**OpenStack Havana安装测试报告**

**目录**

[1. 项目概述 5](#_Toc373330674)

[1.1. 项目目标 5](#_Toc373330675)

[1.2. 测试环境 5](#_Toc373330676)

[1.2.1. 版本说明 5](#_Toc373330677)

[1.2.2. 部署架构 5](#_Toc373330678)

[2. 控制节点安装测试 7](#_Toc373330679)

[2.1. 基础配置 7](#_Toc373330680)

[2.1.1. 系统安装配置 7](#_Toc373330681)

[2.1.2. 安装NTP服务 9](#_Toc373330682)

[2.1.3. MySQL安装配置 10](#_Toc373330683)

[2.1.4. Qpid安装配置 10](#_Toc373330684)

[2.1.5. 安装OpenStack工具包 11](#_Toc373330685)

[2.2. Keystone安装配置 11](#_Toc373330686)

[2.2.1. 初始化Keystone 11](#_Toc373330687)

[2.2.2. 定义Users、Tenants and Roles 12](#_Toc373330688)

[2.2.3. 定义Services 和 API Endpoints 12](#_Toc373330689)

[2.3. Glance安装配置 13](#_Toc373330690)

[2.3.1. 初始化Glance 13](#_Toc373330691)

[2.3.2. 创建User，定义Services 和 API Endpoints 13](#_Toc373330692)

[2.3.3. 配置Glance服务 13](#_Toc373330693)

[2.3.4. Glance测试 15](#_Toc373330694)

[2.4. Nova安装配置 16](#_Toc373330695)

[2.4.1. 初始化Nova 16](#_Toc373330696)

[2.4.2. 创建User，定义Services 和 API Endpoints 16](#_Toc373330697)

[2.4.3. 配置Nova服务 16](#_Toc373330698)

[2.4.4. Nova测试 18](#_Toc373330699)

[2.5. Horizon安装配置 21](#_Toc373330700)

[2.6. Neutron安装配置 21](#_Toc373330701)

[2.6.1. 初始化Neutron 21](#_Toc373330702)

[2.6.2. 创建User，定义Services 和 API Endpoints 22](#_Toc373330703)

[2.6.3. 配置网络服务 22](#_Toc373330704)

[2.6.4. Neutron测试 26](#_Toc373330705)

[2.7. Cinder安装配置 31](#_Toc373330706)

[2.7.1. 初始化Cinder 31](#_Toc373330707)

[2.7.2. 创建User，定义Services 和 API Endpoints 31](#_Toc373330708)

[2.7.3. 配置Cinder服务 32](#_Toc373330709)

[2.7.4. Cinder测试 33](#_Toc373330710)

[2.8. Swift安装配置 34](#_Toc373330711)

[2.8.1 初始化Swift 34](#_Toc373330712)

[2.8.2 创建User，定义Services 和 API Endpoints 34](#_Toc373330713)

[2.8.3 配置Swift 35](#_Toc373330714)

[2.8.4 Cinder+Swift测试 40](#_Toc373330715)

[2.9. Ceilometer安装配置 42](#_Toc373330716)

[2.9.1. 初始化Ceilometer 42](#_Toc373330717)

[2.9.2. 创建User，定义Services 和 API Endpoints 42](#_Toc373330718)

[2.9.3. 配置Ceilometer 43](#_Toc373330719)

[2.9.4. Ceilometer测试（监控glance、nova、neturon、Cinder、Swift） 44](#_Toc373330720)

[2.10. Heat安装配置 48](#_Toc373330721)

[2.10.1. 初始化Heat 48](#_Toc373330722)

[2.10.2. 创建User，定义Services 和 API Endpoints 49](#_Toc373330723)

[2.10.3. 配置Heat 49](#_Toc373330724)

[2.10.4. Heat测试 51](#_Toc373330725)

[3. 添加一个计算节点 54](#_Toc373330726)

[3.1. 基础配置 54](#_Toc373330727)

[3.1.1. 系统安装配置 54](#_Toc373330728)

[3.1.2. 设置时间同步 55](#_Toc373330729)

[3.1.3. 配置libvirtd服务 55](#_Toc373330730)

[3.2. Neutron安装配置 55](#_Toc373330731)

[3.2.1. 初始化neutron-openvswitch 55](#_Toc373330732)

[3.2.2. 配置neutron服务 56](#_Toc373330733)

[3.3. Nova安装配置 58](#_Toc373330734)

[3.3.1. 初始化nova-compute 58](#_Toc373330735)

[3.3.2. 配置nova服务 58](#_Toc373330736)

[4. Havana测试 60](#_Toc373330737)

[4.1. 基本功能测试 60](#_Toc373330738)

[4.1.1. 测试方法 60](#_Toc373330739)

[4.1.2. 测试结论 60](#_Toc373330740)

[4.2. 虚拟机资源调整测试 61](#_Toc373330741)

[4.2.1. 测试方法 61](#_Toc373330742)

[4.2.2. 测试结论 61](#_Toc373330743)

[4.3. 虚拟机迁移测试 62](#_Toc373330744)

[4.3.1. 测试方法 62](#_Toc373330745)

[4.3.2. 测试结论 65](#_Toc373330746)

[4.4. F版到H版升级测试 65](#_Toc373330747)

[4.4.1. 测试方法 65](#_Toc373330748)

[4.4.2. 测试结论 66](#_Toc373330749)

[4.5. 其他新组件的测试 66](#_Toc373330750)

啥都不说了，必须用Python2.7或者更高，推荐2.7，不然解决依赖会让人想哭

# 项目概述

## 项目目标

通过openstack环境安装部署，实现对公司研发生产环境进行统一管理。

## 测试环境

### 版本说明

**操作系统：**

CentOS 6.4

**Openstack版本：**

Havana Release 2013.2

### 部署架构

* **OpenStack+kvm部署架构**



Openstack+kvm安装主机信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **主机名** | **IP地址** | **作用** | **备注** |
| hctrl | 172.21.2.110 | 控制节点+计算节点 | 系统采用CentOS6.4最小化安装，双网卡eth0、eth1 |
| havana | 172.21.2.111 | 计算节点 | 系统采用CentOS6.4最小化安装，双网卡eth0、eth1 |

OpenStack 各组件及组件间的关系：



# 控制节点安装测试

## 基础配置

### 系统安装配置

操作系统使用CentOS6.4最小化安装方式，安装过程省略。本文以下步骤采用网络yum源安装。

1. 导入第三方软件源

# rpm –Uvh http://download.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

# rpm –Uvh [http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86\_64.rpm](http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm)

# yum install http://repos.fedorapeople.org/repos/openstack/openstack-havana/[rdo-release-havana-7.noarch.rpm](http://repos.fedorapeople.org/repos/openstack/openstack-havana/rdo-release-havana-7.noarch.rpm)

注：若使用本地yum源安装，此步骤可省略，但需先将rpm包下载制作成本地源并修改repo文件。

1. 配置/etc/hosts文件

# vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

172.21.2.110 hctrl

1. 网络设置

[root@hctrl ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

HWADDR=00:15:17:C8:35:35

TYPE=Ethernet

UUID=196b23ae-2dda-4d9a-85a4-bb185fb82ca5

ONBOOT=yes

NM\_CONTROLLED=yes

BOOTPROTO=static

IPADDR=172.21.2.110

GATEWAY=172.21.2.1

NETMASK=255.255.255.0

[root@hctrl ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1 //不用配置IP地址，将来做网桥

DEVICE=eth1

HWADDR=00:15:17:C8:35:34

TYPE=Ethernet

UUID=82af9913-4d59-4811-9519-3710d4504a4f

ONBOOT=yes

NM\_CONTROLLED=yes

BOOTPROTO=static

1. 关闭selinux

修改/etc/selinux/config文件中设置SELINUX=disabled ：

# vi /etc/selinux/config

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

SELINUX=disabled

# SELINUXTYPE= can take one of these two values:

# targeted - Targeted processes are protected,

# mls - Multi Level Security protection.

SELINUXTYPE=targeted

1. 修改/etc/sysctl.conf参数

# vi /etc/sysctl.conf

……

net.ipv4.ip\_forward = 1

……

运行以下命令，使其生效：

# sysctl -p

1. 升级系统

# yum -y update

1. 重启机器

# reboot

### 安装NTP服务

1. 安装NTP时钟同步服务器

# yum install -y ntp

1. 编辑/etc/ntp.conf

# vi /etc/ntp.conf

driftfile /var/lib/ntp/drift

restrict default ignore

restrict 127.0.0.1

restrict 172.21.2.0 mask 255.255.255.0 nomodify notrap

server ntp.api.bz

server 127.127.1.0 # local clock

fudge 127.127.1.0 stratum 10

keys /etc/ntp/keys

1. 启动ntp服务，设置开机自启动

# service ntpd start

# chkconfig ntpd on

### MySQL安装配置

1. 安装MySQL

# yum install -y mysql mysql-server MySQL-python

1. 修改mysql启动文件

# vi /etc/my.cnf

[mysqld]

datadir=/data/mysql

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

user=mysql

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

bind-address = 0.0.0.0

[mysqld\_safe]

log-error=/var/log/mysqld.log

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

1. 建立MYSQL的数据存放目录

# lvcreate -L 100G rootvg -n lv\_data

# mkfs.ext4 /dev/mapper/rootvg-lv\_data

# mkdir /data

# echo "/dev/mapper/rootvg-lv\_data /data ext4 defaults 1 0" >> /etc/fstab

# mount /data

# mkdir /data/mysql

# chown -R mysql:mysql /data/mysql/

1. 启动MYSQL服务，设置开机启动

# service mysqld start

# chkconfig mysqld on

1. 修改root用户密码为openstack

# mysqladmin -uroot password 'openstack';history -c

### Qpid安装配置

1. 安装qpid

# yum install -y qpid-cpp-server memcached

1. 修改/etc/qpidd.conf配置文件,将auth设置为no

# vi /etc/qpidd.conf

……

auth=no

1. 启动qpid服务，设置开机自启动

# service qpidd start

# chkconfig qpidd on

### 安装OpenStack工具包

# yum install -y openstack-utils ###自行解决依赖包

## Keystone安装配置

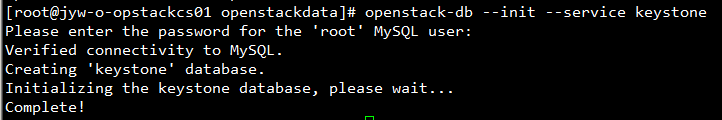
### 初始化Keystone

1. 安装keystone

# yum install -y openstack-keystone ###自行解决依赖包

1. 创建keystone数据库，修改配置文件中的数据库链接

# openstack-db --init --service keystone



# openstack-config --set /etc/keystone/keystone.conf sql connection mysql://keystone:keystone@localhost/keystone

如果有端口指定，使用这条####openstack-config --set /etc/keystone/keystone.conf sql

connection mysql://keystone:keystone@192.168.5.228:3324/keystone //指定数据库端口

1. 使用openssl随即生成一个令牌，将其存储在配置文件中

# export SERVICE\_TOKEN=$(openssl rand -hex 10) //随机生成SERVICE\_TOKEN值，请牢记，保存到文件中

# export SERVICE\_ENDPOINT=http://111.204.241.184:35357/v2.0 //指定链接

# mkdir /root/work //创建目录，用户保存密码文件等

# echo $SERVICE\_TOKEN > /root/work/ks\_admin\_token //保存上面生成的随机密码，以后有用

# cat /root/work/ks\_admin\_token

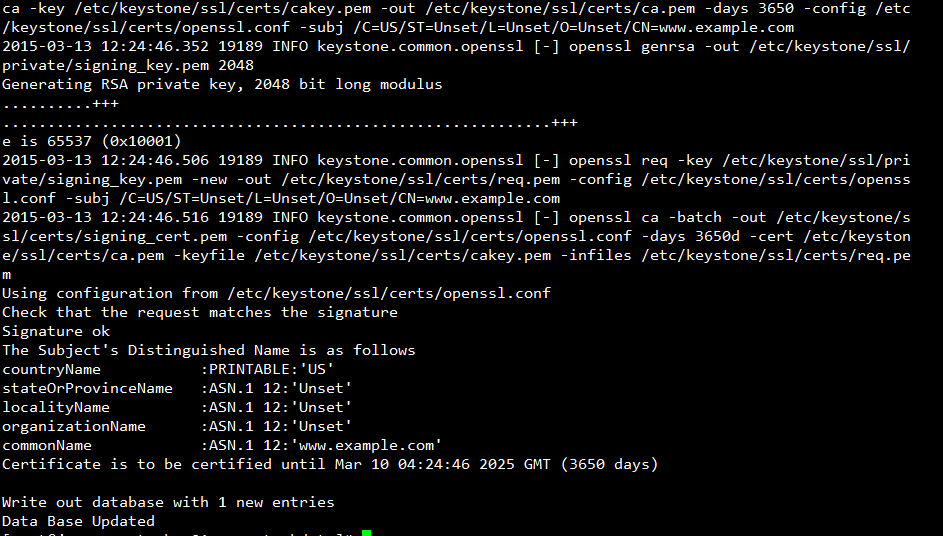
503fbb34e6e0c873efa4

# openstack-config --set /etc/keystone/keystone.conf DEFAULT admin\_token $SERVICE\_TOKEN //将密码写入文件中

注：将生成的SERVICE\_TOKEN值写入文件中保存，以备后续使用，后面涉及到的SERVICE\_TOKEN值都是在ks\_admin\_token文件中获取的。所以一旦写入文件，不要再次运行命令生成SERVICE\_TOKEN，否则前后不一致会为调试带来麻烦。

1. 默认情况下keysonte使用PKI令牌。创建签名密钥和证书。

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



# chown -R keystone:keystone /etc/keystone/\* /var/log/keystone/keystone.log //使用root操作，会是root.keystone

1. 启动keystone服务，设置开机自启动

# service openstack-keystone start // # /etc/init.d/openstack-keystone start

# chkconfig openstack-keystone on

### 定义Users、Tenants and Roles

1. 修改.bash\_profile文件，添加以下参数

[root@hctrl ~]# vi .bash\_profile

……

export OS\_USERNAME=admin

export OS\_TENANT\_NAME=admin

export OS\_PASSWORD=soufun.com

export OS\_AUTH\_URL=http:// 111.204.241.184:5000/v2.0

export SERVICE\_ENDPOINT=http:// 111.204.241.184:35357/v2.0

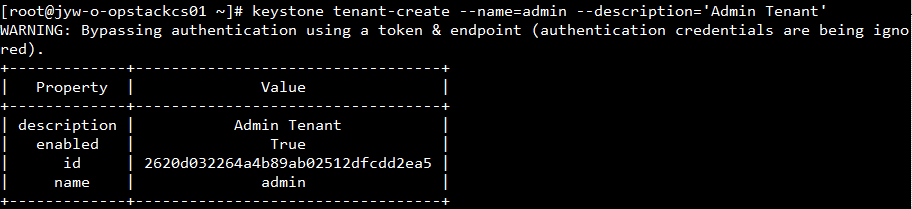
export SERVICE\_TOKEN=503fbb34e6e0c873efa4

执行以下命令使变量即时生效：

[root@hctrl ~]# source .bash\_profile

1. 为管理员用户创建一个tenant，为openstack其他服务的用户创建一个tenant

# keystone tenant-create --name=admin --description='Admin Tenant'

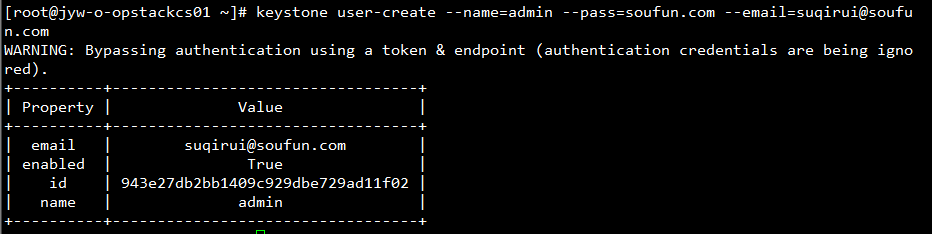


# keystone tenant-create --name=service --description='Service Tenant'



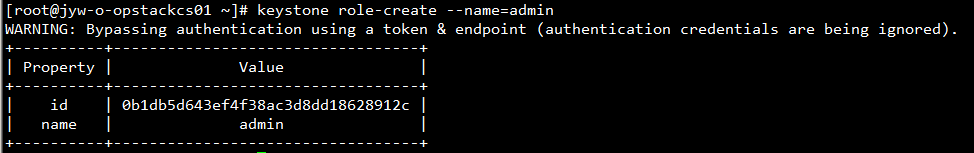
1. 创建一个管理员用户admin，去管理他

# keystone user-create --name=admin --pass=secrete --email=admin@example.com



1. 创建一个管理角色admin

# keystone role-create --name=admin



1. 将角色添加到用户

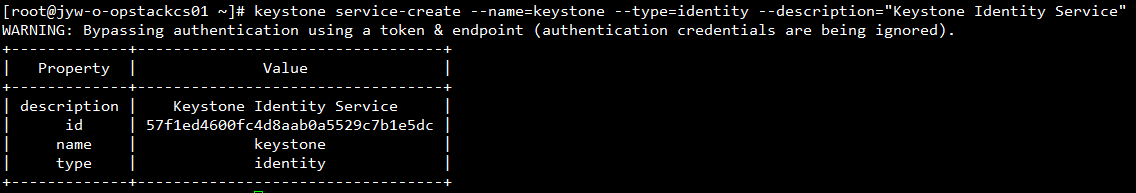
# keystone user-role-add --user=admin --tenant=admin --role=admin



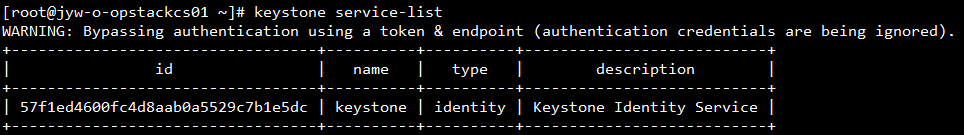
### 定义Services 和 API Endpoints

1. 为keystone创建一个服务

# keystone service-create --name=keystone --type=identity --description="Keystone Identity Service"



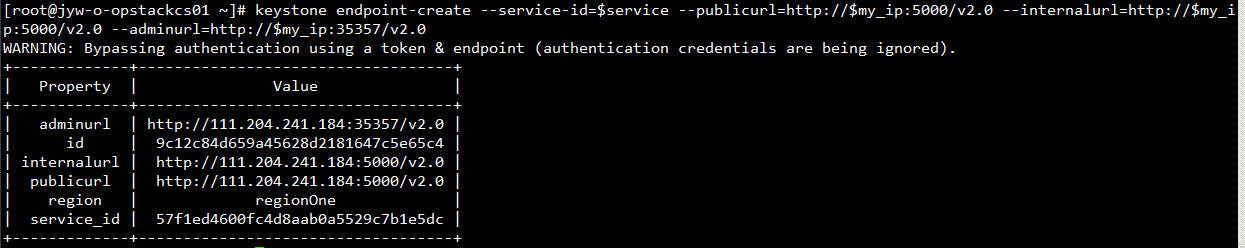
1. 使用服务ID创建一个endpoint



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

# my\_ip= 111.204.241.184

# keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:5000/v2.0 --internalurl=http://$my\_ip:5000/v2.0 --adminurl=http://$my\_ip:35357/v2.0



