第四次实验报告

实验时间：	2020 年 11 月 23 日	实验人：	陈泰良

实验名称：第四次实验

1. 实验任务和目标：

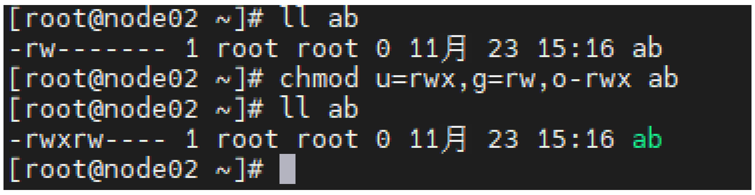
- 熟悉基本命令

实验环境描述：Windows 环境, Vmware 虚拟机

实验结果：

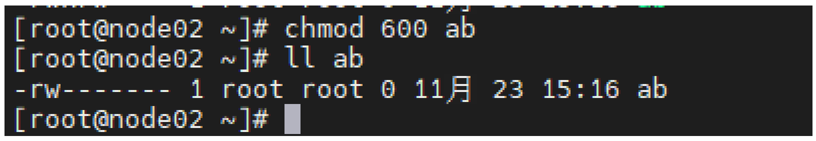
第十章

1、使用文字设定法对/root/ab 文件设置权限，所有者为读取、写入和执行权限，同组用户为读取和写入权限，而其他用户没有任何权限。



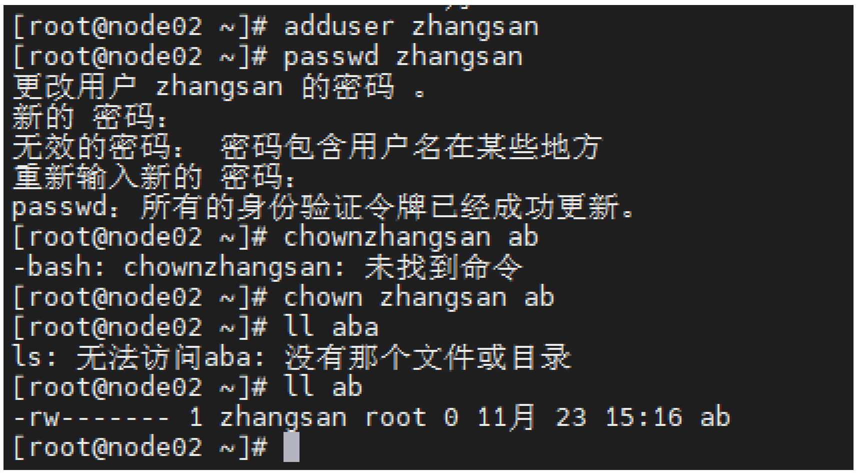
```
[root@node02 ~]# ll ab
-rw------- 1 root root 0 11月 23 15:16 ab
[root@node02 ~]# chmod u=rwx,g=rw,o-rwx ab
[root@node02 ~]# ll ab
-rwxrw---- 1 root root 0 11月 23 15:16 ab
[root@node02 ~]# 
```

2、使用数字设定法设置/root/ab 文件的权限，所有者只拥有读取和写入权限。



```
[root@node02 ~]# chmod 600 ab
[root@node02 ~]# ll ab
-rw------- 1 root root 0 11月 23 15:16 ab
[root@node02 ~]# 
```

