云操作系统应用

CONTENTS

目 录

一、 OpenStack回顾

二、 环境配置

一、 OpenStack回顾

终于正式进入OpenStack操作部分了。从现在开始，我将带着读者一步一步地揭开OpenStack的神秘面纱。

二、 环境配置

初学者需要自行学习centos7操作系统的常用命令

配置网卡文档【网卡配置文档】（见第三章（2）PPT）

配置yum源文档【 YUM源配置文档】（见第三章（3）PPT）

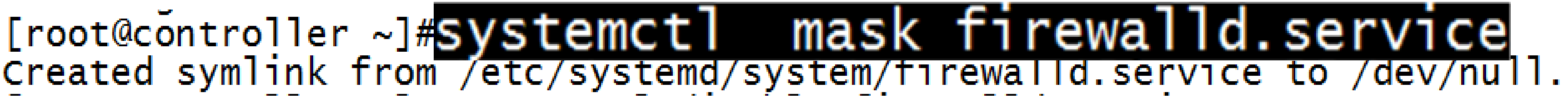
安全配置

1.防火墙设置

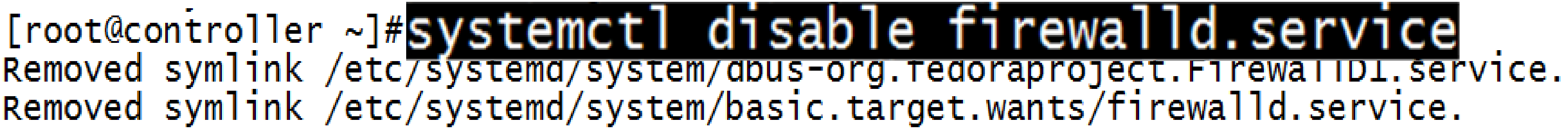
CentOS 7 中默认启用了 Firewall 防火墙，在安装过程中，有些步骤可能会失败，除非你禁用或者修改防火墙规则。在入门学习中，我们将采用关闭防火墙的方法。

关闭控制节点防火墙：

```
# systemctl mask firewalld.service
# systemctl disable firewalld.service
```

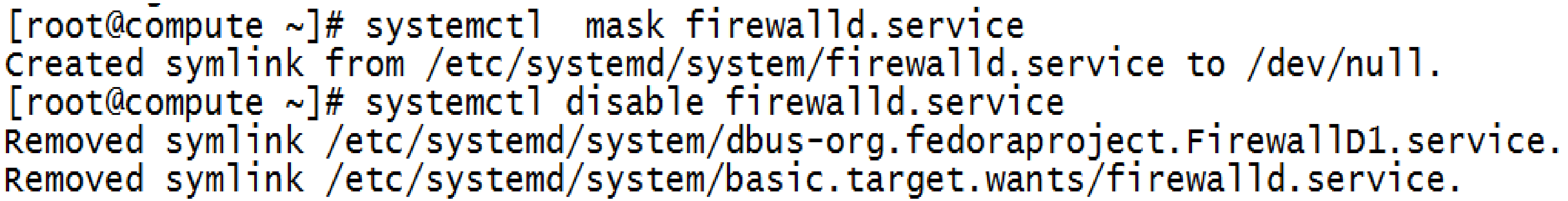


```
[root@controller ~]#systemctl  mask firewalld.service
Created symlink from /etc/systemd/system/firewalld.service to /dev/null.
```



```
[root@controller ~]#systemctl disable firewalld.service
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
Removed symlink /etc/systemd/system/basic.target.wants/firewalld.service.
```

关闭计算节点防火墙：



```
[root@compute ~]# systemctl  mask firewalld.service
Created symlink from /etc/systemd/system/firewalld.service to /dev/null.
[root@compute ~]# systemctl disable firewalld.service
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
Removed symlink /etc/systemd/system/basic.target.wants/firewalld.service.
```

网络配置

网络的连通性非常重要，各个节点需要做到网络互相 Ping 通，使之处于同一个网络中。

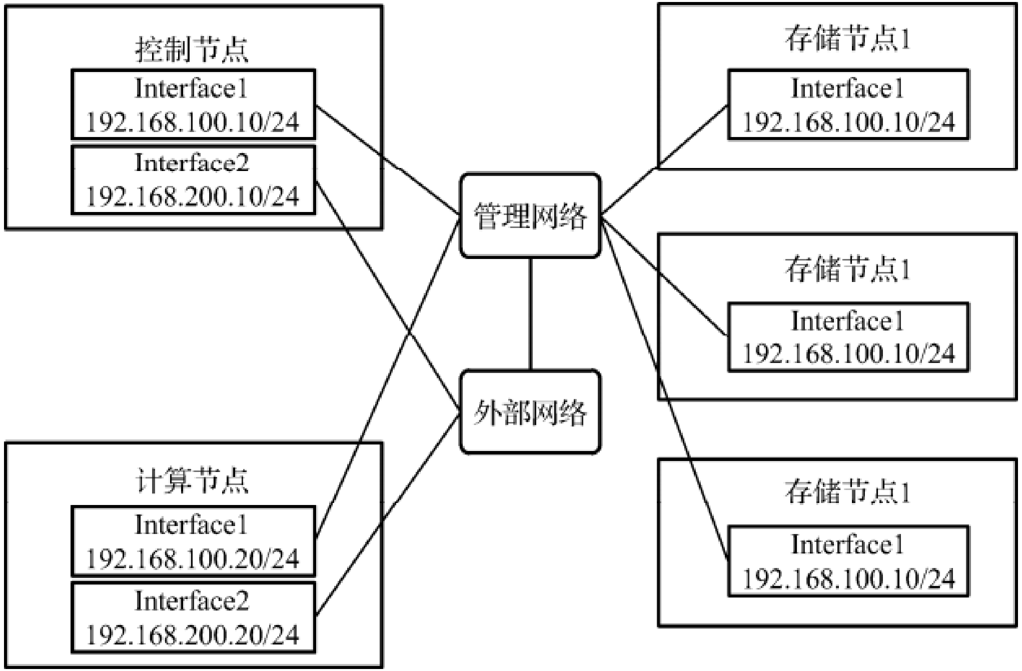


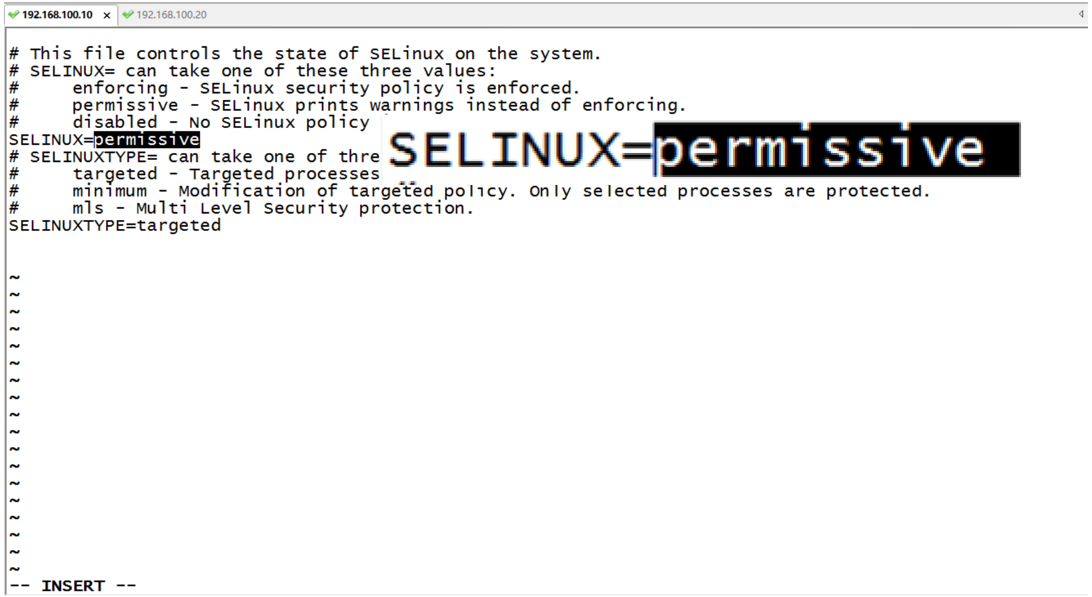
图 3-4　配置图

安全配置

2.Selinux设置（控制节点）

编辑/etc/selinux/config 文件。

```
# vi /etc/selinux/config
  SELINUX=permissive
```

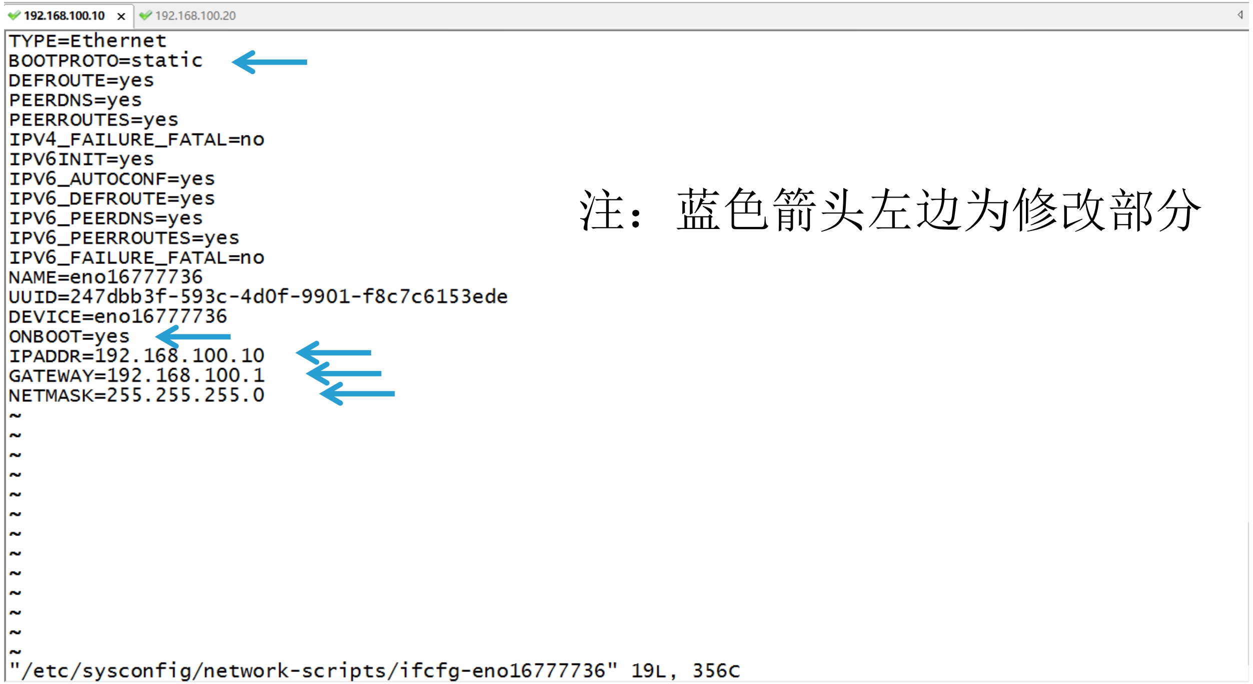


```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy
SELINUX=permissive
# SELINUXTYPE= can take one of thre
#     targeted - Targeted processes
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

网络配置

配置 控制节点网络信息。(注:不同的虚拟机可能网口名称不同。需按照自己的网口名称配置)
网口 eno16777736：作为管理网络使用，配置 IP 为 192.168.100.10/24。
网口 eno33554960：作为外部网络使用，配置 IP 为 192.168.200.10/24。