## Glance安装配置

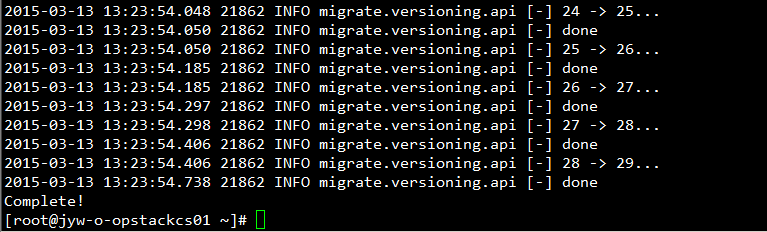
### 初始化Glance

1. 安装glance

# yum install -y openstack-glance

1. 创建glance数据库，修改配置文件中的数据库链接

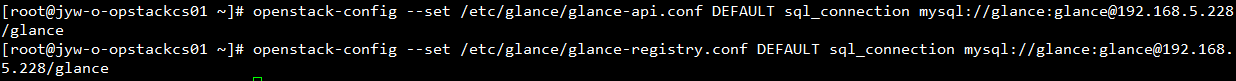
# openstack-db --init --service glance



# openstack-config --set /etc/glance/glance-api.conf DEFAULT sql\_connection mysql://glance:glance@localhost/glance



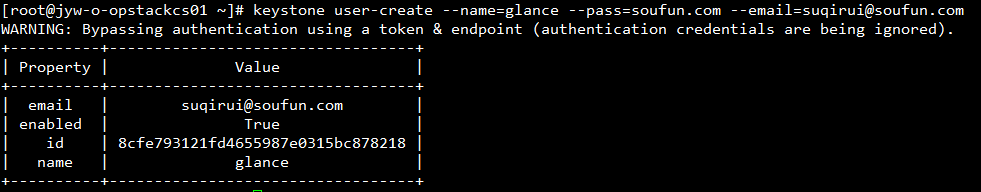
# openstack-config --set /etc/glance/glance-registry.conf DEFAULT sql\_connection mysql://glance:glance@localhost/glance



### 创建User，定义Services 和 API Endpoints

1. 在keystone中为镜像服务创建一个glance用户

# keystone user-create --name=glance --pass=service [--email=glance@example.com](mailto:--email=glance@example.com)

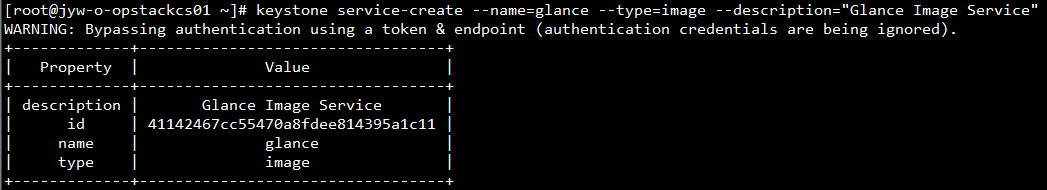


# keystone user-role-add --user=glance --tenant=service --role=admin

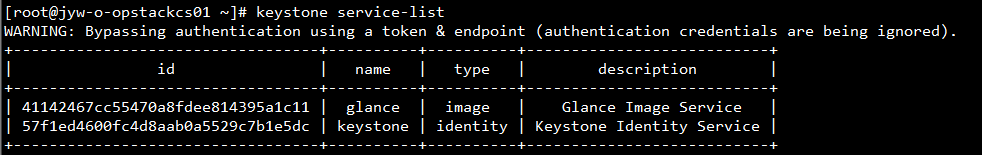


1. 为glance创建一个服务

# keystone service-create --name=glance --type=image --description="Glance Image Service"



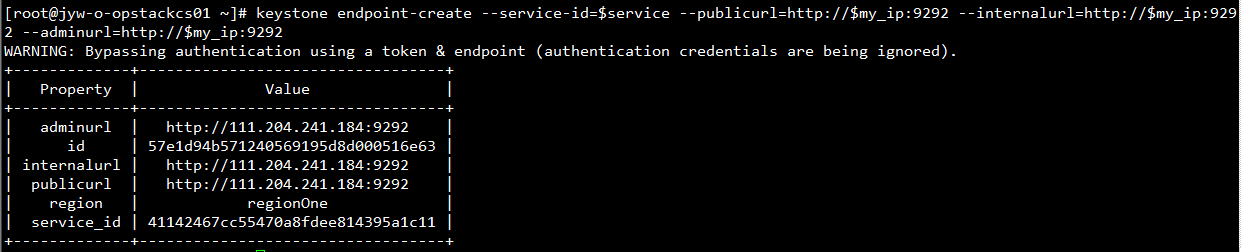
1. 使用服务ID创建一个endpoint



# my\_ip=172.21.2.110

# service=$(keystone service-list | awk '/glance/ {print $2}')

# keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:9292 --internalurl=http://$my\_ip:9292 --adminurl=http://$my\_ip:9292



### 配置Glance服务

1. 将keystone认证信息添加到glance配置文件中

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken auth\_host 111.204.241.184

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

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

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

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

# openstack-config --set /etc/glance/glance-api.conf keystone\_authtoken admin\_password soufun.com

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_host 111.204.241.184

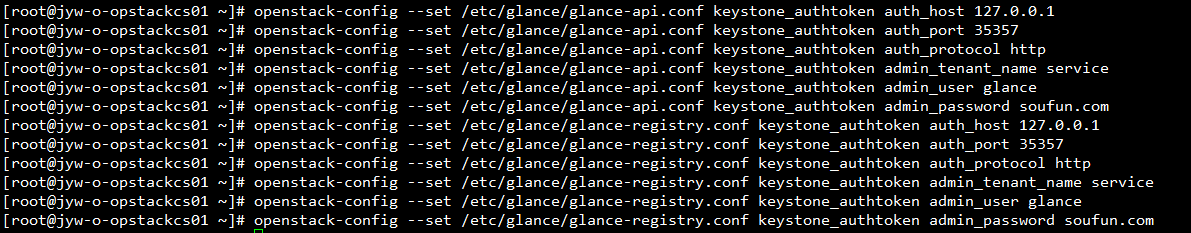
# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken auth\_port 35357

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

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

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

# openstack-config --set /etc/glance/glance-registry.conf keystone\_authtoken admin\_password soufun.com



1. 修改ini文件路径，将keystone认证信息添加到ini文件中

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

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

# openstack-config --set /etc/glance/glance-registry.conf paste\_deploy config\_file /etc/glance/glance-registry-paste.ini

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

# cp /usr/share/glance/glance-api-dist-paste.ini /etc/glance/glance-api-paste.ini

# cp /usr/share/glance/glance-registry-dist-paste.ini /etc/glance/glance-registry-paste.ini

# chown -R root:glance /etc/glance/glance-api-paste.ini

# chown -R root:glance /etc/glance/glance-registry-paste.ini

# openstack-config --set /etc/glance/glance-api-paste.ini filter:authtoken auth\_host 111.204.241.184

# openstack-config --set /etc/glance/glance-api-paste.ini filter:authtoken admin\_tenant\_name service

# openstack-config --set /etc/glance/glance-api-paste.ini filter:authtoken admin\_user glance

# openstack-config --set /etc/glance/glance-api-paste.ini filter:authtoken admin\_password soufun.com

# openstack-config --set /etc/glance/glance-registry-paste.ini filter:authtoken auth\_host 111.204.241.184

# openstack-config --set /etc/glance/glance-registry-paste.ini filter:authtoken admin\_tenant\_name service

# openstack-config --set /etc/glance/glance-registry-paste.ini filter:authtoken admin\_user glance

# openstack-config --set /etc/glance/glance-registry-paste.ini filter:authtoken admin\_password soufun.com

1. 修改镜像文件的存放路径（默认存放在/var/lib/glance目录下，若不需修改，此步骤可省略）

# openstack-config --set /etc/glance/glance-api.conf DEFAULT filesystem\_store\_datadir /data/openstack/lib/glance/images

# openstack-config --set /etc/glance/glance-api.conf DEFAULT scrubber\_datadir /data/openstack/lib/glance/scrubber

# openstack-config --set /etc/glance/glance-api.conf DEFAULT image\_cache\_dir /data/openstack/lib/glance/image-cache

# mkdir /openstack/lib

# cp -r /var/lib/glance/ /openstack/lib/

# chown -R glance:glance /openstack/lib/glance/

1. 启动glance服务，设置开机自启动

# service openstack-glance-api start

# service openstack-glance-registry start

/etc/init.d/openstack-glance-scrubber restart

# chkconfig openstack-glance-api on

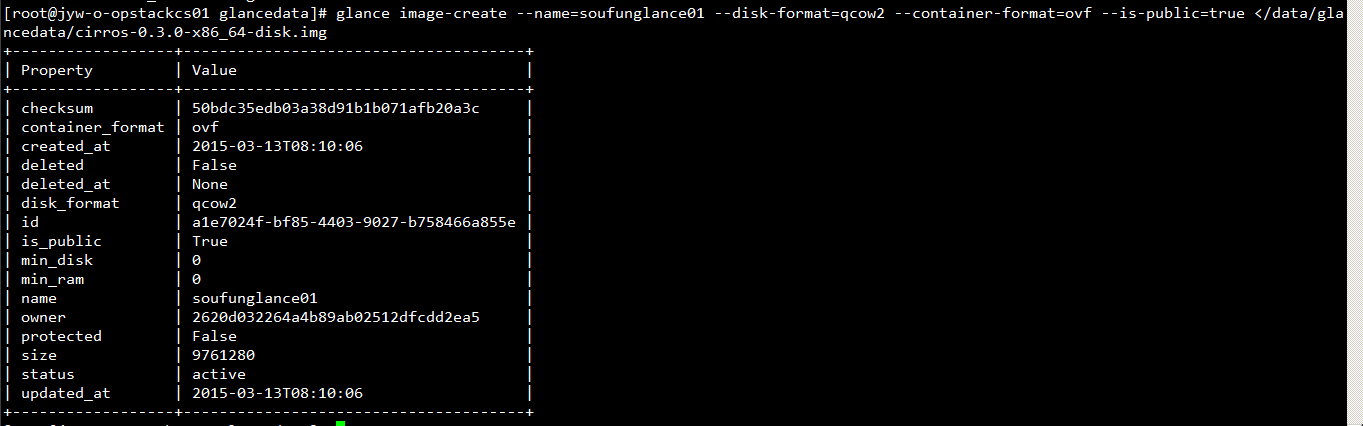
# chkconfig openstack-glance-registry on

for s in openstack-glance-{api,registry,scrubber}; do chkconfig $s on;done

### Glance测试

1. 上传镜像

# glance image-create --name=rhel5.6 --disk-format=qcow2 --container-format=ovf --is-public=true < /media/images/rhelos56.img



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

| Property | Value |

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

| checksum | d54e7667037ece1b64b86d7ea53316b2 |

| container\_format | ovf |

| created\_at | 2013-10-24T06:58:37 |

| deleted | False |

| deleted\_at | None |

| disk\_format | qcow2 |

| id | ee9ee181-e717-4c8b-83c5-afb44d96f301 |

| is\_public | True |

| min\_disk | 0 |

| min\_ram | 0 |

| name | rhel5.6 |

| owner | 4abbad801b1b4bbe89903f4a48c38c58 |

| protected | False |

| size | 1855979520 |

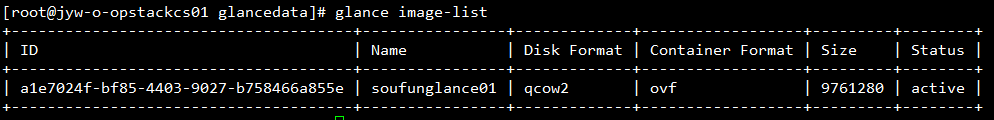
| status | active |

| updated\_at | 2013-10-24T06:58:48 |

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

1. 查看镜像

# glance image-list



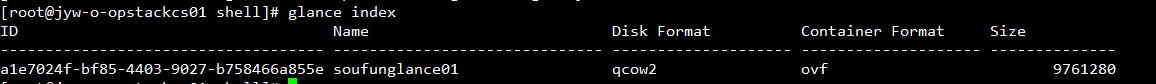
+--------------------------------------+---------+-------------+------------------+------------+--------+

| ID | Name | Disk Format | Container Format | Size | Status |

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

| ee9ee181-e717-4c8b-83c5-afb44d96f301 | rhel5.6 | qcow2 | ovf | 1855979520 | active |

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



## Nova安装配置

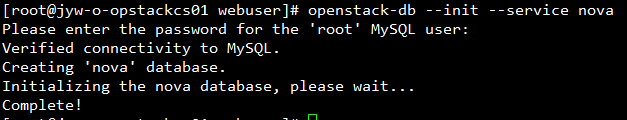
### 初始化Nova

1. 安装nova

# yum install -y openstack-nova

1. 创建nova数据库

# openstack-db --init --service nova //耐心等待，时间较长



### 创建User，定义Services 和 API Endpoints

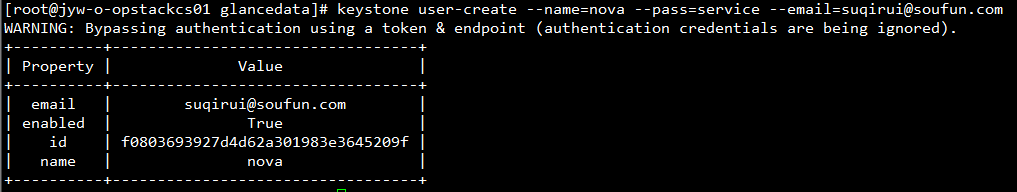
1. 编写脚本

# vi /root/work/nova-user.sh

#!/bin/sh

my\_ip=172.21.2.110

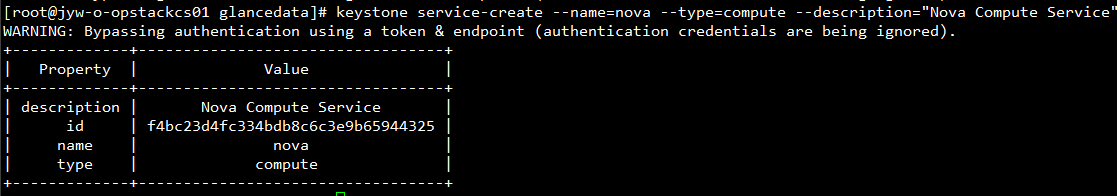
keystone user-create --name=nova --pass=service [--email=suqirui@soufun.com](mailto:--email=suqirui@soufun.com)



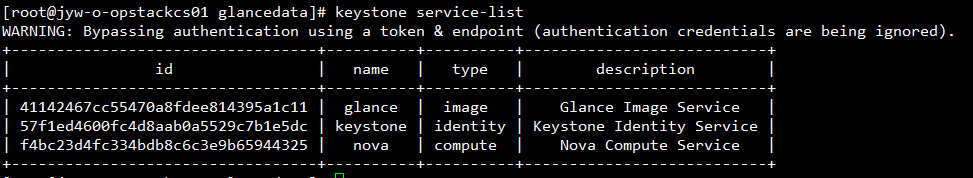
keystone user-role-add --user=nova --tenant=service --role=admin



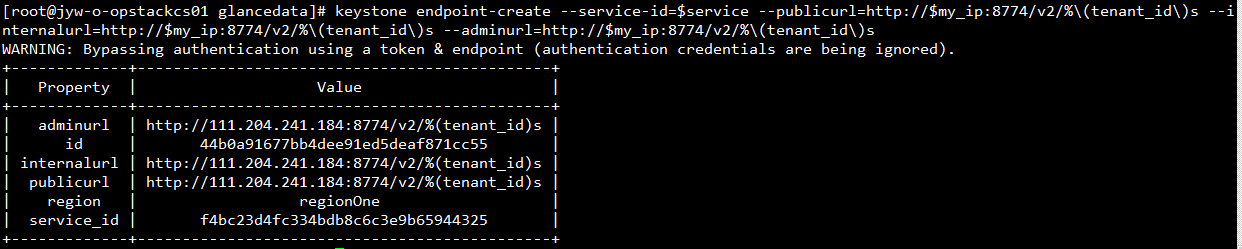
keystone service-create --name=nova --type=compute --description="Nova Compute Service"



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



keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:8774/v2/%\(tenant\_id\)s --internalurl=http://$my\_ip:8774/v2/%\(tenant\_id\)s --adminurl=http://$my\_ip:8774/v2/%\(tenant\_id\)s



1. 运行脚本，创建用户、服务及api endpoint

# sh /root/work/nova-user.sh

### 配置Nova服务

1. 修改nova.conf配置文件

# vi /etc/nova/nova.conf

[DEFAULT]

my\_ip = 111.204.241.184

auth\_strategy = keystone

state\_path = /data/openstack/lib/nova

verbose=True

allow\_resize\_to\_same\_host = true

rpc\_backend = nova.openstack.common.rpc.impl\_qpid

qpid\_hostname = 111.204.241.184

libvirt\_type = kvm

 ­­

vncserver\_listen = $my\_ip

vncserver\_proxyclient\_address = $my\_ip

network\_manager = nova.network.manager.FlatDHCPManager

firewall\_driver = nova.virt.libvirt.firewall.IptablesFirewallDriver

multi\_host = True

flat\_interface = eth1

flat\_network\_bridge = br100

public\_interface = eth0

[hyperv]

[zookeeper]

[osapi\_v3]

[conductor]

[keymgr]

[cells]

[database]

[image\_file\_url]

[baremetal]

[rpc\_notifier2]

[matchmaker\_redis]

[ssl]

[trusted\_computing]

[upgrade\_levels]

[matchmaker\_ring]

[vmware]

[spice]

