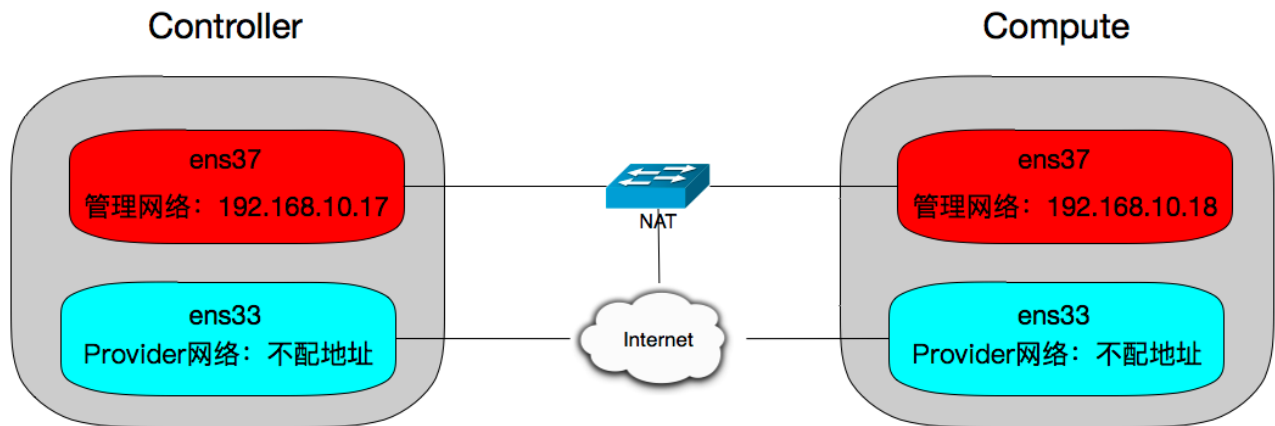


openstack_stein_2node_install

1、系统拓扑图

1. 系统拓扑图



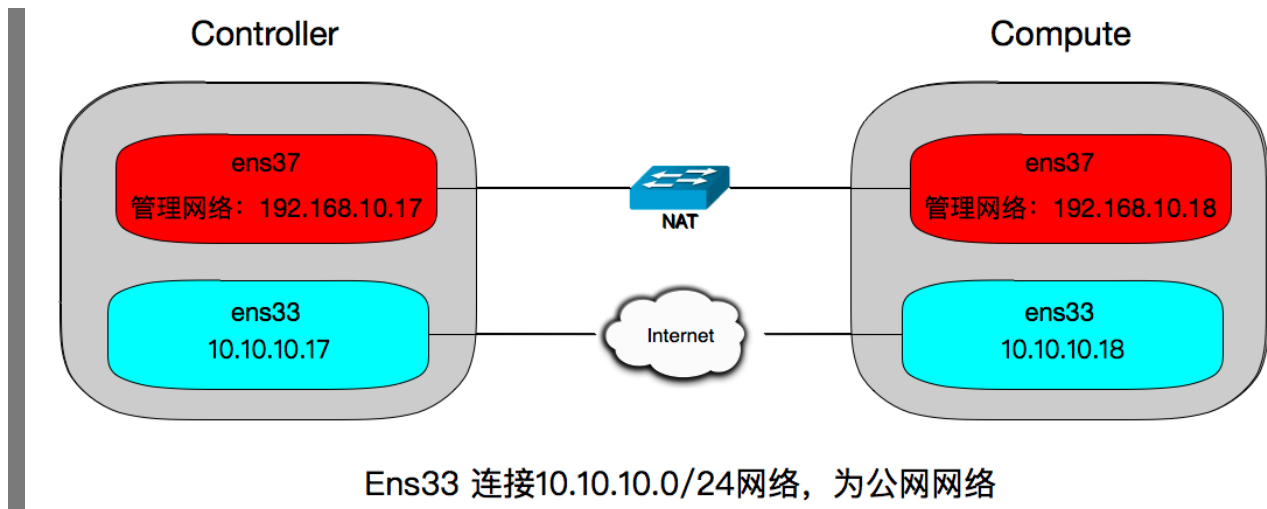
Ens33 连接10.10.10.0/24网络，为公网网络

2. 说明：

- 本次配置为两个节点的openstack
- controller与compute节点都有两块网卡ens33,ens37
- ens37连接192.168.10.0/24网络，此网络为管理网络（Management Network）
- ens33连接10.10.10.0/24网络，此网络为provider网络（provider Network），改网络为实例提供外网。改网络接口不需要配置地址。