```
# vi /etc/sysconfig/network-scripts/ifcfg-eno16777736
```



```
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno16777736
UUID=247dbb3f-593c-4d0f-9901-f8c7c6153ede
DEVICE=eno16777736
ONBOOT=yes
IPADDR=192.168.100.10
GATEWAY=192.168.100.1
NETMASK=255.255.255.0
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/sysconfig/network-scripts/ifcfg-eno16777736" 19L, 356C
```

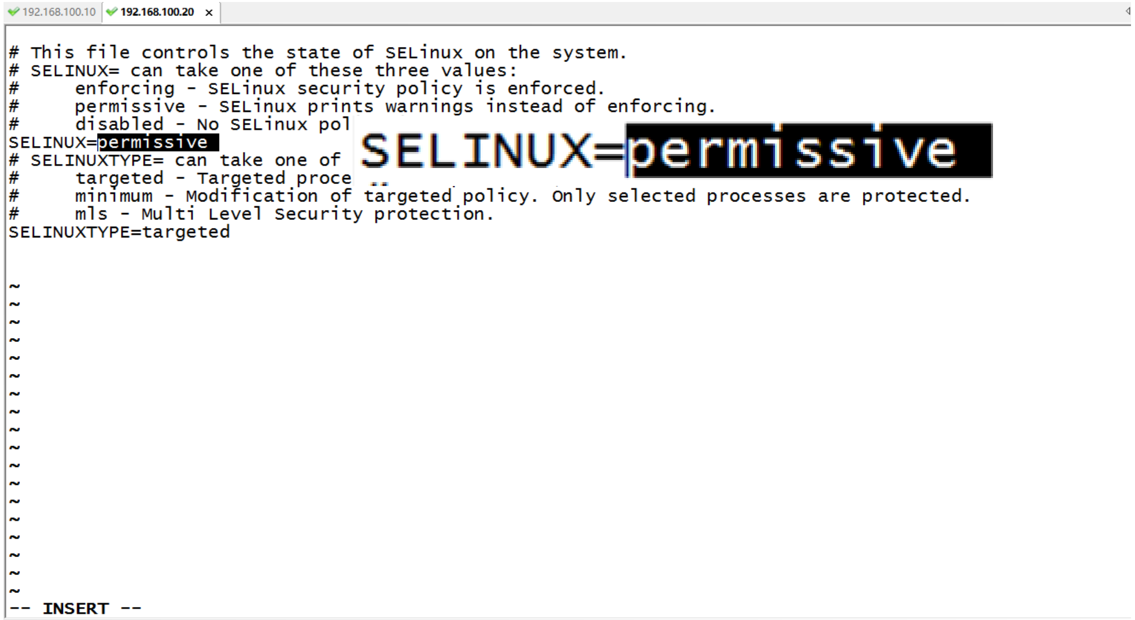
注：蓝色箭头左边为修改部分

安全配置

2.Selinux设置（计算节点）

编辑/etc/selinux/config 文件，修改完成之后**重启虚拟机使防火墙配置生效**。

```
# vi /etc/selinux/config
  SELINUX=permissive
```



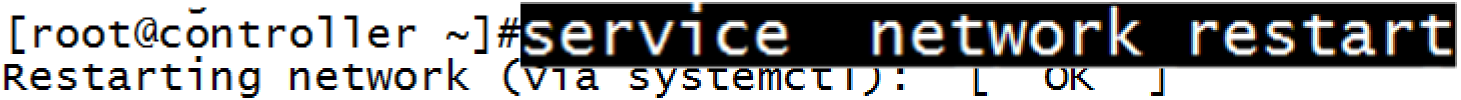
```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux pol
SELINUX=permissive
# SELINUXTYPE= can take one of
#     targeted - Targeted proce
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted


~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

SELINUX=permissive

网络配置

网口 eno16777736：作为管理网络使用，配置 IP 为 192.168.100.10/24。
网口 eno33554960：作为外部网络使用，配置 IP 为 192.168.200.10/24。
修改完成之后请使用以下命令重启生效网卡配置

```
# service  network restart
```

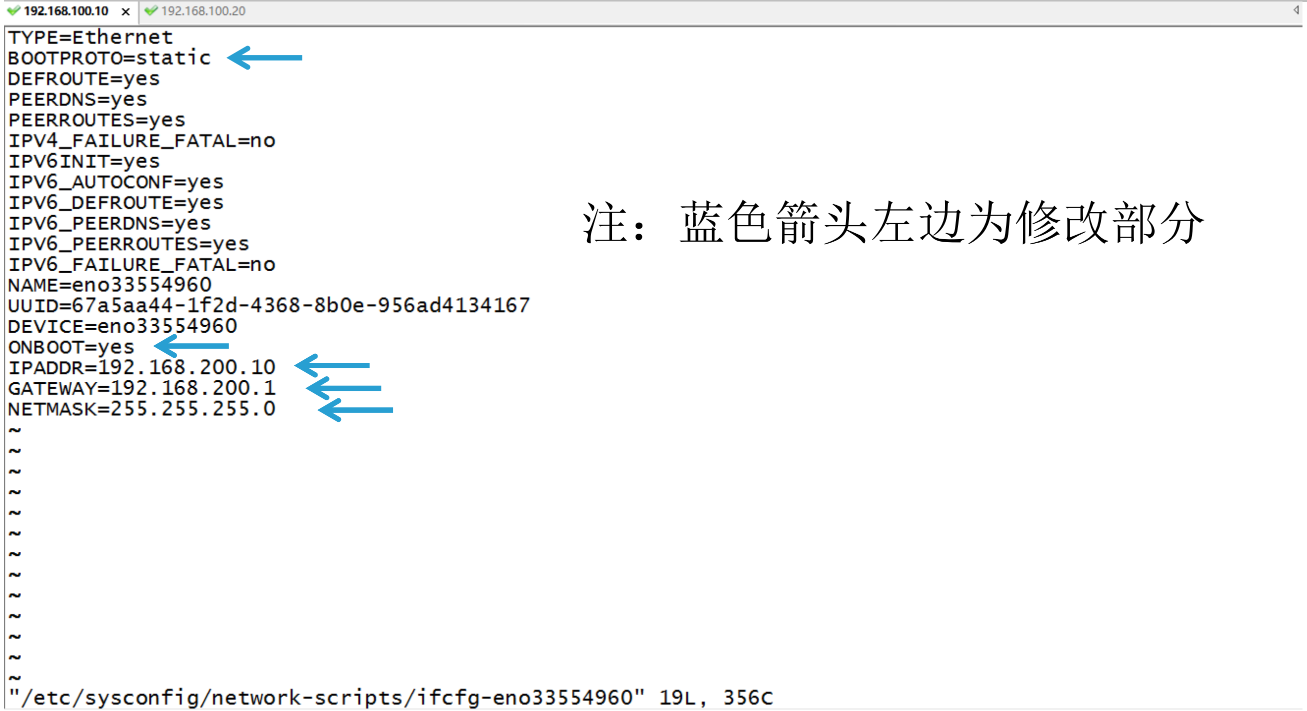


```
[root@controller ~]#service  network restart
Restarting network (via systemctl):  [  OK  ]
```

网络配置

配置 控制节点网络信息。(注:不同的虚拟机可能网口名称不同。需按照自己的网口名称配置)
网口 eno16777736：作为管理网络使用，配置 IP 为 192.168.100.10/24。
网口 eno33554960：作为外部网络使用，配置 IP 为 192.168.200.10/24。

```
# vi /etc/sysconfig/network-scripts/ifcfg-eno33554960
```



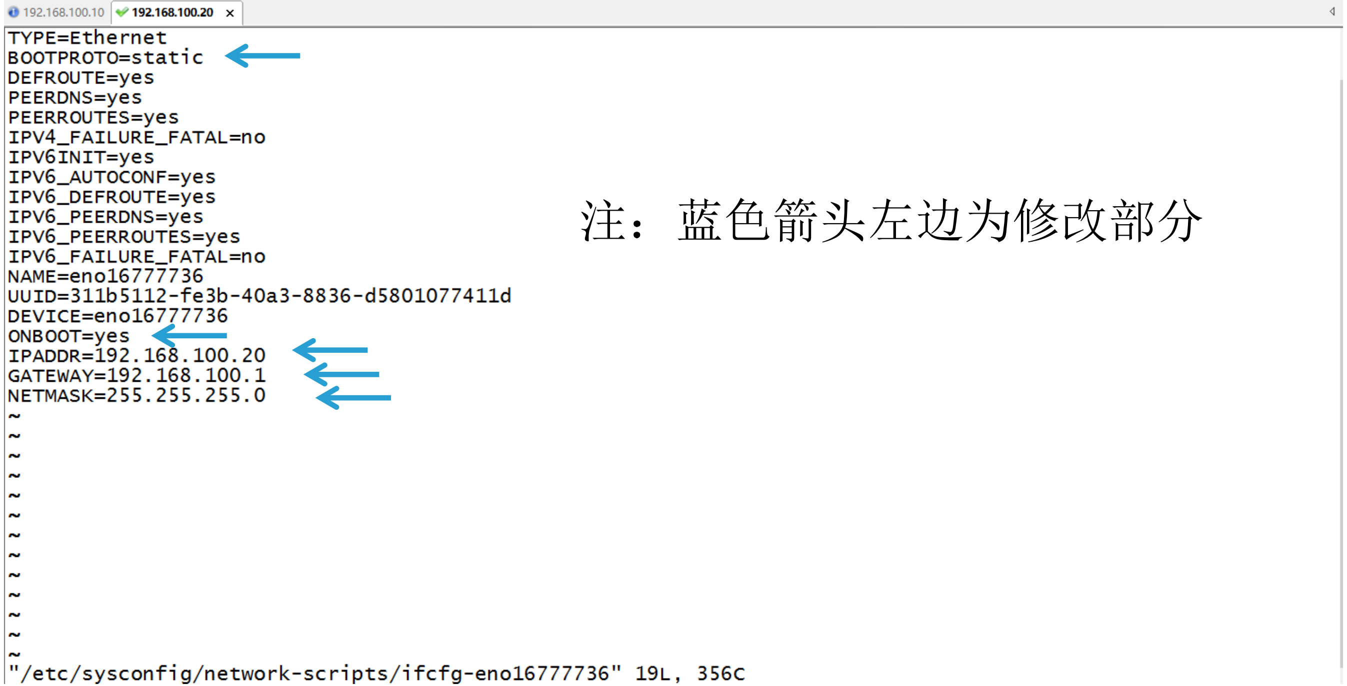
```
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno33554960
UUID=67a5aa44-1f2d-4368-8b0e-956ad4134167
DEVICE=eno33554960
ONBOOT=yes
IPADDR=192.168.200.10
GATEWAY=192.168.200.1
NETMASK=255.255.255.0
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/sysconfig/network-scripts/ifcfg-eno33554960" 19L, 356C
```

