**Openstack\_setup**

安全公司\_技术研究部

姓名：范迪



**北京神州泰岳软件股份有限公司**

目录

目录

[目录 2](#_Toc435196964)

[1 基础环境配置 10](#_Toc435196965)

[1.1 组件信息 10](#_Toc435196966)

[1.1.1 组件功能描述 10](#_Toc435196967)

[1.1.2 组件间关系 11](#_Toc435196968)

[1.1.3 组件部署 11](#_Toc435196969)

[1.2 网络信息 12](#_Toc435196970)

[1.3 硬件信息 13](#_Toc435196971)

[1.4 密码表(本测试环境使用) 14](#_Toc435196972)

[2 Openstack源更新（所有节点都配置） 17](#_Toc435196973)

[3 Ntp server端配置（controller） 18](#_Toc435196974)

[3.1 安装ntp 18](#_Toc435196975)

[3.2 配置/etc/ntp.conf 18](#_Toc435196976)

[3.3 重新启动ntp服务 19](#_Toc435196977)

[3.4 查看ntp配置是否成功 19](#_Toc435196978)

[Ntp client(computer、network、block1节点) 19](#_Toc435196979)

[4 安装mysql数据库\_controller 20](#_Toc435196980)

[4.1 安装数据库包 20](#_Toc435196981)

[4.2 设置root登陆访问的密码 20](#_Toc435196982)

[4.3 编辑mysql配置文件/etc/mysql/my.cnf： 20](#_Toc435196983)

[4.4 重启数据库服务 21](#_Toc435196984)

[4.5 数据库安全初始化 21](#_Toc435196985)

[5 消息队列安装\_controller 24](#_Toc435196986)

[5.1 安装 24](#_Toc435196987)

[5.2 修改guest默认密码 24](#_Toc435196988)

[5.3 开启页面控制台 24](#_Toc435196989)

[5.4 如果是RabbitMQ3.3.0或者更新的版本，需要打开guest的远程访问权限 25](#_Toc435196990)

[5.5 重启rabbitmq 并登陆页面测试 25](#_Toc435196991)

[6 认证组件(keystone)\_controller 27](#_Toc435196992)

[6.1 创建keystone库 27](#_Toc435196993)

[6.2 安装keystone包 28](#_Toc435196994)

[6.3 生成随机令牌 31](#_Toc435196995)

[6.4 编辑配置文件/etc/keystone/keystone.conf 31](#_Toc435196996)

[6.5 数据同步 32](#_Toc435196997)

[6.6 重启keystone服务 32](#_Toc435196998)

[6.7 建计划任务，定期清理过期令牌 33](#_Toc435196999)

[6.8 创建用户角色等 33](#_Toc435197000)

[6.9 创建service entity and API endpoints（表：service 和endpoints） 37](#_Toc435197001)

[6.10 验证keystone可用性 38](#_Toc435197002)

[6.11 新建keystone客户端脚本 40](#_Toc435197003)

[7 安装Image Service(glance)\_controller 42](#_Toc435197004)

[7.1 创建glance库及用户 42](#_Toc435197005)

[7.2 运行keystong客户端脚本（上个章节创建的） 43](#_Toc435197006)

[7.3 创建凭证和glance服务API 43](#_Toc435197007)

[7.4 安装glance 45](#_Toc435197008)

[7.5 配置glance 45](#_Toc435197009)

[7.5.1 /etc/glance/glance-api.conf 45](#_Toc435197010)

[7.5.2 /etc/glance/glance-registry.conf 47](#_Toc435197011)

[7.6 数据同步 48](#_Toc435197012)

[7.7 服务的重新启动 49](#_Toc435197013)

[7.8 验证glance可用性 49](#_Toc435197014)

[8 安装computer service(nova) 52](#_Toc435197015)

[8.1 Controller节点 52](#_Toc435197016)

[8.1.1 创建数据库 52](#_Toc435197017)

[8.1.2 执行keystone客户端脚本 53](#_Toc435197018)

[8.1.3 创建凭证和nova-api 53](#_Toc435197019)

[8.1.4 Nova安装 55](#_Toc435197020)

[8.1.5 配置nova( /etc/nova/nova.conf) 55](#_Toc435197021)

[8.1.6 数据同步 57](#_Toc435197022)

[8.1.7 服务的重新启动 57](#_Toc435197023)

[8.2 Computer节点 58](#_Toc435197024)

[8.2.1 安装包 58](#_Toc435197025)

[8.2.2 配置nova：/etc/nova/nova.conf 58](#_Toc435197026)

[8.2.3 完成安装：/etc/nova/nova-compute.conf 60](#_Toc435197027)

[8.2.4 服务重新启动 61](#_Toc435197028)

[8.3 验证nova可用性 61](#_Toc435197029)

[9 网络组件安装（neutron-server） 63](#_Toc435197030)

[9.1 Controller节点 63](#_Toc435197031)

[9.1.1 创建数据库 63](#_Toc435197032)

[9.1.2 执行keystone客户端脚本 64](#_Toc435197033)

[9.1.3 创建凭证和neutron-api 64](#_Toc435197034)

[9.1.4 网络组件安装 66](#_Toc435197035)

[9.1.5 配置网络组件 66](#_Toc435197036)

[9.1.6 数据同步 71](#_Toc435197037)

[9.1.7 重启新启动服务 71](#_Toc435197038)

[9.1.8 验证neutron可用性 72](#_Toc435197039)

[9.2 Network节点 73](#_Toc435197040)

[9.2.1 计算机属性配置 73](#_Toc435197041)

[9.2.2 安装包 73](#_Toc435197042)

[9.2.3 配置neutron 74](#_Toc435197043)

[9.2.4 服务的重新启动 79](#_Toc435197044)

[9.2.5 验证可用性 79](#_Toc435197045)

[9.3 Computer节点 81](#_Toc435197046)

[9.3.1 计算机属性配置/etc/sysctl.conf 81](#_Toc435197047)

[9.3.2 安装包 81](#_Toc435197048)

[9.3.3 配置网络组件 81](#_Toc435197049)

[9.3.4 服务的重新启动（log-error） 85](#_Toc435197050)

[9.3.5 验证可用性 85](#_Toc435197051)

[9.4 初始化网络 87](#_Toc435197052)

[9.4.1 创建外部网络 87](#_Toc435197053)

[9.4.2 创建用户网络 89](#_Toc435197054)

[10 管理面板安装(dashboard)\_controller 93](#_Toc435197055)

[10.1 安装 93](#_Toc435197056)

[10.2 配置管理面板 93](#_Toc435197057)

[10.3 重新启动服务 94](#_Toc435197058)

[10.4 验证可用性 94](#_Toc435197059)

[11 块存储服务安装 96](#_Toc435197060)

[11.1 Controller端安装（cinder） 96](#_Toc435197061)

[11.1.1 数据库创建 96](#_Toc435197062)

[11.1.2 创建凭证和api 97](#_Toc435197063)

[11.1.3 安装cinder 100](#_Toc435197064)

[11.1.4 配置cinder 100](#_Toc435197065)

[11.1.5 同步数据 102](#_Toc435197066)

[11.1.6 重启服务 102](#_Toc435197067)

[11.2 Block节点 102](#_Toc435197068)

[11.2.1 Ntp配置 102](#_Toc435197069)

[11.2.2 安装lvm包 103](#_Toc435197070)

[11.2.3 创建卷组 103](#_Toc435197071)

[11.2.4 安装cinder-volume python-mysqldb包 104](#_Toc435197072)

[11.2.5 编辑配置文件 104](#_Toc435197073)

[11.2.6 重启服务 106](#_Toc435197074)

[11.3 验证块存储的可用用 106](#_Toc435197075)

[12 对象存储安装(swift) 110](#_Toc435197076)

[12.1 Controller节点 110](#_Toc435197077)

[12.1.1 创建用户信息及api 110](#_Toc435197078)

[12.1.2 安装包 112](#_Toc435197079)

[12.1.3 配置swift 112](#_Toc435197080)

[12.2 Block节点 114](#_Toc435197081)

[12.2.1 配置ip 114](#_Toc435197082)

[12.2.2 配置ntp 114](#_Toc435197083)

[12.2.3 安装rsync 114](#_Toc435197084)

[12.2.4 配置rsync 115](#_Toc435197085)

[12.2.5 重新启动rsync服务 118](#_Toc435197086)

[12.2.6 安装配置swift组件包 118](#_Toc435197087)

[12.2.7 下载配置文件，共3个 118](#_Toc435197088)

[12.2.8 配置swif 119](#_Toc435197089)

[12.2.9 创建Rings(controller节点) 121](#_Toc435197090)

[12.2.1 权限等剩余工作 124](#_Toc435197091)

[12.3 重新启动swift 124](#_Toc435197092)

[12.4 验证可用性 125](#_Toc435197093)

[13 编制模块安装(heat)\_controller 127](#_Toc435197094)

[13.1 创建数据库 127](#_Toc435197095)

[13.2 执行keystone客户端脚本 128](#_Toc435197096)

[13.3 创建凭证和api（9条） 128](#_Toc435197097)

[13.4 安装包 130](#_Toc435197098)

[13.5 配置heat 131](#_Toc435197099)

[13.6 数据同步 131](#_Toc435197100)

[13.7 服务的重新启动 132](#_Toc435197101)

[13.8 验证可用性 132](#_Toc435197102)

[14 安装临时模块 134](#_Toc435197103)

[14.1 Controller节点 134](#_Toc435197104)

[14.1.1 安装mongoDB包 134](#_Toc435197105)

[14.1.2 配置mongodb，重启数据库mongodb 134](#_Toc435197106)

[14.1.3 创建临时库mongo 134](#_Toc435197107)

[14.1.4 执行keystone客户端脚本admin 135](#_Toc435197108)

[14.1.5 创建用户角色api等 135](#_Toc435197109)

[14.1.6 安装临时模块组件包 136](#_Toc435197110)

[14.1.7 生成随机码 136](#_Toc435197111)

[14.1.8 配置临时模块 136](#_Toc435197112)

[14.1.9 重新启动服务 137](#_Toc435197113)

[14.2 Computer节点 138](#_Toc435197114)

[14.2.1 安装ceilometer-agent-compute包 138](#_Toc435197115)

[14.2.2 配置ceilometer 138](#_Toc435197116)

[14.2.3 编辑nova.conf 139](#_Toc435197117)

[14.2.4 重新启动服务 139](#_Toc435197118)

[14.3 配置image（controller节点） 140](#_Toc435197119)

[14.3.1 编辑配置文件/etc/glance/glance-api.conf 140](#_Toc435197120)

[14.3.2 编辑/etc/glance/glance-registry.conf 140](#_Toc435197121)

[14.3.3 重启image服务 140](#_Toc435197122)

[14.4 配置cinder（controller节点、block1节点） 141](#_Toc435197123)

[14.4.1 编辑/etc/cinder/cinder.conf controller节点 141](#_Toc435197124)

[14.4.2 编辑/etc/cinder/cinder.conf block1节点 141](#_Toc435197125)

[14.4.3 重新启动服务 141](#_Toc435197126)

[14.5 配置swift（controller） 142](#_Toc435197127)

[14.5.1 To configure prerequisites（controller） 142](#_Toc435197128)

[14.5.2 To configure notifications（controller） 142](#_Toc435197129)

[14.5.3 给权限 143](#_Toc435197130)

[14.5.4 重启服务（error，最后该pipeline属性没有改动，启动成功） 143](#_Toc435197131)

[15 创建实例 144](#_Toc435197132)

[15.1 创建实例前配置nova 144](#_Toc435197133)

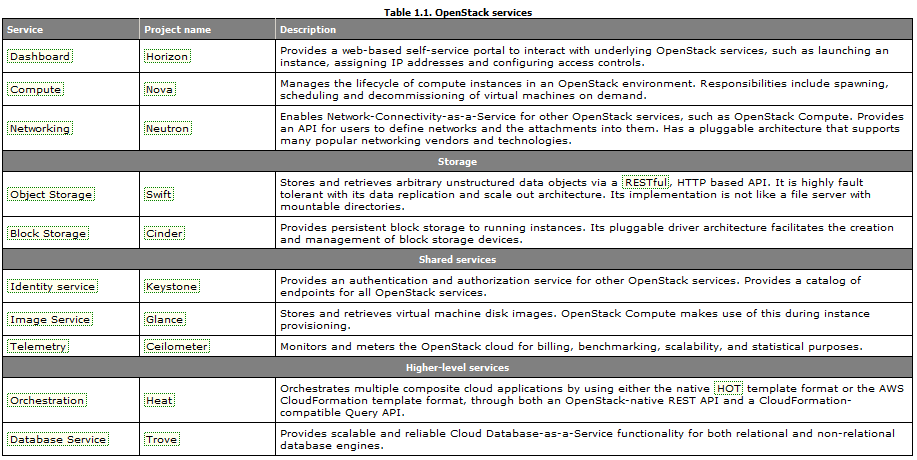
[15.2 创建实例 144](#_Toc435197134)

[15.3 登陆实例 149](#_Toc435197135)

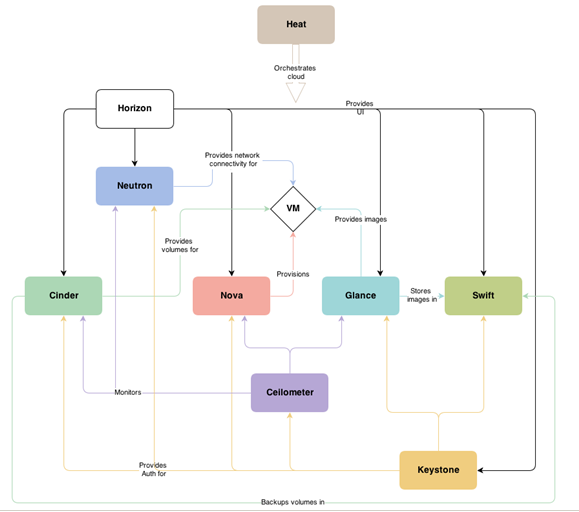
# 基础环境配置

## 组件信息

### 组件功能描述



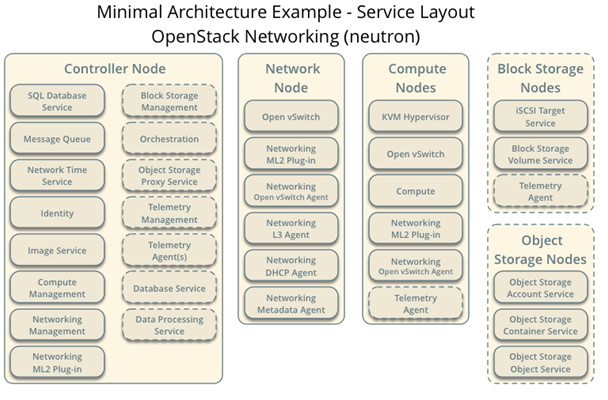
### 组件间关系



### 组件部署

**如下图：**

本环境storage为一个节点block1



## 网络信息

**网络ip列表:**

|  |  |  |
| --- | --- | --- |
| 节点（主机名） | Work Interface (eth1) | Managent Interface( eth0) |
| Controller (unbuntu10) | 10.0.0.10 | 192.168.95.140 |
| Computer1 (computer1) | 10.0.0.11 | 192.168.95.141 |
| Network (network) | 10.0.0.12 | 192.168.95.142 |
| Block1 (block1) | 10.0.0.13 | 192.168.95.143 |

