OpenStack官方文档搭建第一天

# 环境搭建

## 1.虚拟机设置

两台虚拟机，一台是Controller节点，另一台是Compute节点，都采用minimal安装。

root密码都为：wangdage980204

通过命令ip addr 得到ip：

Controller：192.168.112.146

Compute：192.168.112.145

## 2.yum源设置为阿里云

发现下载速度很快，且使用的就是阿里云的yum源，因此不需要更新yum源。

## 3.配置ntp时间同步（两个节点都需要）

注意，在需要配置文件修改的地方，我都将地址改为了Controller节点的IP地址。

[root@controller ~]# yum install chrony -y #安装时间同步服务

[root@controller ~]# vim /etc/chrony.conf

添加子网段 allow 192.168.112.0/24

[root@controller ~]# systemctl restart chronyd.service #重启时间同步服务

[root@controller ~]# systemctl enable chronyd.service #设置时间同步服务开机自启动

## 4.关闭防火墙selinux（两个节点都需要）

关闭防火墙，关闭开机自启动防火墙

systemctl stop firewalld

systemctl disable firewalld

永久关闭selinux

#vi /etc/selinux/config  
#SELINUX=enforcing改为SELINUX=disabled  
重启生效

## 5.安装Train版的yum源（两个节点都需要）

[root@controller ~]# yum install centos-release-openstack-train -y

# （二）Controller节点配置

## 1.安装客户端【controller节点】

[root@controller ~]# yum install python-openstackclient -y

## 2.安装数据库【controller节点】

注意，在需要配置文件修改的地方，我都将地址改为了Controller节点的IP地址。

yum install mariadb mariadb-server python2-PyMySQL

设置配置文件

cat > /etc/my.cnf.d/openstack.cnf << EOF

[mysqld]

bind-address = 192.168.112.146

default-storage-engine = innodb #默认存储引擎

innodb\_file\_per\_table = on #每张表独立表空间文件

max\_connections = 4096 #最大连接数

collation-server = utf8\_general\_ci #默认字符集

character-set-server = utf8

EOF

systemctl enable mariadb.service #开机自启动mysql

systemctl start mariadb.service #开启mysql服务

进入mysql中

mysql\_secure\_installation

后面的操作就是

回车--->n--->一路y

第一个回车是因为首次登陆，询问当前的密码，因为没有设置密码则直接按回车。

n代表没有设置密码。

## 3.安装消息队列服务【controller节点】

[root@controller ~]# yum install rabbitmq-server -y #安装rabbitmq服务器

[root@controller ~]# systemctl enable rabbitmq-server.service #设置rabbitmq开机自启动

[root@controller ~]# systemctl start rabbitmq-server.service #开启rabbitmq服务

创建用户 openstack

[root@controller ~]# rabbitmqctl add\_user openstack RABBIT\_PASS

Creating user "openstack"

授予权限

[root@controller ~]# rabbitmqctl set\_permissions openstack ".\*" ".\*" ".\*"

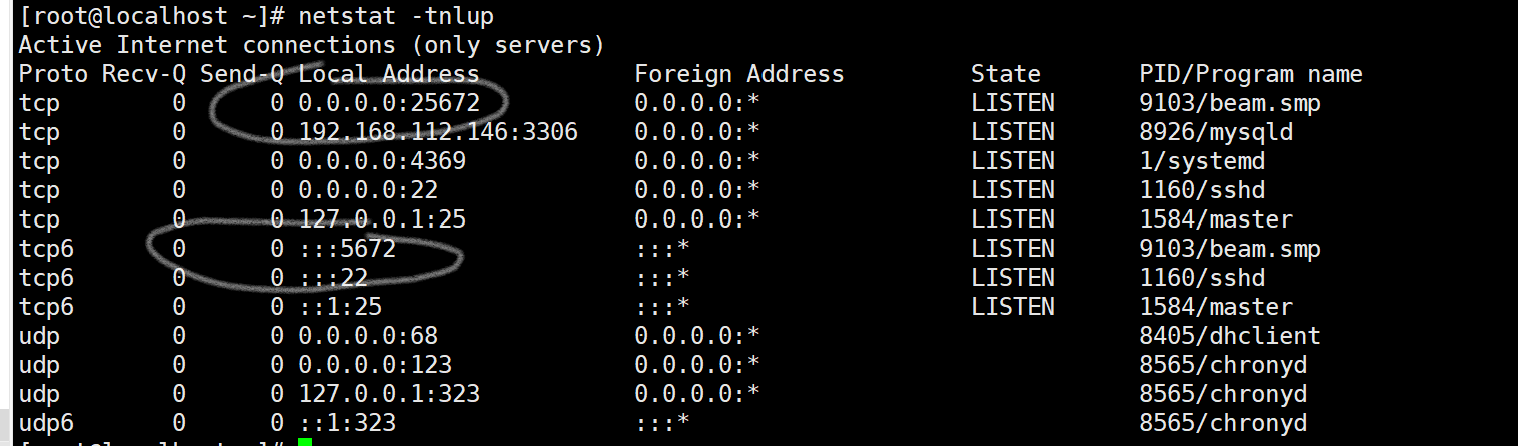
Setting permissions for user "openstack" in vhost "/"