注：蓝色箭头左边为修改部分

网络配置

配置 计算节点网络信息。(注:不同的虚拟机可能网口名称不同。需按照自己的网口名称配置)
网口 eno16777736：作为管理网络使用，配置 IP 为 192.168.100.20/24。
网口 eno33554960：作为外部网络使用，配置 IP 为 192.168.200.20/24。

```
# vi /etc/sysconfig/network-scripts/ifcfg-eno16777736
```



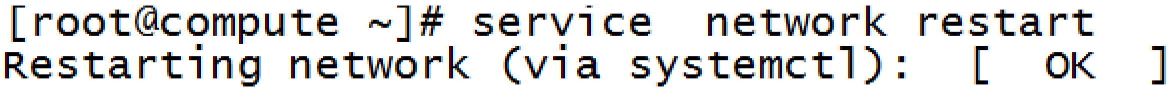
```
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno16777736
UUID=311b5112-fe3b-40a3-8836-d5801077411d
DEVICE=eno16777736
ONBOOT=yes
IPADDR=192.168.100.20
GATEWAY=192.168.100.1
NETMASK=255.255.255.0
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/sysconfig/network-scripts/ifcfg-eno16777736" 19L, 356C
```

注：蓝色箭头左边为修改部分

网络配置

网口 eno16777736：作为管理网络使用，配置 IP 为 192.168.100.10/24。
网口 eno33554960：作为外部网络使用，配置 IP 为 192.168.200.10/24。
修改完成之后请使用以下命令重启生效网卡配置

```
# service  network restart
```

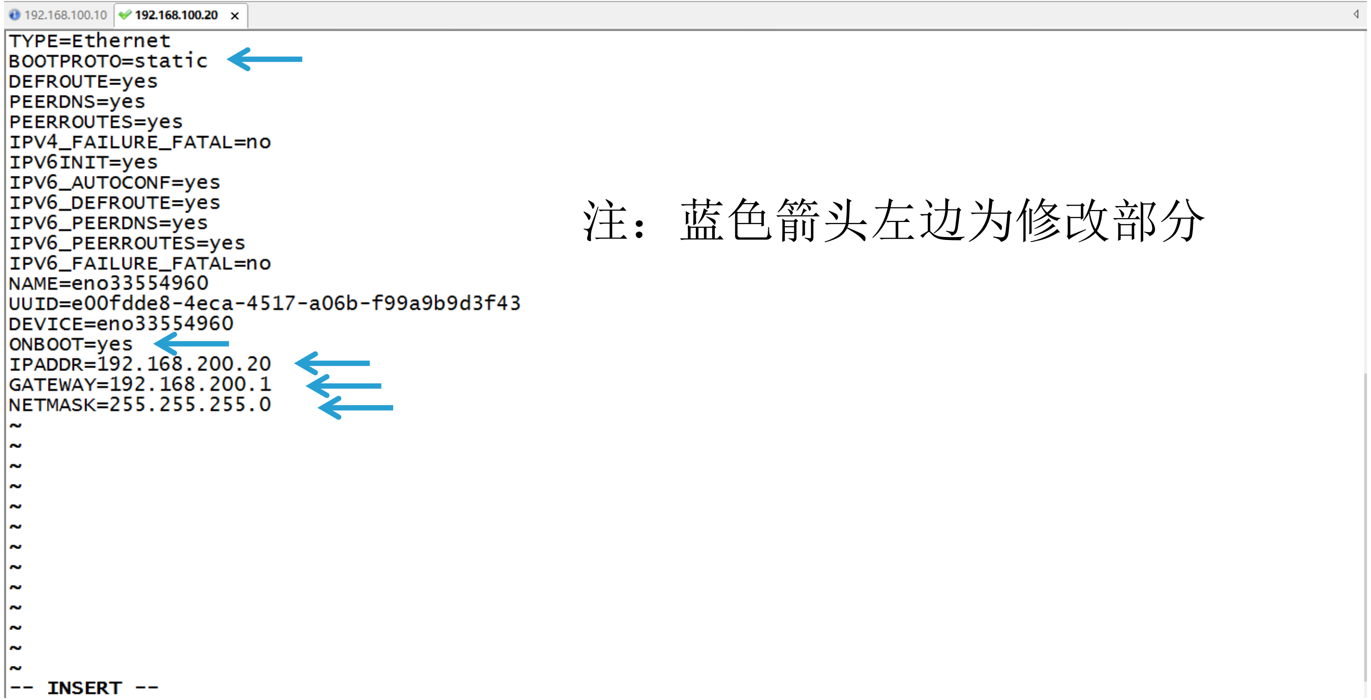


```
[root@compute ~]# service  network restart
Restarting network (via systemctl):  [  OK  ]
```

网络配置

配置 计算节点网络信息。(注:不同的虚拟机可能网口名称不同。需按照自己的网口名称配置)
网口 eno16777736：作为管理网络使用，配置 IP 为 192.168.100.10/24。
网口 eno33554960：作为外部网络使用，配置 IP 为 192.168.200.10/24。

```
# vi /etc/sysconfig/network-scripts/ifcfg-eno33554960
```



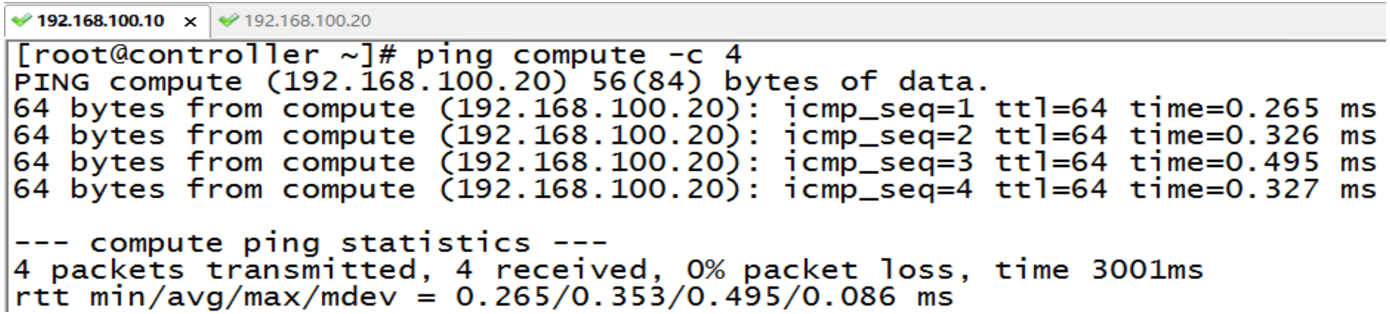
```
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno33554960
UUID=e00fdde8-4eca-4517-a06b-f99a9b9d3f43
DEVICE=eno33554960
ONBOOT=yes
IPADDR=192.168.200.20
GATEWAY=192.168.200.1
NETMASK=255.255.255.0
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

注：蓝色箭头左边为修改部分

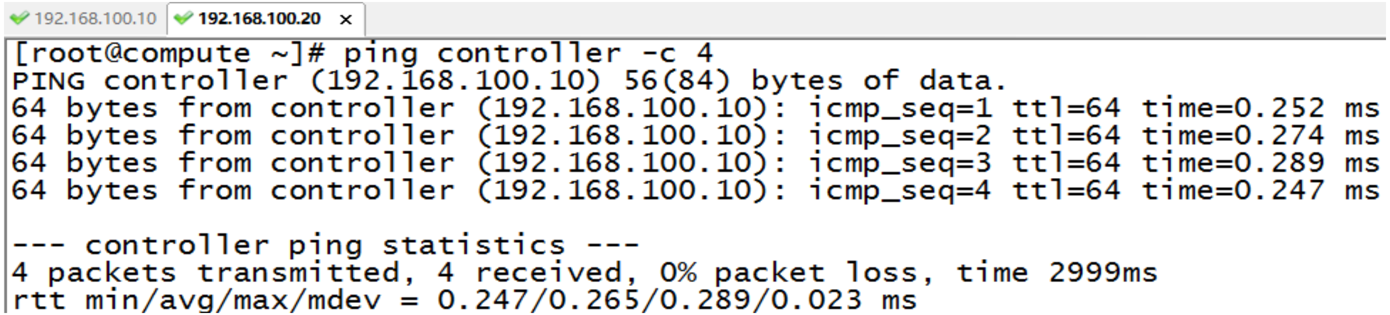
配置主机映射

用ping命令测试连通性。

```
# ping compute -c 4
# ping controller -c 4
```



```
[root@controller ~]# ping compute -c 4
PING compute (192.168.100.20) 56(84) bytes of data.
64 bytes from compute (192.168.100.20): icmp_seq=1 ttl=64 time=0.265 ms
64 bytes from compute (192.168.100.20): icmp_seq=2 ttl=64 time=0.326 ms
64 bytes from compute (192.168.100.20): icmp_seq=3 ttl=64 time=0.495 ms
64 bytes from compute (192.168.100.20): icmp_seq=4 ttl=64 time=0.327 ms

--- compute ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.265/0.353/0.495/0.086 ms
```



```
[root@compute ~]# ping controller -c 4
PING controller (192.168.100.10) 56(84) bytes of data.
64 bytes from controller (192.168.100.10): icmp_seq=1 ttl=64 time=0.252 ms
64 bytes from controller (192.168.100.10): icmp_seq=2 ttl=64 time=0.274 ms
64 bytes from controller (192.168.100.10): icmp_seq=3 ttl=64 time=0.289 ms
64 bytes from controller (192.168.100.10): icmp_seq=4 ttl=64 time=0.247 ms