特别注释：network节点上有第三块网卡，eth2，用于绑定br-ex虚拟网桥，在网络组件搭建中可见,在网络初始化的时候ext-net使用192.168.95.0网段，demo-net使用10.0.1.0网段，这个demo-net网段自己定，这里的controller最好主机名使用controller

**配置临时ip,方法，例如**

$sudo ifconfig eth1 10.0.0.10 netmask 255.255.255.0

**配置hosts文件（所有节点）**

127.0.0.1 localhost

10.0.0.10 ubuntu-10

10.0.0.10 controller

10.0.0.11 computer1

10.0.0.12 network

10.0.0.13 block1

## 硬件信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **节点** | **主机名** | **Ip** | **内存** | **磁盘** | **系统版本** |
| Controller | ubuntu10 | 10.0.0.10  192.168.95.140 | 8G | 50G | Ubuntu 14.04.2 LTS \n \l |
| Computer | computer1 | 10.0.0.10  192.168.95.140 | 4G | 50G | Ubuntu 14.04.2 LTS \n \l |
| Network | network | 10.0.0.10  192.168.95.140 | 4G | 50G | Ubuntu 14.04.2 LTS \n \l |
| Block | block1 | 10.0.0.10  192.168.95.140 | 4G | 50Gx2  其中一块（3个分区sdb1、sdb2、sdb3） | Ubuntu 14.04.2 LTS \n \l |

## 密码表(本测试环境使用)

搭建过程中强烈建议修改复杂密码

| **Table 2.1. Passwords** | |
| --- | --- |
| **Password name** | **Description** |
| Database password (no variable used) | Root password for the database |
| *RABBIT\_PASS* | Password of user guest of RabbitMQ |
| *KEYSTONE\_DBPASS* | Database password of Identity service |
| *DEMO\_PASS* | Password of user demo |
| *ADMIN\_PASS* | Password of user admin |
| *GLANCE\_DBPASS* | Database password for Image Service |
| *GLANCE\_PASS* | Password of Image Service user glance |
| *NOVA\_DBPASS* | Database password for Compute service |
| *NOVA\_PASS* | Password of Compute service user nova |
| *DASH\_DBPASS* | Database password for the dashboard |
| *CINDER\_DBPASS* | Database password for the Block Storage service |
| *CINDER\_PASS* | Password of Block Storage service user cinder |
| *NEUTRON\_DBPASS* | Database password for the Networking service |
| *NEUTRON\_PASS* | Password of Networking service user neutron |
| *HEAT\_DBPASS* | Database password for the Orchestration service |
| *HEAT\_PASS* | Password of Orchestration service user heat |
| *CEILOMETER\_DBPASS* | Database password for the Telemetry service |
| *CEILOMETER\_PASS* | Password of Telemetry service user ceilometer |
| *TROVE\_DBPASS* | Database password of Database service |
| *TROVE\_PASS* | Password of Database Service user trove |

# Openstack源更新（所有节点都配置）

该操作的作用是：使所有节点的安装源一致，从而使安装组件包的版本一致，否则会出现难以解决的问题

# apt-get install ubuntu-cloud-keyring

下面是一行命令，用于添加一个配置文件：

#echo "deb http://ubuntu-cloud.archive.canonical.com/ubuntu" "trusty-updates/juno main" > /etc/apt/sources.list.d/cloudarchive-juno.list

完成安装，这一步安装的东西比较多，时间上会长一点

# apt-get update && apt-get dist-upgrade

# Ntp server端配置（controller）

## 安装ntp

# apt-get install ntp

## 配置/etc/ntp.conf

**主要配置几行如下：**

注释掉的部分：

#server 0.ubuntu.pool.ntp.org

#server 1.ubuntu.pool.ntp.org

#server 2.ubuntu.pool.ntp.org

#server 3.ubuntu.pool.ntp.org

#server ntp.ubuntu.com

添加的部分：

server 127.127.1.0

fudge 127.127.1.0 stratum 10

restrict 10.0.0.0 mask 255.255.255.0 nomodify notrap

## 重新启动ntp服务

#service ntp restart

## 查看ntp配置是否成功

**如下及成功**

# ntpq -p

remote refid st t when poll reach delay offset jitter

=================================================================

\*LOCAL(0) .LOCL. 10 l 19 64 77 0.000 0.000 0.000

# Ntp client(computer、network、block1节点)

创建计划任务 crontab –e 每15分钟与controller同步一次时间

\*/15 \* \* \* \* sudo ntpdate 10.0.0.10

# 安装mysql数据库\_controller

## 安装数据库包

# apt-get install mariadb-server python-mysqldb

## 设置root登陆访问的密码

安装过程中设置：root 密码为openstack

Root登陆的一些技巧 |

Root本地登陆：

#mysql –uroot –popenstsck

简单的sql语句：

查看远程访问权限：

use mysql;

select host,user,password from user;

其中 host的% 是所有远程登录的意思。

## 编辑mysql配置文件/etc/mysql/my.cnf：

修改1（把地址修改成该主机分配的业务ip，请看配置信息章节里面的列表）：

[mysqld]

...

bind-address = 10.0.0.10

修改2(修改字符集)：[mysqld]栏内 最后部分添加如下

[mysqld]

default-storage-engine = innodb

innodb\_file\_per\_table

collation-server = utf8\_general\_ci

init-connect = 'SET NAMES utf8'

character-set-server = utf8

## 重启数据库服务

# service mysql restart

## 数据库安全初始化

包括修改root密码

删除匿名登陆

删除root远程登录

删除test库

重新加载权限

…

root@ubuntu-10:/etc/mysql# mysql\_secure\_installation

/usr/bin/mysql\_secure\_installation: 379: /usr/bin/mysql\_secure\_installation: find\_mysql\_client: not found

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB

SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current

password for the root user. If you've just installed MariaDB, and

you haven't set the root password yet, the password will be blank,

so you should just press enter here.

Enter current password for root (enter for none):

OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB

root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] y

New password:

Re-enter new password:

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone

to log into MariaDB without having to have a user account created for

them. This is intended only for testing, and to make the installation

go a bit smoother. You should remove them before moving into a

production environment.

Remove anonymous users? [Y/n] y

... Success!

Normally, root should only be allowed to connect from 'localhost'. This

ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y

... Success!

By default, MariaDB comes with a database named 'test' that anyone can

access. This is also intended only for testing, and should be removed

before moving into a production environment.

Remove test database and access to it? [Y/n] y

- Dropping test database...

ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist

... Failed! Not critical, keep moving...

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far

will take effect immediately.

Reload privilege tables now? [Y/n] y

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB

installation should now be secure.

Thanks for using MariaDB!

# 消息队列安装\_controller

## 安装

# apt-get install rabbitmq-server

## 修改guest默认密码

# rabbitmqctl change\_password guest RABBIT\_PASS

Changing password for user "guest" ...

...done.

## 开启页面控制台

root@ubuntu-10:/etc/rabbitmq# /usr/sbin/rabbitmq-plugins enable rabbitmq\_management

The following plugins have been enabled:

mochiweb

webmachine

rabbitmq\_web\_dispatch

amqp\_client

rabbitmq\_management\_agent

rabbitmq\_management

Plugin configuration has changed. Restart RabbitMQ for changes to take effect.

root@ubuntu-10:/etc/rabbitmq# service rabbitmq-server restart

\* Restarting message broker rabbitmq-server

...done.

## 如果是RabbitMQ3.3.0或者更新的版本，需要打开guest的远程访问权限

1. 检查rabbitmq版本

# rabbitmqctl status | grep rabbit

我这里的是3.2.4版本，guest有远程的权限，就不用修改了

1. 如果是3.3.0或者更新的，修改配置文件/etc/rabbitmq/rabbitmq.config

[{rabbit, [{loopback\_users, []}]}].

1. 重启服务

#service rabbitmq-server restart

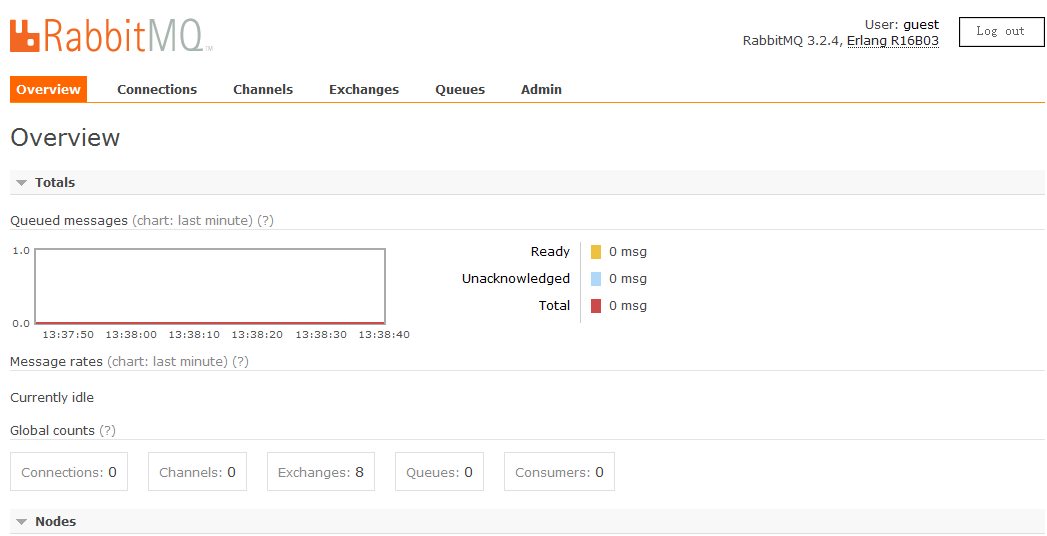
## 重启rabbitmq 并登陆页面测试

# service rabbitmq-server restart

打开与主机网络相通的ie浏览器

<http://ip:15672>





看到如上信息及ok

# 认证组件(keystone)\_controller

## 创建keystone库

Root用户登陆mysql数据库

创建keystone库

授权登陆用户keystone 密码为：KEYSTONE\_DBPASS（本地和远程两项）

操作如下：

root@ubuntu-10:/etc/rabbitmq# mysql -uroot -popenstack

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 38

Server version: 5.5.44-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [mysql]> CREATE DATABASE keystone;

Query OK, 1 row affected (0.00 sec)

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

Query OK, 0 rows affected (0.00 sec)

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

Query OK, 0 rows affected (0.00 sec)

## 安装keystone包

# apt-get install keystone python-keystoneclient

安装过程中出现错误

E: Failed to fetch http://ubuntu-cloud.archive.canonical.com/ubuntu/pool/main/m/migrate/python-migrate\_0.9.1-1ubuntu1~cloud2\_all.deb Cannot initiate the connection to ubuntu-cloud.archive.canonical.com:80 (2001:67c:1360:8c01::23). - connect (101: Network

修改dns为：/etc/resolv.conf

Nameserver 8.8.8.8

# apt-get update

如果还是报错，那么再次执行

# apt-get update

# apt-get install keystone python-keystoneclient

直到正确安装

安装的包如下：

The following extra packages will be installed:

alembic ieee-data libgmp10 libjs-jquery libjs-sphinxdoc libjs-underscore

librabbitmq1 libxslt1.1 libyaml-0-2 python-alembic python-amqp

python-anyjson python-babel python-babel-localedata python-crypto

python-decorator python-dns python-dogpile.cache python-dogpile.core

python-eventlet python-formencode python-greenlet python-iso8601

python-jsonschema python-keystone python-keystonemiddleware python-kombu

python-ldap python-ldappool python-librabbitmq python-lockfile python-lxml

python-mako python-markupsafe python-migrate python-mock python-netaddr

python-oauthlib python-openid python-oslo.config python-oslo.db

python-oslo.i18n python-oslo.messaging python-oslo.serialization

python-oslo.utils python-passlib python-paste python-pastedeploy

python-pastedeploy-tpl python-pastescript python-pbr python-pkg-resources

python-posix-ipc python-prettytable python-pycadf python-repoze.lru

python-routes python-scgi python-setuptools python-six python-sqlalchemy

python-sqlalchemy-ext python-stevedore python-tempita python-tz python-webob

python-yaml ssl-cert

Suggested packages:

python3-alembic javascript-common python-amqp-doc python-crypto-dbg

python-crypto-doc python-egenix-mxdatetime python-greenlet-doc

python-greenlet-dev python-greenlet-dbg python-memcache python-boto

python-beanstalkc python-django python-kombu-doc python-pika python-pymongo

python-ldap-doc python-pyasn1 python-lxml-dbg python-beaker python-mako-doc

python-mock-doc ipython python-netaddr-docs python-pastewebkit

libjs-mochikit libapache2-mod-wsgi libapache2-mod-python libapache2-mod-scgi

python-pgsql python-flup python-cherrypy python-cheetah python-distribute

python-distribute-doc python-sqlalchemy-doc python-psycopg2

python-kinterbasdb python-pymssql python-webob-doc openssl-blacklist

The following NEW packages will be installed:

alembic ieee-data keystone libgmp10 libjs-jquery libjs-sphinxdoc

libjs-underscore librabbitmq1 libxslt1.1 libyaml-0-2 python-alembic

python-amqp python-anyjson python-babel python-babel-localedata

python-crypto python-decorator python-dns python-dogpile.cache

python-dogpile.core python-eventlet python-formencode python-greenlet

python-iso8601 python-jsonschema python-keystone python-keystoneclient

python-keystonemiddleware python-kombu python-ldap python-ldappool

python-librabbitmq python-lockfile python-lxml python-mako python-markupsafe

python-migrate python-mock python-netaddr python-oauthlib python-openid

python-oslo.config python-oslo.db python-oslo.i18n python-oslo.messaging

python-oslo.serialization python-oslo.utils python-passlib python-paste

python-pastedeploy python-pastedeploy-tpl python-pastescript python-pbr

python-posix-ipc python-prettytable python-pycadf python-repoze.lru

python-routes python-scgi python-setuptools python-sqlalchemy

python-sqlalchemy-ext python-stevedore python-tempita python-tz python-webob

python-yaml ssl-cert

## 生成随机令牌

root@ubuntu-10:/etc/rabbitmq# openssl rand -hex 10

f172571661ab6bd8d89f

## 编辑配置文件/etc/keystone/keystone.conf

1. **修改默认值和启用详细日志**

在2行插入如下2行：这串数字是“生成随机令牌章节时生成的”