3. 注意：

- ens33如果开始时不配置网络地址，则ens37必须能够上网，否则不能下载安装包
- 如果ens37不能够上网，只有ens33能够上网，则在配置开始时，对ens33设置ip地址，让controller节点能够连接internet，在openstack配置完成后，删除ens33的ip地址。本例中采用此方案。如下图所示：



2、基本环节搭建

2.1 安全(security)

1. openstack需要各种服务，每个服务都需要设置密码。
2. 设置密码，可以使用以下代码生成随机密码：

```
1 | root@controlller ~]# openssl rand -hex 10
2 | ea4f99cd40067b6dcfad
```

3. openstack需要设置密码的服务

| Password name | Description | 密码 |
|--------------------------------------|---|----------------------|
| Database password (no variable used) | Root password for the database | ea4f99cd40067b6dcfad |
| ADMIN_PASS | Password of user admin | ea4f99cd40067b6dcfad |
| CINDER_DBPASS | Database password for the Block Storage service | ea4f99cd40067b6dcfad |
| CINDER_PASS | Password of Block Storage service user cinder | ea4f99cd40067b6dcfad |
| DASH_DBPASS | Database password for the Dashboard | ea4f99cd40067b6dcfad |
| DEMO_PASS | Password of user demo | ea4f99cd40067b6dcfad |
| GLANCE_DBPASS | Database password for Image service | ea4f99cd40067b6dcfad |

| Password name | Description | 密码 |
|-----------------|--|----------------------|
| GLANCE_PASS | Password of Image service user glance | ea4f99cd40067b6dcfad |
| KEYSTONE_DBPASS | Database password of Identity service | ea4f99cd40067b6dcfad |
| METADATA_SECRET | Secret for the metadata proxy | ea4f99cd40067b6dcfad |
| NEUTRON_DBPASS | Database password for the Networking service | ea4f99cd40067b6dcfad |
| NEUTRON_PASS | Password of Networking service user neutron | ea4f99cd40067b6dcfad |
| NOVA_DBPASS | Database password for Compute service | ea4f99cd40067b6dcfad |
| NOVA_PASS | Password of Compute service user nova | ea4f99cd40067b6dcfad |
| PLACEMENT_PASS | Password of the Placement service user placement | ea4f99cd40067b6dcfad |
| RABBIT_PASS | Password of RabbitMQ user openstack | ea4f99cd40067b6dcfad |

4. 说明:

可以对以上密码设置一个统一的密码，如：openstack

2.2主机网络 (host networking)

2.2.1controller节点配置

1. 管理网络接口配置(ens37)配置： 1. 管理网络接口配置(ens37)配置： g

```

1 | root@controlller ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens37
2 | TYPE=Ethernet
3 | BOOTPROTO=static
4 | DEFROUTE=yegs
5 | PEERDNS=yes
6 | PEERROUTES=yes
7 | IPV4_FAILURE_FATAL=no
8 | IPV6INIT=yes

```

```
9  IPV6_AUTOCONF=yes
10 IPV6_DEFROUTE=yes
11 IPV6_PEERDNS=yes
12 IPV6_PEERROUTES=yes
13 IPV6_FAILURE_FATAL=no
14 UUID=af61a7aa-1a75-3758-8b1d-2e6acf6ace95
15 NAME=ens37
16 DEVICE=ens37
17 ONBOOT=yes
18 IPADDR=192.168.10.17
19 NETMASK=255.255.255.0
```