--- controller ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.247/0.265/0.289/0.023 ms
```

配置Yum源

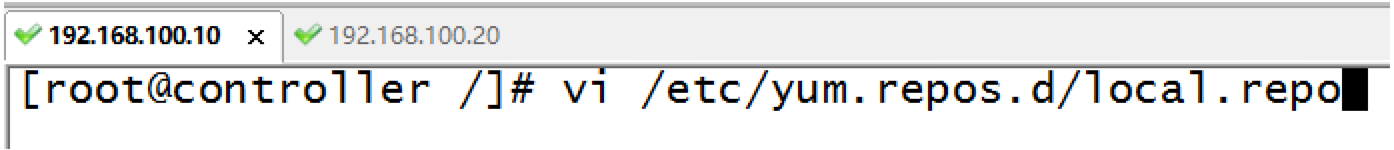
1.控制节点YUM源备份

```
# mv   /etc/yum.repos.d/*    /opt/
```

2.配置repo文件

在控制节点/etc/yum.repo.d/目录下创建local.repo文件

```
# vi /etc/yum.repos.d/local.repo
```



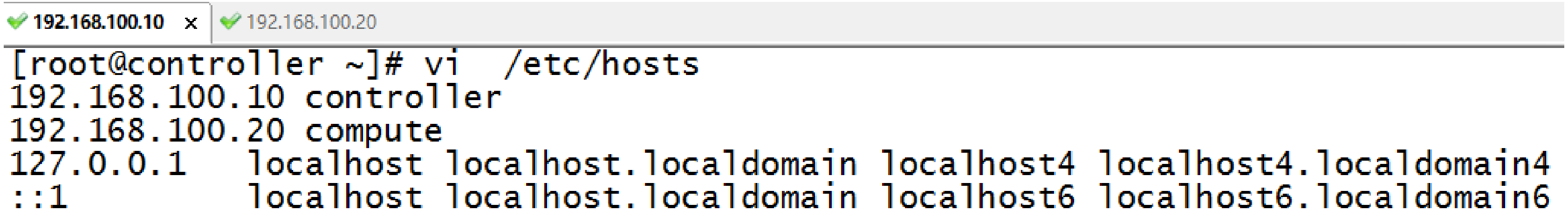
```
[root@controller /]# vi /etc/yum.repos.d/local.repo
```

```
[centos]
name=centos
baseurl=file:///mnt/centos/ （注：具体 yum 源根据真实环境配置）
gpgcheck=0
enabled=1
[mitaka]
name=mitaka
baseurl=file:///mnt/mitaka/Openstack-Mitaka/ （注：具体 yum 源根据真实环境配置）
gpgcheck=0
enabled=1
```

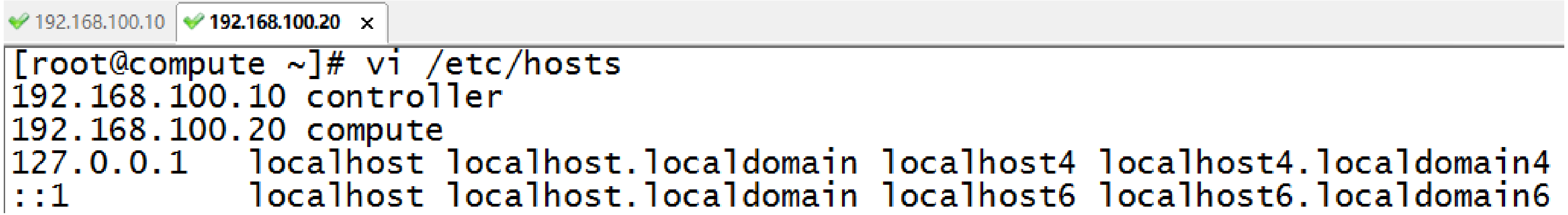
配置主机映射

修改控制节点和计算节点/etc/hosts文件添加以下内容。

```
# vi /etc/hosts
192.168.100.10 controller
192.168.100.20 compute
```



```
[root@controller ~]# vi  /etc/hosts
192.168.100.10 controller
192.168.100.20 compute
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```



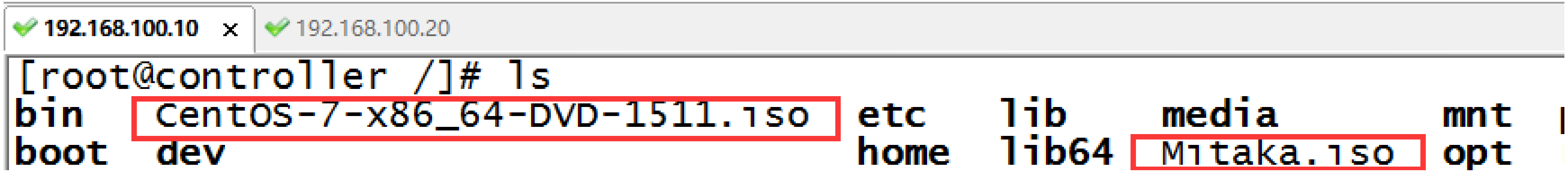
```
[root@compute ~]# vi /etc/hosts
192.168.100.10 controller
192.168.100.20 compute
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

配置Yum源

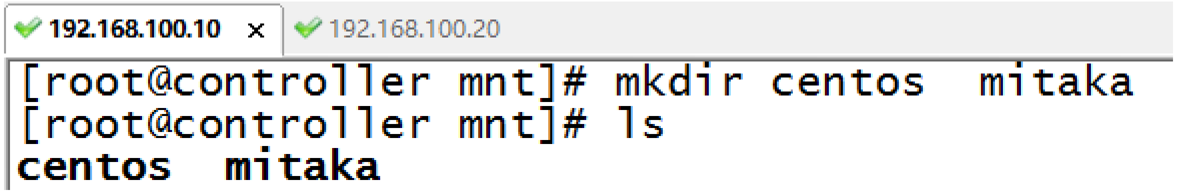
关于软件包和镜像文件的上传请参考【YUM源配置文档】

先使用 SecureFX 工具将所需软件包镜像文件上传至控制节点 “/”目录下，然后进行挂载。
若/mnt/目录下无centos和mitaka文件夹可以通过mkdir命令创建。

```
# mount -o loop /CentOS-7-x86_64-DVD-1511.iso /mnt/centos
# mount -o loop /Mitaka.iso /mnt/mitaka
```



```
[root@controller /]# ls
bin    CentOS-7-x86_64-DVD-1511.iso  etc   lib    media       mnt
boot   dev                           home  lib64  Mitaka.iso  opt
```



```
[root@controller mnt]# mkdir centos  mitaka
[root@controller mnt]# ls
centos  mitaka
```

配置Yum源

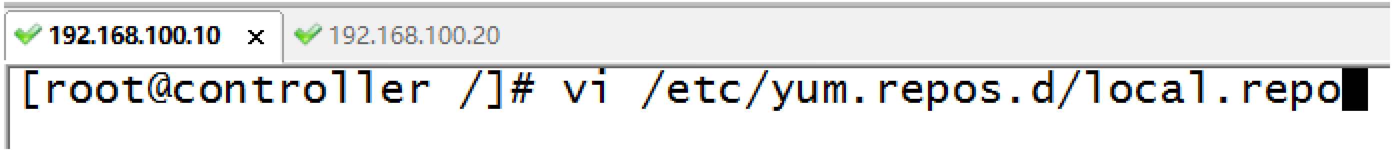
1.计算节点YUM源备份

```
# mv   /etc/yum.repos.d/*    /opt/
```

2.配置repo文件

在计算节点/etc/yum.repo.d/目录下创建local.repo文件

```
# vi /etc/yum.repos.d/local.repo
```



```
[root@controller /]# vi /etc/yum.repos.d/local.repo
```

```
[centos]
name=centos
baseurl=ftp://192.168.100.10/centos/ （注：具体 yum 源根据真实环境配置）
gpgcheck=0
enabled=1
[mitaka]
name=mitaka
 baseurl=ftp://192.168.100.10/mitaka/Openstack-Mitaka/ （注：具体 yum 源根据真实环境配置）
gpcheck=0
enabled=1
```

安装NTP服务

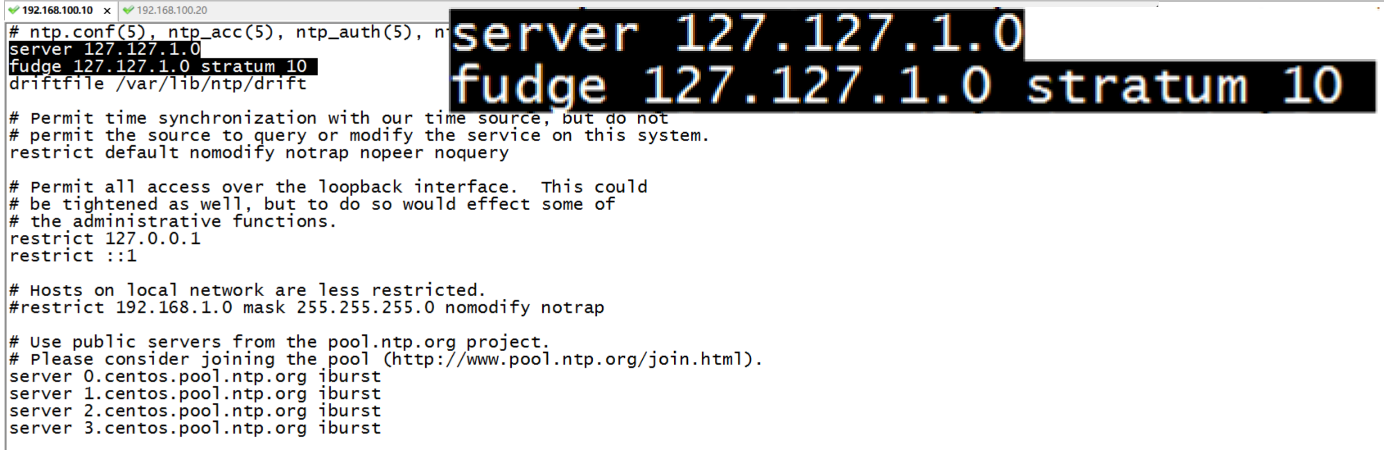
NTP 服务是一种时钟同步服务，在分布式集群中，为了便于同一生命周内不同节点服务的管理，需要各个节点的服务严格的时钟同步，在以下配置中，以控制节点作为时钟服务器，其他节点以控制节点的时钟作为时钟标准调整自己的时钟。

1.在控制节点和计算节点安装NTP服务软件包