[default]

admin\_token = f172571661ab6bd8d89f

verbose = true

1. **修改数据库连接**

在635行左右配置如下，注释原有的连接，添加mysql连接：

[database]

#connection=sqlite:////var/lib/keystone/keystone.db

connection = mysql://keystone:KEYSTONE\_DBPASS@10.0.0.10/keystone

1. **sql驱动**

在1516行左右，插入两行如下：

[token]

provider = keystone.token.providers.uuid.Provider

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

1. **sql撤销驱动**

在1330行左右，插入一行如下：

[revoke]

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

保存文件退出。

## 数据同步

root@ubuntu-10:/etc/keystone# su -s /bin/sh -c "keystone-manage db\_sync" keystone

## 重启keystone服务

root@ubuntu-10:/etc/keystone# service keystone restart

keystone stop/waiting

keystone start/running, process 15269

## 建计划任务，定期清理过期令牌

# (crontab -l -u keystone 2>&1 | grep -q token\_flush) || echo '@hourly /usr/bin/keystone-manage token\_flush >/var/log/keystone/keystone-tokenflush.log 2>&1' >> /var/spool/cron/crontabs/keystone

## 创建用户角色等

**注释重点：**

**第一保证：生成随机令牌，之前生成过！！！请确认**

root@ubuntu-10:/etc/rabbitmq# openssl rand -hex 10

f172571661ab6bd8d89f

**第二保证：修改配置文件，重启服务，这项之前配置过！！！请确认**

/etc/keystone/keystone.conf

[default]

admin\_token = f172571661ab6bd8d89f

#service keystone restart

**第三保证：执行环境变量，等号之间不能有空格**

#export OS\_SERVICE\_TOKEN=f172571661ab6bd8d89f

#export OS\_SERVICE\_ENDPOINT=http://localhost:35357/v2.0

**开始创建**

**1.Create an administrative tenant, user, and role**

root@ubuntu-10:/etc/keystone# keystone tenant-create --name admin --description "Admin Tenant"

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

| Property | Value |

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

| description | Admin Tenant |

| enabled | True |

| id | 185f3d3f7d134c9b87f1045f7d73e8cf |

| name | admin |

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

root@ubuntu-10:/etc/keystone# keystone user-create --name admin --pass ADMIN\_PASS --email EMAIL\_ADDRESS

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

| Property | Value |

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

| email | EMAIL\_ADDRESS |

| enabled | True |

| id | aad71857b78c46269464b064ba8c2cfc |

| name | admin |

| username | admin |

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

root@ubuntu-10:/etc/keystone# keystone role-create --name admin

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

| Property | Value |

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

| id | bda6906df3734cc384e41c6fd96f255e |

| name | admin |

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

root@ubuntu-10:/etc/keystone# keystone user-role-add --user admin --tenant admin --role admin

1. **Create a demo tenant and user**

root@ubuntu-10:/etc/keystone# keystone tenant-create --name demo --description "Demo Tenant"

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

| Property | Value |

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

| description | Demo Tenant |

| enabled | True |

| id | 8293772b681c43cb86e670c42d5d8701 |

| name | demo |

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

root@ubuntu-10:/etc/keystone# keystone user-create --name demo --tenant demo --pass DEMO\_PASS --email EMAIL\_ADDRESS

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

| Property | Value |

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

| email | EMAIL\_ADDRESS |

| enabled | True |

| id | 5e036a7b49fb46b39c839df0aeb3cd56 |

| name | demo |

| tenantId | 8293772b681c43cb86e670c42d5d8701 |

| username | demo |

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

1. **OpenStack services also require a tenant, user, and role**

root@ubuntu-10:/etc/keystone# keystone tenant-create --name service --description "Service Tenant"

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

| Property | Value |

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

| description | Service Tenant |

| enabled | True |

| id | 2e79e3d4585a4511b11f2a988a12a1aa |

| name | service |

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

## 创建service entity and API endpoints（表：service 和endpoints）

这个里面的ip一定要写正确，否则后面验证的时候，找不到项，不能连接，也许只有一个token-get能获取，其他的全不能获取。

root@ubuntu-10:/etc/keystone# keystone service-create --name keystone --type identity --description "OpenStack Identity"

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

| Property | Value |

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

| description | OpenStack Identity |

| enabled | True |

| id | 837f72f3b2d941048e31419505d8b448 |

| name | keystone |

| type | identity |

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

root@ubuntu-10:/etc/keystone# keystone endpoint-create --service-id $(keystone service-list | awk '/ identity / {print $2}') --publicurl http://10.0.0.10:5000/v2.0 \

> --internalurl <http://10.0>.0.10:5000/v2.0 \

> --adminurl http://10.0.0.10:35357/v2.0 \

> --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:35357/v2.0 |

| id | 1c1eaf55655844f68ad9cf8f11b818ba |

| internalurl | http://10.0.0.10:5000/v2.0 |

| publicurl | http://10.0.0.10:5000/v2.0 |

| region | regionOne |

| service\_id | 837f72f3b2d941048e31419505d8b448 |

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

## 验证keystone可用性

先把环境变量都unset，填写参数的形式，相当于客户端去认证,当出现某个参数项，不能连接的问题，那么问题应该出现在上面章节的service和endpoints创建有问题，需要清空service和endpoints两个表，然后重新使用上个章节的命令，插入数据；

验证的项有：

Admin：token-get、tenant-list、user-list、role-list

Demo：token-get、user-list（该项demo账号没有权限访问是正确的）

root@ubuntu-10:/etc/keystone# unset OS\_SERVICE\_TOKEN OS\_SERVICE\_ENDPOINT

root@ubuntu-10:/etc/keystone# keystone --os-tenant-name admin --os-username admin --os-password ADMIN\_PASS --os-auth-url http://10.0.0.10:35357/v2.0 token-get

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

| Property | Value |

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

| expires | 2015-11-03T04:23:32Z |

| id | c73c9988229a41158b5e8c56e8b79da0 |

| tenant\_id | 185f3d3f7d134c9b87f1045f7d73e8cf |

| user\_id | aad71857b78c46269464b064ba8c2cfc |

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

root@ubuntu-10:/etc/keystone# keystone --os-tenant-name admin --os-username admin --os-password ADMIN\_PASS --os-auth-url http://10.0.0.10:35357/v2.0 tenant-list

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

| id | name | enabled |

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

| 185f3d3f7d134c9b87f1045f7d73e8cf | admin | True |

| 8293772b681c43cb86e670c42d5d8701 | demo | True |

| 2e79e3d4585a4511b11f2a988a12a1aa | service | True |

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

root@ubuntu-10:/etc/keystone# keystone --os-tenant-name admin --os-username admin --os-password ADMIN\_PASS --os-auth-url http://10.0.0.10:35357/v2.0 user-list

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

| id | name | enabled | email |

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

| aad71857b78c46269464b064ba8c2cfc | admin | True | EMAIL\_ADDRESS |

| 5e036a7b49fb46b39c839df0aeb3cd56 | demo | True | EMAIL\_ADDRESS |

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

root@ubuntu-10:/etc/keystone# keystone --os-tenant-name admin --os-username admin --os-password ADMIN\_PASS --os-auth-url http://10.0.0.10:35357/v2.0 role-list

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

| id | name |

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

| 9fe2ff9ee4384b1894a90878d3e92bab | \_member\_ |

| bda6906df3734cc384e41c6fd96f255e | admin |

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

root@ubuntu-10:/etc/keystone# keystone --os-tenant-name demo --os-username demo --os-password DEMO\_PASS --os-auth-url http://10.0.0.10:35357/v2.0 token-get

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

| Property | Value |

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

| expires | 2015-11-03T04:24:12Z |

| id | b26a87b16b8b449ea395d2448f21e33e |

| tenant\_id | 8293772b681c43cb86e670c42d5d8701 |

| user\_id | 5e036a7b49fb46b39c839df0aeb3cd56 |

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

root@ubuntu-10:/etc/keystone# keystone --os-tenant-name demo --os-username demo --os-password DEMO\_PASS --os-auth-url http://10.0.0.10:35357/v2.0 user-list

You are not authorized to perform the requested action: admin\_required (HTTP 403)

## 新建keystone客户端脚本

（客户端运行之后，不需要填写账号、密码、地址，就能够使用keystone认证了）

admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

demo-openrc.sh

export OS\_TENANT\_NAME=demo

export OS\_USERNAME=demo

export OS\_PASSWORD=DEMO\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:5000/v2.0

授执行权限：

#chmod +x admin-openrc.sh

#chmod +x demo-openrc.sh

# 安装Image Service(glance)\_controller

## 创建glance库及用户

第一步登陆数据库

第二步创建数据库glance

第三步步授权glance用户登陆数据库（本地和远程）

步骤如下：

root@ubuntu-10:/opt# mysql -uroot -popenstack

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 77

Server version: 5.5.44-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [mysql]> CREATE DATABASE glance;

Query OK, 1 row affected (0.00 sec)

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

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY 'GLANCE\_DBPASS';

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> quit

Bye

## 运行keystong客户端脚本（上个章节创建的）

操作如下：

root@ubuntu-10:/opt# cat admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

root@ubuntu-10:/opt# chmod +x admin-openrc.sh

root@ubuntu-10:/opt# source admin-openrc.sh

## 创建凭证和glance服务API

第一创建glance用户

第二创建glance角色

第三创建条目

第四创建Image Service API

操作如下：

root@ubuntu-10:/opt# keystone user-create --name glance --pass GLANCE\_PASS

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

| Property | Value |

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

| email | |

| enabled | True |

| id | 4f180c3de45a45189b1de0125098f6f8 |

| name | glance |

| username | glance |

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

root@ubuntu-10:/opt# keystone user-role-add --user glance --tenant service --role admin

root@ubuntu-10:/opt#keystone service-create --name glance --type image --description "OpenStack Image Service"

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

| Property | Value |

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

| description | OpenStack Image Service |

| enabled | True |

| id | b8d413424a484672a037313ff4d18420 |

| name | glance |

| type | image |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ image / {print $2}') --publicurl http://10.0.0.10:9292 --internalurl http://10.0.0.10:9292 --adminurl http://10.0.0.10:9292 --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:9292 |

| id | 98a90a728f6e42909eba5534e7537316 |

| internalurl | http://10.0.0.10:9292 |

| publicurl | http://10.0.0.10:9292 |

| region | regionOne |

| service\_id | b8d413424a484672a037313ff4d18420 |

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

## 安装glance

# apt-get install glance python-glanceclient

## 配置glance

需要配置部分涉及的ip地址为controller的IP地址，也就是当前主机的ip地址10.0.0.10

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

1. **DEFAULT配置**

大约第二行的位置插入两行如下：

[DEFAULT]

notification\_driver=noop

verbose=True

1. **数据库配置**

大约209行左右插入一行如下：

[database]

connection=mysql://glance:GLANCE\_DBPASS@10.0.0.10/glance

1. **Keystone配置**

大约386行修改如下：

插入auth\_uri=http://10.0.0.10:5000/v2.0

修改4行

插入flavor=keystone

[keystone\_authtoken]

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri = http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=glance

admin\_password=GLANCE\_PASS

revocation\_cache\_time = 10

[paste\_deploy]

flavor=keystone

1. **glance\_store配置**

大约435行插入一条如下：

[glance\_store]

# List of which store classes and store class locations are

# currently known to glance at startup.

#stores = glance.store.filesystem.Store,

# glance.store.http.Store

# ============ Filesystem Store Options ========================

# Directory that the Filesystem backend store

# writes image data to

default\_store = file

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

### /etc/glance/glance-registry.conf

1. **DEFAULT配置**

大约2行，插入两行

[DEFAULT]

notification\_driver=noop

verbose=True

1. **数据库配置：**

大约129行，插入一行

[database]

connection=mysql://glance:GLANCE\_DBPASS@10.0.0.10/glance

1. **Keystone配置**

大约226行修改配置如下：

[keystone\_authtoken]

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri = http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=glance

admin\_password=GLANCE\_PASS

revocation\_cache\_time = 10

[paste\_deploy]

flavor=keystone

## 数据同步

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

2015-11-03 15:13:23.967 2193 INFO migrate.versioning.api [-] 0 -> 1...

2015-11-03 15:13:23.974 2193 INFO glance.db.sqlalchemy.migrate\_repo.schema [-] creating table images

。。。

。。。

2015-11-03 15:13:36.317 2193 INFO migrate.versioning.api [-] 35 -> 36...

