systemctl start openstack-cinder-volume.service target.service

#### 验证

echo "export OS\_VOLUME\_API\_VERSION=2" | tee -a admin-openrc.sh demo-openrc.sh

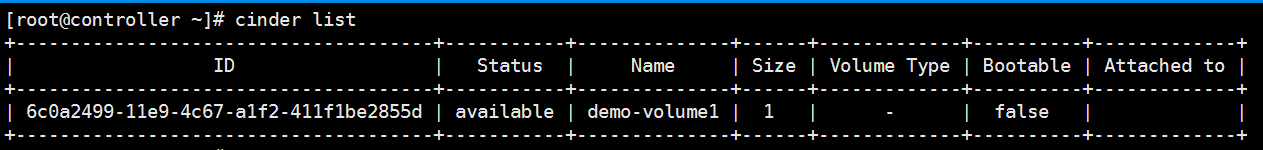
source /root/admin-openrc.sh

cinder service-list

source demo-openrc.sh

cinder create --name demo-volume1 1

cinder list



cinder show 6c0a2499-11e9-4c67-a1f2-411f1be2855d



在虚拟机实例下操作：

sudo fdisk –l

sudo fdisk /dev/vdb

sudo mkfs.ext4 /dev/vdb1

sudo mkdir testdir

sudo mount /dev/vdb1 testdir

## Swift组件安装

### 在controller节点安装

#### 创建账号及服务

source /root/admin-openrc.sh

openstack user create --password-prompt swift

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

openstack service create --name swift \

--description "OpenStack Object Storage" object-store

openstack endpoint create \

--publicurl '[http://controller:8080/v1/AUTH\_%(tenant\_id)s](http://controller:8080/v1/AUTH_%25(tenant_id)s)' \

--internalurl '[http://controller:8080/v1/AUTH\_%(tenant\_id)s](http://controller:8080/v1/AUTH_%25(tenant_id)s)' \

--adminurl [http://controller:8080](http://controller:8080/) \

--region RegionOne \

object-store

#### 安装swift组件

yum install openstack-swift-proxy python-swiftclient python-keystoneclient \

python-keystonemiddleware memcached -y

#### 配置proxy-server

1.下载配置文件模板

curl -o /etc/swift/proxy-server.conf \

<https://git.openstack.org/cgit/openstack/swift/plain/etc/proxy-server.conf-sample?h=stable/kilo>

2.修改配置文件

vi /etc/swift/proxy-server.conf

[DEFAULT]

bind\_port = 8080

user = swift

swift\_dir = /etc/swift

[pipeline:main]

pipeline = catch\_errors gatekeeper healthcheck proxy-logging cache container\_sync bulk ratelimit authtoken keystoneauth container-quotas account-quotas slo dlo proxy-logging proxy-server

[app:proxy-server]

use = egg:swift#proxy

account\_autocreate = true

[filter:keystoneauth]

use = egg:swift#keystoneauth

operator\_roles = admin,user

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:authtoken]

paste.filter\_factory = keystonemiddleware.auth\_token:filter\_factory

auth\_uri = [http://controller:5000](http://controller:5000/)

auth\_url = [http://controller:35357](http://controller:35357/)

auth\_plugin = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = swift

password = sunwu123.

delay\_auth\_decision = true

[filter:cache]

memcache\_servers = 127.0.0.1:11211

### 在object-node-01节点安装

#### 系统配置

1.安装系统组件

yum install xfsprogs rsync -y

2.格式化磁盘

fdisk /dev/sdb

fdisk /dev/sdc

mkfs.xfs /dev/sdb1

mkfs.xfs /dev/sdc1

3.创建文件夹

mkdir -p /srv/node/sdb1

mkdir -p /srv/node/sdc1

4.修改fstab

vi /etc/fstab

/dev/sdb1 /srv/node/sdb1 xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

/dev/sdc1 /srv/node/sdc1 xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

5.挂载磁盘到文件夹

mount /srv/node/sdb1

mount /srv/node/sdc1

6.修改文件同步服务

cp /etc/rsyncd.conf /etc/rsyncd.conf.bak

uid = swift

gid = swift

log file = /var/log/rsyncd.log

pid file = /var/run/rsyncd.pid

address = MANAGEMENT\_INTERFACE\_IP\_ADDRESS

[account]