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_user = nova

admin\_tenant\_name = service

admin\_password = service

因上述配置文件中修改了instance实例存放的位置，还需要作一下操作：

修改路径，设置目录权限：

# cp -r /var/lib/nova/ /openstack/lib/

# chown -R nova:nova /openstack/lib/nova/

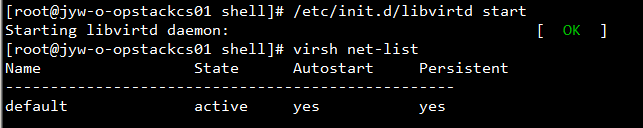
1. 配置libvirtd服务，删除virbr0

启动libvirt服务：

# service libvirtd start

查看net-list，发现default：

# virsh net-list



Name State Autostart Persistent

--------------------------------------------------

default active yes yes

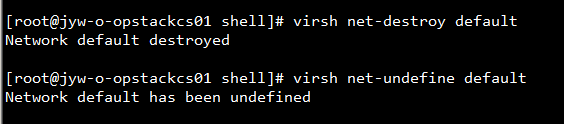
删除default，即virbr0：

# virsh net-destroy default

Network default destroyed

# virsh net-undefine default

Network default has been undefined



重启libvirtd服务，设置开机自启动：

# service libvirtd restart

# chkconfig libvirtd on

1. 启动nova相关服务，设置开机自启动

# service messagebus start

# chkconfig messagebus on

启动nova服务，此处采用network网络服务：

# service openstack-nova-api start

# service openstack-nova-cert start

# service openstack-nova-consoleauth start

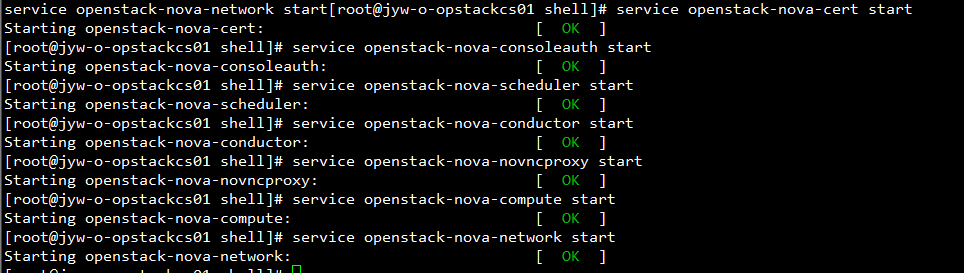
# service openstack-nova-scheduler start

# service openstack-nova-conductor start

# service openstack-nova-novncproxy start

# service openstack-nova-compute start

# service openstack-nova-network start



开机启动项，按上述服务进行设置即可。因后面需要用neutron代替network服务，此处暂不设置启动项。

### Nova测试

本节使用nova-network网络服务进行测试：

1. 创建网络

[root@jyw-o-opstackcs01 img]# nova network-create vmnet --fixed-range-v4=111.204.241.0/25 --bridge-interface=br-int --multi-host=T

ERROR: The server has either erred or is incapable of performing the requested operation. (HTTP 500) (Request-ID: req-721e0d41-5bfe-4c86-bed6-20a6dbfd3333)

//出现此错误，通过修改数据库nova权限，授予nova权限

//修改network\_api\_class=nova.network.api.API

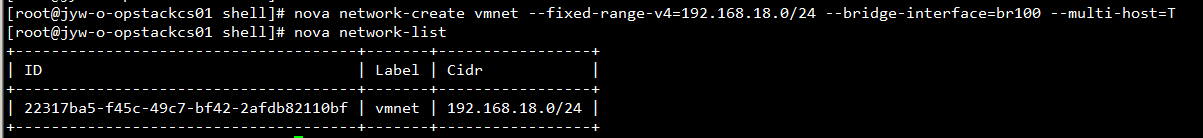
//In order to run "nova network-create vmnet --fixed-range-v4=10.0.0.0/24 --bridge-interface=br100 --multi-host=T",  
I change nova controller's default nova.conf option "network\_api\_class=nova.network.neutronv2.api.API" to "network\_api\_class=nova.network.api.API".

# nova network-create vmnet --fixed-range-v4=10.0.0.0/24 --bridge-interface=br100 --multi-host=T

//地址池必须是给你分配，可以连通的，否则就只是展示而已



# nova network-list //指定地址池



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

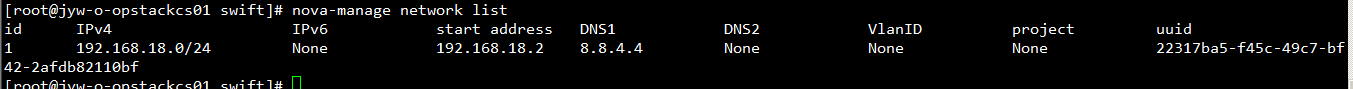
| ID | Label | Cidr |

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

| 148719f9-5b23-450b-9d60-b5376de45e79 | vmnet | 10.0.0.0/24 |

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

# nova-manage network list



id IPv4 IPv6 start address DNS1 DNS2 VlanID project uuid

1 10.0.0.0/24 None 10.0.0.2 8.8.4.4 None None None 148719f9-5b23-450b-9d60-b5376de45e79

1. 设置安全组

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

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

| IP Protocol | From Port | To Port | IP Range | Source Group |

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

| tcp | 22 | 22 | 0.0.0.0/0 | |

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

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

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

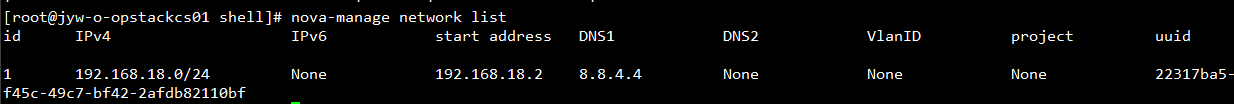
| IP Protocol | From Port | To Port | IP Range | Source Group |

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

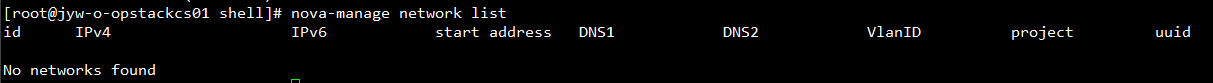
| icmp | -1 | -1 | 0.0.0.0/0 | |

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

# nova-manage network list



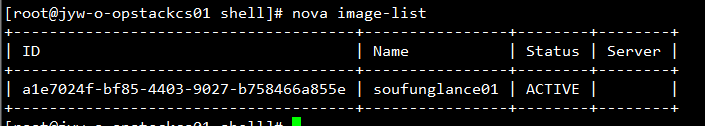
id IPv4 IPv6 start address DNS1 DNS2 VlanID project uuid



1. 创建虚拟机实例

查看可用镜像：

# nova image-list



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

| ID | Name | Status | Server |

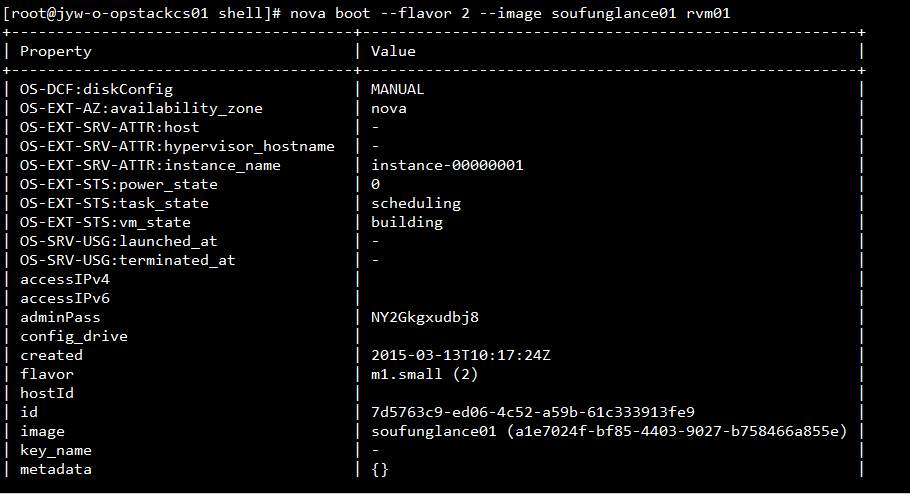
+--------------------------------------+---------+--------+--------+

| ee9ee181-e717-4c8b-83c5-afb44d96f301 | rhel5.6 | ACTIVE | |

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

创建虚拟机实例：

# nova boot --flavor 2 --image rhel5.6 rvm01



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

| Property | Value |

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

| OS-EXT-STS:task\_state | scheduling |

| image | rhel5.6 |

| OS-EXT-STS:vm\_state | building |

| OS-EXT-SRV-ATTR:instance\_name | instance-00000001 |

| OS-SRV-USG:launched\_at | None |

| flavor | m1.small |

| id | fcd6db96-947f-46e7-8978-d4c884b6552e |

| security\_groups | [{u'name': u'default'}] |

| user\_id | 3f59ffb7df79404c91d99fcdd0ee70fe |

| OS-DCF:diskConfig | MANUAL |

| accessIPv4 | |

| accessIPv6 | |

| progress | 0 |

| OS-EXT-STS:power\_state | 0 |

| OS-EXT-AZ:availability\_zone | nova |

| config\_drive | |

| status | BUILD |

| updated | 2013-10-24T08:07:34Z |

| hostId | |

| OS-EXT-SRV-ATTR:host | None |

| OS-SRV-USG:terminated\_at | None |

| key\_name | None |

| OS-EXT-SRV-ATTR:hypervisor\_hostname | None |

| name | rvm01 |

| adminPass | rzMGQ7pyykFT |

| tenant\_id | 4abbad801b1b4bbe89903f4a48c38c58 |

| created | 2013-10-24T08:07:34Z |

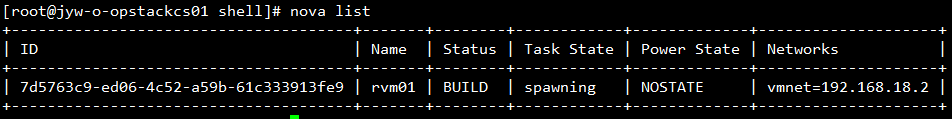
| os-extended-volumes:volumes\_attached | [] |

| metadata | {} |

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

 查看虚拟机实例运行状态：

# nova list



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

| ID | Name | Status | Task State | Power State | Networks |

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

| fcd6db96-947f-46e7-8978-d4c884b6552e | rvm01 | ACTIVE | None | Running | vmnet=10.0.0.2 |

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

测试虚拟机实例的连通性：

# ping 10.0.0.2

PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

64 bytes from 10.0.0.2: icmp\_seq=1 ttl=64 time=0.233 ms

64 bytes from 10.0.0.2: icmp\_seq=2 ttl=64 time=0.223 ms

^C

--- 10.0.0.2 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1199ms

rtt min/avg/max/mdev = 0.223/0.228/0.233/0.005 ms

## Horizon安装配置

1. 安装horizon

# yum install -y openstack-dashboard

1. 修改local\_settings文件

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

DEBUG = True

……

1. 启动http服务，设置开机自启动

# service httpd start

# chkconfig httpd on

1. 重启nova-api服务

# service openstack-nova-api restart

1. 添加防火墙策略

# iptables -I INPUT -p tcp --dport 80 -j ACCEPT

# iptables -I INPUT -p tcp -m multiport --dports 5900:6000 -j ACCEPT

# iptables -I INPUT -p tcp --dport 6080 -j ACCEPT

# service iptables save

Horizon安装完成，使用<http://172.21.2.110/dashboard>访问openstack web界面。

## Neutron安装配置

### 初始化Neutron

1. 安装neutron

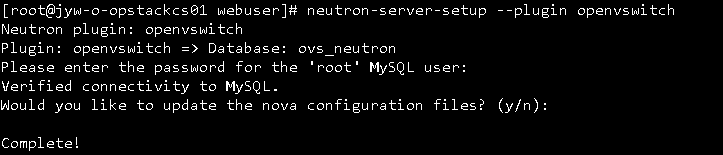
# yum install -y openstack-neutron

1. 安装openvswitch插件

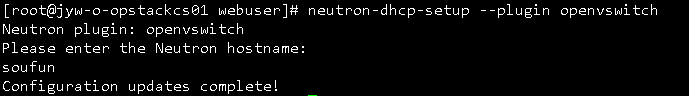
# yum install -y openstack-neutron-openvswitch

1. 创建数据库，配置openvswitch插件

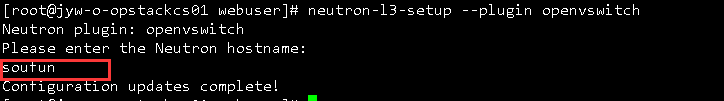
# neutron-server-setup --plugin openvswitch



# neutron-dhcp-setup --plugin openvswitch



# neutron-l3-setup --plugin openvswitch



### 创建User，定义Services 和 API Endpoints

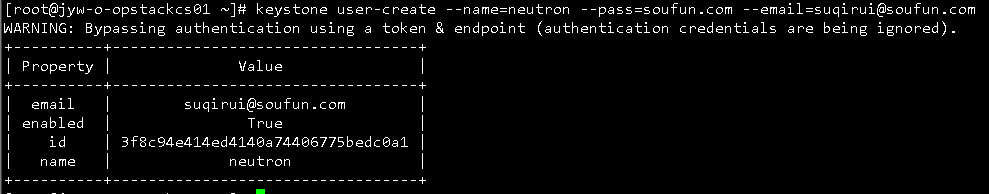
1. 编写脚本

# vi /root/work/neutron-user.sh

#!/bin/sh

my\_ip=172.21.2.110

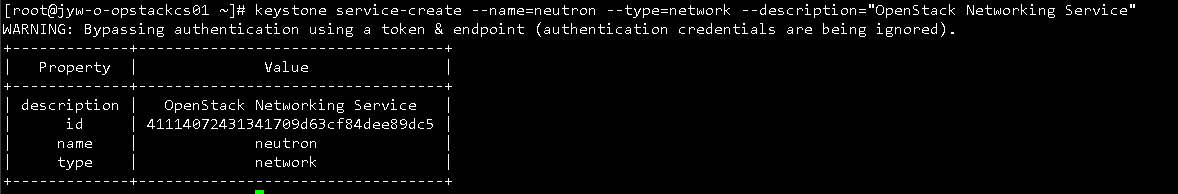
keystone user-create --name=neutron --pass=service [--email=neutron@example.com](mailto:--email=neutron@example.com)



keystone user-role-add --user=neutron --tenant=service --role=admin

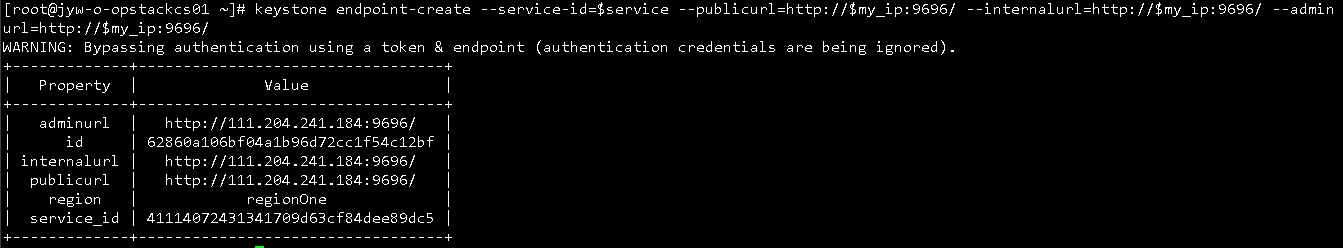


keystone service-create --name=neutron --type=network --description="OpenStack Networking Service"



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

keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:9696/ --internalurl=http://$my\_ip:9696/ --adminurl=http://$my\_ip:9696/



1. 运行脚本，创建用户、服务及api endpoint

# sh /root/work/neutron-user.sh

### 配置网络服务

1. 修改api-paste.ini文件

在api-paste.ini文件中添加keystone认证信息：

# vi /etc/neutron/api-paste.ini

……

[filter:authtoken]

paste.filter\_factory = keystoneclient.middleware.auth\_token:filter\_factory

auth\_host=127.0.0.1

admin\_user=neutron

admin\_tenant\_name=service

admin\_password=service

1. 配置neutron.conf文件

# cat /etc/neutron/neutron.conf | grep -v ^# | grep -v ^$

[DEFAULT]

debug = True

core\_plugin = neutron.plugins.openvswitch.ovs\_neutron\_plugin.OVSNeutronPluginV2

auth\_strategy = keystone

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

qpid\_hostname = localhost

[quotas]

[agent]

root\_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = service

[database]

connection = mysql://neutron:neutron@localhost/ovs\_neutron

[service\_providers]

1. 修改ovs\_neutron\_plugin.ini文件

# cat /etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini | grep -v ^# | grep -v ^$

[ovs]

tenant\_network\_type = vlan

network\_vlan\_ranges = eth1:1000:2999

integration\_bridge = br-int

bridge\_mappings = eth1:br-int

[agent]

[securitygroup]

[DATABASE]

sql\_connection = mysql://neutron:neutron@localhost/ovs\_neutron

[SECURITYGROUP]

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

1. 修改dhcp及l3\_agent相关配置文件

# cat /etc/neutron/dhcp\_agent.ini | grep -v ^# | grep -v ^$

[DEFAULT]

auth\_url = <http://127.0.0.1:5000/v2.0>

admin\_username = neutron

admin\_password = service

admin\_tenant\_name = service

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

use\_namespaces = True

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

# cat /etc/neutron/l3\_agent.ini | grep -v ^# | grep -v ^$

[DEFAULT]

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

external\_network\_bridge = br-int

# cat /etc/neutron/metadata\_agent.ini | grep -v ^# | grep -v ^$

[DEFAULT]

auth\_url = <http://127.0.0.1:5000/v2.0>

auth\_region = regionOne

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = service

1. 修改/etc/libvirt/qemu.conf

在文件末尾添加以下参数：

# vi /etc/libvirt/qemu.conf

……

cgroup\_device\_acl = [

"/dev/null", "/dev/full", "/dev/zero",

"/dev/random", "/dev/urandom",

"/dev/ptmx", "/dev/kvm", "/dev/kqemu",

"/dev/rtc","/dev/hpet","/dev/net/tun",

]

clear\_emulator\_capabilities = 0

user = "root"

group = "root"