2015-11-03 15:13:36.789 2193 INFO migrate.versioning.api [-] done

## 服务的重新启动

# service glance-registry restart

# service glance-api restart

完成安装后清理无用的文件如下：

# rm -f /var/lib/glance/glance.sqlite

## 验证glance可用性

**1创建本地临时目录**

**2下载一个图片到上面创建的目录**

**3执行keystone客户端脚本**

**4上传图片到glance**

**5查询上传的图片信息**

**6删除临时目录**

上面的操作如下，下面是正常的情况：

root@ubuntu-10:/opt# mkdir /tmp/images

root@ubuntu-10:/opt# wget -P /tmp/images http://download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86\_64-disk.img

--2015-11-03 15:21:01-- http://download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86\_64-disk.img

Resolving download.cirros-cloud.net (download.cirros-cloud.net)... 69.163.241.114

Connecting to download.cirros-cloud.net (download.cirros-cloud.net)|69.163.241.114|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 13200896 (13M) [text/plain]

Saving to: ‘/tmp/images/cirros-0.3.3-x86\_64-disk.img’

100%[=================================================================================>] 13,200,896 84.7KB/s in 1m 55s

2015-11-03 15:22:57 (112 KB/s) - ‘/tmp/images/cirros-0.3.3-x86\_64-disk.img’ saved [13200896/13200896]

root@ubuntu-10:/opt# ls

admin-openrc.sh

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# glance image-create --name "cirros-0.3.3-x86\_64" --file /tmp/images/cirros-0.3.3-x86\_64-disk.img --disk-format qcow2 --container-format bare --is-public True --progress

[=============================>] 100%

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

| Property | Value |

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

| checksum | 133eae9fb1c98f45894a4e60d8736619 |

| container\_format | bare |

| created\_at | 2015-11-03T07:24:00 |

| deleted | False |

| deleted\_at | None |

| disk\_format | qcow2 |

| id | e91c40fa-6858-411e-b8a4-b10e6b7c3d34 |

| is\_public | True |

| min\_disk | 0 |

| min\_ram | 0 |

| name | cirros-0.3.3-x86\_64 |

| owner | 185f3d3f7d134c9b87f1045f7d73e8cf |

| protected | False |

| size | 13200896 |

| status | active |

| updated\_at | 2015-11-03T07:24:00 |

| virtual\_size | None |

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

root@ubuntu-10:/opt# glance image-list

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

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

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

| e91c40fa-6858-411e-b8a4-b10e6b7c3d34 | cirros-0.3.3-x86\_64 | qcow2 | bare | 13200896 | active |

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

root@ubuntu-10:/opt# rm -r /tmp/images

# 安装computer service(nova)

## Controller节点

### 创建数据库

第一步登陆数据库

第二步创建数据库nova

第三步步授权nova用户登陆数据库（本地和远程）

步骤如下：

root@ubuntu-10:/opt# mysql -uroot -popenstack

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 90

Server version: 5.5.44-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [mysql]> create database nova;

Query OK, 1 row affected (0.00 sec)

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

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY 'NOVA\_DBPASS';

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> quit

Bye

### 执行keystone客户端脚本

Keystone安装里面查找

root@ubuntu-10:/opt# ls

admin-openrc.sh

root@ubuntu-10:/opt# source admin-openrc.sh

### 创建凭证和nova-api

第一创建nova用户

第二创建nova角色

第三创建条目

第四创建Image Service API

操作如下：

root@ubuntu-10:/opt# keystone user-create --name nova --pass NOVA\_PASS

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

| Property | Value |

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

| email | |

| enabled | True |

| id | c0e496b50ccb4dc59e3618014482114b |

| name | nova |

| username | nova |

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

root@ubuntu-10:/opt# keystone user-role-add --user nova --tenant service --role admin

root@ubuntu-10:/opt# keystone service-create --name nova --type compute --description "OpenStack Compute"

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

| Property | Value |

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

| description | OpenStack Compute |

| enabled | True |

| id | 10262849ac9549118d9b375af7a91091 |

| name | nova |

| type | compute |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ compute / {print $2}') --publicurl http://10.0.0.10:8774/v2/%\(tenant\_id\)s --internalurl http://10.0.0.10:8774/v2/%\(tenant\_id\)s --adminurl http://10.0.0.10:8774/v2/%\(tenant\_id\)s --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:8774/v2/%(tenant\_id)s |

| id | 8e33aa7a52a94f599ea31ddd57eaa25d |

| internalurl | http://10.0.0.10:8774/v2/%(tenant\_id)s |

| publicurl | http://10.0.0.10:8774/v2/%(tenant\_id)s |

| region | regionOne |

| service\_id | 10262849ac9549118d9b375af7a91091 |

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

### Nova安装

# apt-get install nova-api nova-cert nova-conductor nova-consoleauth nova-novncproxy nova-scheduler python-novaclient

### 配置nova( /etc/nova/nova.conf)

编辑/etc/nova/nova.conf

1. Default配置
2. Database配置
3. Keystone配置
4. Glance配置

配置文件内容如下：插入如下的黄色部分，其中ip地址为controller地址，也就是当前主机IP地址10.0.0.10

[DEFAULT]

dhcpbridge\_flagfile=/etc/nova/nova.conf

dhcpbridge=/usr/bin/nova-dhcpbridge

logdir=/var/log/nova

state\_path=/var/lib/nova

lock\_path=/var/lock/nova

force\_dhcp\_release=True

libvirt\_use\_virtio\_for\_bridges=True

verbose=True

ec2\_private\_dns\_show\_ip=True

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

enabled\_apis=ec2,osapi\_compute,metadata

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

my\_ip = 10.0.0.10

vncserver\_listen=10.0.0.10

vncserver\_proxyclient\_address=10.0.0.10

verbose = True

[database]

connection=mysql://nova:NOVA\_DBPASS@10.0.0.10/nova

[keystone\_authtoken]

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=nova

admin\_password=NOVA\_PASS

[glance]

host=10.0.0.10

### 数据同步

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

2015-11-03 16:21:47.374 4223 INFO migrate.versioning.api [-] 215 -> 216...

2015-11-03 16:22:18.321 4223 INFO migrate.versioning.api [-] done

。。。

。。。

2015-11-03 16:22:27.441 4223 INFO migrate.versioning.api [-] 253 -> 254...

2015-11-03 16:22:27.774 4223 INFO migrate.versioning.api [-] done

### 服务的重新启动

# service nova-api restart

# service nova-cert restart

# service nova-consoleauth restart

# service nova-scheduler restart

# service nova-conductor restart

# service nova-novncproxy restart

删除没有用的文件

# rm -f /var/lib/nova/nova.sqlite

## Computer节点

### 安装包

# apt-get install nova-compute sysfsutils

### 配置nova：/etc/nova/nova.conf

**编辑/etc/nova/nova.conf**

本机ip是computer1:10.0.0.11

Controller ip:10.0.0.10

配置如下：

[DEFAULT]

dhcpbridge\_flagfile=/etc/nova/nova.conf

dhcpbridge=/usr/bin/nova-dhcpbridge

logdir=/var/log/nova

state\_path=/var/lib/nova

lock\_path=/var/lock/nova

force\_dhcp\_release=True

libvirt\_use\_virtio\_for\_bridges=True

#verbose=True

ec2\_private\_dns\_show\_ip=True

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

enabled\_apis=ec2,osapi\_compute,metadata

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

my\_ip=10.0.0.11

vnc\_enabled=True

vncserver\_listen=0.0.0.0

vncserver\_proxyclient\_address=10.0.0.11

novncproxy\_base\_url=http://10.0.0.10:6080/vnc\_auto.html

verbose=True

[keystone\_authtoken]

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

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=nova

admin\_password=NOVA\_PASS

[glance]

host=10.0.0.10

### 完成安装：/etc/nova/nova-compute.conf

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

**编辑配置文件：/etc/nova/nova-compute.conf**

添加如下：

[DEFAULT]

compute\_driver=libvirt.LibvirtDriver

[libvirt]

virt\_type=kvm

virt\_type=qemu

### 服务重新启动

#service nova-compute restart

删除无用文件，一次即可

# rm -f /var/lib/nova/nova.sqlite

## 验证nova可用性

登陆controller 10.0.0.10

执行客户端脚本

查看服务列表

查看image服务列表

操作如下：正确的配置结果如下

root@ubuntu-10:/opt# cat admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# nova service-list

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

| Id | Binary | Host | Zone | Status | State | Updated\_at | Disabled Reason |

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

| 1 | nova-cert | ubuntu-10 | internal | enabled | up | 2015-11-05T08:30:36.000000 | - |

| 2 | nova-consoleauth | ubuntu-10 | internal | enabled | up | 2015-11-05T08:30:32.000000 | - |

| 3 | nova-scheduler | ubuntu-10 | internal | enabled | up | 2015-11-05T08:30:39.000000 | - |

| 4 | nova-conductor | ubuntu-10 | internal | enabled | up | 2015-11-05T08:30:35.000000 | - |

| 6 | nova-compute | computer1 | nova | enabled | up | 2015-11-05T08:30:33.000000 | - |

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

root@ubuntu-10:/opt# nova image-list

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

| ID | Name | Status | Server |

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

| e91c40fa-6858-411e-b8a4-b10e6b7c3d34 | cirros-0.3.3-x86\_64 | ACTIVE | |

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

# 网络组件安装（neutron-server）

## Controller节点

### 创建数据库

第一步登陆数据库

第二步创建数据库nova

第三步步授权nova用户登陆数据库（本地和远程）

root@ubuntu-10:/opt# mysql -uroot -popenstack

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 110

Server version: 5.5.44-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [mysql]> CREATE DATABASE neutron;

Query OK, 1 row affected (0.00 sec)

MariaDB [mysql]>

MariaDB [mysql]> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' IDENTIFIED BY 'NEUTRON\_DBPASS';

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' IDENTIFIED BY 'NEUTRON\_DBPASS';

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> quit

Bye

### 执行keystone客户端脚本

Keystone安装步骤里面寻找

root@ubuntu-10:/opt# ls

admin-openrc.sh

root@ubuntu-10:/opt# source admin-openrc.sh

### 创建凭证和neutron-api

第一创建neutron用户

第二创建neutron角色

第三创建条目

第四创建neutron API

操作如下：

root@ubuntu-10:/opt# keystone user-create --name neutron --pass NEUTRON\_PASS

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

| Property | Value |

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

| email | |

| enabled | True |

| id | efececcc0e8e47a6875551344dfbe7ce |

| name | neutron |

| username | neutron |

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

root@ubuntu-10:/opt# keystone user-role-add --user neutron --tenant service --role admin

root@ubuntu-10:/opt# keystone service-create --name neutron --type network --description "OpenStack Networking"

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

| Property | Value |

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

| description | OpenStack Networking |

| enabled | True |

| id | 6a714a7dec864d47aaad5c3fd189a362 |

| name | neutron |

| type | network |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ network / {print $2}') --publicurl http://10.0.0.10:9696 --adminurl http://10.0.0.10:9696 --internalurl http://10.0.0.10:9696 --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:9696 |

| id | 05a2bf19a5a84d36b0f2ef255e10e487 |

| internalurl | http://10.0.0.10:9696 |

| publicurl | http://10.0.0.10:9696 |

| region | regionOne |

| service\_id | 6a714a7dec864d47aaad5c3fd189a362 |

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

### 网络组件安装

# apt-get install neutron-server neutron-plugin-ml2 python-neutronclient

### 配置网络组件

#### /etc/neutron/neutron.conf

**Default配置**

在第二行，没有的就添加，有的就修改

[DEFAULT]

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

core\_plugin=ml2

service\_plugins=router

allow\_overlapping\_ips=True

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

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

nova\_url = http://10.0.0.10:8774/v2

nova\_admin\_auth\_url=http://10.0.0.10:35357/v2.0

nova\_region\_name=regionOne

nova\_admin\_username=nova

nova\_admin\_tenant\_id=SERVICE\_TENANT\_ID

nova\_admin\_password=NOVA\_PASS

verbose = True

**Keystone配置**

大约在587行左右，修改配置如下：

[keystone\_authtoken]

auth\_host=127.0.0.1

auth\_port=35357

auth\_protocol = http

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=neutron

admin\_password=NEUTRON\_PASS

**Database配置**

大约598行左右，添加一行，注释一样如下：

[database]

connection=mysql://neutron:NEUTRON\_DBPASS@10.0.0.10:3306/neutron

# This line MUST be changed to actually run the plugin.

# Example:

# connection = mysql://root:pass@127.0.0.1:3306/neutron

# Replace 127.0.0.1 above with the IP address of the database used by the

# main neutron server. (Leave it as is if the database runs on this host.)

#connection = sqlite:////var/lib/neutron/neutron.sqlite

#### /etc/neutron/plugins/ml2/ml2\_conf.ini

**Ml2配置**

第一行添加如下：

[ml2]

type\_drivers=flat,gre

tenant\_network\_types=gre

mechanism\_drivers=openvswitch

**ml2\_type\_gre配置**

大约50行左右，插入如下：

[ml2\_type\_gre]

tunnel\_id\_ranges=1:1000

**securitygroup配置**

大约68行左右，插入如下3行：

[securitygroup]

enable\_security\_group=True

enable\_ipset=True

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

#### /etc/nova/nova.conf

**Default配置**

在[default]里面最后添加4行如下：

[DEFAULT]

dhcpbridge\_flagfile=/etc/nova/nova.conf

dhcpbridge=/usr/bin/nova-dhcpbridge

logdir=/var/log/nova

state\_path=/var/lib/nova

lock\_path=/var/lock/nova

force\_dhcp\_release=True

libvirt\_use\_virtio\_for\_bridges=True

verbose=True

ec2\_private\_dns\_show\_ip=True

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

enabled\_apis=ec2,osapi\_compute,metadata

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

my\_ip = 10.0.0.10

vncserver\_listen=10.0.0.10

vncserver\_proxyclient\_address=10.0.0.10

verbose = True

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

security\_group\_api=neutron

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

firewall\_driver=nova.virt.firewall.NoopFirewallDriver

**Neutron配置**

在文件的最后天机如下几行：

[neutron]

url=http://10.0.0.10:9696

auth\_strategy=keystone

admin\_auth\_url=http://10.0.0.10:35357/v2.0

admin\_tenant\_name=service

admin\_username=neutron

admin\_password=NEUTRON\_PASS

### 数据同步

# su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugins/ml2/ml2\_conf.ini upgrade juno" neutron

### 重启新启动服务

**Computer组件重启**

# service nova-api restart

# service nova-scheduler restart

# service nova-conductor restart

**网络组件重启**

# service neutron-server restart

### 验证neutron可用性

注释：如果出现的ext-list 不是16个 那么就是有问题

检查配置文件

**执行keystone客户端脚本**

root@ubuntu-10:/opt# cat admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

root@ubuntu-10:/opt# source admin-openrc.sh

**列表查看 共16个**

root@ubuntu-10:/opt# neutron ext-list

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

| alias | name |

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

| security-group | security-group |

| l3\_agent\_scheduler | L3 Agent Scheduler |

| ext-gw-mode | Neutron L3 Configurable external gateway mode |

| binding | Port Binding |

| provider | Provider Network |

| agent | agent |

| quotas | Quota management support |

| dhcp\_agent\_scheduler | DHCP Agent Scheduler |

| l3-ha | HA Router extension |

| multi-provider | Multi Provider Network |

| external-net | Neutron external network |

| router | Neutron L3 Router |

| allowed-address-pairs | Allowed Address Pairs |

| extraroute | Neutron Extra Route |

| extra\_dhcp\_opt | Neutron Extra DHCP opts |

| dvr | Distributed Virtual Router |

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

## Network节点

登陆10.0.0.13 network

### 计算机属性配置

开启路由转发等 /etc/sysctl.conf 配置下面3行，并执行命令使其生效

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

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

net.ipv4.ip\_forward=1

root@network:~# sysctl –p

### 安装包

#apt-get install neutron-plugin-ml2 neutron-plugin-openvswitch-agent neutron-l3-agent neutron-dhcp-agent