通过netstat查询rabbitmq服务启动与否

可是Centos7中默认没有安装netstat，因此首先查询出netstat在本机中的包为net-tools，然后使用yum-y install net-tools 安装即可。

通过netstat查询rabbitmq服务启动与否

netstat -tnlup



出现了5672端口和25672端口，则说明rabbbitmq安装成功。

## 4.安装memcache【controller节点】

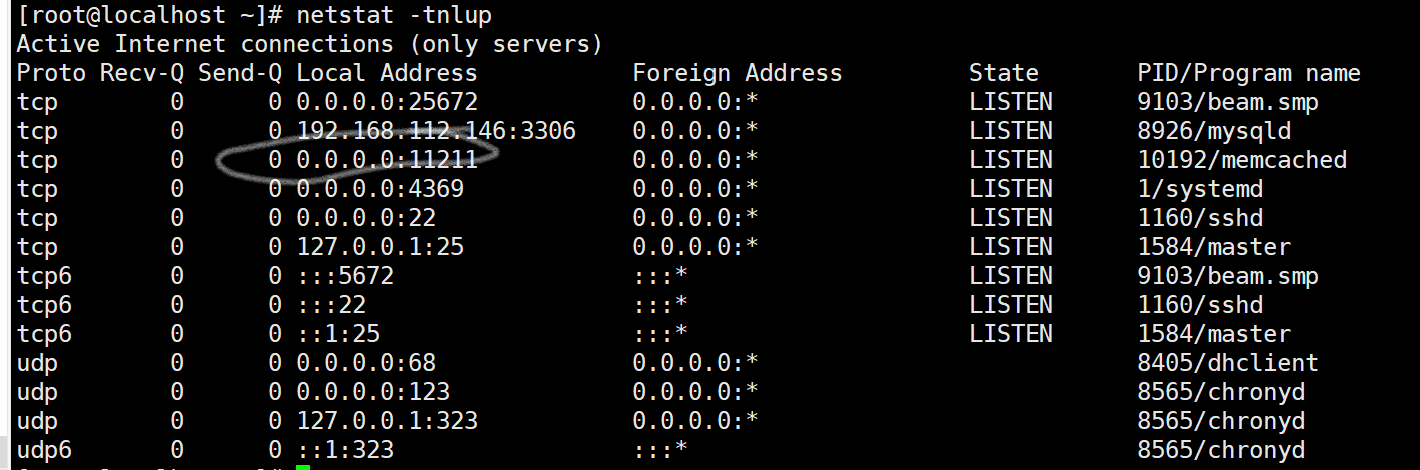
yum install memcached python-memcached #安装memcache服务

sed -i '/OPTIONS/c\OPTIONS="-l 0.0.0.0"' /etc/sysconfig/memcached

systemctl enable memcached.service #设置开机自启动

systemctl start memcached.service #开启服务

安装和启动好之后，同样使用netstat -tnlup查看端口情况，看到11211端口有程序在侦听则表示memcache安装成功



## 5.安装etcd【controller节点】

注意，在需要配置文件修改的地方，我都将地址改为了Controller节点的IP地址。

yum install etcd #安装etcd服务

cp -a /etc/etcd/etcd.conf{,.bak}

etcd配置文件

cat > /etc/etcd/etcd.conf <<EOF

#[Member]