重新启动libvirtd服务

# service libvirtd restart

1. 修改nova.conf文件

# cat /etc/nova/nova.conf

[DEFAULT]

my\_ip = 172.21.2.110

auth\_strategy = keystone

state\_path = /openstack/lib/nova

verbose=True

allow\_resize\_to\_same\_host = true

rpc\_backend = nova.openstack.common.rpc.impl\_qpid

qpid\_hostname = 172.21.2.110

libvirt\_type = kvm

glance\_api\_servers = 172.21.2.110:9292

novncproxy\_base\_url = <http://172.21.2.110:6080/vnc_auto.html>

vncserver\_listen = 172.21.2.110

vncserver\_proxyclient\_address = 172.21.2.110

vnc\_enabled = true

vnc\_keymap = en-us

network\_manager = nova.network.manager.FlatDHCPManager

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

multi\_host = True

flat\_interface = eth1

flat\_network\_bridge = br100

public\_interface = eth0

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

neutron\_admin\_username = neutron

neutron\_admin\_password = service

neutron\_admin\_auth\_url = <http://127.0.0.1:5000/v2.0>

neutron\_auth\_strategy = keystone

neutron\_admin\_tenant\_name = service

neutron\_url = <http://localhost:9696/>

security\_group\_api = neutron

libvirt\_vif\_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver

### libvirt\_vif\_driver=nova.virt.libvirt.vif.LibvirtGenericVIFDriver

neutron\_ovs\_bridge = br-int

[hyperv]

[zookeeper]

[osapi\_v3]

[conductor]

[keymgr]

[cells]

[database]

[image\_file\_url]

[baremetal]

[rpc\_notifier2]

[matchmaker\_redis]

[ssl]

[trusted\_computing]

[upgrade\_levels]

[matchmaker\_ring]

[vmware]

[spice]

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_user = nova

admin\_tenant\_name = service

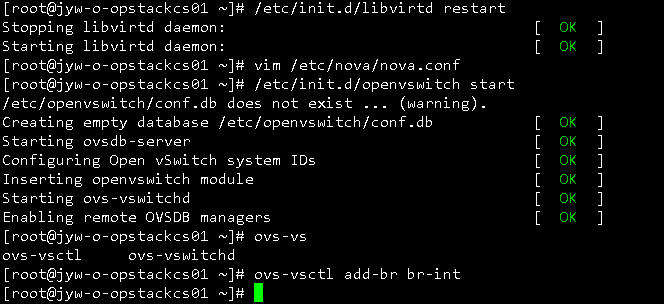
admin\_password = service

1. 启动openvswitch服务，添加br-int

# service openvswitch start

# chkconfig openvswitch on

# ovs-vsctl add-br br-int



1. 停止nova-network，启动neutron，重启nova服务

# service openstack-nova-network stop

# service neutron-server start

# service neutron-dhcp-agent start

# service neutron-openvswitch-agent start

# service openstack-nova-api restart

# service openstack-nova-cert restart

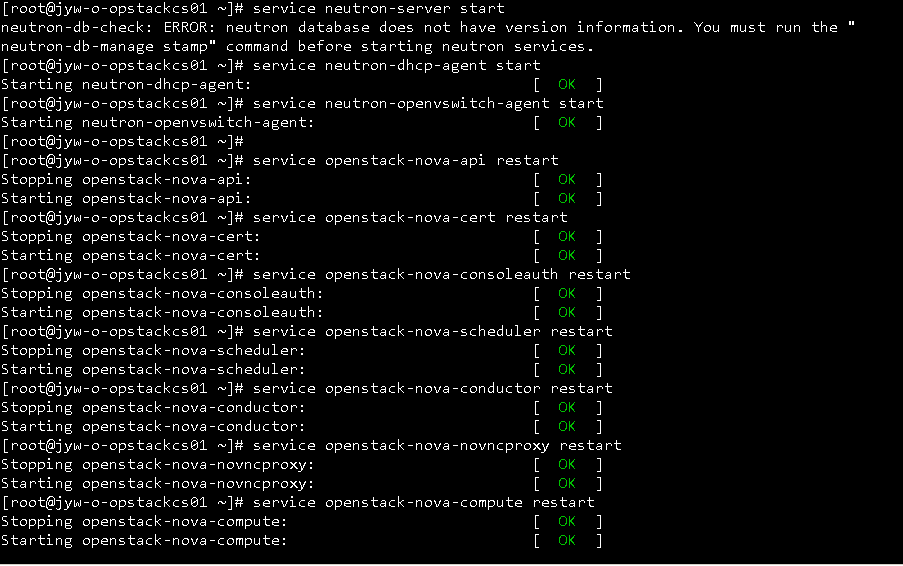
# service openstack-nova-consoleauth restart

# service openstack-nova-scheduler restart

# service openstack-nova-conductor restart

# service openstack-nova-novncproxy restart

# service openstack-nova-compute restart



# neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugin.ini stamp Havana

############# neutron-db-manage --config-file /usr/share/neutron/neutron-dist.conf --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugin.ini stamp head

# service neutron-server start

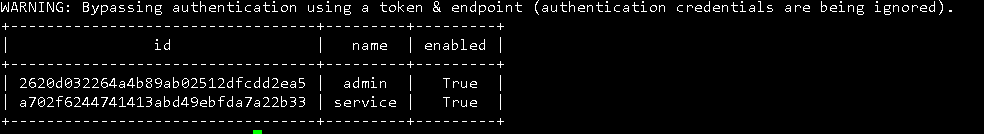


### Neutron测试

1. 创建网络

查看tenant：

# keystone tenant-list



WARNING: Bypassing authentication using a token & endpoint (authentication credentials are being ignored).

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

| id | name | enabled |

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

| 4abbad801b1b4bbe89903f4a48c38c58 | admin | True |

| 876f3a86b6a145679b61f6aaf863f9c4 | service | True |

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

创建网络：

# neutron net-create --tenant-id 4abbad801b1b4bbe89903f4a48c38c58 sharednet1 --shared --provider:network\_type vlan --provider:physical\_network eth1 --provider:segmentation\_id 1024

##################neutron net-create --tenant-id 861eee7d92ea48aeb289f1ce1ce3d946 net-ex --shared --provider:network\_type flat --provider:physical\_network eth1 --router:external True ###支持浮动IP --router:external True

[root@jyw-o-opstackcs01 nova]# neutron net-create --tenant-id 63a2c21f54364dafb0548c6875937bb6 net-ex --shared --provider:network\_type vlan --provider:physical\_network eth1 --provider:segmentation\_id 1024

400-{u'NeutronError': {u'message': u'Invalid input for operation: Unknown provider:physical\_network eth1.', u'type': u'InvalidInput', u'detail': u''}} 此处错误为配置文件中配置错误，具体位置如下：

/etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini:network\_vlan\_ranges = physnet1:1000:2999

/etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini:# Example: network\_vlan\_ranges = physnet1:1000:2999

/etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini:bridge\_mappings = physnet1:br-int

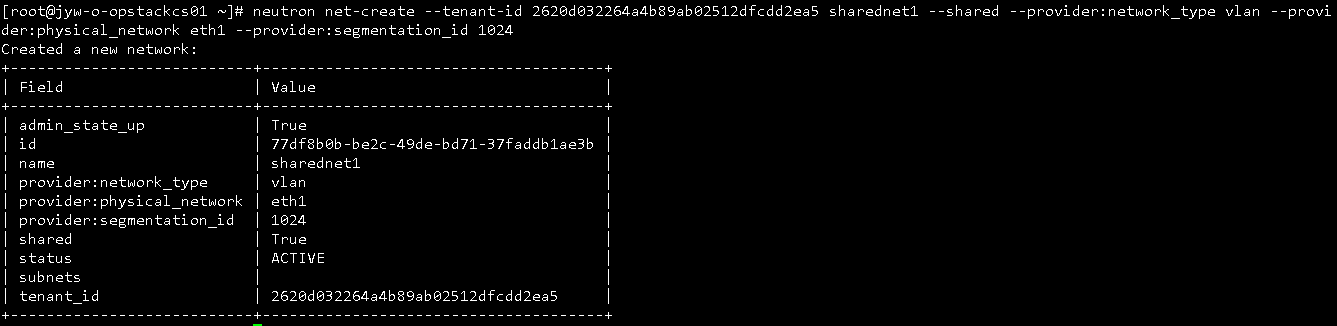
/etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini:# Example: bridge\_mappings = physnet1:br-eth1

/etc/neutron/plugin.ini:network\_vlan\_ranges = physnet1:1000:2999

/etc/neutron/plugin.ini:# Example: network\_vlan\_ranges = physnet1:1000:2999

/etc/neutron/plugin.ini:bridge\_mappings = physnet1:br-int

/etc/neutron/plugin.ini:# Example: bridge\_mappings = physnet1:br-eth1



Created a new network:

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

| Field | Value |

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

| admin\_state\_up | True |

| id | cd33634c-ae3b-40be-9ab6-2724e49d39a2 |

| name | sharednet1 |

| provider:network\_type | vlan |

| provider:physical\_network | eth1 |

| provider:segmentation\_id | 1024 |

| shared | True |

| status | ACTIVE |

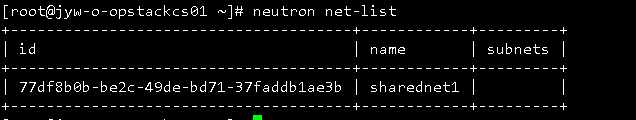
| subnets | |

| tenant\_id | 4abbad801b1b4bbe89903f4a48c38c58 |

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

查看网络信息：

# neutron net-list



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

| id | name | subnets |

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

| cd33634c-ae3b-40be-9ab6-2724e49d39a2 | sharednet1 | |

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

// Neutron net-delete name 删除网络

1. 创建子网

创建子网：

# neutron subnet-create --ip\_version 4 --tenant-id 4abbad801b1b4bbe89903f4a48c38c58 sharednet1 20.0.0.0/24



Created a new subnet:

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

| Field | Value |

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

| allocation\_pools | {"start": "20.0.0.2", "end": "20.0.0.254"} |

| cidr | 20.0.0.0/24 |

| dns\_nameservers | |

| enable\_dhcp | True |

| gateway\_ip | 20.0.0.1 |

| host\_routes | |

| id | 1def1908-60f4-469e-a1ae-f97621e7af40 |

| ip\_version | 4 |

| name | |

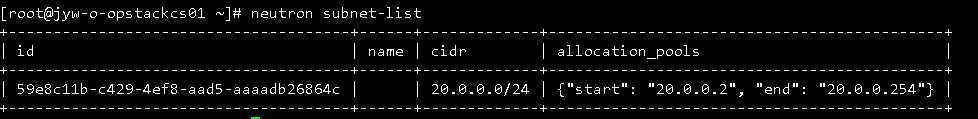
| network\_id | cd33634c-ae3b-40be-9ab6-2724e49d39a2 |

| tenant\_id | 4abbad801b1b4bbe89903f4a48c38c58 |

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

查看子网信息：

# neutron subnet-list



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

| id | name | cidr | allocation\_pools |

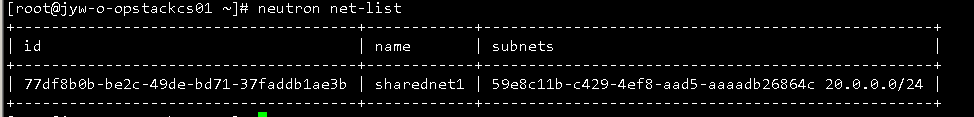
+--------------------------------------+------+-------------+--------------------------------------------+

| 1def1908-60f4-469e-a1ae-f97621e7af40 | | 20.0.0.0/24 | {"start": "20.0.0.2", "end": "20.0.0.254"} |

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

查看网络信息：

# neutron net-list



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

| id | name | subnets |

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

| cd33634c-ae3b-40be-9ab6-2724e49d39a2 | sharednet1 | 1def1908-60f4-469e-a1ae-f97621e7af40 20.0.0.0/24 |

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

1. 设置安全组

查看默认安全组：

# neutron security-group-list



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

| id | name | description |

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

| b2917b6f-5cf6-4e3d-b2e7-fb0a2ac4f78c | default | default |

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

设置tcp 规则：

# neutron security-group-rule-create --protocol tcp --port-range-min 22 --port-range-max 22 --direction ingress default



Created a new security\_group\_rule:

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

| Field | Value |

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

| direction | ingress |

| ethertype | IPv4 |

| id | 0cf0241d-d2a8-48b3-9756-c27ad4794c00 |

| port\_range\_max | 22 |

| port\_range\_min | 22 |

| protocol | tcp |

| remote\_group\_id | |

| remote\_ip\_prefix | |

| security\_group\_id | b2917b6f-5cf6-4e3d-b2e7-fb0a2ac4f78c |

| tenant\_id | 4abbad801b1b4bbe89903f4a48c38c58 |

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

设置icmp规则：

# neutron security-group-rule-create --protocol icmp --direction ingress default



Created a new security\_group\_rule:

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

| Field | Value |

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

| direction | ingress |

| ethertype | IPv4 |

| id | 56a7b9d3-f97f-4949-8a8f-825be2f360ce |

| port\_range\_max | |

| port\_range\_min | |

| protocol | icmp |

| remote\_group\_id | |

| remote\_ip\_prefix | |

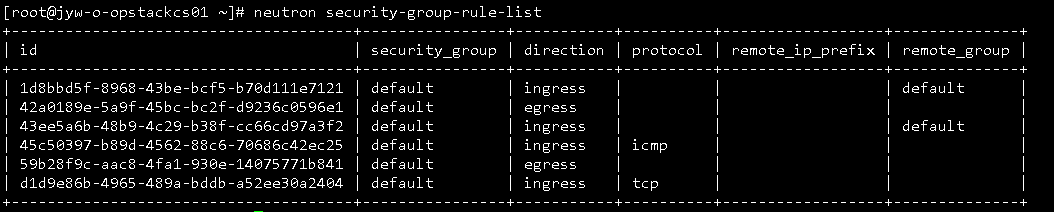
| security\_group\_id | b2917b6f-5cf6-4e3d-b2e7-fb0a2ac4f78c |

| tenant\_id | 4abbad801b1b4bbe89903f4a48c38c58 |

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

查看安全组规则：

# neutron security-group-rule-list



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

| id | security\_group | direction | protocol | remote\_ip\_prefix | remote\_group |

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

| 0cf0241d-d2a8-48b3-9756-c27ad4794c00 | default | ingress | tcp | | |

| 566c2d07-8644-44dc-ac4b-0070a1ac9a4f | default | ingress | | | default |

| 56a7b9d3-f97f-4949-8a8f-825be2f360ce | default | ingress | icmp | | |

| 75006ade-98a6-4818-b77d-01365734b777 | default | egress | | | |

| 77366149-b90e-489c-8834-cb900b1da185 | default | ingress | | | default |

| f06833d2-15b9-478b-918f-3f622aa2fe2d | default | egress | | | |

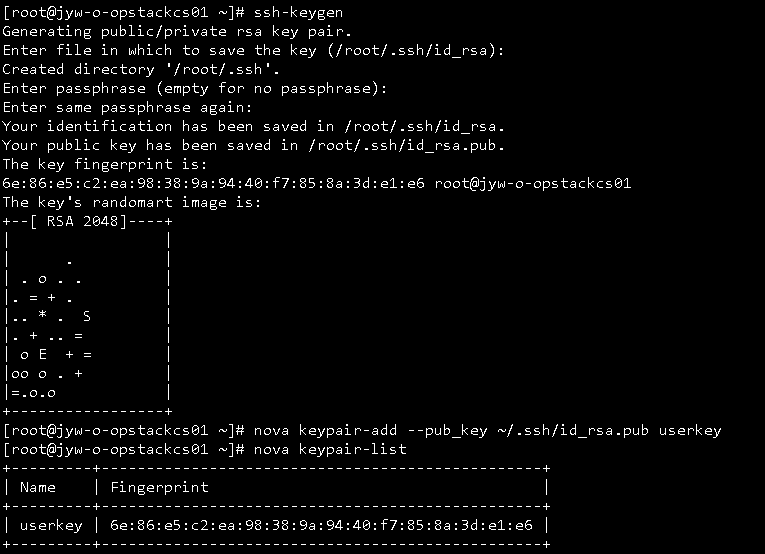
+--------------------------------------+----------------+-----------+----------+------------------+--------------+

1. 创建密钥

# ssh-keygen

# nova keypair-add --pub\_key ~/.ssh/id\_rsa.pub userkey

# nova keypair-list



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

| Name | Fingerprint |

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

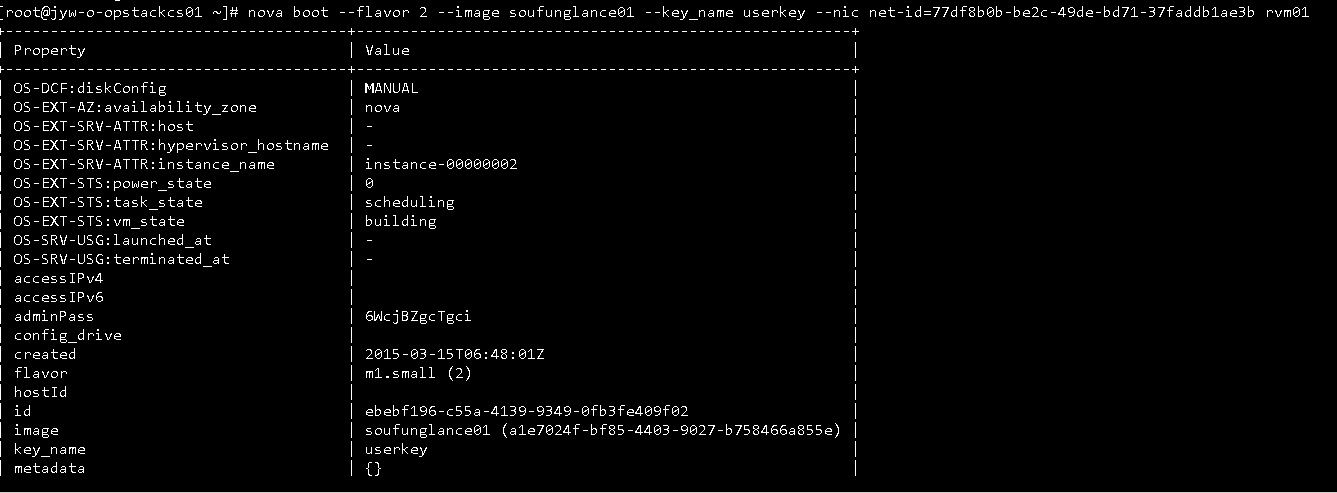
| userkey | ea:bc:d8:ad:1a:50:7a:1c:d7:d5:9a:05:99:aa:c5:ee |