```
## yum   install  ntp  -y
```

2.配置控制节点/etc/ntp.conf文件，添加以下内容

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```



```
# ntp.conf(5), ntp_acc(5), ntp_auth(5), n
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict ::1

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
```

server 127.127.1.0
fudge 127.127.1.0 stratum 10

安装vsftpd服务

1.在控制节点安装vsftpd服务

```
# yum  install  –y  vsftpd
```

2.配置vsftpd服务

修改/etc/vsftpd/vsftpd.conf文件

```
# vi /etc/vsftpd/vsftpd.conf
  anon_root=/mnt
```

3.启动vsftpd服务

```
# systemctl   start   vsftpd.service
```

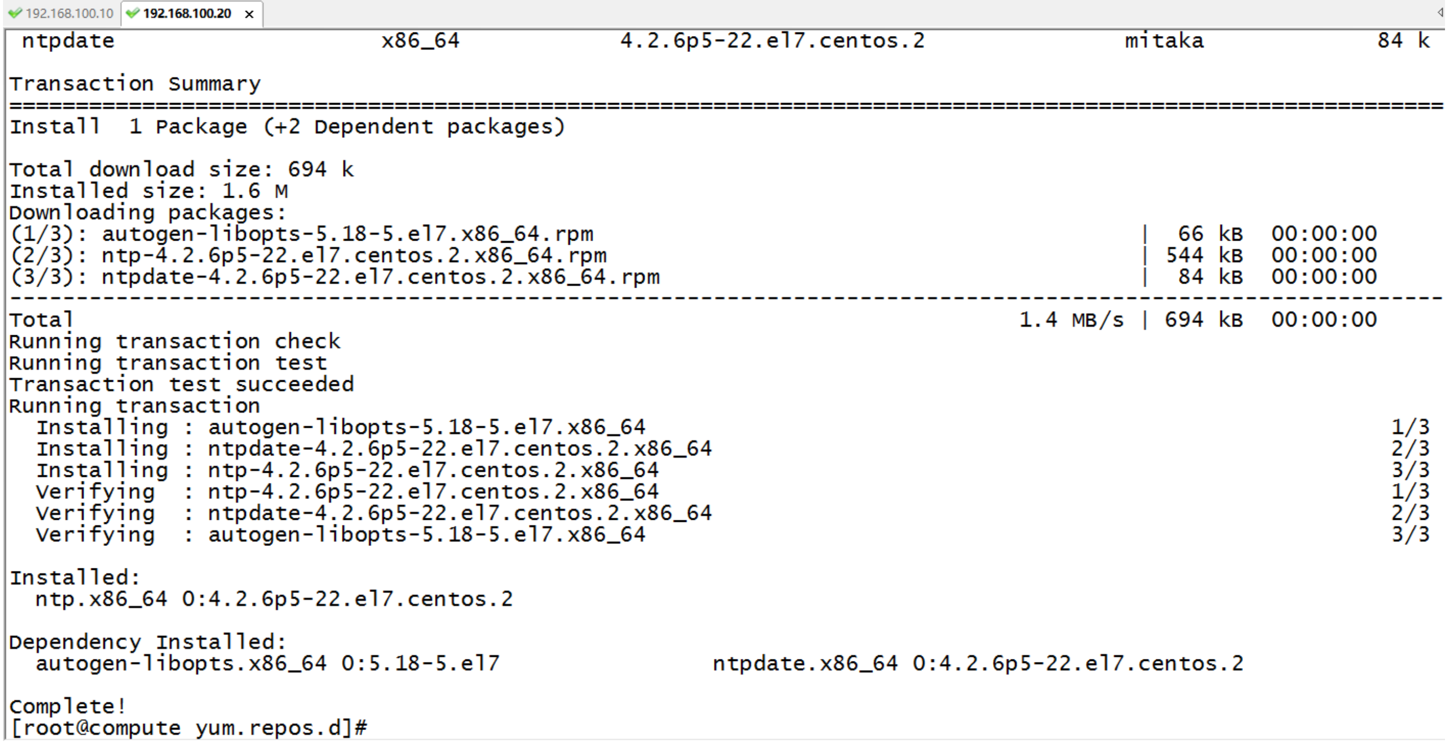
4.清理缓存

```
# yum clean all
```

安装NTP服务

1.计算节点安装NTP服务。

```
# yum  install  ntp  -y
```

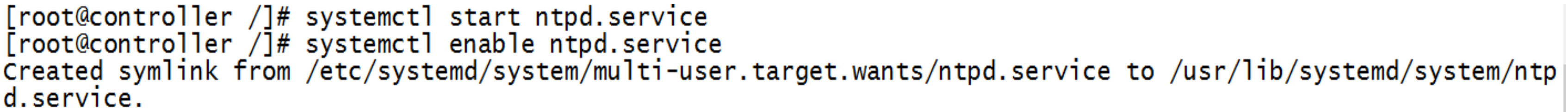


```
 ntpdate                   x86_64           4.2.6p5-22.el7.centos.2                    mitaka             84 k

Transaction Summary
============================================================================================================
Install  1 Package (+2 Dependent packages)

Total download size: 694 k
Installed size: 1.6 M
Downloading packages:
(1/3): autogen-libopts-5.18-5.el7.x86_64.rpm                                         |  66 kB  00:00:00
(2/3): ntp-4.2.6p5-22.el7.centos.2.x86_64.rpm                                        | 544 kB  00:00:00
(3/3): ntpdate-4.2.6p5-22.el7.centos.2.x86_64.rpm                                    |  84 kB  00:00:00
------------------------------------------------------------------------------------------------------------
Total                                                                       1.4 MB/s | 694 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : autogen-libopts-5.18-5.el7.x86_64                                                       1/3
  Installing : ntpdate-4.2.6p5-22.el7.centos.2.x86_64                                                  2/3
  Installing : ntp-4.2.6p5-22.el7.centos.2.x86_64                                                      3/3
  Verifying  : ntp-4.2.6p5-22.el7.centos.2.x86_64                                                      1/3
  Verifying  : ntpdate-4.2.6p5-22.el7.centos.2.x86_64                                                  2/3
  Verifying  : autogen-libopts-5.18-5.el7.x86_64                                                       3/3

Installed:
  ntp.x86_64 0:4.2.6p5-22.el7.centos.2

Dependency Installed:
  autogen-libopts.x86_64 0:5.18-5.el7                ntpdate.x86_64 0:4.2.6p5-22.el7.centos.2

Complete!
[root@compute yum.repos.d]#
```

安装NTP服务

1.设置NTP服务开机自启动和启动NTP服务。

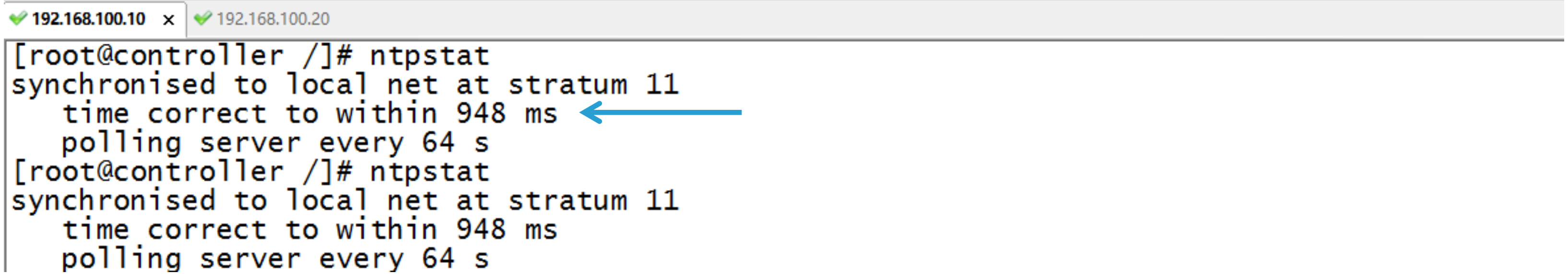
```
# systemctl start ntpd.service
# systemctl enable ntpd.service
```



```
[root@controller /]# systemctl start ntpd.service
[root@controller /]# systemctl enable ntpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to /usr/lib/systemd/system/ntp
d.service.
```

2.查看服务状态。(注:蓝色箭头指向的位置数字可能是不同的，对后续操作无影响。)

```
# ntpstat
```

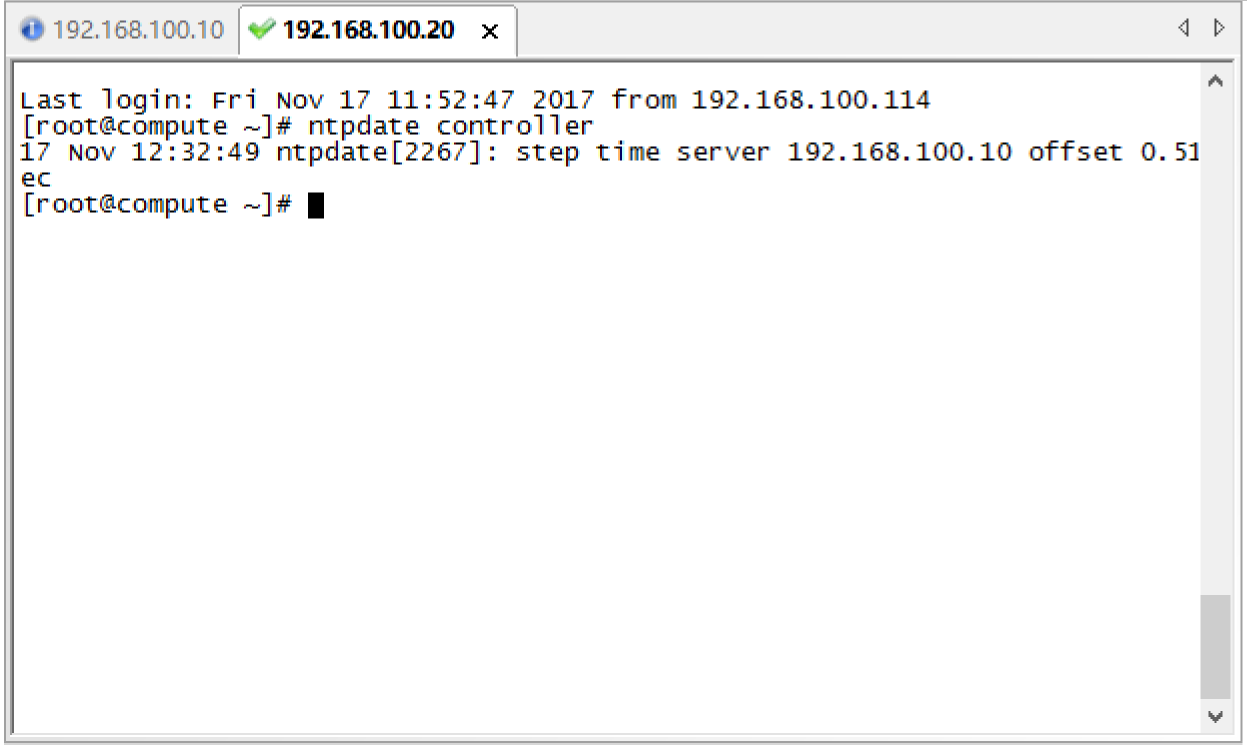


```
[root@controller /]# ntpstat
synchronised to local net at stratum 11
   time correct to within 948 ms
   polling server every 64 s
[root@controller /]# ntpstat
synchronised to local net at stratum 11
   time correct to within 948 ms
   polling server every 64 s
```

安装NTP服务

1.配置计算节点