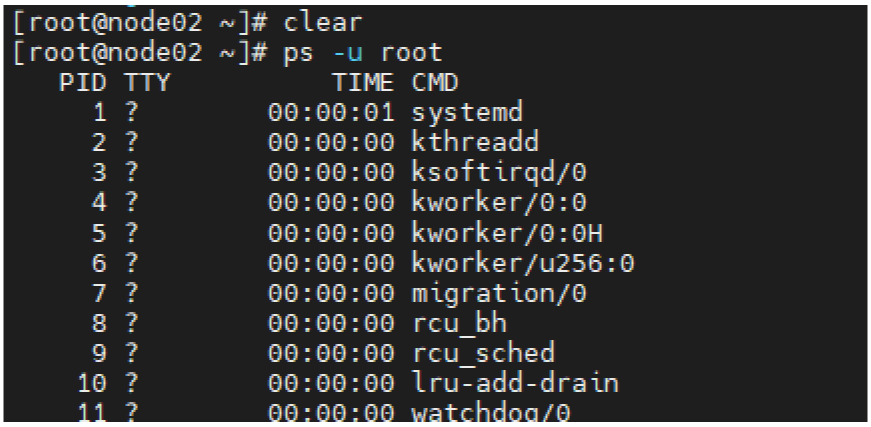
3、将/root/ab 文件的所有者更改为用户 zhangsan。



```
[root@node02 ~]# adduser zhangsan
[root@node02 ~]# passwd zhangsan
更改用户 zhangsan 的密码 。
新的 密码：
无效的密码： 密码包含用户名在某些地方
重新输入新的 密码：
passwd：所有的身份验证令牌已经成功更新。
[root@node02 ~]# chownzhangsan ab
-bash: chownzhangsan: 未找到命令
[root@node02 ~]# chown zhangsan ab
[root@node02 ~]# ll aba
ls: 无法访问aba: 没有那个文件或目录
[root@node02 ~]# ll ab
-rw------- 1 zhangsan root 0 11月 23 15:16 ab
[root@node02 ~]# 
```

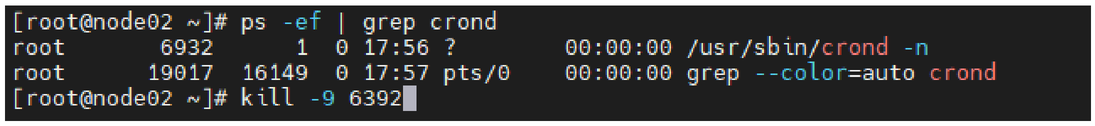
第十一章

1、使用 ps 命令显示 root 用户的进程。



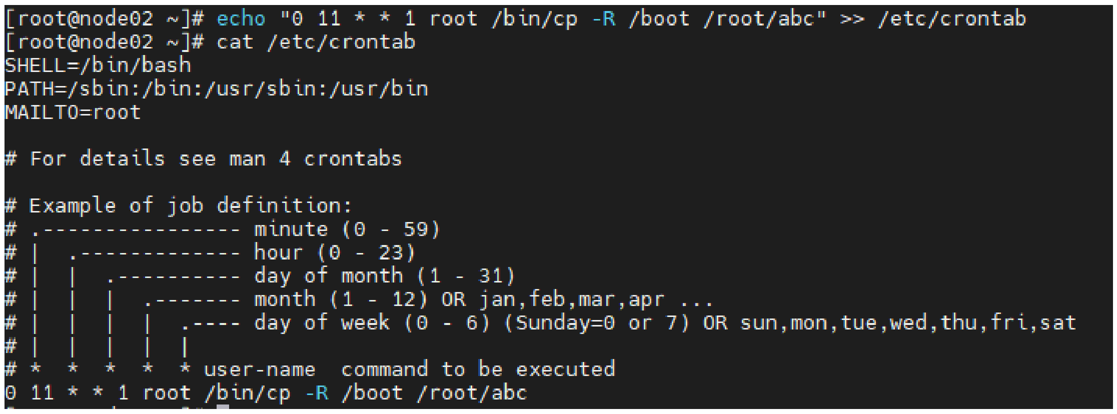
```
[root@node02 ~]# clear
[root@node02 ~]# ps -u root
   PID TTY          TIME CMD
     1 ?        00:00:01 systemd
     2 ?        00:00:00 kthreadd
     3 ?        00:00:00 ksoftirqd/0
     4 ?        00:00:00 kworker/0:0
     5 ?        00:00:00 kworker/0:0H
     6 ?        00:00:00 kworker/u256:0
     7 ?        00:00:00 migration/0
     8 ?        00:00:00 rcu_bh
     9 ?        00:00:00 rcu_sched
    10 ?        00:00:00 lru-add-drain
    11 ?        00:00:00 watchdog/0
```