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

1. 创建虚拟机实例

创建虚拟机实例：

# nova boot --flavor 2 --image rhel5.6 --key\_name userkey --nic net-id=cd33634c-ae3b-40be-9ab6-2724e49d39a2 rvm01



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

| Property | Value |

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

| OS-EXT-STS:task\_state | scheduling |

| image | rhel5.6 |

| OS-EXT-STS:vm\_state | building |

| OS-EXT-SRV-ATTR:instance\_name | instance-00000011 |

| OS-SRV-USG:launched\_at | None |

| flavor | m1.small |

| id | d5ebe319-678a-4e08-9a45-b9f4a084bcf7 |

| security\_groups | [{u'name': u'default'}] |

| user\_id | 3f59ffb7df79404c91d99fcdd0ee70fe |

| OS-DCF:diskConfig | MANUAL |

| accessIPv4 | |

| accessIPv6 | |

| progress | 0 |

| OS-EXT-STS:power\_state | 0 |

| OS-EXT-AZ:availability\_zone | nova |

| config\_drive | |

| status | BUILD |

| updated | 2013-11-05T03:05:01Z |

| hostId | |

| OS-EXT-SRV-ATTR:host | None |

| OS-SRV-USG:terminated\_at | None |

| key\_name | userkey |

| OS-EXT-SRV-ATTR:hypervisor\_hostname | None |

| name | rvm01 |

| adminPass | dZEtJoWa5jiD |

| tenant\_id | 4abbad801b1b4bbe89903f4a48c38c58 |

| created | 2013-11-05T03:05:00Z |

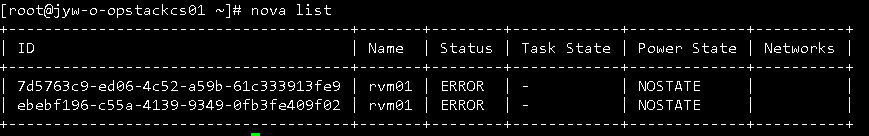
| os-extended-volumes:volumes\_attached | [] |

| metadata | {} |

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

查看虚拟机实例列表：

# nova list



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

| ID | Name | Status | Task State | Power State | Networks |

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

| d5ebe319-678a-4e08-9a45-b9f4a084bcf7 | rvm01 | ACTIVE | None | Running | sharednet1=20.0.0.2 |

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

查看运行中的虚拟机：

# virsh list



Id Name State

----------------------------------------------------

1 instance-00000011 running

## Cinder安装配置

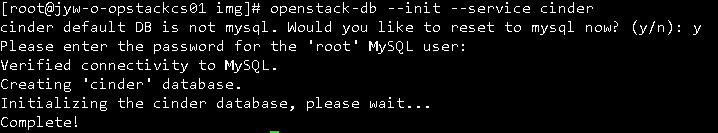
### 初始化Cinder

1. 安装Cinder

# yum install -y openstack-cinder openstack-selinux

1. 创建Cinder数据库

# openstack-db --init --service cinder



### 创建User，定义Services 和 API Endpoints

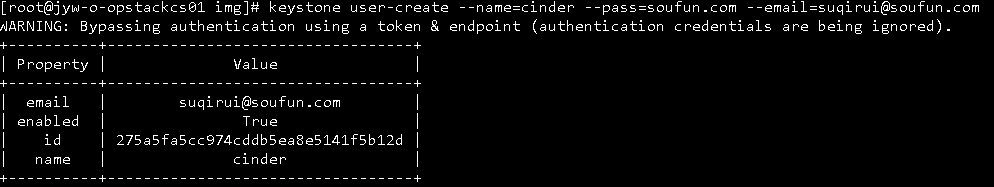
1. 编写脚本

# vi /root/work/cinder-user.sh

#!/bin/sh

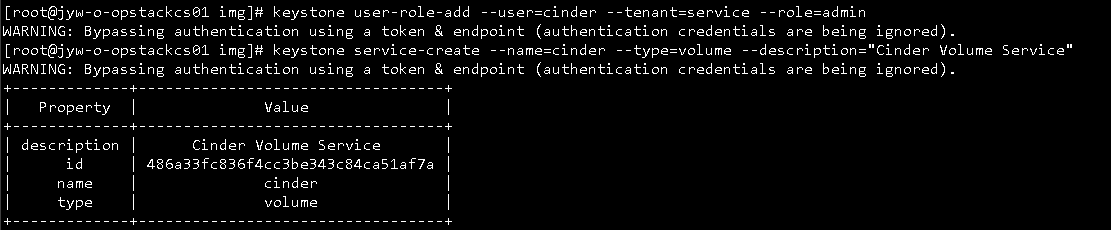
my\_ip=172.21.2.110

keystone user-create --name=cinder --pass=service [--email=cinder@example.com](mailto:--email=cinder@example.com)



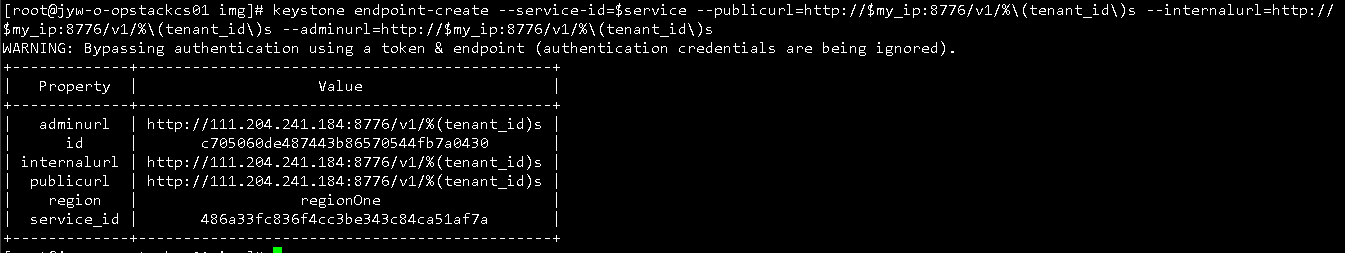
keystone user-role-add --user=cinder --tenant=service --role=admin

keystone service-create --name=cinder --type=volume --description="Cinder Volume Service"

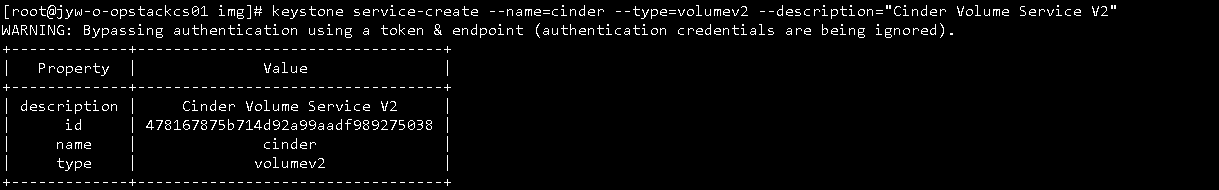


service=$(keystone service-list | awk '/cinder/ {print $2}')

keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:8776/v1/%\(tenant\_id\)s --internalurl=http://$my\_ip:8776/v1/%\(tenant\_id\)s --adminurl=http://$my\_ip:8776/v1/%\(tenant\_id\)s

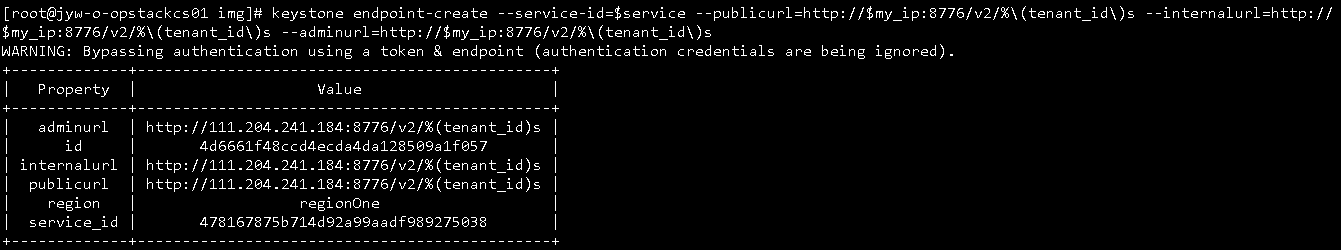


// keystone service-create --name=cinder --type=volumev2 --description="Cinder Volume Service V2"



// service=$(keystone service-list | awk '/volumev2/ {print $2}')

// keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:8776/v2/%\(tenant\_id\)s --internalurl=http://$my\_ip:8776/v2/%\(tenant\_id\)s --adminurl=http://$my\_ip:8776/v2/%\(tenant\_id\)s



1. 运行脚本，创建用户、服务及api endpoint

# sh /root/work/cinder-user.sh

### 配置Cinder服务

1. 创建分区

# fdisk /dev/sdb

依次输入：n p 1 1 24320 w （根据实际情况进行设置）

# pvcreate /dev/sdb1

# vgcreate cinder-volumes /dev/sdb1 //逻辑卷组

1. 修改cinder.conf文件

# cat /etc/cinder/cinder.conf

[DEFAULT]

log\_dir = /var/log/cinder

state\_path = /var/lib/cinder

lock\_path = /var/lib/cinder/tmp

volumes\_dir=/etc/cinder/volumes

iscsi\_helper = tgtadm ///

connection = mysql://cinder:cinder@localhost/cinder

notification\_driver = cinder.openstack.common.notifier.rpc\_notifier

control\_exchange = cinder

rpc\_backend = cinder.openstack.common.rpc.impl\_qpid

qpid\_hostname = 172.21.2.110

auth\_strategy = keystone

1. 修改api-paste.ini文件

# vi /etc/cinder/api-paste.ini

[filter:authtoken]

paste.filter\_factory = keystoneclient.middleware.auth\_token:filter\_factory

auth\_host = 172.21.2.110

auth\_port = 35357

auth\_protocol = http

admin\_user = cinder

admin\_tenant\_name = service

admin\_password = service

1. 修改targets.conf文件

# grep -q /etc/cinder/volumes /etc/tgt/targets.conf || sed -i '1iinclude /etc/cinder/volumes/\*' /etc/tgt/targets.conf

1. 启动cinder服务

# service tgtd start

# chkconfig tgtd on

# service openstack-cinder-api start

# service openstack-cinder-scheduler start

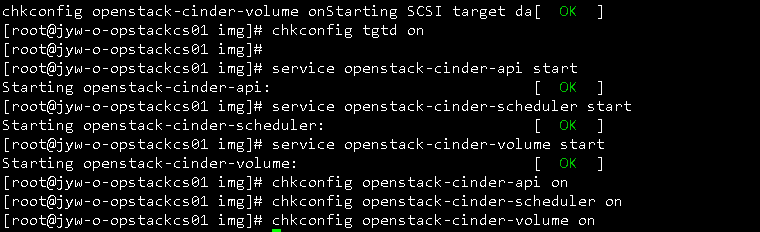
# service openstack-cinder-volume start

# chkconfig openstack-cinder-api on

# chkconfig openstack-cinder-scheduler on

# chkconfig openstack-cinder-volume on

for s in openstack-cinder-{api,scheduler,volume}; do chkconfig $s on;done



### Cinder测试

1. 创建卷

cinder list  查看目前的EBS块存储

# cinder create --display-name lv01 2

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

| Property | Value |

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

| attachments | [] |

| availability\_zone | nova |

| bootable | false |

| created\_at | 2013-11-06T03:30:13.945688 |

| display\_description | None |

| display\_name | lv01 |

| id | 0b883ef7-fc7e-4792-b731-0d9409823dd6 |

| metadata | {} |

| size | 2 |

| snapshot\_id | None |

| source\_volid | None |

| status | creating |

| volume\_type | None |

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

1. 查看卷列表

# cinder list

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

| ID | Status | Display Name | Size | Volume Type | Bootable | Attached to |

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

| 0b883ef7-fc7e-4792-b731-0d9409823dd6 | available | lv01 | 2 | None | false | |

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

## Swift安装配置

### 初始化Swift

1. 安装Swift

# yum install -y openstack-swift openstack-swift-proxy openstack-swift-account openstack-swift-container openstack-swift-object xinetd xfsprogs

1. 存储设备

# fdisk /dev/sdb

依次输入：n p 2 回车 48640 w （根据实际情况进行设置）

# mkfs.xfs -i size=1024 /dev/sdb2

# echo "/dev/sdb2 /swift/node xfs loop,noatime,nodiratime,nobarrier,logbufs=8 0 0" >> /etc/fstab

# mkdir -p /swift/node

# mount /swift/node

# chown -R swift:swift /swift

### 创建User，定义Services 和 API Endpoints

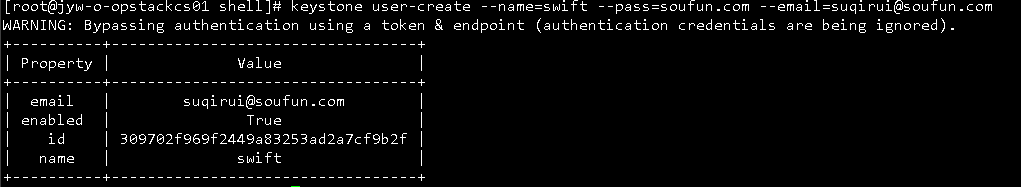
1. 编写脚本

# vi /root/work/swift-user.sh

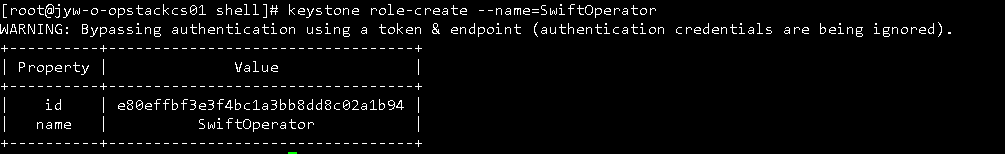
#!/bin/sh

my\_ip=172.21.2.110

keystone user-create --name=swift --pass=service [--email=swift@example.com](mailto:--email=swift@example.com)



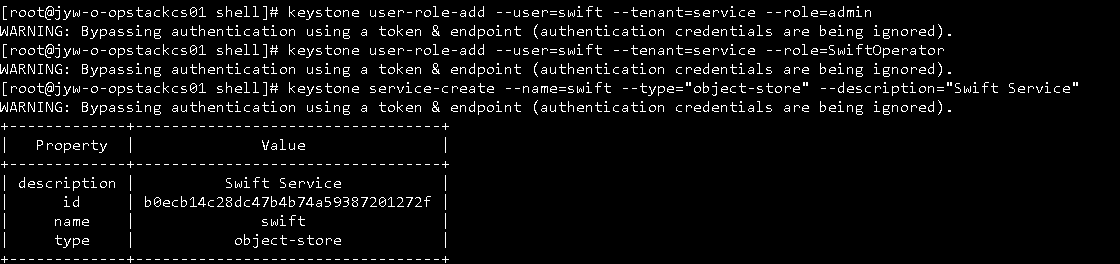
keystone role-create --name=SwiftOperator



keystone user-role-add --user=swift --tenant=service --role=admin

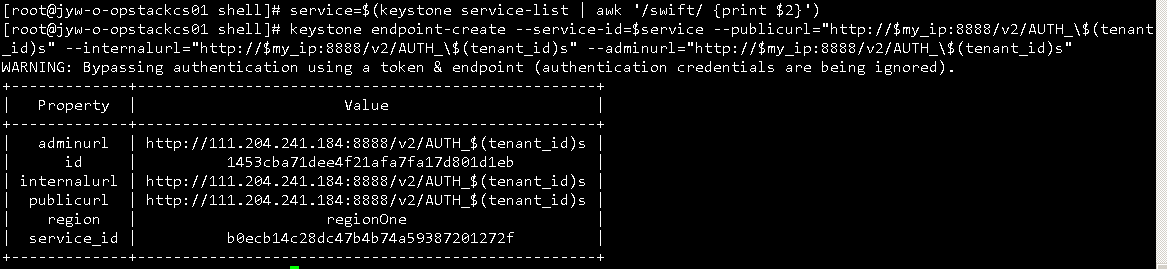
keystone user-role-add --user=swift --tenant=service --role=SwiftOperator

keystone service-create --name=swift --type="object-store" --description="Swift Service"



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

keystone endpoint-create --service-id=$service --publicurl="http://$my\_ip:8888/v2/AUTH\_\$(tenant\_id)s" --internalurl="http://$my\_ip:8888/v2/AUTH\_\$(tenant\_id)s" --adminurl=<http://$my_ip:8888/v2/AUTH_\$(tenant_id)s>



1. 运行脚本，创建用户、服务及api endpoint

# sh /root/work/swift-user.sh

### 配置Swift

1. 配置rsync

创建rsync.conf文件：

# vi /etc/rsync.conf

uid = swift

gid = swift

log file = /var/log/rsyncd.log

pid file = /var/run/rsyncd.pid

address = 127.0.0.1

[account]

max connections = 2

path = /swift/node

read only = false

lock file = /var/lock/account.lock

[container]

max connections = 2

path = /swift/node

read only = false

lock file = /var/lock/container.lock

[object]

max connections = 2

path = /swift/node

read only = false

lock file = /var/lock/object.lock

启用rsync：

# sed -i 's/yes/no/' /etc/xinetd.d/rsync

# sed -i 's/IPv6/IPv4/' /etc/xinetd.d/rsync

启动xinetd：

# service xinetd start

设置开机启动：

# chkconfig xinetd on

1. 配置swift.conf文件

# vi /etc/swift/swift.conf

[swift-hash]

swift\_hash\_path\_suffix = `od -t x8 -N 8 -A n </dev/random`

1. 配置proxy

# vi /etc/swift/proxy-server.conf

[DEFAULT]

bind\_port = 8888

workers = 3

user = swift

log\_name = proxy

log\_level = DEBUG

log\_facility = LOG\_LOCAL0

[pipeline:main]

pipeline = healthcheck cache authtoken keystone proxy-server

[app:proxy-server]

use = egg:swift#proxy

allow\_account\_management = true

account\_autocreate = true

[filter:cache]

use = egg:swift#memcache

memcache\_servers = 127.0.0.1:11211

[filter:catch\_errors]

use = egg:swift#catch\_errors

[filter:healthcheck]

use = egg:swift#healthcheck

[filter:keystone]

use = egg:swift#keystoneauth

operator\_roles = admin, SwiftOperator

is\_admin = true

cache = swift.cache

[filter:authtoken]

paste.filter\_factory = keystoneclient.middleware.auth\_token:filter\_factory

admin\_tenant\_name = service

admin\_user = swift

admin\_password = service

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

signing\_dir = /tmp/keystone-signing-swift

1. 配置storage

修改account-server.conf：

# cat /etc/swift/account-server.conf

[DEFAULT]

bind\_port = 6002

workers = 2

user = swift

devices = /swift/node

mount\_check = false

log\_name = swift-account

log\_facility = LOG\_LOCAL1

log\_level = DEBUG

[pipeline:main]

pipeline = account-server

[app:account-server]

use = egg:swift#account

[account-replicator]

vm\_test\_mode = no

[account-auditor]