2. provider 网络接口(ens33) 配置

```
1  root@controlller ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens33
2  TYPE=Ethernet
3  BOOTPROTO=static
4  DEFROUTE=yes
5  PEERDNS=yes
6  PEERROUTES=yes
7  IPV4_FAILURE_FATAL=no
8  IPV6INIT=yes
9  IPV6_AUTOCONF=yes
10 IPV6_DEFROUTE=yes
11 IPV6_PEERDNS=yes
12 IPV6_PEERROUTES=yes
13 IPV6_FAILURE_FATAL=no
14 NAME=ens33
15 DEVICE=ens33
16 ONBOOT=yes
17 IPADDR=10.10.10.17
18 NETMASK=255.255.255.0
19 GATEWAY=10.10.10.2
20 DNS1=8.8.8.8
```

3. /etc/hosts配置

```
1  root@controlller ~]# cat /etc/hosts
2  127.0.0.1    localhost localhost.localdomain localhost4 localhost4.local
```

```
domain4
3  ::1          localhost localhost.localdomain localhost6 localhost6.local
domain6
4  192.168.10.17 controller
5  192.168.10.18 compute
```

说明:

设置管理网络地址，不要设置provider网络地址

2.2.2 Compute节点配置

1. 管理网络接口(ens37) 配置

```
1  [root@compute ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens37
2  TYPE=Ethernet
3  BOOTPROTO=static
4  DEFROUTE=yes
5  PEERDNS=yes
6  PEERROUTES=yes
7  IPV4_FAILURE_FATAL=no
8  IPV6INIT=yes
9  IPV6_AUTOCONF=yes
10 IPV6_DEFROUTE=yes
11 IPV6_PEERDNS=yes
12 IPV6_PEERROUTES=yes
13 IPV6_FAILURE_FATAL=no
14 UUID=af61a7aa-1a75-3758-8b1d-2e6acf6ace95
15 NAME=ens37
16 DEVICE=ens37
17 ONBOOT=yes
18 IPADDR=192.168.10.18
19 NETMASK=255.255.255.02
```

2 . provider 网络接口(ens33) 配置:

```
1  [root@compute ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens33
2  TYPE=Ethernet
3  BOOTPROTO=static
4  DEFROUTE=yes
5  PEERDNS=yes
```

```
6 PEERROUTES=yes
7 IPV4_FAILURE_FATAL=no
8 IPV6INIT=yes
9 IPV6_AUTOCONF=yes
10 IPV6_DEFROUTE=yes
11 IPV6_PEERDNS=yes
12 IPV6_PEERROUTES=yes
13 IPV6_FAILURE_FATAL=no
14 NAME=ens33
15 DEVICE=ens33
16 ONBOOT=yes
17 IPADDR=10.10.10.18
18 NETMASK=255.255.255.0
19 GATEWAY=10.10.10.2
20 DNS1=8.8.8.8
```

3. /etc/hosts配置

拷贝 controller节点 /etc/hosts文件到compute节点，命令如下：

```
1 [root@compute ~]# scp root@192.168.10.17:/etc/hosts /etc/hosts
2 The authenticity of host '192.168.10.17 (192.168.10.17)' can't be estab
  lished.
3 ECDSA key fingerprint is SHA256:YHPabnLNSR7B1jly0ZgI2PxXAasnYD9K0lsQRM3
  8kTk.
4 ECDSA key fingerprint is MD5:8d:be:fd:de:3a:1d:20:dc:e1:ae:b4:17:c3:a5:
  2f:e4.
5 Are you sure you want to continue connecting (yes/no)? yes
6 Warning: Permanently added '192.168.10.17' (ECDSA) to the list of known
  hosts.
7 root@192.168.10.17's password:
8 hosts                                100% 205   182.9KB/s   0
  0:00
```