```
# ntpdate controlle
```



```
Last login: Fri Nov 17 11:52:47 2017 from 192.168.100.114
[root@compute ~]# ntpdate controller
17 Nov 12:32:49 ntpdate[2267]: step time server 192.168.100.10 offset 0.51
ec
[root@compute ~]# 
```

注： NTP 服务需要在每个节点上安装。并与控制节点同步。

安装OpenStack包

1.在计算节点和控制节点安装OpenStack包

```
# yum   install   python-openstackclient  -y
# yum   install   openstack-selinux   -y
# yum   upgrade --skip-broken   -y
```

安装并配置SQL数据库

SQL数据库作为基础或扩展服务产生的数据存放的地方，数据库运行在控制节点上。OpenStack支持的数据库有MySQL、MariaDB以及PostgreSQL等其他数据库。本次安装采用MariaDB数据库。
小助手：本次安装会大量地编辑配置文件，但是很多配置文件有许多以#开头的注释文件 或者空格的命令，不容易找到自己需要编辑的模块。可采用下面的命令，删除#和空格的命令。

```
# cat  file  | grep  -v  ^#  | grep  -v  ^$  >  newfile
```

安装并配置SQL数据库

以下步骤均在控制节点操作

1.备份/etc/my.cnf文件

```
# cp   /etc/my.cnf    / etc/my.cnf.bak
```

2.删除#和空格的命令

```
# cat  /etc/my.cnf.bak  |  grep   -v  ^# | grep -v  ^$ >  /etc/my.cnf
```

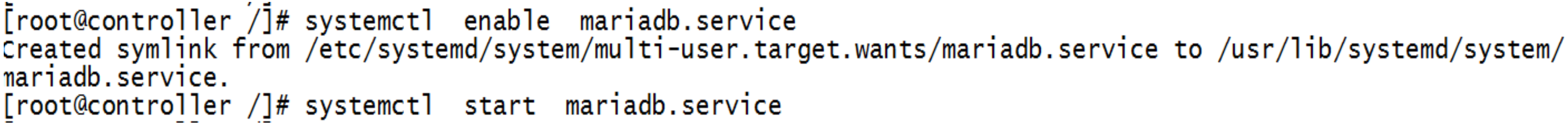
3.安装数据库MariaDB

```
# yum   install   mariadb   mariadb-server   python2-PyMySQL   -y
```

安装并配置SQL数据库

5.启动数据库并设置开机自启动

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

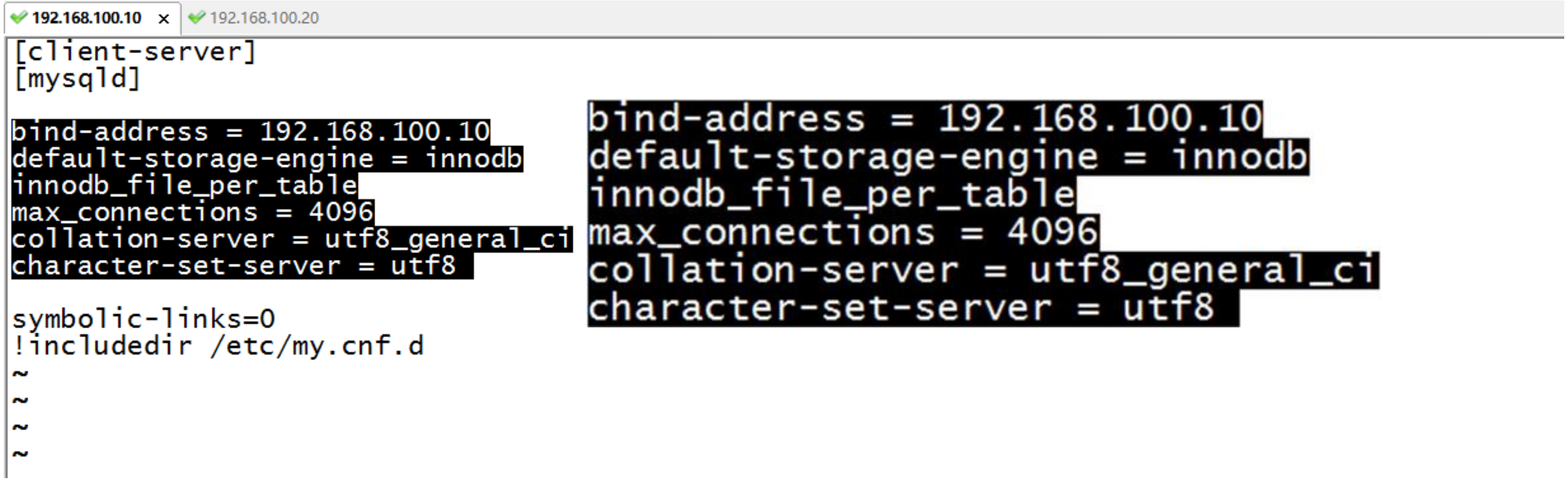


```
[root@controller /]# systemctl  enable  mariadb.service
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/
mariadb.service.
[root@controller /]# systemctl  start  mariadb.service
```

安装并配置SQL数据库

4.编辑/etc/my.cnf文件并在在[mysqld]部分添加以下内容

```
bind-address = 192.168.100.10
default-storage-engine = innodb
innodb_file_per_table
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```



```
[client-server]
[mysqld]

bind-address = 192.168.100.10
default-storage-engine = innodb
innodb_file_per_table
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8

symbolic-links=0
!includedir /etc/my.cnf.d
~
~
~
~
```

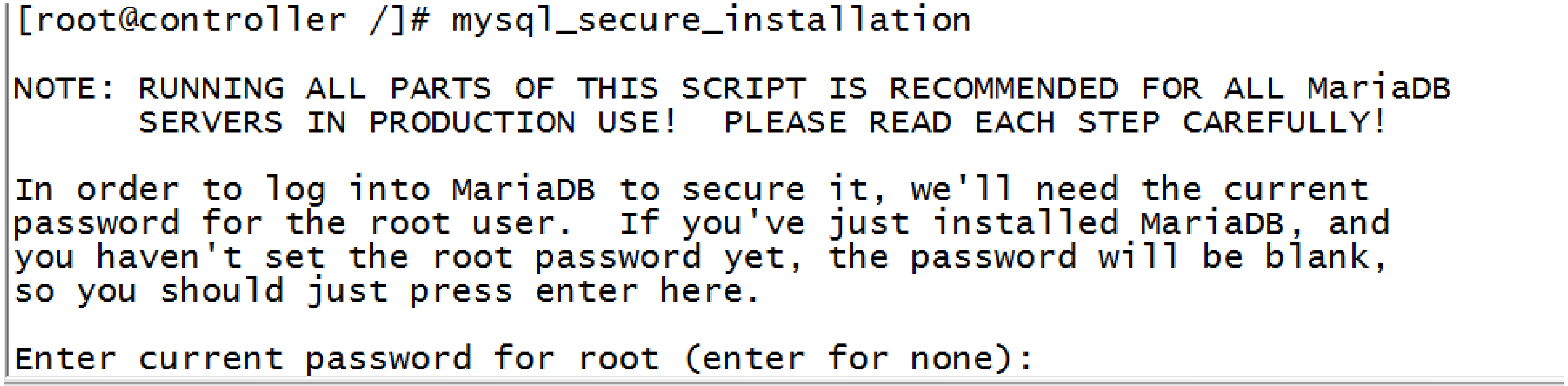
bind-address = 192.168.100.10
default-storage-engine = innodb
innodb_file_per_table
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8

安装并配置SQL数据库

6.运行 “mysql_secure_installation” 脚本。初始化数据库并 设置密码

```
# mysql_secure_installation
```

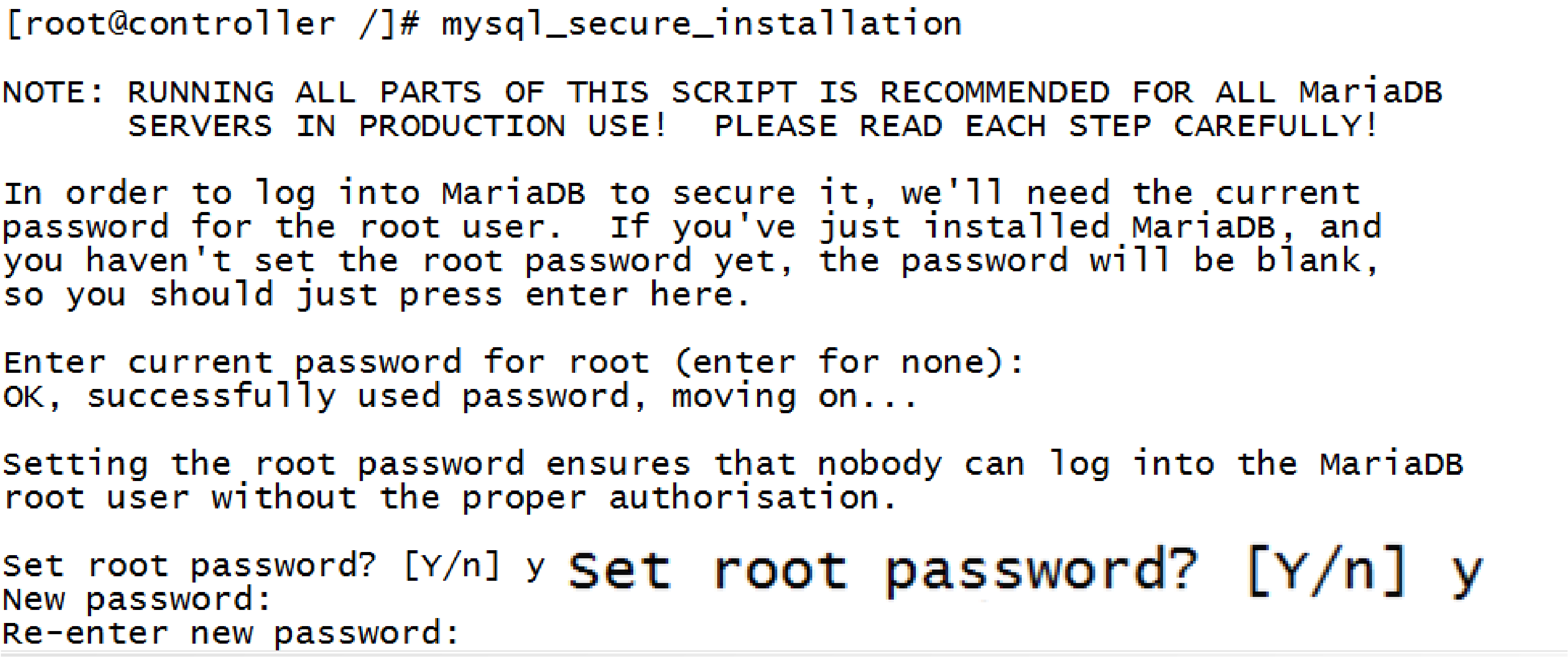
1) 第一次输入为回车，因为没有密码



```
[root@controller /]# mysql_secure_installation

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):
```