[account-reaper]

修改container-server.conf：

# cat /etc/swift/container-server.conf

[DEFAULT]

bind\_port = 6001

workers = 2

user = swift

devices = /swift/node

mount\_check = false

log\_name = swift-container

log\_facility = LOG\_LOCAL1

log\_level = DEBUG

[pipeline:main]

pipeline = container-server

[app:container-server]

use = egg:swift#container

[container-replicator]

vm\_test\_mode = no

[container-updater]

[container-auditor]

[container-sync]

修改object-server.conf：

# cat /etc/swift/object-server.conf

[DEFAULT]

bind\_port = 6000

workers = 2

user = swift

devices = /swift/node

mount\_check = false

log\_name = swift-object

log\_facility = LOG\_LOCAL1

log\_level = DEBUG

[pipeline:main]

pipeline = object-server

[app:object-server]

use = egg:swift#object

[object-replicator]

vm\_test\_mode = no

[object-updater]

[object-auditor]

1. 配置日志

# vi /etc/rsyslog.d/10-swift.conf

local0.\* /var/log/swift/proxy.log

local1.\* /var/log/swift/storage1.log

# mkdir -p /var/log/swift

# chown -R swift:swift /var/log/swift

# service rsyslog restart

1. 配置Ring

进入/etc/swift目录：

# cd /etc/swift/

创建Ring：

# swift-ring-builder account.builder create 18 1 1

# swift-ring-builder container.builder create 18 1 1

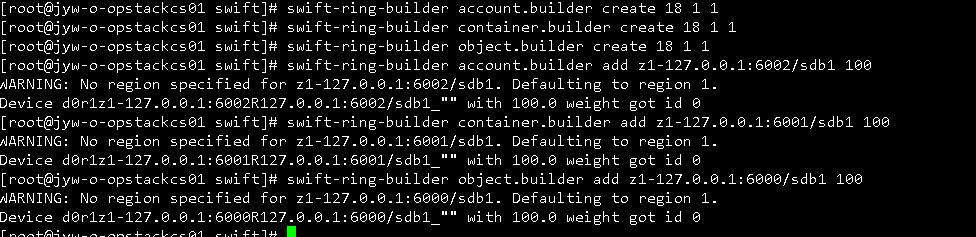
# swift-ring-builder object.builder create 18 1 1

向环里添加存储设备：

# swift-ring-builder account.builder add z1-127.0.0.1:6002/sdb2 100

# swift-ring-builder container.builder add z1-127.0.0.1:6001/sdb2 100

# swift-ring-builder object.builder add z1-127.0.0.1:6000/sdb2 100

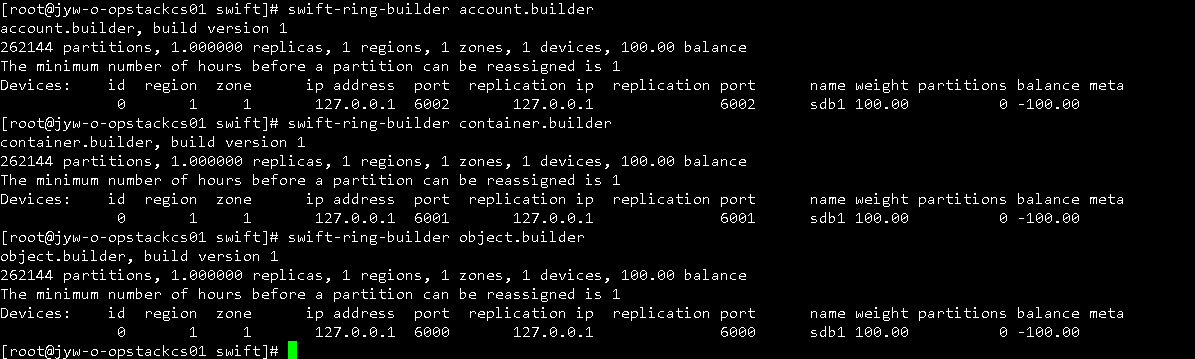


检验每个环里的条目：

# swift-ring-builder account.builder

# swift-ring-builder container.builder

# swift-ring-builder object.builder

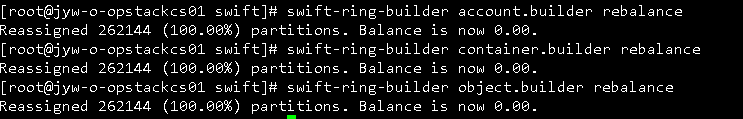


平衡环：

# swift-ring-builder account.builder rebalance

# swift-ring-builder container.builder rebalance

# swift-ring-builder object.builder rebalance



1. 设置目录权限

# chown -R swift:swift /etc/swift

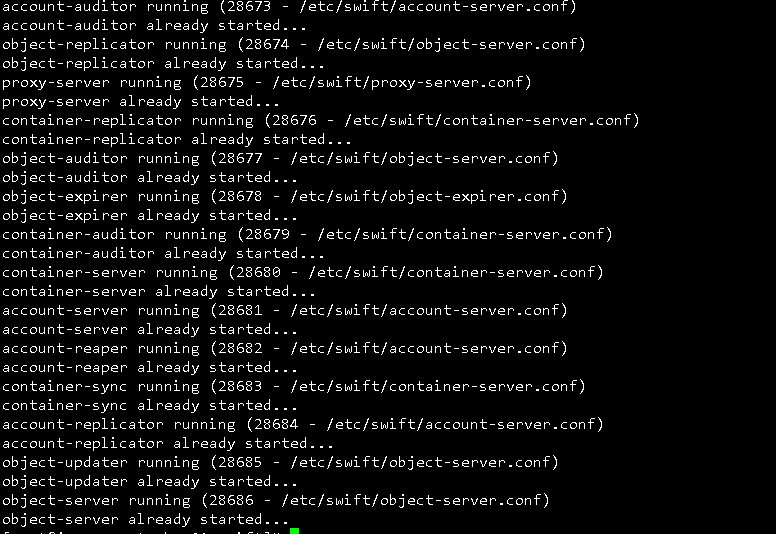
1. 启动swift服务

# service memcached start

# chkconfig memcached on

# swift-init all start

// No handlers could be found for logger "proxy"



测试

新建container

swift post testcontainer

上传文件

swift upload testcontainer test.txt

查询一级目录

swift list

查询testcontainer目录下面的文件

swift list testcontainer

### Cinder+Swift测试

Swift作为Cinder的备份后端使用方法如下：

1. 修改Cinder的配置文件

# cat /etc/cinder/cinder.conf

[DEFAULT]

log\_dir = /var/log/cinder

state\_path = /var/lib/cinder

lock\_path = /var/lib/cinder/tmp

volumes\_dir=/etc/cinder/volumes

iscsi\_helper = tgtadm

connection = mysql://cinder:cinder@localhost/cinder

notification\_driver = cinder.openstack.common.notifier.rpc\_notifier

control\_exchange = cinder

rpc\_backend = cinder.openstack.common.rpc.impl\_qpid

qpid\_hostname = 172.21.2.110

auth\_strategy = keystone

backup\_swift\_url=http://172.21.2.110:8888/v2/AUTH\_

backup\_swift\_auth=127.0.0.1

backup\_swift\_user=swift

backup\_swift\_key=service

1. 重新启动Cinder服务

# service openstack-cinder-api restart

# service openstack-cinder-scheduler restart

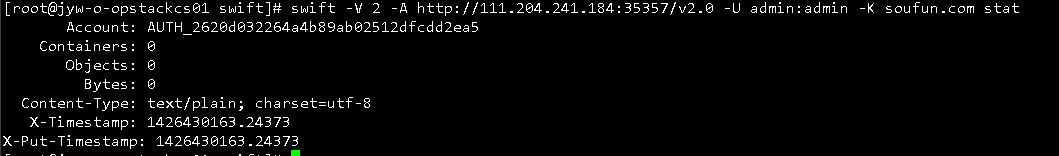
# service openstack-cinder-volume restart

1. 启动Cinder 备份服务

# service openstack-cinder-backup start

1. 查看swift的存储信息

swift -V 2 -A http://111.204.241.184:35357/v2.0 -U admin:admin -K soufun.com stat



Account: AUTH\_4abbad801b1b4bbe89903f4a48c38c58

Containers: 0

Objects: 0

Bytes: 0

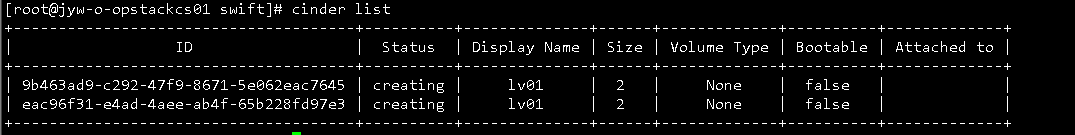
Content-Type: text/plain; charset=utf-8

X-Timestamp: 1384329984.86445

X-Put-Timestamp: 1384329984.86445

1. 测试Cinder备份功能

# cinder list



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

| ID | Status | Display Name | Size | Volume Type | Bootable | Attached to |

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

| 0b883ef7-fc7e-4792-b731-0d9409823dd6 | available | lv01 | 2 | None | false | |

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

# cinder backup-create --display-name t\_lv01 lv01

# cinder backup-list

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

| ID | Volume ID | Status | Name | Size | Object Count | Container |

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

| 3a5d77e6-1750-4985-8be6-1ecf7feb29e3 | 0b883ef7-fc7e-4792-b731-0d9409823dd6 | available | t\_lv01 | 2 | 42 | volumebackups |

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

1. 查看swift存储信息

# swift -V 2 -A <http://172.21.2.110:35357/v2.0> -U admin:admin -K secrete stat

Account: AUTH\_4abbad801b1b4bbe89903f4a48c38c58

Containers: 1

Objects: 42

Bytes: 2100176

Accept-Ranges: bytes

X-Timestamp: 1384332027.11336

Content-Type: text/plain; charset=utf-8

# swift -V 2 -A <http://172.21.2.110:35357/v2.0> -U admin:admin -K secrete list



volumebackups

# swift -V 2 -A <http://172.21.2.110:35357/v2.0> -U admin:admin -K secrete list volumebackups

volume\_0b883ef7-fc7e-4792-b731-0d9409823dd6/20131113084027/az\_nova\_backup\_3a5d77e6-1750-4985-8be6-1ecf7feb29e3-00001

volume\_0b883ef7-fc7e-4792-b731-0d9409823dd6/20131113084027/az\_nova\_backup\_3a5d77e6-1750-4985-8be6-1ecf7feb29e3-00002

………….

volume\_0b883ef7-fc7e-4792-b731-0d9409823dd6/20131113084027/az\_nova\_backup\_3a5d77e6-1750-4985-8be6-1ecf7feb29e3-00041

volume\_0b883ef7-fc7e-4792-b731-0d9409823dd6/20131113084027/az\_nova\_backup\_3a5d77e6-1750-4985-8be6-1ecf7feb29e3\_metadata

## Ceilometer安装配置

### 初始化Ceilometer

1. 安装mongodb

# yum install -y mongo-10gen\*

1. 安装ceilometer

# yum install -y openstack-ceilometer\*

### 创建User，定义Services 和 API Endpoints

1. 编写脚本

# vi /root/work/ceilometer-user.sh

#!/bin/sh

my\_ip=172.21.2.110

keystone user-create --name=ceilometer --pass=service --email=ceilometer@example.com

keystone user-role-add --user=ceilometer --tenant=service --role=admin

keystone service-create --name=ceilometer --type=metering --description="Ceilometer Metering Service"

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

keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:8777/ --internalurl=http://$my\_ip:8777/ --adminurl=http://$my\_ip:8777/

1. 运行脚本，创建用户、服务及api endpoint

# sh /root/work/ceilometer-user.sh

### 配置Ceilometer

1. 使用openssl随即生成一个令牌

# meter\_token=$(openssl rand -hex 10)

# echo $meter\_token

3f763fa88a74ac72a679

# echo $meter\_token > /root/work/meter\_token.txt

1. 修改ceilometer配置文件

# vi /etc/ceilometer/ceilometer.conf

[DEFAULT]

verbose = True

debug = True

#os\_auth\_url = <http://127.0.0.1:5000/v2.0>

#os\_tenant\_name = admin

#os\_password = secrete

#os\_username = admin

#policy\_file = /etc/ceilometer/policy.json

rpc\_backend=ceilometer.openstack.common.rpc.impl\_qpid

qpid\_hostname = 172.21.2.110

auth\_strategy = keystone

[publisher\_rpc]

metering\_secret = 3f763fa88a74ac72a679

[ssl]

[database]

connection = mongodb://localhost:27017/ceilometer

[alarm]

[rpc\_notifier2]

[api]

[service\_credentials]

[dispatcher\_file]

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_user = ceilometer

admin\_tenant\_name = service

admin\_password = service

[collector]

[matchmaker\_ring]

[matchmaker\_redis]

1. 启动mongodb，设置开机自启动

# service mongod start

# chkconfig mongod on

1. 启动ceilometer，设置开机自启动

# service openstack-ceilometer-api start

# service openstack-ceilometer-alarm-evaluator start

# service openstack-ceilometer-alarm-notifier start

# service openstack-ceilometer-central start

# service openstack-ceilometer-collector start

# service openstack-ceilometer-compute start

# chkconfig openstack-ceilometer-api on

# chkconfig openstack-ceilometer-alarm-evaluator on

# chkconfig openstack-ceilometer-alarm-notifier on

# chkconfig openstack-ceilometer-central on

# chkconfig openstack-ceilometer-collector on

# chkconfig openstack-ceilometer-compute on

### Ceilometer测试（监控glance、nova、neturon、Cinder、Swift）

1. Glance配置Ceilometer监控

修改glance-api文件配置notifier\_strategy参数值，重新启动glance服务：

# vi /etc/glance/glance-api.conf

……

notifier\_strategy=qpid

# service openstack-glance-api restart

# service openstack-glance-registry restart

1. Nova配置Ceilometer监控

修改nova.conf文件，重启启动nova-compute服务：

# cat /etc/nova/nova.conf

[DEFAULT]

debug=true

my\_ip = 172.21.2.110

auth\_strategy = keystone

state\_path = /openstack/lib/nova

verbose=True

allow\_resize\_to\_same\_host = true

rpc\_backend = nova.openstack.common.rpc.impl\_qpid

qpid\_hostname = 172.21.2.110

libvirt\_type = kvm

glance\_api\_servers = 172.21.2.110:9292

novncproxy\_base\_url = http://172.21.2.110:6080/vnc\_auto.html

vncserver\_listen = 172.21.2.110

vncserver\_proxyclient\_address = 172.21.2.110

vnc\_enabled = true

vnc\_keymap = en-us

network\_manager = nova.network.manager.FlatDHCPManager

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

multi\_host = True

flat\_interface = eth1

flat\_network\_bridge = br100

public\_interface = eth0

instance\_usage\_audit = True

instance\_usage\_audit\_period = hour

notify\_on\_state\_change = vm\_and\_task\_state

notification\_driver = nova.openstack.common.notifier.rpc\_notifier

notification\_driver = ceilometer.compute.nova\_notifier

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

neutron\_admin\_username = neutron

neutron\_admin\_password = service

neutron\_admin\_auth\_url = http://127.0.0.1:5000/v2.0

neutron\_auth\_strategy = keystone

neutron\_admin\_tenant\_name = service

neutron\_url = http://localhost:9696/

security\_group\_api = neutron

libvirt\_vif\_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver

#libvirt\_vif\_driver=nova.virt.libvirt.vif.LibvirtGenericVIFDriver

neutron\_ovs\_bridge = br-int

[hyperv]

[zookeeper]

[osapi\_v3]

[conductor]

[keymgr]

[cells]

[database]

[image\_file\_url]

[baremetal]

[rpc\_notifier2]

[matchmaker\_redis]

[ssl]

[trusted\_computing]

[upgrade\_levels]

[matchmaker\_ring]

[vmware]

[spice]

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_user = nova

admin\_tenant\_name = service

admin\_password = service

# service openstack-nova-compute restart

1. Neutron配置Ceilometer监控

修改neutron.conf配置文件，重启neutron服务：

# cat /etc/neutron/neutron.conf | grep -v ^# | grep -v ^$

[DEFAULT]

core\_plugin = neutron.plugins.openvswitch.ovs\_neutron\_plugin.OVSNeutronPluginV2

auth\_strategy = keystone

control\_exchange = neutron

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

qpid\_hostname = localhost

notification\_driver = neutron.openstack.common.notifier.rpc\_notifier

[quotas]

[agent]

root\_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = service

[database]

connection = mysql://neutron:neutron@localhost/ovs\_neutron

[service\_providers]