max connections = 2

path = /srv/node/

read only = false

lock file = /var/lock/account.lock

[container]

max connections = 2

path = /srv/node/

read only = false

lock file = /var/lock/container.lock

[object]

max connections = 2

path = /srv/node/

read only = false

lock file = /var/lock/object.lock

7.启动文件同步服务

systemctl enable rsyncd.service

systemctl start rsyncd.service

#### 安装

yum install openstack-swift-account openstack-swift-container \

openstack-swift-object -y

#### 配置swift

1.下载配置文件模板

curl -o /etc/swift/account-server.conf \

curl -o /etc/swift/container-server.conf \

curl -o /etc/swift/object-server.conf \

curl -o /etc/swift/container-reconciler.conf \

curl -o /etc/swift/object-expirer.conf \

curl -o /etc/swift/swift.conf \

https://git.openstack.org/cgit/openstack/swift/plain/etc/swift.conf-sample?h=stable/kilo

2.修改/etc/swift/account-server.conf

[DEFAULT]

bind\_ip = 本机IP

bind\_port = 6002

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon account-server

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:recon]

use = egg:swift#recon

recon\_cache\_path = /var/cache/swift

3.修改/etc/swift/container-server.conf

[DEFAULT]

bind\_ip = 本机IP

bind\_port = 6001

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon container-server

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:recon]

use = egg:swift#recon

recon\_cache\_path = /var/cache/swift

4.修改/etc/swift/object-server.conf

[DEFAULT]

bind\_ip = 本机IP

bind\_port = 6000

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon object-server

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:recon]

use = egg:swift#recon

recon\_cache\_path = /var/cache/swift

recon\_lock\_path = /var/lock

5.修改/etc/swift/swift.conf

[swift-hash]

swift\_hash\_path\_suffix = HASH\_PATH\_SUFFIX

swift\_hash\_path\_prefix = HASH\_PATH\_PREFIX

[storage-policy:0]

name = Policy-0

default = yes

5.文件夹授权

chown -R swift:swift /srv/node

mkdir -p /var/cache/swift

chown -R swift:swift /var/cache/swift

chown -R swift:swift /etc/swift

### 在object-node-02节点安装

# 参照2.5.1章节安装。配置文件可以自己从object-node-01节点拷贝过来，只需要修改配置文件中的IP地址即可。

### controller节点创建环（Ring）

P地址分别为object-node-01和object-node-02所对应的IP

1. 创建账号环（Account Ring）

swift-ring-builder account.builder create 10 3 1

swift-ring-builder account.builder add r1z1-192.168.1.34:6002/sdb1 100

swift-ring-builder account.builder add r1z1-192.168.1.34:6002/sdc1 100

swift-ring-builder account.builder add r1z2-192.168.1.35:6002/sdb1 100

swift-ring-builder account.builder add r1z2-192.168.1.35:6002/sdc1 100

swift-ring-builder account.builder

swift-ring-builder account.builder rebalance

2.创建容器环（Container Ring)

swift-ring-builder container.builder create 10 3 1

swift-ring-builder container.builder create 10 3 1

swift-ring-builder container.builder add r1z1-192.168.1.34:6001/sdb1 100

swift-ring-builder container.builder add r1z1-192.168.1.34:6001/sdc1 100

swift-ring-builder container.builder add r1z2-192.168.1.35:6001/sdb1 100

swift-ring-builder container.builder add r1z2-192.168.1.35:6001/sdc1 100

swift-ring-builder container.builder

swift-ring-builder container.builder rebalance

3.创建对象环（Object Ring)

swift-ring-builder object.builder create 10 3 1

swift-ring-builder object.builder create 10 3 1

swift-ring-builder object.builder add r1z1-192.168.1.34:6000/sdb1 100

swift-ring-builder object.builder add r1z1-192.168.1.34:6000/sdc1 100

swift-ring-builder object.builder add r1z2-192.168.1.35:6000/sdb1 100

swift-ring-builder object.builder add r1z2-192.168.1.35:6000/sdc1 100

swift-ring-builder object.builder

swift-ring-builder object.builder rebalance

3.启动服务(controller节点执行)