安装并配置SQL数据库

2)第二次输入为 y，然后设置数据库密码000000

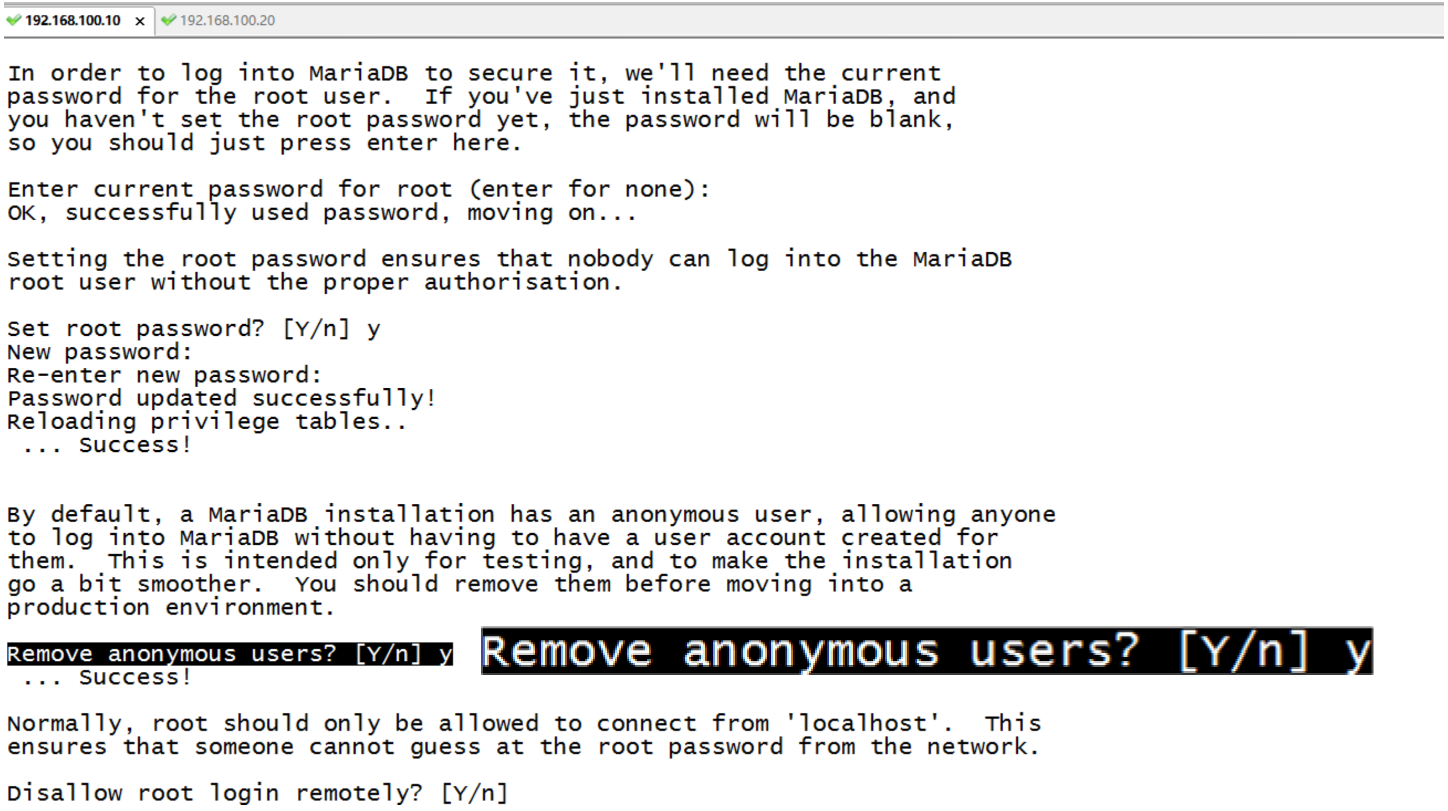


```
[root@controller /]# mysql_secure_installation

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.

Set root password? [Y/n] y
New password:
Re-enter new password:
```

Set root password? [Y/n] y

安装并配置SQL数据库

3)第三次输入为 y

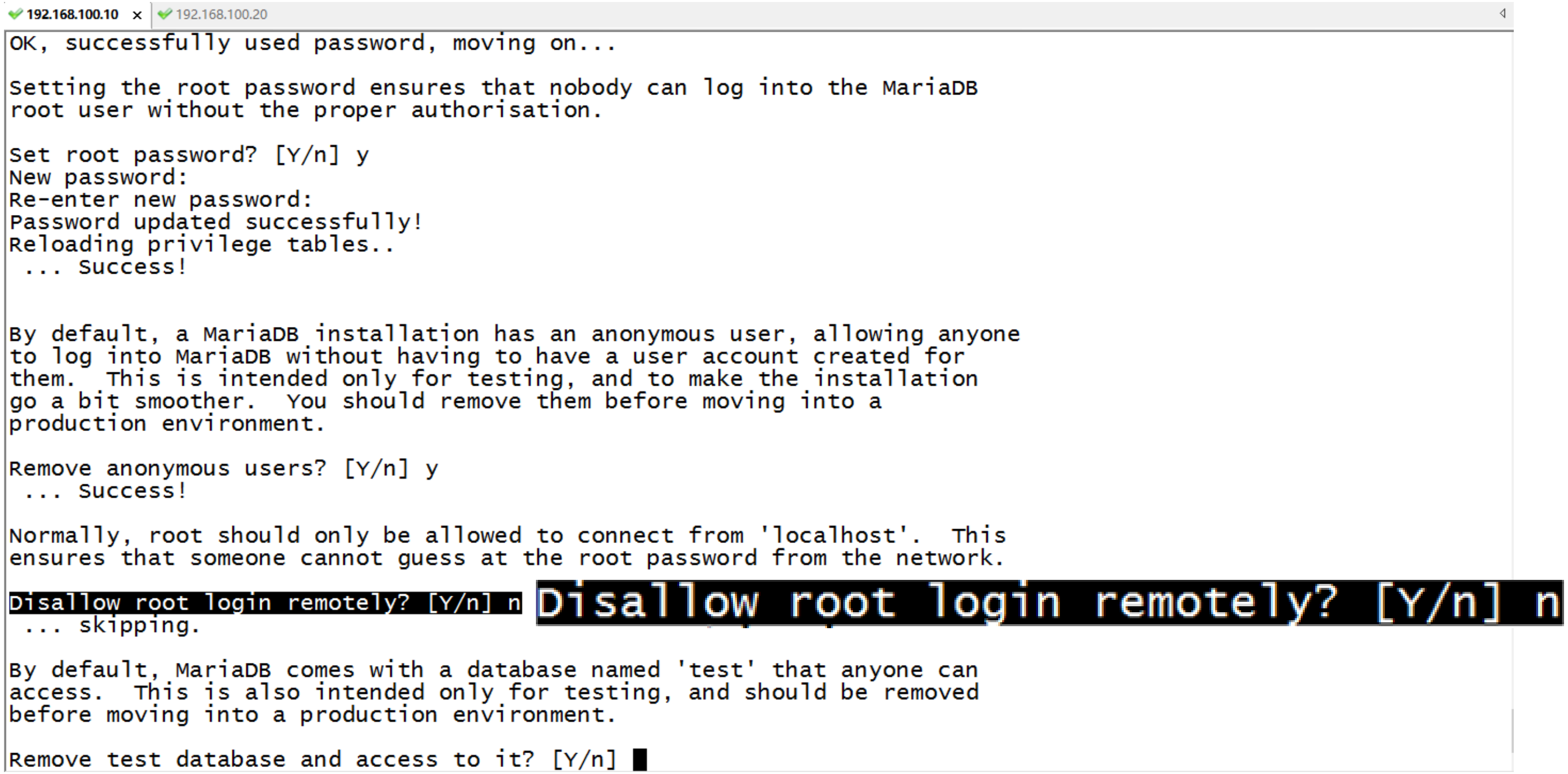


```
In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

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.

Set 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]
```