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

ETCD\_LISTEN\_PEER\_URLS="http://192.168.112.146:2380"

ETCD\_LISTEN\_CLIENT\_URLS="http://192.168.112.146:2379"

ETCD\_NAME="controller" #[Clustering]

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://192.168.112.146:2380" ETCD\_ADVERTISE\_CLIENT\_URLS="http://192.168.112.146:2379" ETCD\_INITIAL\_CLUSTER="controller=http://192.168.112.146:2380" ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster-01"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

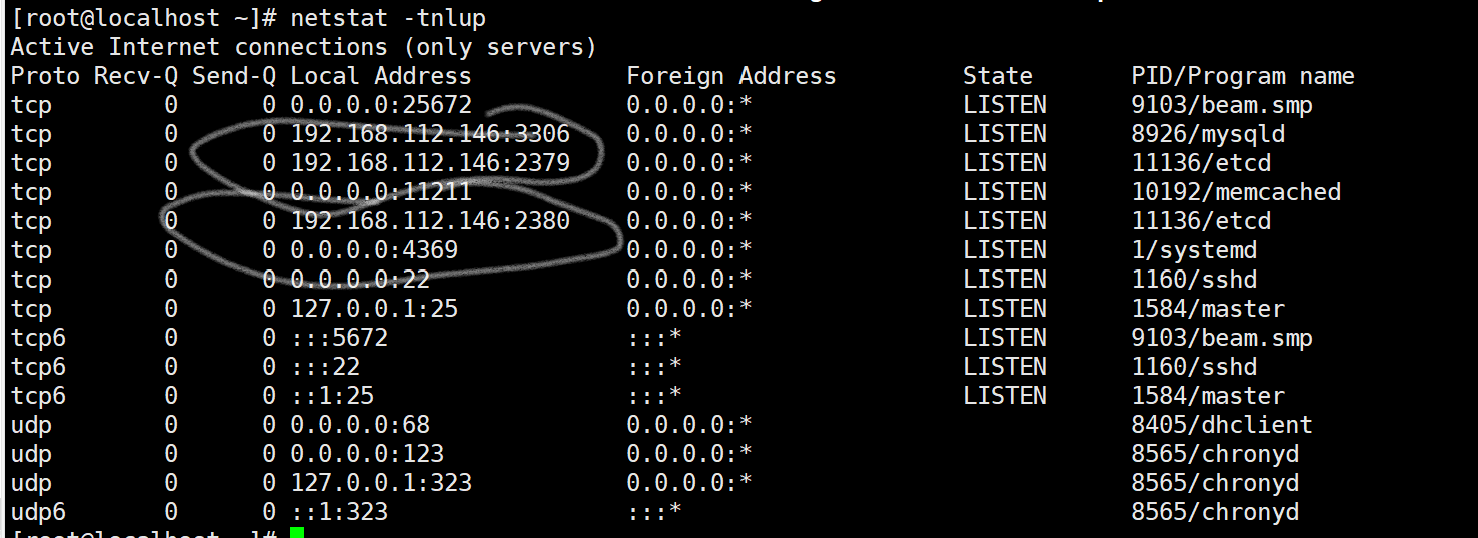
EOF

注意，配置文件中不应该有其他的#注释，会导致文件解析失败。

systemctl enable etcd #开机自启动

systemctl start etcd #启动服务

安装和启动好之后，同样使用netstat -tnlup查看端口情况，看到2379和2380端口有程序在侦听则表示etcd安装成功



## 6.etcd介绍

etcd是CoreOS团队于2013年6月发起的开源项目，它的目标是构建一个高可用的分布式键值(key-value)数据库。etcd内部采用raft协议作为一致性算法，etcd基于Go语言实现。

etcd作为服务发现系统，有以下的特点：

* 简单：安装配置简单，而且提供了HTTP API进行交互，使用也很简单
* 安全：支持SSL证书验证
* 快速：根据官方提供的benchmark数据，单实例支持每秒2k+读操作
* 可靠：采用raft算法，实现分布式系统数据的可用性和一致性