2.2.3测试连通型

1. controller 能否ping通 compute节点与外网

```
1 [root@controlller ~]# ping compute
2 PING compute (192.168.10.18) 56(84) bytes of data.
```

```

3 64 bytes from compute (192.168.10.18): icmp_seq=1 ttl=64 time=0.639 ms
4 64 bytes from compute (192.168.10.18): icmp_seq=2 ttl=64 time=0.483 ms
5 64 bytes from compute (192.168.10.18): icmp_seq=3 ttl=64 time=0.682 ms
6 64 bytes from compute (192.168.10.18): icmp_seq=4 ttl=64 time=0.833 ms
7 64 bytes from compute (192.168.10.18): icmp_seq=5 ttl=64 time=0.744 ms
8 --- compute ping statistics ---
9 5 packets transmitted, 5 received, 0% packet loss, time 4004ms
10 rtt min/avg/max/mdev = 0.483/0.676/0.833/0.117 ms
11 [root@controlller ~]# ping 10.10.10.18
12 PING 10.10.10.18 (10.10.10.18) 56(84) bytes of data.
13 64 bytes from 10.10.10.18: icmp_seq=1 ttl=64 time=0.771 ms
14 64 bytes from 10.10.10.18: icmp_seq=2 ttl=64 time=0.550 ms
15 64 bytes from 10.10.10.18: icmp_seq=3 ttl=64 time=0.678 ms
16 ^C
17 --- 10.10.10.18 ping statistics ---
18 3 packets transmitted, 3 received, 0% packet loss, time 2005ms
19 rtt min/avg/max/mdev = 0.550/0.666/0.771/0.093 ms
20 [root@controlller ~]# ping www.baidu.com
21 PING www.a.shifen.com (183.232.231.174) 56(84) bytes of data.
22 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=1 ttl=128 ti
    me=42.8 ms
23 64 bytes from 183.232.231.174 (183.232.231.174): icmp_seq=2 ttl=128 ti
    me=42.6 ms
24 ^C
25 --- www.a.shifen.com ping statistics ---
26 2 packets transmitted, 2 received, 0% packet loss, time 1005ms
27 rtt min/avg/max/mdev = 42.662/42.762/42.862/0.100 ms

```

2. compute节点能否 ping通 controller节点与外网

```

1 [root@compute ~]# ping controller
2 PING controller (192.168.10.17) 56(84) bytes of data.
3 64 bytes from controller (192.168.10.17): icmp_seq=1 ttl=64 time=0.497
    ms
4 64 bytes from controller (192.168.10.17): icmp_seq=2 ttl=64 time=0.734
    ms
5 ^C
6 --- controller ping statistics ---
7 2 packets transmitted, 2 received, 0% packet loss, time 1001ms
8 rtt min/avg/max/mdev = 0.497/0.615/0.734/0.121 ms
9 [root@compute ~]# ping 10.10.10.17

```

```
10 | PING 10.10.10.17 (10.10.10.17) 56(84) bytes of data.
11 | 64 bytes from 10.10.10.17: icmp_seq=1 ttl=64 time=0.478 ms
12 | 64 bytes from 10.10.10.17: icmp_seq=2 ttl=64 time=0.598 ms
13 | ^C
14 | --- 10.10.10.17 ping statistics ---
15 | 2 packets transmitted, 2 received, 0% packet loss, time 1001ms
16 | rtt min/avg/max/mdev = 0.478/0.538/0.598/0.060 ms
17 | [root@compute ~]# ping www.baidu.com
18 | PING www.a.shifen.com (183.232.231.172) 56(84) bytes of data.
19 | 64 bytes from 183.232.231.172 (183.232.231.172): icmp_seq=1 ttl=128 ti
    | me=29.2 ms
20 | 64 bytes from 183.232.231.172 (183.232.231.172): icmp_seq=2 ttl=128 ti
    | me=30.2 ms
21 | ^C
22 | --- www.a.shifen.com ping statistics ---
23 | 2 packets transmitted, 2 received, 0% packet loss, time 1003ms
24 | rtt min/avg/max/mdev = 29.231/29.720/30.210/0.518 ms
```

说明:

所有测试必须能够ping通

2.3 配置NTP服务

2.3.1 controller节点配置

1. 安装chrony包

```
1 | # yum install chrony
```

2. 配置/etc/chrony.conf文件

增加两行内容:

```
1 | server 192.168.10.17 iburst
2 | allow 192.168.10.0/24
```

3. 设置服务自启动与启动服务

```
1 | # systemctl enable chronyd.service
```



```
2 | # systemctl start chronyd.service
```