### 配置neutron

本机的ip为 10.0.0.12 主机名为network

Controller的ip为10.0.0.10

#### 配置命令行组件：/etc/neutron/neutron.conf

Default配置，大约第二行配置如下：

[DEFAULT]

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

core\_plugin=ml2

service\_plugins=router

allow\_overlapping\_ips=True

verbose=True

keystone\_authtoken、database配置，大约576行，去掉127.0.0.1，注释掉connect数据库连接，不需要数据库

[keystone\_authtoken]

auth\_port = 35357

auth\_protocol = http

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=neutron

admin\_password=NEUTRON\_PASS

[database]

# This line MUST be changed to actually run the plugin.

# Example:

# connection = mysql://root:pass@127.0.0.1:3306/neutron

# Replace 127.0.0.1 above with the IP address of the database used by the

# main neutron server. (Leave it as is if the database runs on this host.)

#connection = sqlite:////var/lib/neutron/neutron.sqlite

#### 配置ml2插件：/etc/neutron/plugins/ml2/ml2\_conf.ini

配置如下：

[ml2]

type\_drivers=flat,gre

tenant\_network\_types=gre

mechanism\_drivers=openvswitch

[ml2\_type\_flat]

flat\_networks=external

[ml2\_type\_vlan]

[ml2\_type\_gre]

tunnel\_id\_ranges=1:1000

[ml2\_type\_vxlan]

[securitygroup]

enable\_security\_group=True

enable\_ipset=True

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

[ovs]

local\_ip=10.0.0.12

enable\_tunneling=True

bridge\_mappings=external:br-ex

[agent]

tunnel\_types=gre

#### 配置l3agent：/etc/neutron/l3\_agent.ini

配置如下：

[DEFAULT]

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

use\_namespaces=True

external\_network\_bridge=br-ex

router\_delete\_namespaces=True

verbose=True

#### 配置dhcp-agent：/etc/neutron/dhcp\_agent.ini

[DEFAULT]

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

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

use\_namespaces=True

dhcp\_delete\_namespaces=True

verbose=True

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

#### 配置/etc/neutron/dnsmasq-neutron.conf并执行命令pkill dnsmasq

配置如下：创建文件，并添加一行

dhcp-option-force=26,1454

# pkill dnsmasq

#### 配置metadata-agent：/etc/neutron/metadata\_agent.ini

##### 网络节点配置

[DEFAULT]

auth\_url=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

auth\_region=regionOne

admin\_tenant\_name=service

admin\_user=neutron

admin\_password=NEUTRON\_PASS

nova\_metadata\_ip=10.0.0.10

metadata\_proxy\_shared\_secret=METADATA\_SECRET

verbose=True

##### controller节点控制/etc/nova/nova.conf

登陆10.0.0.10

neutron配置，添加如下两行：METADATA\_SECRET这个值与网络节点配置对应

添加完之后

执行重新服务的命令：# service nova-api restart

[neutron]

url=http://10.0.0.10:9696

auth\_strategy=keystone

admin\_auth\_url=http://10.0.0.10:35357/v2.0

admin\_tenant\_name=service

admin\_username=neutron

admin\_password=NEUTRON\_PASS

service\_metadata\_proxy=True

metadata\_proxy\_shared\_secret=METADATA\_SECRET

#### 配置ovs服务

# service openvswitch-switch restart

# ovs-vsctl add-br br-ex

下面的eth2外部网卡

# ovs-vsctl add-port br-ex eth2

### 服务的重新启动

# service openvswitch-switch restart

# service neutron-plugin-openvswitch-agent restart

# service neutron-l3-agent restart

# service neutron-dhcp-agent restart

# service neutron-metadata-agent restart

### 验证可用性

登陆controller 10.0.0.10

在这里面 如果出现问题 ，全部的插件都不可用 就去检查controller的配置文件

如果单个不可用，就去查单个的配置文件

如下操作：

#source admin-openrc.sh

#neutron agent-list

root@ubuntu-10:/opt# neutron agent-list

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

| id | agent\_type | host | alive | admin\_state\_up | binary |

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

| 226c5e84-570b-4971-b7bc-7b28dbdcdbd9 | Open vSwitch agent | network | :-) | True | neutron-openvswitch-agent |

| 27778bc7-c196-49e8-a228-8425223e089a | L3 agent | network | :-) | True | neutron-l3-agent |

| b07a5bc1-485e-4f31-b2d8-0e0b77e0be60 | DHCP agent | network | :-) | True | neutron-dhcp-agent |

| e0300341-9d35-4198-997f-d64018040b97 | Open vSwitch agent | computer1 | :-) | True | neutron-openvswitch-agent |

| f36ef335-4389-4bd0-834a-7b8b8de7874f | Metadata agent | network | :-) | True | neutron-metadata-agent |

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

## Computer节点

### 计算机属性配置/etc/sysctl.conf

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

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

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

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

使设置生效

root@computer1:~# sysctl -p

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

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

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

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

### 安装包

apt-get install neutron-plugin-ml2 neutron-plugin-openvswitch-agent

### 配置网络组件

#### 编辑配置文件/etc/neutron/neutron.conf

Default配置，大约第二行配置如下：

[DEFAULT]

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

core\_plugin=ml2

service\_plugins=router

allow\_overlapping\_ips=True

verbose=True

keystone\_authtoken、database配置，大约576行，去掉127.0.0.1，注释掉connect数据库连接，不需要数据库

[keystone\_authtoken]

auth\_port = 35357

auth\_protocol = http

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=neutron

admin\_password=NEUTRON\_PASS

[database]

# This line MUST be changed to actually run the plugin.

# Example:

# connection = mysql://root:pass@127.0.0.1:3306/neutron

# Replace 127.0.0.1 above with the IP address of the database used by the

# main neutron server. (Leave it as is if the database runs on this host.)

#connection = sqlite:////var/lib/neutron/neutron.sqlite

#### 配置ml2插件：/etc/neutron/plugins/ml2/ml2\_conf.ini

配置如下：

[ml2]

type\_drivers=flat,gre

tenant\_network\_types=gre

mechanism\_drivers=openvswitch

[ml2\_type\_flat]

flat\_networks=external

[ml2\_type\_vlan]

[ml2\_type\_gre]

tunnel\_id\_ranges=1:1000

[ml2\_type\_vxlan]

[securitygroup]

enable\_security\_group=True

enable\_ipset=True

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

[ovs]

local\_ip=10.0.0.11

enable\_tunneling=True

bridge\_mappings=external:br-ex

[agent]

tunnel\_types=gre

#### 编辑配置文件/etc/nova/nova.conf

修改default，和neutron中的配置如下，如果没有的就添加:

[DEFAULT]

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

security\_group\_api=neutron

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

wall\_driver=nova.virt.firewall.NoopFirewallDriver

[neutron]

url=http://10.0.0.10:9696

auth\_strategy=keystone

admin\_auth\_url=http://10.0.0.10:35357/v2.0

admin\_tenant\_name=service

admin\_username=neutron

admin\_password=NEUTRON\_PASS

### 服务的重新启动（log-error）

重启nova

#service nova-compute restart

重启网络插件

#service openvswitch-switch restart

#service neutron-plugin-openvswitch-agent restart

在tail -0f openvswitch-agent.log

里面有报错如下：

2015-11-09 11:38:19.801 305 ERROR neutron.plugins.openvswitch.agent.ovs\_neutron\_agent [req-1f5f99a5-e7b5-4cb0-9ba4-f3d95734bf84 None] Bridge br-ex for physical network external does not exist. Agent terminated!

### 验证可用性

登陆controller，出现了 computer1的agent

如下操作：

root@ubuntu-10:/opt# ls

admin-openrc.sh demo-openrc.sh

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# neutron agent-list

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

| id | agent\_type | host | alive | admin\_state\_up | binary |

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

| 226c5e84-570b-4971-b7bc-7b28dbdcdbd9 | Open vSwitch agent | network | :-) | True | neutron-openvswitch-agent |

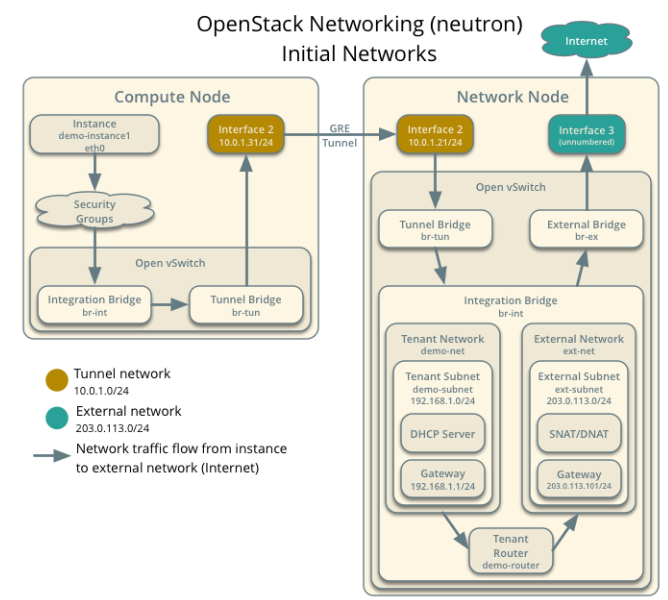
| b07a5bc1-485e-4f31-b2d8-0e0b77e0be60 | DHCP agent | network | :-) | True | neutron-dhcp-agent |

| e0300341-9d35-4198-997f-d64018040b97 | Open vSwitch agent | computer1 | :-) | True | neutron-openvswitch-agent |

| f36ef335-4389-4bd0-834a-7b8b8de7874f | Metadata agent | network | :-) | True | neutron-metadata-agent |

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

## 初始化网络



### 创建外部网络

执行keystone客户端脚本

执行两行创建命令

要配置外部网络，里面的ip端换成开始时的外部网络，例如本例子是192.168.95.0/24

管理口

操作如下：

root@ubuntu-10:/opt# cat admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# neutron net-create ext-net --router:external True --provider:physical\_network external --provider:network\_type flat

Created a new network:

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

| Field | Value |

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

| admin\_state\_up | True |

| id | 4d3304c2-1b46-42d7-af17-fe807c8e9e9b |

| name | ext-net |

| provider:network\_type | flat |

| provider:physical\_network | external |

| provider:segmentation\_id | |

| router:external | True |

| shared | False |

| status | ACTIVE |

| subnets | |

| tenant\_id | 185f3d3f7d134c9b87f1045f7d73e8cf |

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

root@ubuntu-10:/opt# neutron subnet-create ext-net --name ext-subnet --allocation-pool start=192.168.95.2,end=192.168.95.254 --disable-dhcp --gateway 192.168.95.1 192.168.95.0/24

Created a new subnet:

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

| Field | Value |

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

| allocation\_pools | {"start": "192.168.95.2", "end": "192.168.95.254"} |

| cidr | 192.168.95.0/24 |

| dns\_nameservers | |

| enable\_dhcp | False |

| gateway\_ip | 192.168.95.1 |

| host\_routes | |

| id | 459c0aa4-ea31-4525-a7b4-2610c502c0e3 |

| ip\_version | 4 |

| ipv6\_address\_mode | |

| ipv6\_ra\_mode | |

| name | ext-subnet |

| network\_id | 4d3304c2-1b46-42d7-af17-fe807c8e9e9b |

| tenant\_id | 185f3d3f7d134c9b87f1045f7d73e8cf |

### 创建用户网络

执行keystone客户端脚本demo

两条创建网络命令

这里的网络还是使用192.168.95.0/24 网关为192.168.95.1

root@ubuntu-10:/opt# cat demo-openrc.sh

export OS\_TENANT\_NAME=demo

export OS\_USERNAME=demo

export OS\_PASSWORD=DEMO\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:5000/v2.0

root@ubuntu-10:/opt# source demo-openrc.sh

root@ubuntu-10:/opt# neutron net-create demo-net

Created a new network:

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

| Field | Value |

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

| admin\_state\_up | True |

| id | c83f3b5d-2aa6-48ec-b195-1455501324cc |

| name | demo-net |

| router:external | False |

| shared | False |

| status | ACTIVE |

| subnets | |

| tenant\_id | 8293772b681c43cb86e670c42d5d8701 |

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

root@ubuntu-10:/opt#neutron subnet-create demo-net --name demo-subnet --gateway 10.0.1.1 10.0.1.0/24

Created a new subnet:

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

| Field | Value |

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

| allocation\_pools | {"start": "10.0.1.2", "end": "10.0.1.254"} |

| cidr | 10.0.1.0/24 |

| dns\_nameservers | |

| enable\_dhcp | True |

| gateway\_ip | 10.0.1.1 |

| host\_routes | |

| id | 88d3584d-2457-46f5-9891-ed6979dd3c12 |

| ip\_version | 4 |

| ipv6\_address\_mode | |

| ipv6\_ra\_mode | |

| name | demo-subnet |

| network\_id | c83f3b5d-2aa6-48ec-b195-1455501324cc |

| tenant\_id | 8293772b681c43cb86e670c42d5d8701 |

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

root@ubuntu-10:/opt# neutron router-create demo-router

Created a new router:

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

| Field | Value |

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

| admin\_state\_up | True |

| external\_gateway\_info | |

| id | 86be78d7-afee-45e3-ab9d-2abae83df2a6 |

| name | demo-router |

| routes | |

| status | ACTIVE |

| tenant\_id | 8293772b681c43cb86e670c42d5d8701 |

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

root@ubuntu-10:/opt# neutron router-interface-add demo-router demo-subnet

Added interface e9a46d83-9779-4f81-8ee9-c215a2d256b0 to router demo-router.

root@ubuntu-10:/opt# neutron router-gateway-set demo-router ext-net

Set gateway for router demo-router

# 管理面板安装(dashboard)\_controller

## 安装

#apt-get install openstack-dashboard apache2 libapache2-mod-wsgi memcached python-memcache

## 配置管理面板

**编辑配置文件/etc/openstack-dashboard/local\_settings.py**

**配置openstack\_host**

133行左右：

OPENSTACK\_HOST = "10.0.0.10"

**配置访问**

557行左右

ALLOWED\_HOSTS = ['\*']

**配置caches**

103行左右

CACHES = {

'default': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache', 'LOCATION': '127.0.0.1:11211',

}

}

**配置时区**

254行左右

TIME\_ZONE = "UTC"

## 重新启动服务

# service apache2 restart

# service memcached restart

## 验证可用性

重要：如果页面报500错误，然后<http://ip>这个还能访问，

那么就是配置文件里面的时区搞错了，UTC 或者 CN，修改之后重启上面两个服务就可以了，登陆成功，看到面板信息就是可以了

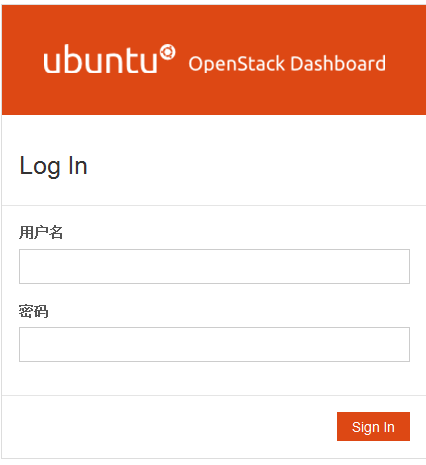
UTC世界标准时间，CN中国上海 ，与服务器controller配置相同

登陆页面：

<http://ip/horizon>

例如：

<http://192.168.95.140/horizon>



输入admin 和密码进行登陆



# 块存储服务安装

## Controller端安装（cinder）

### 数据库创建

操作如下：

root@ubuntu-10:/opt# mysql -uroot -popenstack

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 1142

Server version: 5.5.44-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [mysql]> CREATE DATABASE cinder;

Query OK, 1 row affected (0.00 sec)

MariaDB [mysql]> GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'localhost' IDENTIFIED BY 'CINDER\_DBPASS';

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'%' IDENTIFIED BY 'CINDER\_DBPASS';

Query OK, 0 rows affected (0.01 sec)

MariaDB [mysql]> quit

Bye

### 创建凭证和api

先执行keystone客户端脚本

第一创建cinder用户

第二创建cinder角色

第三创建条目

第四创建cinder-API

操作如下：

root@ubuntu-10:/opt# cat admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# keystone user-create --name cinder --pass CINDER\_PASS

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

| Property | Value |

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

| email | |

| enabled | True |

| id | 9c37f2c22cbb4817afbd9f35c5dae72a |

| name | cinder |

| username | cinder |

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

root@ubuntu-10:/opt# keystone user-role-add --user cinder --tenant service --role admin

root@ubuntu-10:/opt# keystone service-create --name cinder --type volume --description "OpenStack Block Storage"

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

| Property | Value |

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

| description | OpenStack Block Storage |

| enabled | True |

| id | 42630ecb06b94cd989cfed1ac7088060 |

| name | cinder |

| type | volume |

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

root@ubuntu-10:/opt# keystone service-create --name cinderv2 --type volumev2 --description "OpenStack Block Storage"

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

| Property | Value |

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

| description | OpenStack Block Storage |

| enabled | True |

| id | 79e602b5d0f84855a8db3a0ee5f60098 |

| name | cinderv2 |

| type | volumev2 |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ volume / {print $2}') --publicurl http://10.0.0.10:8776/v1/%\(tenant\_id\)s --internalurl http://10.0.0.10:8776/v1/%\(tenant\_id\)s --adminurl http://10.0.0.10:8776/v1/%\(tenant\_id\)s --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:8776/v1/%(tenant\_id)s |

| id | 052ca560a4484460ab9dedaa0dfd04b5 |

| internalurl | http://10.0.0.10:8776/v1/%(tenant\_id)s |

| publicurl | http://10.0.0.10:8776/v1/%(tenant\_id)s |

| region | regionOne |

| service\_id | 42630ecb06b94cd989cfed1ac7088060 |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ volumev2 / {print $2}') --publicurl http://10.0.0.10:8776/v2/%\(tenant\_id\)s --internalurl http://10.0.0.10:8776/v2/%\(tenant\_id\)s --adminurl http://10.0.0.10:8776/v2/%\(tenant\_id\)s --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:8776/v2/%(tenant\_id)s |

| id | 55303e74846b4c1a8b51336731224567 |

| internalurl | http://10.0.0.10:8776/v2/%(tenant\_id)s |

| publicurl | http://10.0.0.10:8776/v2/%(tenant\_id)s |

| region | regionOne |

| service\_id | 79e602b5d0f84855a8db3a0ee5f60098 |

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

### 安装cinder

#apt-get install cinder-api cinder-scheduler python-cinderclient

### 配置cinder

编辑配置文件/etc/cinder/cinder.conf：

Default配置、Database配置

配置如下：

[DEFAULT]

rootwrap\_config = /etc/cinder/rootwrap.conf

api\_paste\_confg = /etc/cinder/api-paste.ini

iscsi\_helper = tgtadm

volume\_name\_template = volume-%s

volume\_group = cinder-volumes

#verbose = True

#auth\_strategy = keystone

state\_path = /var/lib/cinder

lock\_path = /var/lock/cinder

volumes\_dir = /var/lib/cinder/volumes

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

my\_ip=10.0.0.10

verbose=True

glance\_host=10.0.0.10

[keystone\_authtoken]

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=cinder

admin\_password=CINDER\_PASS

[database]

connection=mysql://cinder:CINDER\_DBPASS@10.0.0.10/cinder

### 同步数据

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

### 重启服务

# service cinder-scheduler restart

# service cinder-api restart

删除无用的文件，删一次即可

# rm -f /var/lib/cinder/cinder.sqlite

## Block节点

### Ntp配置

计划任务：crontab –e

1 \*/2 \* \* \* sudo ntpdate 10.0.0.10

### 安装lvm包

root@block1:/etc# apt-get install lvm2

Reading package lists... Done

Building dependency tree

Reading state information... Done

lvm2 is already the newest version.

0 upgraded, 0 newly installed, 0 to remove and 150 not upgraded.

### 创建卷组

例子主机有一块新的硬盘sdb 并进行4个分区，这次操作将会把第一个分区sdb1做成卷组cinder-volumes

root@block1:~# fdisk -l /dev/sdb

Disk /dev/sdb: 53.7 GB, 53687091200 bytes

255 heads, 63 sectors/track, 6527 cylinders, total 104857600 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x8084b8f4

Device Boot Start End Blocks Id System

/dev/sdb1 2048 20973567 10485760 83 Linux

/dev/sdb2 20973568 41945087 10485760 83 Linux

/dev/sdb3 41945088 62916607 10485760 83 Linux

/dev/sdb4 62916608 104857599 20970496 83 Linux

root@block1:~# pvcreate /dev/sdb1

Physical volume "/dev/sdb1" successfully created

root@block1:~# vgcreate cinder-volumes /dev/sdb1

Volume group "cinder-volumes" successfully created

### 安装cinder-volume python-mysqldb包

# apt-get install cinder-volume python-mysqldb

### 编辑配置文件

/etc/cinder/cinder.conf

**Default配置**

Myip为10.0.0.13 自己本机的ip

Keystone认证配置ip为controller的ip：10.0.0.10

**Datbase配置**

数据配置指向controller

配置如下：

[DEFAULT]

rootwrap\_config = /etc/cinder/rootwrap.conf

api\_paste\_confg = /etc/cinder/api-paste.ini

iscsi\_helper = tgtadm

volume\_name\_template = volume-%s

volume\_group = cinder-volumes

#verbose = True

#auth\_strategy = keystone

state\_path = /var/lib/cinder

lock\_path = /var/lock/cinder

volumes\_dir = /var/lib/cinder/volumes

rpc\_backend=rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

auth\_strategy=keystone

my\_ip=10.0.0.13

verbose=True

glance\_host=10.0.0.10

[keystone\_authtoken]

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=cinder

admin\_password=CINDER\_PASS

[database]

connection=mysql://cinder:CINDER\_DBPASS@10.0.0.10/cinder

### 重启服务

# service tgt restart

# service cinder-volume restart

删除无用的文件，一次即可

# rm -f /var/lib/cinder/cinder.sqlite

## 验证块存储的可用用

第一步查询服务列表：如果服务的状态显示是down那么，就是ntp的问题，如果还是没有解决，看cinder的日志/var/log/cinder/

第二步创建vlume的时候，创建出来的vlume的状态一直是createing ，且块存储的日志出现争议等，就是版本controller安装的cinder与block1安装的cinder版本不一致

查看版本方式dpkg -l|grep cinder

可以使用第一章的 1.4 Openstack源更新（all）更新cinder的版本

登陆controller，10.0.0.10 进行操作验证

操作如下：

root@ubuntu-10:/opt# ls

admin-openrc.sh demo-openrc.sh

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# cinder service-list

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

| Binary | Host | Zone | Status | State | Updated\_at | Disabled Reason |

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

| cinder-scheduler | ubuntu-10 | nova | enabled | up | 2015-11-10T01:48:34.000000 | None |

| cinder-volume | block1 | nova | enabled | up | 2015-11-10T01:48:35.000000 | None |

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

root@ubuntu-10:/opt# source demo-openrc.sh

root@ubuntu-10:/opt# cinder create --display-name demo-volume1 1

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

| Property | Value |

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

| attachments | [] |

| availability\_zone | nova |

| bootable | false |

| created\_at | 2015-11-10T01:51:32.008753 |

| display\_description | None |

| display\_name | demo-volume1 |

| encrypted | False |

| id | f2237433-0a58-43ae-8320-95d7297ca5c7 |

| metadata | {} |

| size | 1 |

| snapshot\_id | None |

| source\_volid | None |

| status | creating |

| volume\_type | None |

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

root@ubuntu-10:/opt# cinder list

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

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

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

| f2237433-0a58-43ae-8320-95d7297ca5c7 | available | demo-volume1 | 1 | None | false | |

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

创建成功的日志：10.0.0.13

2015-11-10 09:51:34.865 16041 INFO cinder.volume.flows.manager.create\_volume [req-cdc46251-3cd6-467e-a933-821a7cb2b525 5e036a7b49fb46b39c839df0aeb3cd56 8293772b681c43cb86e670c42d5d8701 - - -] Volume f2237433-0a58-43ae-8320-95d7297ca5c7: being created as raw with specification: {'status': u'creating', 'volume\_size': 1, 'volume\_name': u'volume-f2237433-0a58-43ae-8320-95d7297ca5c7'}

2015-11-10 09:51:35.675 16041 INFO cinder.volume.flows.manager.create\_volume [req-cdc46251-3cd6-467e-a933-821a7cb2b525 5e036a7b49fb46b39c839df0aeb3cd56 8293772b681c43cb86e670c42d5d8701 - - -] Volume volume-f2237433-0a58-43ae-8320-95d7297ca5c7 (f2237433-0a58-43ae-8320-95d7297ca5c7): created successfully

2015-11-10 09:51:42.103 16041 INFO cinder.volume.manager [-] Updating volume status

2015-11-10 09:51:42.359 16041 INFO cinder.volume.manager [-] Updating volume replication status.

2015-11-10 09:52:42.104 16041 INFO cinder.volume.manager [-] Updating volume status

2015-11-10 09:52:42.442 16041 INFO cinder.volume.manager [-] Updating volume replication status.