systemctl enable openstack-swift-proxy.service

systemctl start openstack-swift-proxy.service

systemctl restart memcached.service

4.启动服务（object-node-01, object-node-02节点执行）

systemctl enable openstack-swift-account.service openstack-swift-account-auditor.service \

openstack-swift-account-reaper.service openstack-swift-account-replicator.service

systemctl start openstack-swift-account.service openstack-swift-account-auditor.service \

openstack-swift-account-reaper.service openstack-swift-account-replicator.service

systemctl enable openstack-swift-container.service openstack-swift-container-auditor.service \

openstack-swift-container-replicator.service openstack-swift-container-updater.service

systemctl start openstack-swift-container.service openstack-swift-container-auditor.service \

openstack-swift-container-replicator.service openstack-swift-container-updater.service

systemctl enable openstack-swift-object.service openstack-swift-object-auditor.service \

openstack-swift-object-replicator.service openstack-swift-object-updater.service

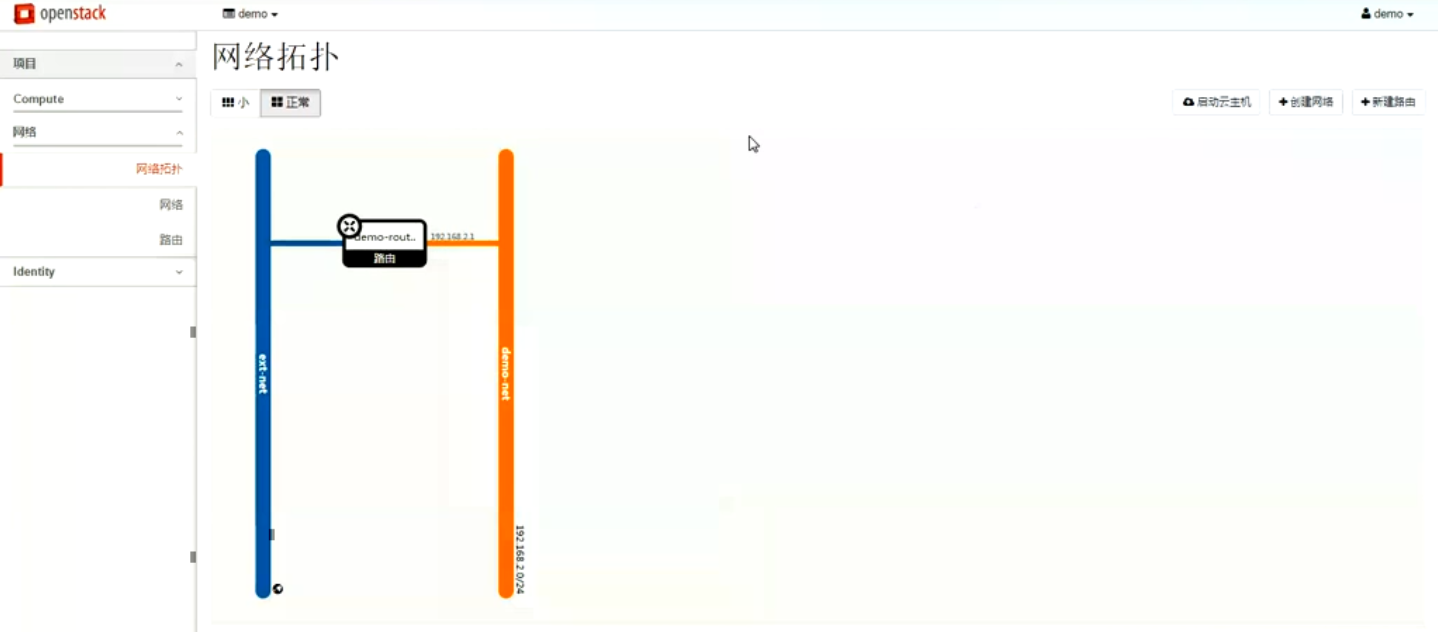
systemctl start openstack-swift-object.service openstack-swift-object-auditor.service \