# service neutron-server restart

# service neutron-dhcp-agent restart

# service neutron-openvswitch-agent restart

1. Cinder配置Ceilometer监控

修改cinder配置文件，重启服务：

# cat /etc/cinder/cinder.conf

[DEFAULT]

log\_dir = /var/log/cinder

state\_path = /var/lib/cinder

lock\_path = /var/lib/cinder/tmp

volumes\_dir=/etc/cinder/volumes

iscsi\_helper = tgtadm

connection = mysql://cinder:cinder@localhost/cinder

notification\_driver = cinder.openstack.common.notifier.rpc\_notifier

control\_exchange = cinder

rpc\_backend = cinder.openstack.common.rpc.impl\_qpid

qpid\_hostname = 172.21.2.110

auth\_strategy = keystone

backup\_swift\_url=http://172.21.2.110:8888/v2/AUTH\_

backup\_swift\_auth=127.0.0.1

backup\_swift\_user=swift

backup\_swift\_key=service

# service openstack-cinder-api restart

# service openstack-cinder-scheduler restart

# service openstack-cinder-volume restart

1. Swift配置Ceilometer监控

创建ResellerAdmin角色，将该角色赋给ceilometer用户：

# keystone role-create --name=ResellerAdmin

# keystone user-role-add --user=ceilometer --tenant=service --role=ResellerAdmin

修改proxy-server.conf配置文件，添加以下内容：

# vi /etc/swift/proxy-server.conf

……

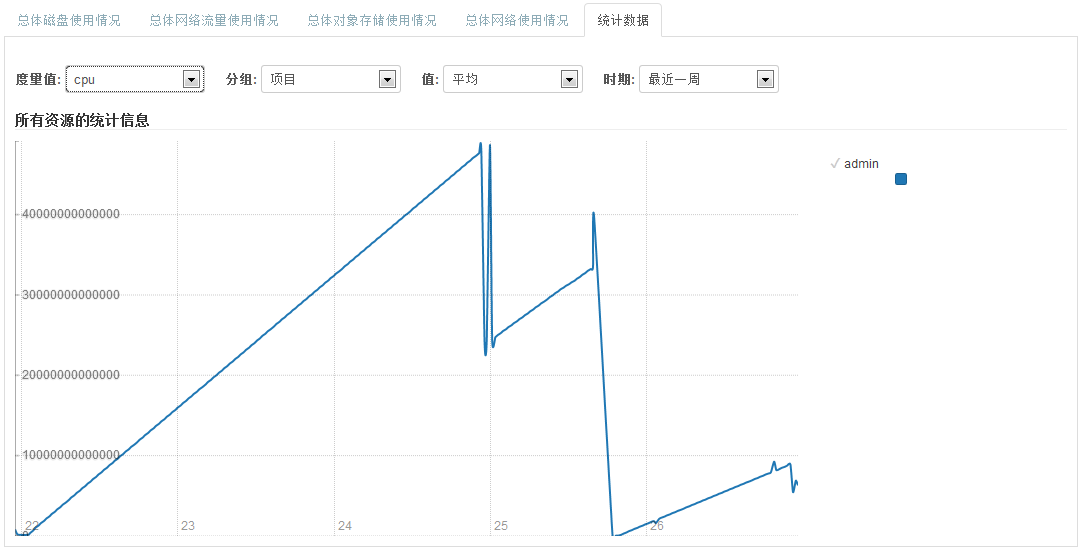
[filter:ceilometer]

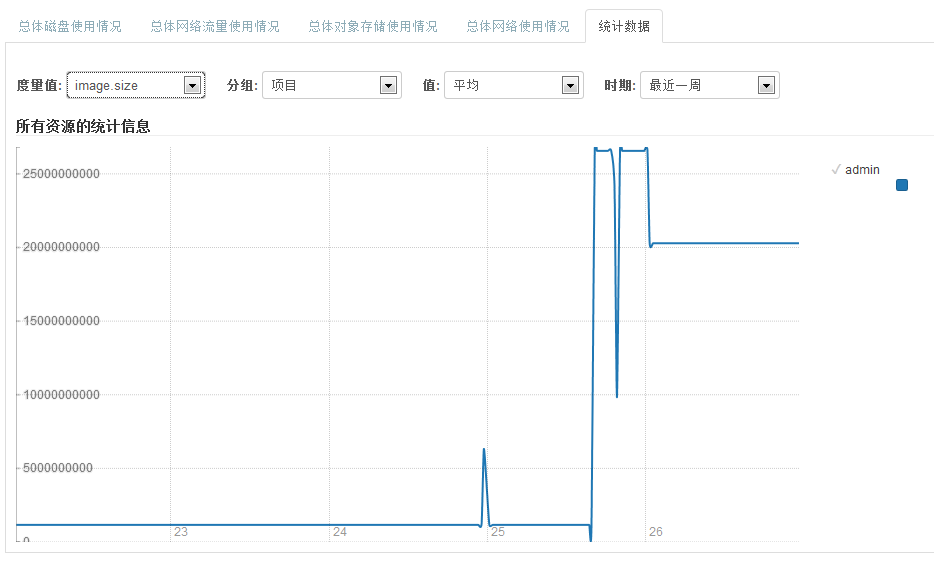
use = egg:ceilometer#swift

重新启动proxy服务：

# service openstack-swift-proxy restart

1. 通过web界面查看ceilometer统计信息





## Heat安装配置

### 初始化Heat

1. 安装Heat

# yum install -y openstack-heat\*

1. 创建Heat数据库

# heat-db-setup rpm -y -r openstack

### 创建User，定义Services 和 API Endpoints

1. 编写脚本

# vi /root/work/heat-user.sh

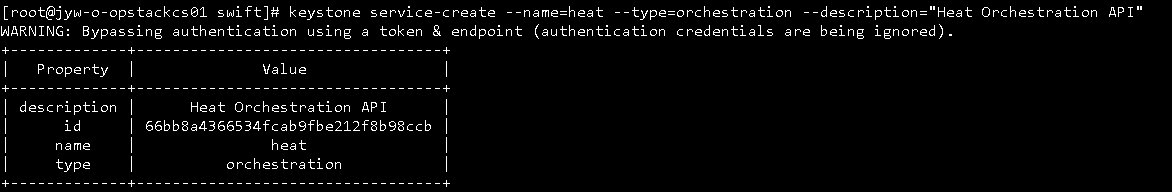
#!/bin/sh

my\_ip=172.21.2.110

keystone user-create --name=heat --pass=service --email=heat@example.com

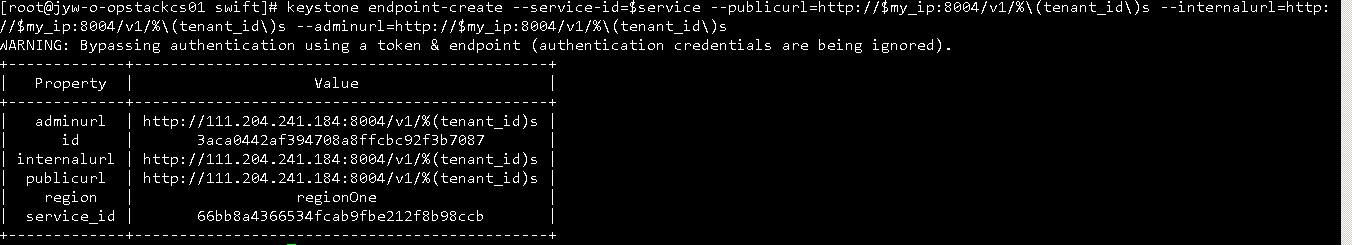
keystone user-role-add --user=heat --tenant=service --role=admin

keystone service-create --name=heat --type=orchestration --description="Heat Orchestration API"



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

keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:8004/v1/%\(tenant\_id\)s --internalurl=http://$my\_ip:8004/v1/%\(tenant\_id\)s --adminurl=http://$my\_ip:8004/v1/%\(tenant\_id\)s

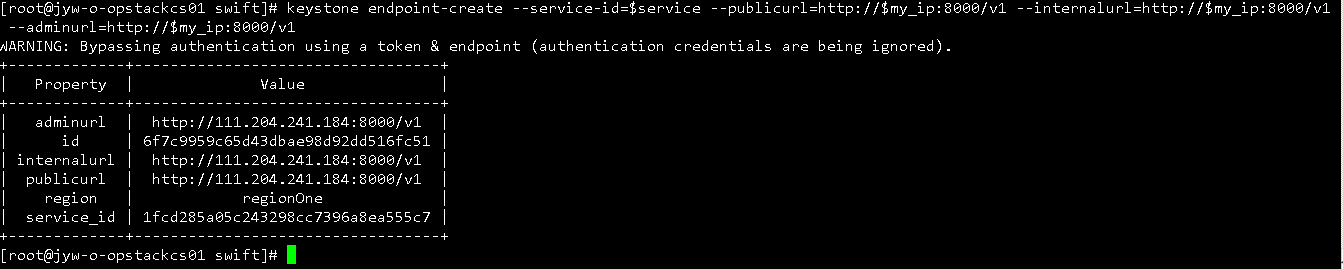


keystone service-create --name=heat-cfn --type=cloudformation --description="Heat CloudFormation API"



service=$(keystone service-list | awk '/heat-cfn/ {print $2}')

keystone endpoint-create --service-id=$service --publicurl=http://$my\_ip:8000/v1 --internalurl=http://$my\_ip:8000/v1 --adminurl=http://$my\_ip:8000/v1



1. 运行脚本，创建用户、服务及api endpoint

# sh /root/work/heat-user.sh

### 配置Heat

1. 配置heat.conf

使用openssl随即生成一个令牌，将其存储在配置文件中:

# heat\_token=$(openssl rand -hex 10)

# echo $heat\_token

a3348271784502b6fcc3

# echo $heat\_token > /root/work/heat\_token.txt

# vi /etc/heat/heat.conf

[DEFAULT]

sql\_connection=mysql://heat:heat@localhost/heat

host=172.21.2.110

heat\_metadata\_server\_url=http://127.0.0.1:8000

heat\_waitcondition\_server\_url=http://127.0.0.1:8000/v1/waitcondition

heat\_watch\_server\_url=http://127.0.0.1:8003

instance\_connection\_is\_secure=0

instance\_connection\_https\_validate\_certificates=1

heat\_stack\_user\_role=heat\_stack\_user

#### auth\_encryption\_key=a3348271784502b6fcc3

db\_backend=heat.db.sqlalchemy.api

debug=True

rpc\_backend=heat.openstack.common.rpc.impl\_qpid

qpid\_hostname=172.21.2.110

[ssl]

[database]

[paste\_deploy]

api\_paste\_config=/etc/heat/api-paste.ini

[rpc\_notifier2]

[ec2authtoken]

[heat\_api\_cloudwatch]

[heat\_api]

[heat\_api\_cfn]

[auth\_password]

[matchmaker\_ring]

[matchmaker\_redis]

[keystone\_authtoken]

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_user = heat

admin\_tenant\_name = service

admin\_password = secrete

因上述配置文件中，api-paste.ini的路径设置为/etc/heat目录下，需要执行以下操作：

# cp /usr/share/heat/api-paste-dist.ini /etc/heat/api-paste.ini

# chown -R root:heat /etc/heat/api-paste.ini

1. 修改api-paste.ini文件

# vi /etc/heat/api-paste.ini

[filter:authtoken]

paste.filter\_factory = heat.common.auth\_token:filter\_factory

auth\_host = 127.0.0.1

auth\_port = 35357

auth\_protocol = http

admin\_user = heat

admin\_tenant\_name = service

admin\_password = secrete

1. 启动heat服务

# service openstack-heat-api start

# service openstack-heat-api-cfn start

# service openstack-heat-api-cloudwatch start

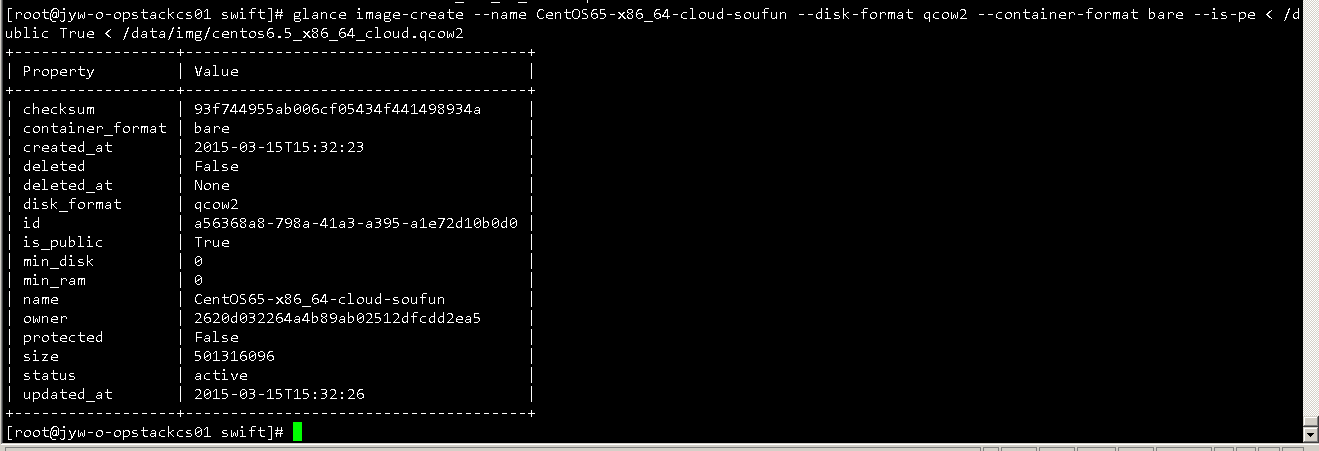
# service openstack-heat-engine start

for s in openstack-heat-{api,api-cfn,api-cloudwatch,engine};do chkconfig $s on;done

### Heat测试

1. 上传Heat测试用的镜像

# glance image-create --name F17-x86\_64-cfntools --disk-format qcow2 --container-format bare --is-public True < /media/images/F17-x86\_64-cfntools.qcow2



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

| Property | Value |

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

| checksum | afab0f79bac770d61d24b4d0560b5f70 |

| container\_format | bare |

| created\_at | 2013-10-25T02:51:26 |

| deleted | False |

| deleted\_at | None |

| disk\_format | qcow2 |

| id | e03cbba0-07cb-4741-8987-b8d7a50260d9 |

| is\_public | True |

| min\_disk | 0 |

| min\_ram | 0 |

| name | F17-x86\_64-cfntools |

| owner | 4abbad801b1b4bbe89903f4a48c38c58 |

| protected | False |

| size | 476704768 |

| status | active |

| updated\_at | 2013-10-25T02:51:29 |

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

# glance image-list



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

| ID | Name | Disk Format | Container Format | Size | Status |

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

| e03cbba0-07cb-4741-8987-b8d7a50260d9 | F17-x86\_64-cfntools | qcow2 | bare | 476704768 | active |

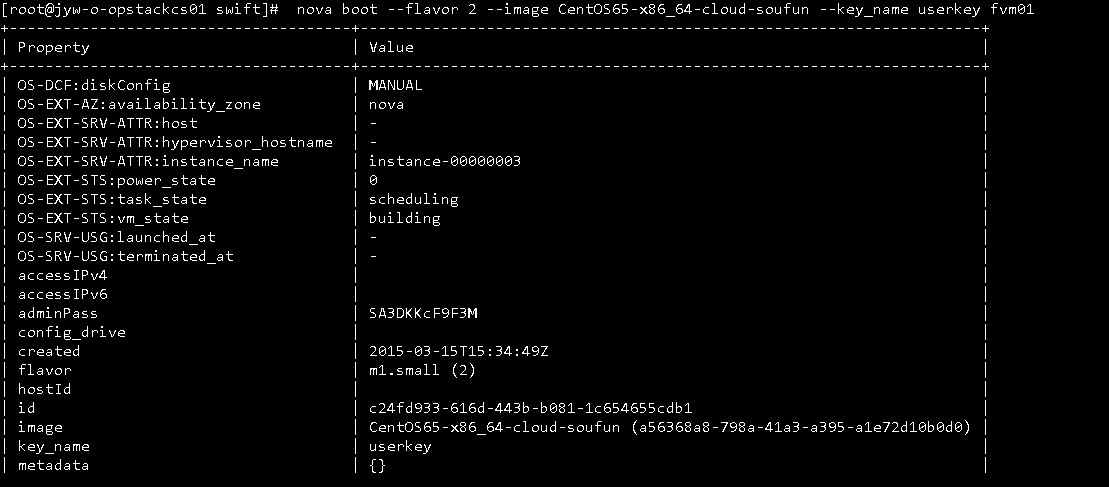
| ee9ee181-e717-4c8b-83c5-afb44d96f301 | rhel5.6 | qcow2 | ovf | 1855979520 | active |

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

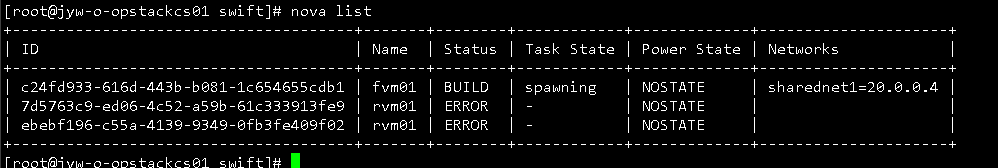
1. 创建一个虚拟机实例

# nova boot --flavor 2 --image F17-x86\_64-cfntools --key\_name userkey fvm01

#nova boot --flavor 2 --image CentOS65-x86\_64-cloud-soufun --key\_name userkey --nic net-id=77df8b0b-be2c-49de-bd71-37faddb1ae3b rvm01



# nova list



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

| ID | Name | Status | Task State | Power State | Networks |

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

| 41fb07f7-bb06-45c0-a534-469635dd0113 | fvm01 | ACTIVE | None | Running | vmnet=10.0.0.2 |

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

登录虚拟机实例：

# ssh -i userkey root@10.0.0.2

Warning: Identity file userkey not accessible: No such file or directory.

The authenticity of host '10.0.0.2 (10.0.0.2)' can't be established.

RSA key fingerprint is b0:9e:36:11:9f:2d:ff:12:47:24:e6:12:9c:30:86:d9.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.0.0.2' (RSA) to the list of known hosts.

[root@fvm01 ~]# cd /var/cache/yum/x86\_64/17/updates/packages/

[root@fvm01 packages]# ll

total 0

[root@fvm01 ~]# exit;

logout

Connection to 10.0.0.2 closed.

1. 创建heat stack

# heat stack-create mystack --template-file=/media/images/WordPress\_Single\_Instance.template --parameter="InstanceType=m1.small;KeyName=userkey;LinuxDistribution=F17"

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

| id | stack\_name | stack\_status | creation\_time |

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

| b06f7e3a-69b5-4182-b5fd-fe250c7735b3 | mystack | CREATE\_IN\_PROGRESS | 2013-10-25T07:01:55Z |

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

 查看stack列表：

# heat stack-list

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

| id | stack\_name | stack\_status | creation\_time |

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

| b06f7e3a-69b5-4182-b5fd-fe250c7735b3 | mystack | CREATE\_COMPLETE | 2013-10-25T07:01:55Z |

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

查看mystack详细信息：

# heat stack-show mystack

1. 查看resource

# heat resource-list mystack

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

| logical\_resource\_id | resource\_type | resource\_status | updated\_time |

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

| WikiDatabase | AWS::EC2::Instance | CREATE\_COMPLETE | 2013-10-25T07:02:12Z |

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

查看resource详细信息：

# heat resource-show mystack WikiDatabase

1. 查看event

# heat event-list mystack

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

| logical\_resource\_id | id | resource\_status\_reason | resource\_status | event\_time |

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

| WikiDatabase | 5 | state changed | CREATE\_IN\_PROGRESS | 2013-10-25T07:01:55Z |

| WikiDatabase | 6 | state changed | CREATE\_COMPLETE | 2013-10-25T07:02:12Z |

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

1. 查看nova虚拟机列表

# nova list

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

| ID | Name | Status | Task State | Power State | Networks |

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

| 41fb07f7-bb06-45c0-a534-469635dd0113 | fvm01 | ACTIVE | None | Running | vmnet=10.0.0.2 |

| b2556253-f41f-48d1-9221-48bb8bd6316b | mystack-WikiDatabase-mq6q6753xjcm | ACTIVE | None | Running | vmnet=10.0.0.4 |

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

虚拟机mystack-WikiDatabase-mq6q6753xjcm是由heat stack创建生成的，它也是由F17-x86\_64-cfntools.qcow2镜像为模板生成的，与fvm01相比多了mysql相关的安装文件而已。

1. 登录虚拟机mystack-WikiDatabase-mq6q6753xjcm

# ssh -i userkey ec2-user@10.0.0.4

Warning: Identity file userkey not accessible: No such file or directory.