2.3.2 compute节点配置

1. 安装chrony包

```
1 | # yum install chrony
```

2. 配置/etc/chrony.conf文件 删除所有的server，增加1行

```
1 | server controller iburst
```

3. 设置服务自启动与启动服务

```
1 | # systemctl enable chronyd.service
2 | # systemctl start chronyd.service
```

2.3.3 测试

1. controller节点

```
1 | [root@controlller ~]# chronyc sources
2 | 210 Number of sources = 5
3 | MS Name/IP address          Stratum Poll Reach LastRx Last sample
4 | =====
5 | ^- controller                3    6   377    33   -16us[ -16us] +/-
6 |   21ms
7 | ^* 203.107.6.88              2    8   377    90  -176us[ -227us] +/-
8 |   21ms
9 | ^? ntp1.ams1.nl.leaseweb.net 2    9     0   39m   +14ms[ -9633s] +/-
10 |   202ms
11 | ^- ntp.xtom.nl               2    7   377    18  +6113us[+6113us] +/-
12 |   111ms
13 | ^- electrode.felixc.at       3    7   307    89   +12ms[ +12ms] +/-
14 |   182
```

2. compute节点

```
1 root@compute yum.repos.d]# chronyc sources
2 210 Number of sources = 1
3 MS Name/IP address          Stratum Poll Reach LastRx Last sample
4 =====
5 ^? controller                0    7    0    -    +0ns[  +0ns] +/-
   -    0ns
```

2.4 openstack packages配置

2.4.1 安装openstack repository

对于所有节点（controller,compute节点）安装

1. stein版本

```
1 [root@controller ~]# yum install centos-release-openstack-stein
```

2. rocky版本

```
1 [root@controller ~]# yum install centos-release-openstack-rocky
```

3. queen版本

```
1 [root@controller ~]# yum install centos-release-openstack-queen
```

安装完成后，在/etc/yum.repos.d文件夹下新增以下的repo文件：

```
1 root@controller yum.repos.d]# ls
2 CentOS-NFS-Ganesha-28.repo  CentOS-OpenStack-stein.repo
3 CentOS-Ceph-Nautilus.repo  CentOS-QEMU-EV.repo
4 CentOS-Storage-common.repo
```

2.4.2 安装其他包

1. 更新系统

```
1 | [root@controlller yum.repos.d]# yum upgrade
```

2. 安装openstack client

```
1 | [root@controlller yum.repos.d]# yum install python-openstackclient
```

3. 安装 openstack-selinux

对于centos来说，sexlinux，默认处于打开状态，有两种解决办法：

(1) 禁止sexlinux，打开/etc/selinux/conf,设置：

```
1 | SELINUX=disabled
```

(2)安装openstack-selinux，解决selinux限制

```
1 | # yum install openstack-selinux
```

2.5 安装数据库（mariadb）

在controller节点安装数据库

2.5.1 安装mariadb

```
1 | # yum install mariadb mariadb-server python2-PyMySQL
```

2.5.2 配置mysql

1. 在/etc/my.cnf.d下创建文件openstack.cnf,内容如下：

```
1 |  
2 | bind-address = 192.168.10.17
```

```
3
4 default-storage-engine = innodb
5 innodb_file_per_table = on
6 max_connections = 4096
7 collation-server = utf8_general_ci
8 character-set-server = utf8
```

注意：如果启动客户端报错，请在bind-address行前面增加一行，内容为：[mysqld]

2. 配置自启动与启动mysql服务

```
1 # systemctl enable mariadb.service
2 # systemctl start mariadb.service
```