# 对象存储安装(swift)

## Controller节点

### 创建用户信息及api

执行keystone客户端脚本admin

创建用户

创建角色

创建条目

创建api

操作如下：

root@ubuntu-10:/opt# cat admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# keystone user-create --name swift --pass SWIFT\_PASS

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

| Property | Value |

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

| email | |

| enabled | True |

| id | e1899ca9573d4fec988b5b8e70466f89 |

| name | swift |

| username | swift |

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

root@ubuntu-10:/opt# keystone user-role-add --user swift --tenant service --role admin

root@ubuntu-10:/opt# keystone service-create --name swift --type object-store --description "OpenStack Object Storage"

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

| Property | Value |

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

| description | OpenStack Object Storage |

| enabled | True |

| id | 1b4a2467c4334c39bfe7d12ac000cf10 |

| name | swift |

| type | object-store |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ object-store / {print $2}') --publicurl 'http://10.0.0.10:8080/v1/AUTH\_%(tenant\_id)s' --internalurl 'http://10.0.0.10:8080/v1/AUTH\_%(tenant\_id)s' --adminurl http://10.0.0.10:8080 --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:8080 |

| id | b7c201326460403789a36f2bd2df772d |

| internalurl | http://10.0.0.10:8080/v1/AUTH\_%(tenant\_id)s |

| publicurl | http://10.0.0.10:8080/v1/AUTH\_%(tenant\_id)s |

| region | regionOne |

| service\_id | 1b4a2467c4334c39bfe7d12ac000cf10 |

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

### 安装包

#apt-get install swift swift-proxy python-swiftclient python-keystoneclient python-keystonemiddleware memcached

### 配置swift

**创建目录及下载配置文件**

root@ubuntu-10:/opt# mkdir /etc/swift

root@ubuntu-10:/opt# curl -o /etc/swift/proxy-server.conf https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/proxy-server.conf-sample

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 24714 100 24714 0 0 23632 0 0:00:01 0:00:01 --:--:-- 23649

**编辑配置文件/etc/swift/proxy-server.conf**

**Default配置，大约第二行，如下：**

[DEFAULT]

# bind\_ip = 0.0.0.0

bind\_port = 8080

# bind\_timeout = 30

# backlog = 4096

swift\_dir = /etc/swift

user = swift

**pipeline:main和app:proxy-server配置，大约78行左右**

**非常重要，顺序和个数必须一样pipeline = healthcheck cache authtoken keystoneauth proxy-logging proxy-server**

[pipeline:main]

pipeline = healthcheck cache authtoken keystoneauth proxy-logging proxy-server

[app:proxy-server]

use=egg:swift#proxy

allow\_account\_management=true

account\_autocreate=true

**filter:keystoneauth配置，大约284行**

[filter:keystoneauth]

use=egg:swift#keystoneauth

operator\_roles=admin,\_member\_

**filter:authtoken配置，大约271行左右**

[filter:authtoken]

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

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=swift

admin\_password=SWIFT\_PASS

delay\_auth\_decision=true

**filter:cache配置，大约320行左右**

[filter:cache]

use = egg:swift#memcache

memcache\_servers = 127.0.0.1:11211

## Block节点

### 配置ip

### 配置ntp

### 安装rsync

# apt-get install xfsprogs rsync

### 配置rsync

#### 分区及挂载

**我为对象存储准备了两个分区，如下**

root@block1:~# fdisk -l /dev/sdb

/dev/sdb2 20973568 41945087 10485760 83 Linux

/dev/sdb3 41945088 62916607 10485760 83 Linux

**格式化分区**

root@block1:~# mkfs.xfs /dev/sdb2

meta-data=/dev/sdb2 isize=256 agcount=4, agsize=655360 blks

= sectsz=512 attr=2, projid32bit=0

data = bsize=4096 blocks=2621440, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0

log =internal log bsize=4096 blocks=2560, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

root@block1:~# mkfs.xfs /dev/sdb3

meta-data=/dev/sdb3 isize=256 agcount=4, agsize=655360 blks

= sectsz=512 attr=2, projid32bit=0

data = bsize=4096 blocks=2621440, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0

log =internal log bsize=4096 blocks=2560, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

**创建挂载点**

root@block1:~# mkdir -p /srv/node/sdb2

root@block1:~# mkdir -p /srv/node/sdb3

**配置开机自动挂载项，及挂载两个分区**

#vi /etc/fstab 后面添加两行

/dev/sdb2 /srv/node/sdb2 xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

/dev/sdb3 /srv/node/sdb3 xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

手动挂载分区，由于在开机挂载项里面，所有命令可以简单化

root@block1:~# mount /dev/sdb2

root@block1:~# mount /dev/sdb3

df –h 查看挂载情况

#### 编辑配置文件/etc/rsyncd.conf

添加如下内容

uid=swift

gid=swift

log file=/var/log/rsyncd.log

pid file=/var/run/rsyncd.pid

address=10.0.0.13

[account]

max connections=2

path=/srv/node/

read only=false

lock file=/var/lock/account.lock

[container]

max connections=2

path=/srv/node/

read only=false

lock file=/var/lock/container.lock

[object]

max connections=2

path=/srv/node/

read only=false

lock file=/var/lock/object.lock

#### 编辑配置文件 /etc/default/rsync

大约第8行修改如下：

RSYNC\_ENABLE=true

### 重新启动rsync服务

#service rsync restart

### 安装配置swift组件包

#apt-get install swift swift-account swift-container swift-object

### 下载配置文件，共3个

#curl -o /etc/swift/account-server.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/account-server.conf-sample>

#curl -o /etc/swift/container-server.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/container-server.conf-sample>

#curl -o /etc/swift/object-server.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/object-server.conf-sample>

#curl -o /etc/swift/swift.conf https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/swift.conf-sample

### 配置swif

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

修改如下[中括号]下面的项

[DEFAULT]

bind\_ip = 0.0.0.0

bind\_port = 6002

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon account-server

[filter:recon]

use = egg:swift#recon

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

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

修改如下[中括号]下面的项

[DEFAULT]

bind\_ip = 0.0.0.0

bind\_port = 6001

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon container-server

[filter:recon]

use = egg:swift#recon

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

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

修改如下[中括号]下面的项

[DEFAULT]

bind\_ip = 0.0.0.0

bind\_port = 6000

# bind\_timeout = 30

# backlog = 4096

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon object-server

[filter:recon]

use = egg:swift#recon

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

#### /etc/swift/swift.conf

修改如下[中括号]下面的项，并拷贝该文件到每一个存储swift节点和controller

把block1上的10.0.0.13上配置好的/etc/swift/swift.conf

Cp到 controller上一份

[swift-hash]

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

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

[storage-policy:0]

name = Policy-0

default = yes

### 创建Rings(controller节点)

#### 创建account ring

**进入controller**

#cd /etc/swift

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

#swift-ring-builder account.builder add r1z1-10.0.0.13:6002/sdb2 100

#swift-ring-builder account.builder add r1z1-10.0.0.13:6002/sdb3 100

**验证可用性：**

root@ubuntu-10:/etc/swift# swift-ring-builder account.builder

account.builder, build version 2

1024 partitions, 3.000000 replicas, 1 regions, 1 zones, 2 devices, 100.00 balance

The minimum number of hours before a partition can be reassigned is 1

Devices: id region zone ip address port replication ip replication port name weight partitions balance meta

0 1 1 10.0.0.13 6002 10.0.0.13 6002 sdb2 100.00 0 -100.00

1 1 1 10.0.0.13 6002 10.0.0.13 6002 sdb3 100.00 0 -100.00

Reblance：

root@ubuntu-10:/etc/swift# swift-ring-builder account.builder rebalance

Reassigned 1024 (100.00%) partitions. Balance is now 0.00.

#### 创建container ring

**进入controller**

#cd /etc/swift

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

# swift-ring-builder container.builder add r1z1-10.0.0.13:6001/sdb2 100

# swift-ring-builder container.builder add r1z1-10.0.0.13:6001/sdb3 100

**验证可用性：**

root@ubuntu-10:/etc/swift# swift-ring-builder container.builder

container.builder, build version 2

1024 partitions, 3.000000 replicas, 1 regions, 1 zones, 2 devices, 100.00 balance

The minimum number of hours before a partition can be reassigned is 1

Devices: id region zone ip address port replication ip replication port name weight partitions balance meta

0 1 1 10.0.0.13 6001 10.0.0.13 6001 sdb2 100.00 0 -100.00

1 1 1 10.0.0.13 6001 10.0.0.13 6001 sdb3 100.00 0 -100.00

Reblance：

root@ubuntu-10:/etc/swift# swift-ring-builder container.builder rebalance

Reassigned 1024 (100.00%) partitions. Balance is now 0.00.

#### 创建object ring

**进入controller**

#cd /etc/swift

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

# swift-ring-builder object.builder add r1z1-10.0.0.13:6000/sdb2 100

# swift-ring-builder object.builder add r1z1-10.0.0.13:6000/sdb3 100

**验证可用性：**

root@ubuntu-10:/etc/swift# swift-ring-builder object.builder

object.builder, build version 2

1024 partitions, 3.000000 replicas, 1 regions, 1 zones, 2 devices, 100.00 balance

The minimum number of hours before a partition can be reassigned is 1

Devices: id region zone ip address port replication ip replication port name weight partitions balance meta

0 1 1 10.0.0.13 6000 10.0.0.13 6000 sdb2 100.00 0 -100.00

1 1 1 10.0.0.13 6000 10.0.0.13 6000 sdb3 100.00 0 -100.00

Reblance：

root@ubuntu-10:/etc/swift# swift-ring-builder object.builder rebalance

Reassigned 1024 (100.00%) partitions. Balance is now 0.00.

#### Cp文件到其他swift节点

Copy the account.ring.gz, container.ring.gz, and object.ring.gz files to the /etc/swift directory on each storage node and any additional nodes running the proxy service.

**登陆controller节点向block1 cp文件**

#Cd /etc/swift

#scp \*.gz [user-100@10.0.0.13:/etc/swift/](mailto:user-100@10.0.0.13:/etc/swift/)

**登陆block1节点给文件添加权限**

#cd /etc/swift

#root@block1:/etc/swift# chown swift:swift \*.gz

#root@block1:/etc/swift# chmod 777 \*.gz

### 权限等剩余工作

把block1上的10.0.0.13上配置好的/etc/swift/swift.conf

Cp到 controller上一份

**有swift的节点都需要执行（controller、block1）**

#chown -R swift:swift /etc/swift

# chmod -R 777 /etc/swift/

**Block1节点需要执行**

# chown -R swift:swift /srv/node

# chmod -R 777 /srv/node

# mkdir -p /var/cache/swift

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

# chmod -R 777 /var/cache/swift

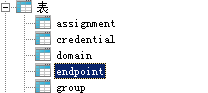
## 重新启动swift

**进入controller节点，和所有运行proxy 服务的节点 10.0.0.10**

# service memcached restart

# service swift-proxy restart（出现问题 job失败时，用这个命令测试# swift-init main start

本环节出现了问题，8080端口被占用一直启动不了（被dashboard占用了），修改8081端口，服务启动

8080端口占用 还需要修改，进入数据库）

把之前创建的api 端口8080修改成8081

或者重新创建api（controller节点的创建用户信息及api章节的最后一步keystone endpoint-create）

**进入存储模块 10.0.0.13**

#service rsync restart

#swift-init all restart

## 验证可用性

root@ubuntu-10:/opt# source demo-openrc.sh

root@ubuntu-10:/opt# swift stat

Account: AUTH\_8293772b681c43cb86e670c42d5d8701

Containers: 0

Objects: 0

Bytes: 0

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

X-Timestamp: 1447234789.19696

X-Trans-Id: txa0cd594d57af466788595-0056430ce4

X-Put-Timestamp: 1447234789.19696

root@ubuntu-10:/opt# touch test.txt

root@ubuntu-10:/opt# swift upload demo-container1 test.txt

test.txt

root@ubuntu-10:/opt# swift list

demo-container1

root@ubuntu-10:/opt# rm -rf test.txt

root@ubuntu-10:/opt# ls

admin-openrc.sh demo-container1 demo-openrc.sh

root@ubuntu-10:/opt# swift download demo-container1

test.txt [auth 0.197s, headers 0.306s, total 0.306s, 0.000 MB/s]

root@ubuntu-10:/opt# ls

admin-openrc.sh demo-container1 demo-openrc.sh test.txt

# 编制模块安装(heat)\_controller

## 创建数据库

第一步登陆数据库

第二步创建数据库heat

第三步步授权heat用户登陆数据库（本地和远程）

步骤如下：

root@ubuntu-10:~# mysql -uroot -popenstack

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 934

Server version: 5.5.44-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [mysql]> CREATE DATABASE heat;

Query OK, 1 row affected (0.01 sec)

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

Query OK, 0 rows affected (0.03 sec)

MariaDB [mysql]> GRANT ALL PRIVILEGES ON heat.\* TO 'heat'@'%' IDENTIFIED BY 'HEAT\_DBPASS';

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> quit

Bye

## 执行keystone客户端脚本

root@ubuntu-10:/opt# cat admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://10.0.0.10:35357/v2.0

root@ubuntu-10:/opt# source admin-openrc.sh

## 创建凭证和api（9条）

root@ubuntu-10:/opt# keystone user-create --name heat --pass HEAT\_PASS

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

| Property | Value |

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

| email | |

| enabled | True |

| id | 5c08f45f086a4f2d8ceec39181541437 |

| name | heat |

| username | heat |

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

root@ubuntu-10:/opt# keystone user-role-add --user heat --tenant service --role admin

root@ubuntu-10:/opt# keystone role-create --name heat\_stack\_owner

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

| Property | Value |

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

| id | 47f7b1ab7ded4cfc87d2b2a8231627c6 |

| name | heat\_stack\_owner |

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

root@ubuntu-10:/opt#keystone user-role-add --user demo --tenant demo --role heat\_stack\_owner

root@ubuntu-10:/opt# keystone role-create --name heat\_stack\_user

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

| Property | Value |

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

| id | 6e23f6f8f6d24e6ea93d713631e55a17 |

| name | heat\_stack\_user |

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

root@ubuntu-10:/opt# keystone service-create --name heat --type orchestration --description "Orchestration"

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

| Property | Value |

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

| description | Orchestration |

| enabled | True |

| id | 011c3f763acc410e863844ee6e50dde3 |

| name | heat |

| type | orchestration |

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

root@ubuntu-10:/opt#keystone service-create --name heat-cfn --type cloudformation --description "Orchestration"

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

| Property | Value |

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

| description | Orchestration |

| enabled | True |

| id | 04a092f00ad14dc7a50f517d9c536a35 |

| name | heat-cfn |

| type | cloudformation |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ orchestration / {print $2}') --publicurl http://10.0.0.10:8004/v1/%\(tenant\_id\)s --internalurl http://10.0.0.10:8004/v1/%\(tenant\_id\)s --adminurl http://10.0.0.10:8004/v1/%\(tenant\_id\)s --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:8004/v1/%(tenant\_id)s |

| id | 67499996e7474dbbbe2d286a18455c65 |

| internalurl | http://10.0.0.10:8004/v1/%(tenant\_id)s |

| publicurl | http://10.0.0.10:8004/v1/%(tenant\_id)s |

| region | regionOne |

| service\_id | 011c3f763acc410e863844ee6e50dde3 |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ cloudformation / {print $2}') --publicurl http://10.0.0.10:8000/v1 --internalurl http://10.0.0.10:8000/v1 --adminurl http://10.0.0.10:8000/v1 --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:8000/v1 |

| id | 5e564a4838254bf48617559c8eda28c9 |

| internalurl | http://10.0.0.10:8000/v1 |

| publicurl | http://10.0.0.10:8000/v1 |

| region | regionOne |

| service\_id | 04a092f00ad14dc7a50f517d9c536a35 |

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

## 安装包

#apt-get install heat-api heat-api-cfn heat-engine python-heatclient

缺少各种依赖关系报错

最后的命令如下：

#apt-get install libjs-sphinxdoc=1.2.2+dfsg-1ubuntu1 sphinx-common python-sphinx python-heat heat-common heat-api heat-api-cfn heat-engine python-heatclient

## 配置heat

编辑/etc/heat/heat.conf,default第一行，database大约第857行左右，keystone\_authtoken大约1087行左右，ec2authtoken大约964行左右