# （三）安装Keystone认证服务

## 1.创建keystone数据库并授权

mysql -uroot #登录root用户

CREATE DATABASE keystone; #创建数据库keystone

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY 'KEYSTONE\_DBPASS'; #给keystone数据库的所有表赋予权限，keystone用户在命令行登录模式下有所有权限，注意密码是‘KEYSTONE\_DBPASS’

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

## 2. 安装keystone软件包

1 #keystone软件包名openstack-keystone

2 #安装httpd和mod\_wsgi的原因是，社区主推apache+keystone

3 #openstack-keystone本质就是一款基于wsgi协议的web app,而httpd本质就是一个兼容wsgi协议的web server，所以我们需要为httpd安装mod\_wsgi模块

[root@controller ~]# yum install openstack-keystone httpd mod\_wsgi

## 3. 修改配置文件

cp -a /etc/keystone/keystone.conf{,.bak} #复制keystone的配置文件到.bak文件中

grep -Ev "^$|#" /etc/keystone/keystone.conf.bak > /etc/keystone/keystone.conf #从.bak文件中搜索出相应的配置信息到conf文件中，精简conf配置文件

[root@controller ~]# yum install -y openstack-utils #安装openstack-utils工具

让openstack-keystone能够知道如何连接到后端的数据库keystone

mysql+pymysql：pymysql是一个python库，使用python可以操作mysql原生sql

在keystone配置文件中设置keystone数据库连接的密码（具体可以查询keystone.conf配置文件），其实就是修改keystone.conf配置文件

格式： 用户名:密码@mysql地址/哪个库

注意在mysql数据库中创建密码的时候加上了引号“123”而在配置文件中不要加上引号，不然会保错

注意，地址写IP地址，不要写controller

[root@controller ~]# openstack-config --set /etc/keystone/keystone.conf database connection mysql+pymysql://keystone:KEYSTONE\_DBPASS@192.168.112.146/keystone

在keystone配置你文件中设置token令牌

[root@controller ~]# openstack-config --set /etc/keystone/keystone.conf token provider fernet

 可能有的时候大家在部署的时候会遇到报错，怎么办，虽然我没有遇到，但是如果各位遇到了不要慌

|  |  |
| --- | --- |
| 1 | tail -f -n 20 /var/log/keystone/keystone.log |

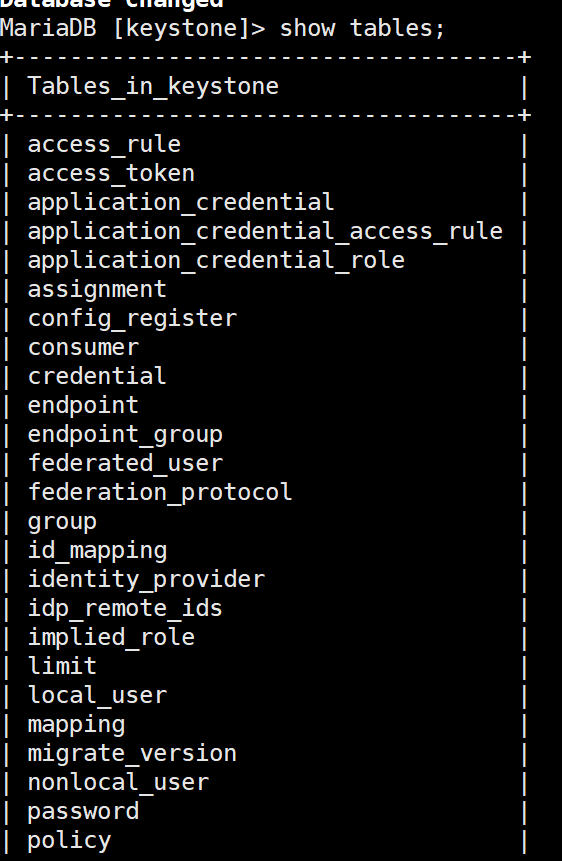
## 4. 填充数据库

su命令即作为root执行命令，-s 表示运行shell，-c表示运行shell命令，最后的keystone表示keystone用户

初始化数据库，之所以要初始化，是因为python的orm对象关系映射，需要初始化来生成数据库表结构。什么是orm（Object Relation Mapping）的缩写叫对象关系映射。