3. 安装数据库,设置root用户密码为：123456

```
1 [root@controlller my.cnf.d]# mysql_secure_installation
2
3 NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
4     SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!
5
6 In order to log into MariaDB to secure it, we'll need the current
7 password for the root user.  If you've just installed MariaDB, and
8 you haven't set the root password yet, the password will be blank,
9 so you should just press enter here.
10
11 Enter current password for root (enter for none):
12 OK, successfully used password, moving on...
13
14 Setting the root password ensures that nobody can log into the MariaDB
15 root user without the proper authorisation.
16
17 Set root password? [Y/n] Y
18 New password:
19 Re-enter new password:
20 Password updated successfully!
21 Reloading privilege tables..
22     ... Success!
23
24
```

```
25 By default, a MariaDB installation has an anonymous user, allowing any
26 one
27 to log into MariaDB without having to have a user account created for
28 them. This is intended only for testing, and to make the installation
29 go a bit smoother. You should remove them before moving into a
30 production environment.
31
32 Remove anonymous users? [Y/n] Y
33 ... Success!
34
35 Normally, root should only be allowed to connect from 'localhost'. Th
36 is
37 ensures that someone cannot guess at the root password from the networ
38 k.
39
40 Disallow root login remotely? [Y/n] n
41 ... skipping.
42
43 By default, MariaDB comes with a database named 'test' that anyone can
44 access. This is also intended only for testing, and should be removed
45 before moving into a production environment.
46
47 Remove test database and access to it? [Y/n] Y
48 - Dropping test database...
49 ... Success!
50 - Removing privileges on test database...
51 ... Success!
52
53 Reloading the privilege tables will ensure that all changes made so fa
54 r
55 will take effect immediately.
56
57 Reload privilege tables now? [Y/n] Y
58 ... Success!
59
60 Cleaning up...
61
62 All done! If you've completed all of the above steps, your MariaDB
63 installation should now be secure.
64
65 Thanks for using MariaDB!
```

4. 设置客户端免密登录

正常情况下使用mysql客户端命令登录mysql需要输入用户名与密码，可以在/etc/my.cnf.d/client.conf下配置：

```
1 [client]
2
3 # This group is not read by mysql client library,
4 # If you use the same .cnf file for MySQL and MariaDB,
5 # use it for MariaDB-only client options
6 user=root
7 password="123456"
```

这样只有在终端中输入mysql就可以直接登录了

```
1 root@controlller my.cnf.d]# mysql
2 Welcome to the MariaDB monitor.  Commands end with ; or \g.
3 Your MariaDB connection id is 22
4 Server version: 10.3.10-MariaDB MariaDB Server
5
6 Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
7
8 Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
9
10 MariaDB [(none)]>
```

2.6安装消息队列

openstack中使用消息队列，消息队列用于controller节点与compute之间的消息通信，openstack使用消息队列来协调各种服务之间的操作与状态信息，消息队列服务通常安装在controller节点上。消息队列软件有：RabbitMQ，Qpid，ZeroMQ,本次使用RabbitMQ

在controller节点上安装

2.6.1安装rabbitmq

```
1 [root@controlller ~]# yum install rabbitmq-server -y
```

2.6.2 设置开机自启动与启动服务

```
1 root@controlller ~]# systemctl enable rabbitmq-server.service
2 Created symlink from /etc/systemd/system/multi-user.target.wants/rabbitmq-server.service to /usr/lib/systemd/system/rabbitmq-server.service.
3 [root@controlller ~]# systemctl start rabbitmq-server.service
4 [root@controlller ~]#
```

2.6.3 设置消息队列用户名与密码

1. 设置用户与密码

```
1 [root@controlller ~]# rabbitmqctl add_user openstack ea4f99cd40067b6dcfad
2 Creating user "openstack"
```

2. 查看设置的用户与密码

```
1 root@controlller ~]# rabbitmqctl list_users
2 Listing users
3 openstack []
4 guest [administrator]
```

2.6.4 设置用户权限

1. 语法:

```
1 rabbitmqctl set_permissions <user> <conf> <write> <read>
```

2. 说明:

- 是指为某个用户设置权限
- 设置配置权限
- 设置写权限
- 设置读权限

3. 对openstack用户设置权限

```
1 [root@controlller ~]# rabbitmqctl set_permissions openstack ".*" ".*" ".  
*"
2 Setting permissions for user "openstack" in vhost "/"
```

说明:

对openstack用户设置了所有的配置，写与读的权限

4. 查看openstack用户的权限

```
1 [root@controlller ~]# rabbitmqctl list_user_permissions openstack
2 Listing permissions for user "openstack"
3 /      .*      .*      .*
```

2.7 安装缓存服务(MemCached)

openstack为其他服务提供身份识别服务，需要使用Memcached来缓存tokens，缓存服务通常安装在controller节点上。

在controller节点上安装

2.7.1 安装memcached包

```
1 [root@controlller ~]# yum install memcached python-memcached -y
```

2.7.2 配置memcached

编辑/etc/sysconfig/memcached文件，修改：

```
1 OPTIONS="-l 127.0.0.1,:::1"
```

为：

```
1 OPTIONS="-l 127.0.0.1,:::1,controller"
```

2.7.3 设置自启动与启动服务


```
1 [root@controlller ~]# systemctl enable memcached.service
2 Created symlink from /etc/systemd/system/multi-user.target.wants/memcached.service to /usr/lib/systemd/system/memcached.service.
3 [root@controlller ~]# systemctl start memcached.service
4 [root@controlller ~]#
```

查看服务是否启动成功：

```
1 [root@controlller ~]# systemctl status memcached.service
```

2.8 配置Etcd

Etcd是一种可靠的分布式键值对管理软件，可以存储分布式键锁，存储配置信息，跟踪服务的是否存活等应用场景。openstack服务会用到Etcd软件。

etcd服务安装在controller节点上

2.8.1 安装包

```
1 [root@controlller ~]# yum install etcd
```

2.8.2 配置etcd.conf

编辑/etc/etcd/etcd.conf,内容如下：

```
1  #[Member]
2  ETCD_DATA_DIR="/var/lib/etcd/default.etcd"
3  ETCD_LISTEN_PEER_URLS="http://192.168.10.17:2380"
4  ETCD_LISTEN_CLIENT_URLS="http://192.168.10.17:2379"
5  ETCD_NAME="controller"
6  #[Clustering]
7  ETCD_INITIAL_ADVERTISE_PEER_URLS="http://192.168.10.17:2380"
8  ETCD_ADVERTISE_CLIENT_URLS="http://192.168.10.17:2379"
9  ETCD_INITIAL_CLUSTER="controller=http://192.168.10.17:2380"
10 ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster-01"
11 ETCD_INITIAL_CLUSTER_STATE="new"
```

2.8.3 设置自启动与启动服务

```
1 # systemctl enable etcd
2 # systemctl start etcd
```

查看服务是否启动成功:

x

```
1 [root@controlller ~]# systemctl status etcd
2 • etcd.service - Etcd Server
3   Loaded: loaded (/usr/lib/systemd/system/etcd.service; enabled; vend
or preset: disabled)
4   Active: active (running) since 日 2019-10-13 23:01:17 CST; 1min 0s
ago
5   Main PID: 43139 (etcd)
6   CGroup: /system.slice/etcd.service
7           └─43139 /usr/bin/etcd --name=controller --data-dir=/var/li
b/etcd/d...
8
9 10月 13 23:01:17 controlller etcd[43139]: 850ca1f0fc28aca8 received Ms
gVote...2
10 10月 13 23:01:17 controlller etcd[43139]: 850ca1f0fc28aca8 became lead
er at...2
11 10月 13 23:01:17 controlller etcd[43139]: raft.node: 850ca1f0fc28aca8
elect...2
12 10月 13 23:01:17 controlller etcd[43139]: published {Name:controller C
lient...b
13 10月 13 23:01:17 controlller etcd[43139]: setting up the initial clust
er ve...3
14 10月 13 23:01:17 controlller systemd[1]: Started Etcd Server.
15 10月 13 23:01:17 controlller etcd[43139]: set the initial cluster vers
ion t...3
16 10月 13 23:01:17 controlller etcd[43139]: enabled capabilities for ver
sion 3.3
17 10月 13 23:01:17 controlller etcd[43139]: ready to serve client reques
ts
18 10月 13 23:01:17 controlller etcd[43139]: serving insecure client requ
ests ...!
19 Hint: Some lines were ellipsized, use -l to show in full.
```