[ec2-user@mystack-wikidatabase-mq6q6753xjcm ~]$ sudo su - root

[ec2-user@mystack-wikidatabase-mq6q6753xjcm ~]$ cd /var/cache/yum/x86\_64/17/updates/packages/

[ec2-user@mystack-wikidatabase-mq6q6753xjcm packages]$ ll

total 0

-rw-r--r--. 1 root root 0 Oct 25 03:03 httpd-2.2.23-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 httpd-tools-2.2.23-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 libxslt-1.1.26-10.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 mysql-5.5.31-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 mysql-libs-5.5.31-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 mysql-server-5.5.31-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 perl-Compress-Raw-Bzip2-2.052-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 perl-Compress-Raw-Zlib-2.052-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 perl-IO-Compress-2.052-1.fc17.noarch.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 perl-IO-Tty-1.10-8.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 perl-Net-HTTP-6.06-1.fc17.noarch.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 perl-Net-SSLeay-1.48-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-cli-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-common-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-enchant-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-IDNA\_Convert-0.8.0-1.fc17.noarch.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-mbstring-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-mysql-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-mysqlnd-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-pdo-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-PHPMailer-5.2.6-1.fc17.noarch.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-process-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-simplepie-1.3.1-3.fc17.noarch.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 php-xml-5.4.15-1.fc17.x86\_64.rpm

-rw-r--r--. 1 root root 0 Oct 25 03:03 wordpress-3.5.1-2.fc17.noarch.rpm

[ec2-user@mystack-wikidatabase-mq6q6753xjcm ~]$ exit;

logout

Connection to 10.0.0.4 closed.

由上述的/var/cache/yum/x86\_64/17/updates/packages/目录可见，heat stack创建的虚拟机多了mysql的相关安装rpm包。

# 添加一个计算节点

**yum install openstack-utils**

**yum install openstack-selinux**

## 基础配置

### 系统安装配置

参见[2.1.1 系统安装配置](#_系统安装配置)

### 设置时间同步

1. 时间同步

# ntpdate 172.21.2.110

1. 设置计划任务

# crontab -e

0 23 \* \* \* ntpdate 172.21.2.110 >> /var/log/ntpdate.log

### 配置libvirtd服务

安装libvirt：

# yum install -y qemu-kvm libvirt

启动libvirt服务：

# service libvirtd start

查看net-list，发现default：

# virsh net-list

Name State Autostart Persistent

--------------------------------------------------

default active yes yes

删除default，即virbr0：

# virsh net-destroy default

Network default destroyed

# virsh net-undefine default

Network default has been undefined

重启libvirtd服务，设置开机自启动：

# service libvirtd restart

# chkconfig libvirtd on

## Neutron安装配置

### 初始化neutron-openvswitch

1. 安装openvswitch插件

# yum install -y openstack-neutron-openvswitch

### 配置neutron服务

1. 修改api-paste.ini文件

在api-paste.ini文件中添加keystone认证信息：

# vi /etc/neutron/api-paste.ini

……

[filter:authtoken]

paste.filter\_factory = keystoneclient.middleware.auth\_token:filter\_factory

auth\_host=172.21.2.110

admin\_user=neutron

admin\_tenant\_name=service

admin\_password=service

1. 配置neutron.conf文件

# cat /etc/neutron/neutron.conf | grep -v ^# | grep -v ^$

[DEFAULT]

core\_plugin = neutron.plugins.openvswitch.ovs\_neutron\_plugin.OVSNeutronPluginV2

auth\_strategy = keystone

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

qpid\_hostname = 172.21.2.110

control\_exchange = neutron

notification\_driver = neutron.openstack.common.notifier.rpc\_notifier

[quotas]

[agent]

root\_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf

[keystone\_authtoken]

auth\_host = 172.21.2.110

auth\_port = 35357

auth\_protocol = http

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = service

[database]

connection = mysql://neutron:neutron@172.21.2.110/ovs\_neutron

[service\_providers]

1. 修改ovs\_neutron\_plugin.ini文件

# cat /etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini | grep -v ^# | grep -v ^$

[ovs]

tenant\_network\_type = vlan

network\_vlan\_ranges = eth1:1000:2999

integration\_bridge = br-int

bridge\_mappings = eth1:br-int

[agent]

[securitygroup]

[DATABASE]

sql\_connection = mysql://neutron:neutron@172.21.2.110/ovs\_neutron

[SECURITYGROUP]

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

1. 修改dhcp及l3\_agent相关配置文件

# cat /etc/neutron/dhcp\_agent.ini | grep -v ^# | grep -v ^$

[DEFAULT]

auth\_url = <http://172.21.2.110:5000/v2.0>

admin\_username = neutron

admin\_password = service

admin\_tenant\_name = service

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

use\_namespaces = False

# cat /etc/neutron/l3\_agent.ini | grep -v ^# | grep -v ^$

[DEFAULT]

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

external\_network\_bridge = br-int

# cat /etc/neutron/metadata\_agent.ini | grep -v ^# | grep -v ^$

[DEFAULT]

auth\_url = [http://172.21.2.110:5000/v2.0](http://127.0.0.1:5000/v2.0)

auth\_region = RegionOne

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = service

1. 修改/etc/libvirt/qemu.conf

在文件末尾添加以下参数：

# vi /etc/libvirt/qemu.conf

……

cgroup\_device\_acl = [

"/dev/null", "/dev/full", "/dev/zero",

"/dev/random", "/dev/urandom",

"/dev/ptmx", "/dev/kvm", "/dev/kqemu",

"/dev/rtc","/dev/hpet","/dev/net/tun",

]

clear\_emulator\_capabilities = 0

user = "root"

group = "root"

重新启动libvirtd服务

# service libvirtd restart

1. 启动openvswitch服务，添加br-int

# service openvswitch start

# chkconfig openvswitch on

# ovs-vsctl add-br br-int

1. 启动neutron服务

# service neutron-openvswitch-agent start

## Nova安装配置

### 初始化nova-compute

# yum install -y openstack-nova-compute

### 配置nova服务

1. 修改nova.conf配置文件

# vi /etc/nova/nova.conf

[DEFAULT]

my\_ip = 172.21.2.111 //计算节点

auth\_strategy = keystone

state\_path = /openstack/lib/nova

verbose=True

allow\_resize\_to\_same\_host = true

rpc\_backend = nova.openstack.common.rpc.impl\_qpid

qpid\_hostname = 172.21.2.110

libvirt\_type = kvm

glance\_api\_servers = 172.21.2.110:9292

novncproxy\_base\_url = http://172.21.2.110:6080/vnc\_auto.html

vncserver\_listen = 172.21.2.111

vncserver\_proxyclient\_address = 172.21.2.111

vnc\_enabled = true

vnc\_keymap = en-us

network\_manager = nova.network.manager.FlatDHCPManager

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

multi\_host = True

flat\_interface = eth1

flat\_network\_bridge = br100

public\_interface = eth0

instance\_usage\_audit = True

instance\_usage\_audit\_period = hour

notify\_on\_state\_change = vm\_and\_task\_state

notification\_driver = nova.openstack.common.notifier.rpc\_notifier

notification\_driver = ceilometer.compute.nova\_notifier

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

neutron\_admin\_username = admin

neutron\_admin\_password = secrete

neutron\_admin\_auth\_url = http://172.21.2.110:5000/v2.0

neutron\_auth\_strategy = keystone

neutron\_admin\_tenant\_name = admin

neutron\_url = http://172.21.2.110:9696/

security\_group\_api = neutron

#libvirt\_vif\_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver

libvirt\_vif\_driver=nova.virt.libvirt.vif.LibvirtGenericVIFDriver

neutron\_ovs\_bridge = br-int

host=havana

default\_availability\_zone=nova

[hyperv]

[zookeeper]

[osapi\_v3]

[conductor]

[keymgr]

[cells]

[database]

connection=mysql://nova:nova@192.168.5.228/nova

[image\_file\_url]

[baremetal]

[rpc\_notifier2]

[matchmaker\_redis]

[ssl]

[trusted\_computing]

[upgrade\_levels]

[matchmaker\_ring]

[vmware]

[spice]

[keystone\_authtoken]

auth\_host = 172.21.2.110

auth\_port = 35357

auth\_protocol = http

admin\_user = nova

admin\_tenant\_name = service

admin\_password = service

因上述配置文件中修改了instance实例存放的位置，还需要作一下操作：

修改路径，设置目录权限：

# cp -r /var/lib/nova/ /openstack/lib/

# chown -R nova:nova /openstack/lib/nova/

1. 启动nova相关服务，设置开机自启动

# service messagebus start

# chkconfig messagebus on

启动nova服务

# service openstack-nova-compute start

# Havana测试

## 基本功能测试

### 测试方法

各组件基本功能的测试参见，第2章节中的测试部分。

### 测试结论

Openstack的基本功能（例如镜像管理、虚拟机创建、删除、安全组设置、网络创建等各组件的基本使用）性能都比较稳定，Havana版本的基本功能的测试均通过，不存在使用问题。

## 虚拟机资源调整测试

### 测试方法

虚拟机资源调整，包括内存、vcpu、硬盘等资源的调整。Kvm中虚拟机资源调整，不论内存还是硬盘均采用相同的调整方法，测试如下：

1. 查看虚拟机规格：

# nova flavor-list

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

| ID | Name | Memory\_MB | Disk | Ephemeral | Swap | VCPUs | RXTX\_Factor | Is\_Public |

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

| 1 | m1.tiny | 512 | 1 | 0 | | 1 | 1.0 | True |

| 2 | m1.small | 2048 | 20 | 0 | | 1 | 1.0 | True |

| 3 | m1.medium | 4096 | 40 | 0 | | 2 | 1.0 | True |

| 4 | m1.large | 8192 | 80 | 0 | | 4 | 1.0 | True |

| 5 | m1.xlarge | 16384 | 160 | 0 | | 8 | 1.0 | True |

| 6 | t1.small | 4096 | 20 | 0 | | 1 | 1.0 | True |

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

1. 创建虚拟机实例

# nova boot --flavor 2 --image rhel5.6 rvm01

1. 调整虚拟机资源

# nova resize rvm01 6

1. 确认是否调整

# nova resize-confirm rvm01 //确认调整

# nova resize-revert rvm01 //取消调整

### 测试结论

经过多次测试，H版虚拟机资源调整的功能较之前的F/G版的性能明显得到提升，kvm虚拟机实例在作资源调整过程中花费的时候与虚拟机实例使用的磁盘大小有关， 测试结果如下：

|  |  |
| --- | --- |
| 虚拟机实例大小 | 调整所用的时间 |
| 48G | 14m |
| 94G | 26m |
| 180G | 54m |

## 虚拟机迁移测试

Openstack中虚拟机迁移也实现了离线迁移和在线迁移两种。

### 测试方法

* **离线迁移测试方法**

1、修改用户nova，允许su成nova用户：

# usermod -s /bin/bash nova

2、通过nova用户来用ssh-keygen创建公钥：

# su - nova

# ssh-keygen -t rsa

输入后，会提示创建.ssh/id\_rsa、id\_rsa.pub的文件，其中第一个为密钥，第二个为公钥。过程中会要求输入密码，为了ssh访问过程无须密码，可以直接回车。

3、修改authorized\_keys权限为600

# mv /var/lib/nova/.ssh/id\_rsa.pub /var/lib/nova/.ssh/authorized\_keys

# chmod 600 /var/lib/nova/.ssh/authorized\_keys

4、将id\_rsa，authorized\_keys拷贝到其它计算节点

# su - root

# scp /var/lib/nova/.ssh/id\_rsa /var/lib/nova/.ssh/authorized\_keys 172.21.2.111:/var/lib/nova/.ssh/

# chown nova.nova /var/lib/nova/.ssh/authorized\_keys

# chown nova.nova /var/lib/nova/.ssh/id\_rsa

# chmod 600 /var/lib/nova/.ssh/authorized\_keys

# chmod 600 /var/lib/nova/.ssh/id\_rsa

5、测试信任关系

# su - nova

# ssh 172.21.2.111

6、迁移

[root@hctrl ~]# nova boot --flavor 2 --image rhel5.6 --availability\_zone nova:havana rvm01

[root@hctrl ~]# nova list

[root@hctrl ~]# nova migrate rvm01

[root@hctrl ~]# nova resize-confirm rvm01

* **在线迁移**

**控制节点：hctrl（172.21.2.110）**

**计算节点：hctrl（172.21.2.110）、havana（172.21.2.111）**

1. 设置共享存储

[root@hctrl ~]# vi /etc/exports

/openstack/lib/nova/instances/ 172.21.2.111(rw,no\_root\_squash,no\_all\_squash,sync)

[root@hctrl ~]# exportfs -r

[root@hctrl ~]# service nfs start

[root@havana ~]# showmount -e 172.21.2.110

Export list for 172.21.2.110:

/openstack/lib/nova/instances 172.21.2.111

[root@havana ~]# mount 172.21.2.110:/openstack/lib/nova/instances /openstack/lib/nova/instances

1. 建立计算节点间的信任关系

设置方法参加离线迁移中的1-5步。

1. 在计算节点上安装vncserver

[root@havana ~]# yum install -y vnc vnc-server tigervnc-server

[root@havana ~]# vi /etc/sysconfig/vncservers

VNCSERVERS="1:root 2:nova"

以vnc使用用户登录系统，创建vnc密码：

[root@havana ~]# vncserver

You will require a password to access your desktops.

Password:输入密码

Verify:再次输入密码

xauth: creating new authority file /root/.Xauthority

New 'havana:1 (root)' desktop is havana:1

Creating default startup script /root/.vnc/xstartup

Starting applications specified in /root/.vnc/xstartup

Log file is /root/.vnc/havana:1.log

查看已打开的vnc服务，关闭服务：

[root@havana ~]# vncserver -list

TigerVNC server sessions:

X DISPLAY # PROCESS ID

:1 32329

[root@havana ~]# vncserver -kill :1

Killing Xvnc process ID 32329

切换到nova用户：

[root@havana ~]# su - nova

-bash-4.1$ vncserver

You will require a password to access your desktops.

Password:

Verify:

xauth: creating new authority file /var/lib/nova/.Xauthority

New 'havana:2 (nova)' desktop is havana:2

Creating default startup script /var/lib/nova/.vnc/xstartup

Starting applications specified in /var/lib/nova/.vnc/xstartup

Log file is /var/lib/nova/.vnc/havana:2.log

-bash-4.1$ vncserver -list

TigerVNC server sessions:

X DISPLAY # PROCESS ID

:2 32562

-bash-4.1$ vncserver -kill :2

Killing Xvnc process ID 32562

-bash-4.1$ exit

logout

设置防护墙策略：

[root@havana ~]# iptables -I INPUT -p tcp --dport 5901 -j ACCEPT

[root@havana ~]# iptables -I INPUT -p tcp --dport 5902 -j ACCEPT

[root@havana ~]# service iptables save

1. 修改nova.conf文件

# vi /etc/nova/nova.conf

vncserver\_listen = 0.0.0.0

1. 重新启动nova-compute服务

# service openstack-nova-compute restart

1. 创建虚拟机实例

[root@hctrl ~]# nova boot --flavor 2 --image rhel5.6 --availability\_zone nova:havana rvm01

[root@hctrl ~]# nova live-migration rvm01 hctrl

### 测试结论

Havana版本的虚拟机迁移测试通过。

## F版到H版升级测试

### 测试方法

F版与H版数据库变化较大，无法直接进行升级，采用虚拟机快照备份的方式从F版将虚拟机迁移到H版。测试方法如下：

1. 在F版环境中，将需要迁移的虚拟机进行快照

[root@folsom ~]# nova image-create rvm01 rvm01-snap

1. 查看H版导入的项目id

[root@hctrl ~]# keystone tenant-list

1. 导出glance中虚拟机快照的相关信息

use glance;

select id,name,size,status,is\_public,created\_at,updated\_at,deleted\_at,deleted,disk\_format,container\_format,checksum,'$tenant\_id',min\_disk,min\_ram,protected from images where id='$image\_id' into outfile '$filepath1';

select id,location,updated\_at,updated\_at,deleted\_at,deleted,'{}' from images where id='$image\_id' into outfile '$filepath2';

其中$image\_id为快照的id，$filepath1及$filepath2为导出数据存放的路径，$tenant\_id为H版导入的tenantid

1. 将导出的数据文件拷贝到H版环境中，导入H版数据库中

use glance;

load data local infile '$filepath1' into table images(id,name,size,status,is\_public,created\_at,updated\_at,deleted\_at,deleted,disk\_format,container\_format,checksum,owner,min\_disk,min\_ram,protected);

load data local infile '$filepath2' into table image\_locations(image\_id,value,created\_at,updated\_at,deleted\_at,deleted,meta\_data);

1. 将F版rvm01-snap的images文件拷贝到H版的glance路径下

为节省时间，可将F版glance的images路径共享，挂载到H版glance的images路径上。

1. 在H版的环境下，通过使用rvm01-snap镜像来恢复虚拟机rvm01

[root@hctrl ~]# nova boot --flavor 2 --image rvm01-snap rvm01

### 测试结论

经测试，此种通过虚拟机快照备份的方式将F版的虚拟机实例迁移到H版的环境中升级方式可行。虚拟机实例可恢复到创建快照备份的时间点。

## 其他新组件的测试

其他新组件如ceilometer、heat等的测试参见[2.9.4Ceilometer测试](#_Ceilometer测试（监控glance、nova、neturon、)及[2.10.4Heat测试](#_Heat测试)。

这两个组件目前功能可使用，针对ceilometer有一定的采集数据，目前测试过程中监控一个计算节点没问题，但是监控多个计算节点时，采集程序出现问题，无法同时采集多个计算节点的数据，目前还未解决。

Heat的基本功能测试通过，目前对其原理了解有限，具体的使用场景不清楚。

yum install git --disablerepo=rpmforge //排除使用rpmforge源来更新，因为该源不稳定

yum install -y openstack-keystone --disablerepo=rpmforge