openstack-swift-object-replicator.service openstack-swift-object-updater.service

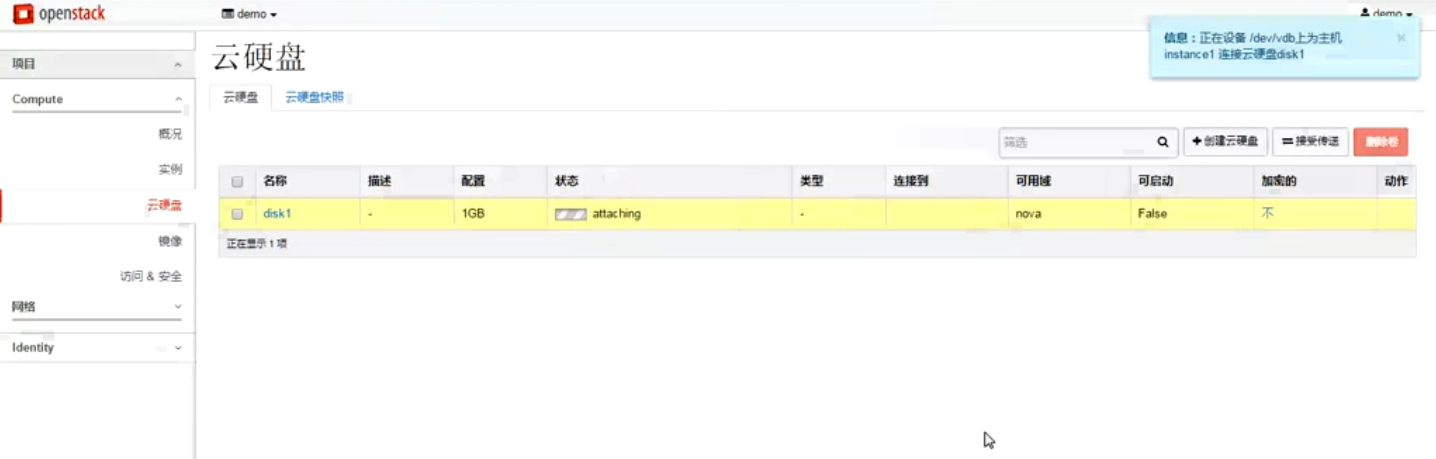
配置完成之后，即可在浏览器中打开查看，分别如下图所示。



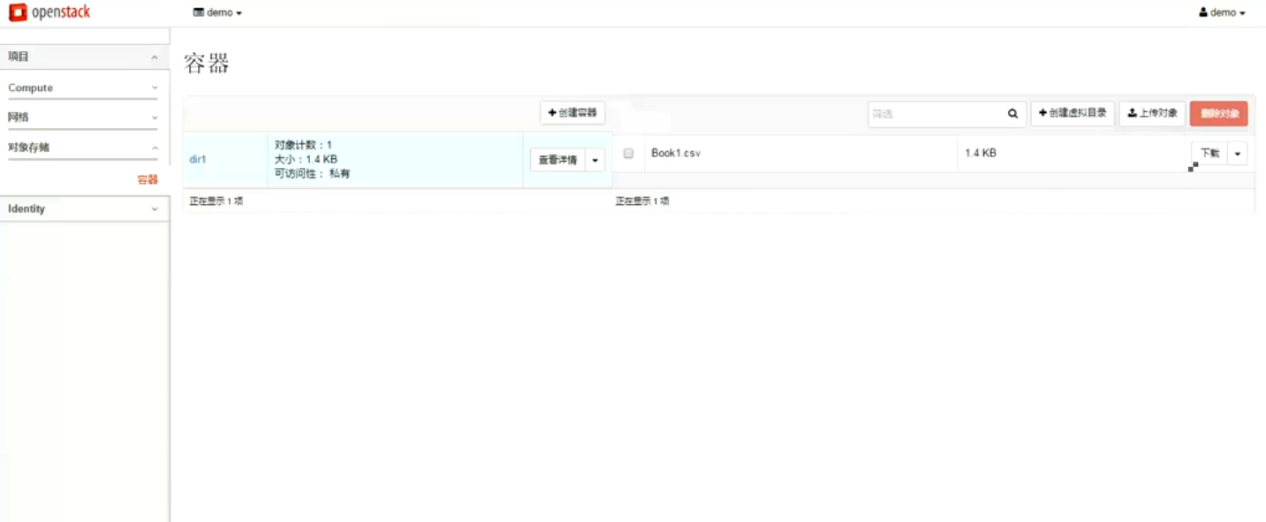












到此 openstack 安装结束。