[root@controller ~]# su -s /bin/sh -c "keystone-manage db\_sync" keystone

在这一步之后，去keystone数据库查看，如果数据库初始化成功会生成很多数据表：



初始化Fernet密钥存储库 这是新版本的OpenStack的新功能，在Train版本下，keystone不再使用简单的字符串作为临时token，而是使用下面创建的fernet的用户来运行keystone。同时，keystone也不再对管理员用户和普通用户的服务端点区分使用不同的端口5000和35357，而是只使用5000端口不再使用35357端口。

[root@controller ~]# keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

[root@controller ~]# keystone-manage credential\_setup --keystone-user keystone --keystone-group keystone

上面的两个步骤是keystone对自己授权的一个过程，创建了一个keystone用户与一个keystone组。并对这个用户和组授权。因为keystone是对其他组件认证的服务，那么它自己就合格么？所以它先要对自己进行一下认证

## 5.修改apache配置

修改主机名，方便操作

hostnamectl set-hostname controller

配置hosts

192.168.112.146 controller

修改httpd服务器配置

[root@controller ~]# echo "ServerName controller" >> /etc/httpd/conf/httpd.conf

创建wsgi配置文件软链接 ，为mod\_wsgi模块添加配置文件，直接拷贝模块文件或者做软连接都可以，

wsgi：apache关于mod\_wsgi模块的配置文件

keystone：用来连接keystone

[root@controller ~]# ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/

wsgi-keystone.conf文件中的500端口说明keystone要开放这个端口，apache才能把请求交给keystone

启动和开机自启动apache

[root@controller ~]# systemctl enable httpd.service

[root@controller ~]# systemctl start httpd.service

## 6本质就是在为keystone创建catalog

注意，将controller都修改为IP地址

Perform the basic bootstrap process.（初始化keystone操作）

[root@controller ~]# keystone-manage bootstrap --bootstrap-password ADMIN\_PASS \

--bootstrap-admin-url http://192.168.112.146:5000/v3/ \

--bootstrap-internal-url http://192.168.112.146:5000/v3/ \

--bootstrap-public-url http://192.168.112.146:5000/v3/ \

--bootstrap-region-id RegionOne

## 7.初始化环境变量

同样，要将controller转换为IP地址

[root@controller ~]# cat >> ~/.bashrc << EOF

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS #就是keystone-manage中设定的--bootstrap-password

export OS\_PROJECT\_NAME=admin #项目名　project <-- 用户 <--角色 若想让用户获取权限必须要指定用户所在的项目是哪个

export OS\_USER\_DOMAIN\_NAME=Default #默认域

export OS\_PROJECT\_DOMAIN\_NAME=Default

export OS\_AUTH\_URL=http://192.168.112.146:5000/v3 #认证url

export OS\_IDENTITY\_API\_VERSION=3 #指定版本信息

EOF

source命令用法[#](https://www.cnblogs.com/shuiche/p/9436126.html" \l "source%E5%91%BD%E4%BB%A4%E7%94%A8%E6%B3%95)

source FileName

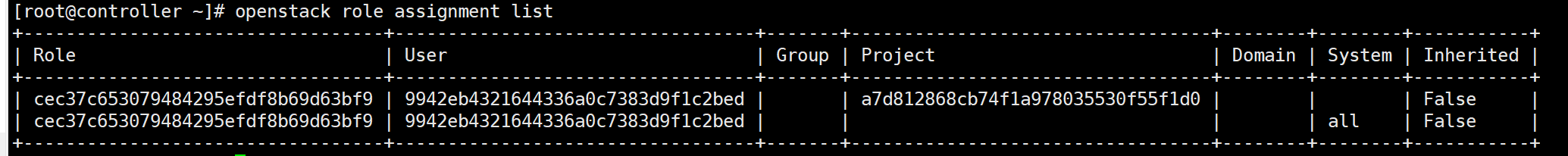
source命令作用[#](https://www.cnblogs.com/shuiche/p/9436126.html" \l "source%E5%91%BD%E4%BB%A4%E4%BD%9C%E7%94%A8)

在当前bash环境下读取并执行FileName中的命令。

\*注：该命令通常用命令“.”来替代。

[root@controller ~]# source ~/.bashrc #读取刚刚更新的文件

验证初始化操作成功：



[root@controller ~]# openstack token issue