Remove anonymous users? [Y/n] y

安装并配置SQL数据库

4)第四次输入为 n

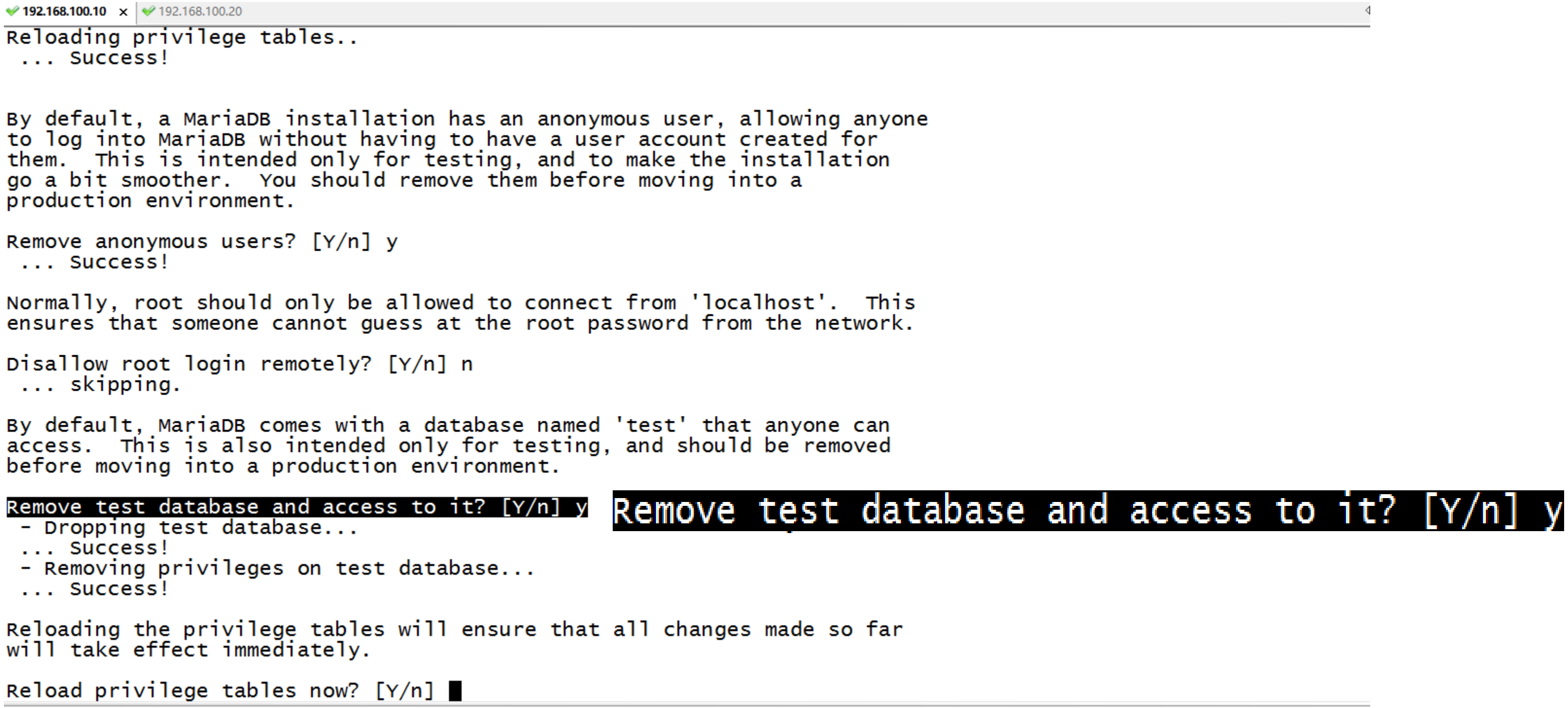


```
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set 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] n
 ... skipping.

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] █
```

Disallow root login remotely? [Y/n] n

安装并配置SQL数据库

5)第五次输入为 y

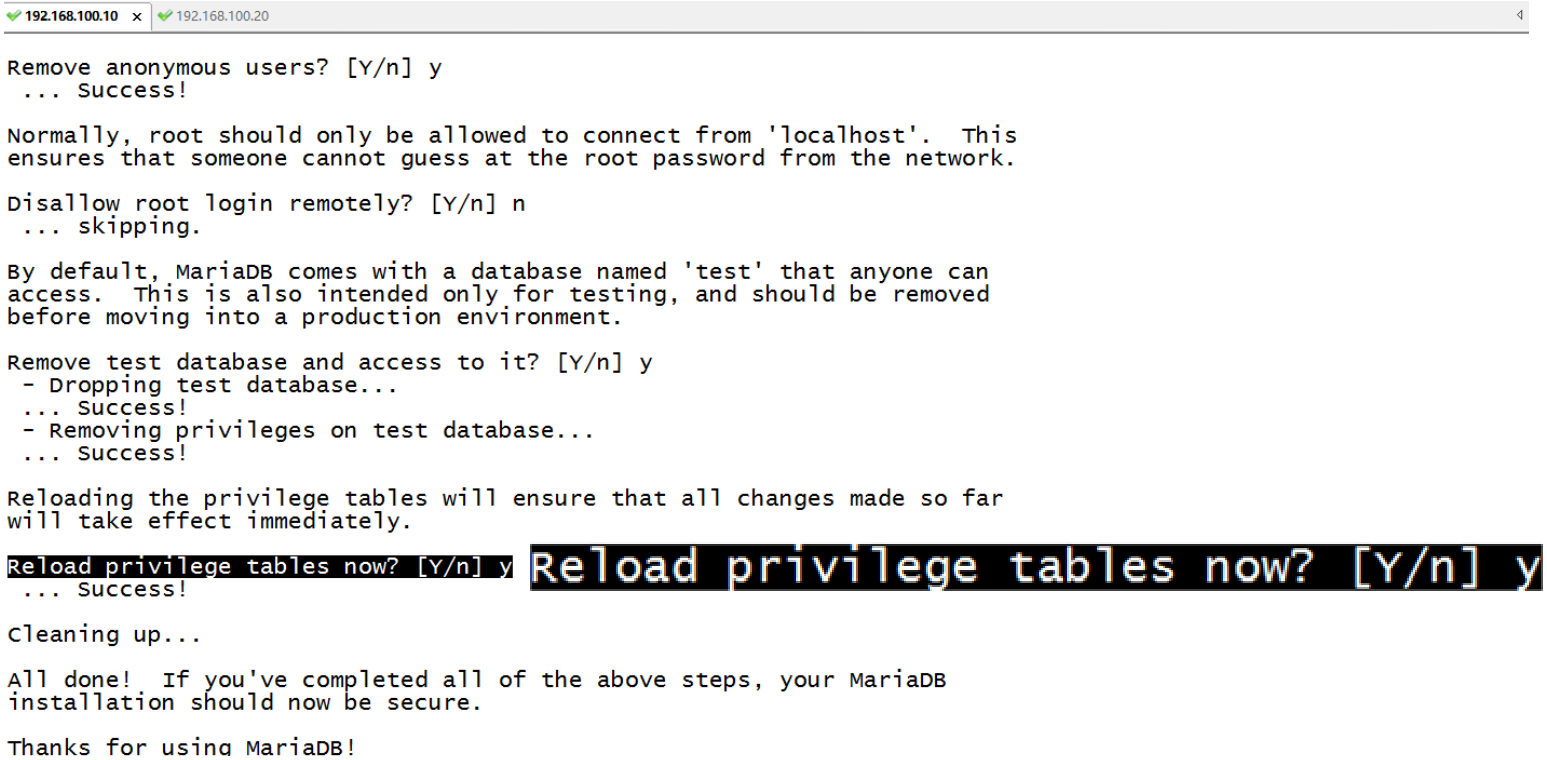


```
Reloading privilege tables..
 ... Success!


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] n
 ... skipping.

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...
 ... Success!
 - 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] 
```

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

安装并配置SQL数据库

6)第六次输入为 y



```
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] n
 ... skipping.

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...
 ... Success!
 - 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!
```

Reload privilege tables now? [Y/n] y

安装并配置消息服务器

OpenStack 使用 message queue 协调操作和各服务的状态信息。消息队列服务本次部署在控 制节点上。OpenStack 支持的几种消息队列服务包括 RabbitMQ、Qpid 和 ZeroMQ。我们采用安 装 RabbitMQ 消息队列服务。

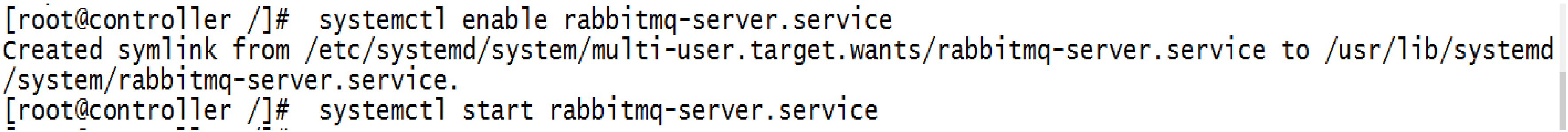
以下操作都是在控制节点完成：

1.安装消息队列服务

```
#  yum install rabbitmq-server  -y
```

启动服务并设置开机自启动

```
#  systemctl   enable   rabbitmq-server.service
#  systemctl   start   rabbitmq-server.service
```

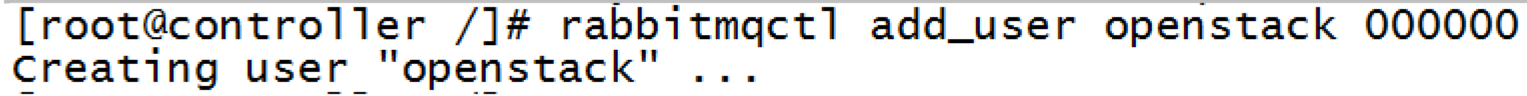


```
[root@controller /]#  systemctl enable rabbitmq-server.service
Created symlink from /etc/systemd/system/multi-user.target.wants/rabbitmq-server.service to /usr/lib/systemd
/system/rabbitmq-server.service.
[root@controller /]#  systemctl start rabbitmq-server.service
```

安装并配置消息服务器

2.创建rabbitmq用户并设置权限

```
# rabbitmqctl add_user openstack 000000
```



```
[root@controller /]# rabbitmqctl add_user openstack 000000
Creating user "openstack" ...
```

3.给OpenStack用户授予读/写权限

```
#  rabbitmqctl set_permissions openstack ".*" ".*" ".*"
```

安装Memcached

认证服务认证缓存使用 Memcached 缓存令牌，缓存服务 memecached 运行在控制节点上。 在生产部署中，建议联合启用防火墙、认证和加密保证它的安全。

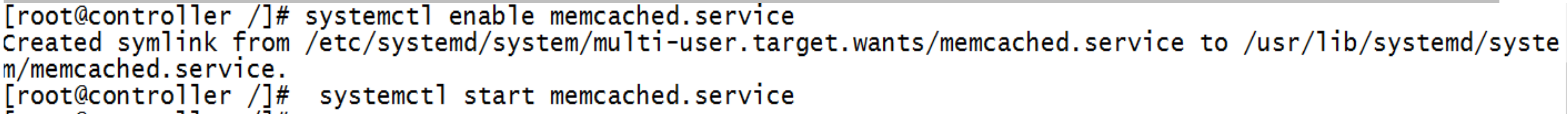
以下操作都是在控制节点完成：

安装Memcached服务

```
# yum install memcached python-memcached -y
```

启动服务并设置开机自启动

```
# systemctl enable memcached.service
#  systemctl start memcached.service
```



```
[root@controller /]# systemctl enable memcached.service
Created symlink from /etc/systemd/system/multi-user.target.wants/memcached.service to /usr/lib/systemd/syste
m/memcached.service.
[root@controller /]#  systemctl start memcached.service
```

谢谢观看