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host=10.0.0.10

rabbit\_password=RABBIT\_PASS

heat\_metadata\_server\_url=http://10.0.0.10:8000

heat\_waitcondition\_server\_url=http://10.0.0.10:8000/v1/waitcondition

verbose=True

[database]

connection = mysql://heat:HEAT\_DBPASS@10.0.0.10/heat

[keystone\_authtoken]

auth\_uri=http://10.0.0.10:5000/v2.0

identity\_uri=http://10.0.0.10:35357

admin\_tenant\_name=service

admin\_user=heat

admin\_password=HEAT\_PASS

[ec2authtoken]

auth\_uri = http://10.0.0.10:5000/v2.0

## 数据同步

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

## 服务的重新启动

#service heat-api restart

#service heat-api-cfn restart

#service heat-engine restart

删除无用文件，一次即可

# rm -f /var/lib/heat/heat.sqlite

## 验证可用性

操作如下：

root@ubuntu-10:/opt# source demo-openrc.sh

root@ubuntu-10:/opt# cat test-stack.yml

heat\_template\_version: 2014-10-16

description: A simple server.

parameters:

ImageID:

type: string

description: Image use to boot a server

NetID:

type: string

description: Network ID for the server

resources:

server:

type: OS::Nova::Server

properties:

image: { get\_param: ImageID }

flavor: m1.tiny

networks:

- network: { get\_param: NetID }

outputs:

private\_ip:

description: IP address of the server in the private network

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

root@ubuntu-10:/opt# NET\_ID=$(nova net-list | awk '/ demo-net / { print $2 }')

root@ubuntu-10:/opt# heat stack-create -f test-stack.yml -P "ImageID=cirros-0.3.3-x86\_64;NetID=$NET\_ID" testStack

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

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

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

| e8ecd44e-ff57-4004-ae21-8e7385e57ca9 | testStack | CREATE\_IN\_PROGRESS | 2015-11-13T07:33:34Z |

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

root@ubuntu-10:/opt# heat stack-list

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

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

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

| e8ecd44e-ff57-4004-ae21-8e7385e57ca9 | testStack | CREATE\_COMPLETE | 2015-11-13T07:33:34Z |

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

# 安装临时模块

## Controller节点

### 安装mongoDB包

#apt-get install mongodb-server mongodb-clients python-pymongo

### 配置mongodb，重启数据库mongodb

**编辑配置文件/etc/mongodb.conf**

bind\_ip = 10.0.0.10

**重启数据库mongodb**

#service mongodb restart

### 创建临时库mongo

root@ubuntu-10:~# mongo --host 10.0.0.10 --eval '

> db = db.getSiblingDB("ceilometer");

> db.addUser({user: "ceilometer",

> pwd: "CEILOMETER\_DBPASS",

> roles: [ "readWrite", "dbAdmin" ]})'

MongoDB shell version: 2.4.9

connecting to: 10.0.0.10:27017/test

{

"user" : "ceilometer",

"pwd" : "447c1db3b92df9035684b39ad9fa2185",

"roles" : [

"readWrite",

"dbAdmin"

],

"\_id" : ObjectId("564440a1c518a17cf2eda194")

}

### 执行keystone客户端脚本admin

#source admin-openrc.sh

### 创建用户角色api等

root@ubuntu-10:/opt# keystone user-create --name ceilometer --pass CEILOMETER\_PASS

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

| Property | Value |

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

| email | |

| enabled | True |

| id | 63029545c0524c3db95d3e3829454cd6 |

| name | ceilometer |

| username | ceilometer |

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

root@ubuntu-10:/opt# keystone user-role-add --user ceilometer --tenant service --role admin

root@ubuntu-10:/opt# keystone service-create --name ceilometer --type metering --description "Telemetry"

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

| Property | Value |

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

| description | Telemetry |

| enabled | True |

| id | 0aefd6a7dfe64a16923dd17b812aab4e |

| name | ceilometer |

| type | metering |

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

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/

root@ubuntu-10:/opt# keystone endpoint-create --service-id $(keystone service-list | awk '/ metering / {print $2}') --publicurl http://10.0.0.10:8777 --internalurl http://10.0.0.10:8777 --adminurl http://10.0.0.10:8777 --region regionOne

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

| Property | Value |

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

| adminurl | http://10.0.0.10:8777 |

| id | 13c88e3168a746f899cd90ba7aa68d32 |

| internalurl | http://10.0.0.10:8777 |

| publicurl | http://10.0.0.10:8777 |

| region | regionOne |

| service\_id | 0aefd6a7dfe64a16923dd17b812aab4e |

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

### 安装临时模块组件包

#apt-get install ceilometer-api ceilometer-collector ceilometer-agent-central ceilometer-agent-notification ceilometer-alarm-evaluator ceilometer-alarm-notifier python-ceilometerclient

### 生成随机码

root@ubuntu-10:~# openssl rand -hex 10

5f68ddcf3acfa962b3a9

### 配置临时模块

**编辑配置文件/etc/ceilometer/ceilometer.conf**

Default第一行

keystone\_authtoken大约900行左右

database大约709行

service\_credentials大约1142行

publisher大约1104行 写入的是上面生成的随机码

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.0.0.10

rabbit\_password = RABBIT\_PASS

auth\_strategy = keystone

verbose = True

[keystone\_authtoken]

auth\_uri = http://10.0.0.10:5000/v2.0

identity\_uri = http://10.0.0.10:35357

admin\_tenant\_name = service

admin\_user = ceilometer

admin\_password = CEILOMETER\_PASS

[database]

connection = mongodb://ceilometer:CEILOMETER\_DBPASS@10.0.0.10:27017/ceilometer

[service\_credentials]

os\_auth\_url = http://10.0.0.10:5000/v2.0

os\_username = ceilometer

os\_tenant\_name = service

os\_password = CEILOMETER\_PASS

[publisher]

metering\_secret = 5f68ddcf3acfa962b3a9

### 重新启动服务

#service mongodb restart

# service ceilometer-agent-central restart

# service ceilometer-agent-notification restart

# service ceilometer-api restart

# service ceilometer-collector restart

# service ceilometer-alarm-evaluator restart

# service ceilometer-alarm-notifier restart

## Computer节点

### 安装ceilometer-agent-compute包

#apt-get install ceilometer-agent-compute

### 配置ceilometer

**编辑配置文件/etc/ceilometer/ceilometer.conf**

Default第一行

keystone\_authtoken第897行左右

service\_credentials第1142行左右

publisher第1104行左右 填写controller节点生成的随机码

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.0.0.10

rabbit\_password = RABBIT\_PASS

verbose = True

[keystone\_authtoken]

auth\_uri = http://10.0.0.10:5000/v2.0

identity\_uri = http://10.0.0.10:35357

admin\_tenant\_name = service

admin\_user = ceilometer

admin\_password = CEILOMETER\_PASS

[service\_credentials]

os\_auth\_url = http://10.0.0.10:5000/v2.0

os\_username = ceilometer

os\_tenant\_name = service

os\_password = CEILOMETER\_PASS

[publisher]

metering\_secret = 5f68ddcf3acfa962b3a9

### 编辑nova.conf

编辑 /etc/nova/nova.conf 在default里面添加如下：

[DEFAULT]

instance\_usage\_audit = True

instance\_usage\_audit\_period = hour

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

notification\_driver = messagingv2

### 重新启动服务

#service ceilometer-agent-compute restart

#service nova-compute restart

## 配置image（controller节点）

### 编辑配置文件/etc/glance/glance-api.conf

添加如下4行

[DEFAULT]

notification\_driver = messagingv2

rpc\_backend = rabbit

rabbit\_host=10.0.0.10

rabbit\_password = RABBIT\_PASS

### 编辑/etc/glance/glance-registry.conf

添加如下4行

[DEFAULT]

notification\_driver = messagingv2

rpc\_backend = rabbit

rabbit\_host=10.0.0.10

rabbit\_password = RABBIT\_PASS

### 重启image服务

#service glance-registry restart

#service glance-api restart

## 配置cinder（controller节点、block1节点）

### 编辑/etc/cinder/cinder.conf controller节点

添加如下两行

[DEFAULT]

control\_exchange = cinder

notification\_driver = messagingv2

### 编辑/etc/cinder/cinder.conf block1节点

添加如下两行

[DEFAULT]

control\_exchange = cinder

notification\_driver = messagingv2

### 重新启动服务

**Controller节点：**

# service cinder-api restart

# service cinder-scheduler restart

**Block1节点：**

# service cinder-volume restart

## 配置swift（controller）

### To configure prerequisites（controller）

下面的配置注意，最后一行命令 里面的role字符串是 上一行命令生成的id

这里是75afb368b1024f3ba3628423ef46ea18

root@ubuntu-10:/opt# source admin-openrc.sh

root@ubuntu-10:/opt# keystone role-create --name ResellerAdmin

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

| Property | Value |

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

| id | 75afb368b1024f3ba3628423ef46ea18 |

| name | ResellerAdmin |

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

root@ubuntu-10:/opt#keystone user-role-add --tenant service --user ceilometer --role 75afb368b1024f3ba3628423ef46ea18

### To configure notifications（controller）

**编辑/etc/swift/proxy-server.conf**

[filter:keystoneauth]大约37行 修改

pipeline:main大约10行 修改

filter:ceilometer最后一行添加

[filter:keystoneauth]

use=egg:swift#keystoneauth

operator\_roles=admin,\_member\_,ResellerAdmin

[pipeline:main]

pipeline = healthcheck cache authtoken keystoneauth proxy-logging proxy-server

[filter:ceilometer]

use = egg:ceilometer#swift

log\_level = WARN

### 给权限

#usermod -a -G ceilometer swift

### 重启服务（error，最后该pipeline属性没有改动，启动成功）

#service swift-proxy restart

# 创建实例

## 创建实例前配置nova

进入computer1节点

编辑/etc/nova/nova.conf

在[default]最后添加两行如下：

vif\_plugging\_is\_fatal=false

vif\_plugging\_timeout=0

否则报错：

Computer节点/var/log/nova/ nova-compute.log有报错，并且实例创建失败error，错误开始，如下:

2015-11-13 13:51:01.163 26523 WARNING nova.virt.libvirt.driver [-] Timeout waiting for vif plugging callback for instance 6d6bd6a1-39d4-4bd1-9fbf-b190ba840a87

## 创建实例

登陆controller节点

步骤：

**1.执行keystone客户端脚本demo**

**2.创建ssh-keygen并添加到nova**

**3.创建实例**

nova boot --flavor m1.tiny --image cirros-0.3.3-x86\_64 --nic net-id=c83f3b5d-2aa6-48ec-b195-1455501324cc --security-group default --key-name demo-key demo-instance2

net-id为neutron net-list命令查看到的demo-net的id

**4.查看实例的登陆地址**

#nova get-vnc-console demo-instance2 novnc

**操作如下：**

root@ubuntu-10:/opt# source demo-openrc.sh

root@ubuntu-10:/opt# ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

58:90:98:c4:ae:35:99:11:aa:50:77:3e:e1:2a:ee:b6 root@ubuntu-10

The key's randomart image is:

+--[ RSA 2048]----+

| .o+++. |

| . o=+.o |

|. .. ++ . |

|.. \*. + |

|. .o... S |

| ... |

| . |

| .. |

| .E. |

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

root@ubuntu-10:/opt# nova keypair-add --pub-key ~/.ssh/id\_rsa.pub demo-key

root@ubuntu-10:/opt# nova keypair-list

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

| Name | Fingerprint |

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

| demo-key | 58:90:98:c4:ae:35:99:11:aa:50:77:3e:e1:2a:ee:b6 |

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

root@ubuntu-10:/opt# 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 |

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

root@ubuntu-10:/opt# nova image-list

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

| ID | Name | Status | Server |

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

| e91c40fa-6858-411e-b8a4-b10e6b7c3d34 | cirros-0.3.3-x86\_64 | ACTIVE | |

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

root@ubuntu-10:/opt# neutron net-list

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

| id | name | subnets |

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

| c83f3b5d-2aa6-48ec-b195-1455501324cc | demo-net | 88d3584d-2457-46f5-9891-ed6979dd3c12 10.0.1.0/24 |

| 21bf8144-ab8a-4b1e-97f7-74e2137dc96e | ext-net | bc1e4606-25f2-4024-8663-0f0fe631a266 |

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

root@ubuntu-10:/opt# nova secgroup-list

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

| Id | Name | Description |

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

| 6d1ffeaa-156f-4bf1-92e4-2f3820f6254b | default | default |

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

root@ubuntu-10:/opt# nova boot --flavor m1.tiny --image cirros-0.3.3-x86\_64 --nic net-id=c83f3b5d-2aa6-48ec-b195-1455501324cc --security-group default --key-name demo-key demo-instance2

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

| Property | Value |

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

| OS-DCF:diskConfig | MANUAL |

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

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

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

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

| OS-SRV-USG:launched\_at | - |

| OS-SRV-USG:terminated\_at | - |

| accessIPv4 | |

| accessIPv6 | |

| adminPass | NdbkMo6FBrex |

| config\_drive | |

| created | 2015-11-13T06:01:15Z |

| flavor | m1.tiny (1) |

| hostId | |

| id | 83722ed6-f2c0-4db6-b45f-2f5a8442e320 |

| image | cirros-0.3.3-x86\_64 (e91c40fa-6858-411e-b8a4-b10e6b7c3d34) |

| key\_name | demo-key |

| metadata | {} |

| name | demo-instance2 |

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

| progress | 0 |

| security\_groups | default |

| status | BUILD |

| tenant\_id | 8293772b681c43cb86e670c42d5d8701 |

| updated | 2015-11-13T06:01:15Z |

| user\_id | 5e036a7b49fb46b39c839df0aeb3cd56 |

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

root@ubuntu-10:/opt# nova get-vnc-console demo-instance2 novnc

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

| Type | Url |

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

| novnc | http://10.0.0.10:6080/vnc\_auto.html?token=56b4e177-e2bc-4056-83d4-a010401578d0 |

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

## 登陆实例

本环境controller的ip为

业务ip：10.0.0.10

管理ip：192.168.95.140

登陆时请使用firefox

http://controller\_IP:6080/vnc\_auto.html?token=56b4e177-e2bc-4056-83d4-a010401578d0