2、强制杀死 crond 进程。



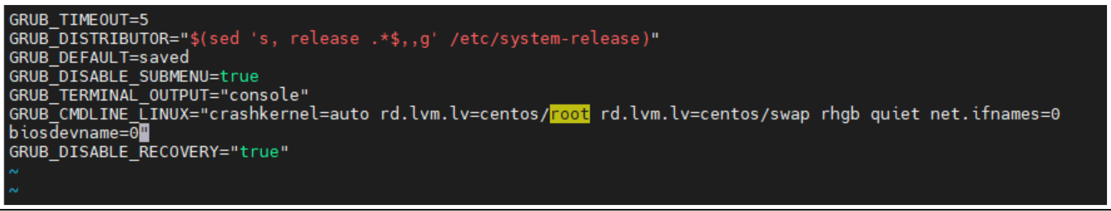
```
[root@node02 ~]# ps -ef | grep crond
root       6932      1  0 17:56 ?        00:00:00 /usr/sbin/crond -n
root      19017  16149  0 17:57 pts/0    00:00:00 grep --color=auto crond
[root@node02 ~]# kill -9 6392
```

3、修改/etc/crontab 文件实现自动化，使得每星期一的 11:00 将/boot 目录及其子目录和文件复制到/root/abc 目录下。

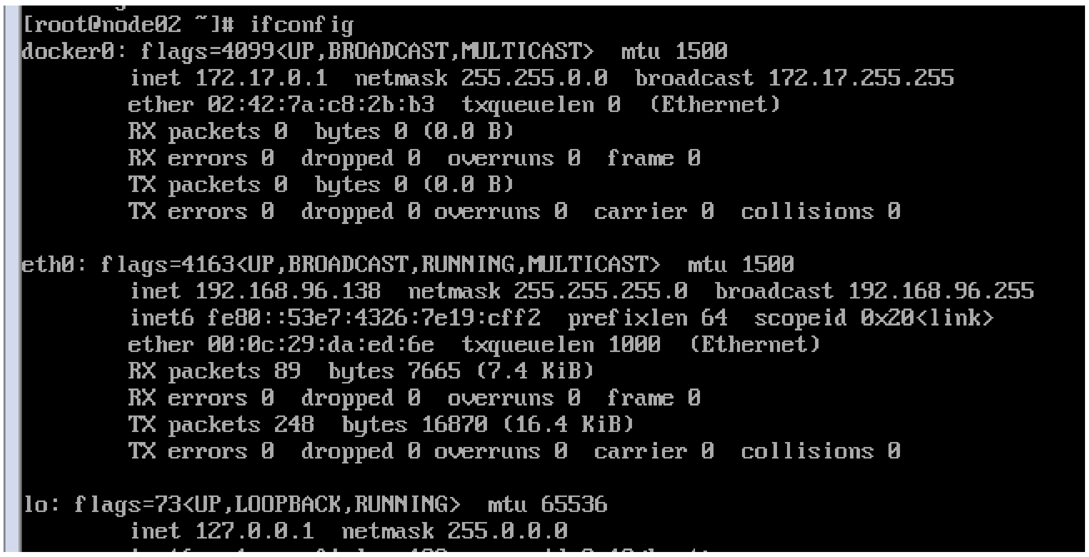


```
[root@node02 ~]# echo "0 11 * * 1 root /bin/cp -R /boot /root/abc" >> /etc/crontab
[root@node02 ~]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
0 11 * * 1 root /bin/cp -R /boot /root/abc
```

4、将网卡名称 eno16777736 更改为 eth0。（根据自己安装系统的实际情况选做）

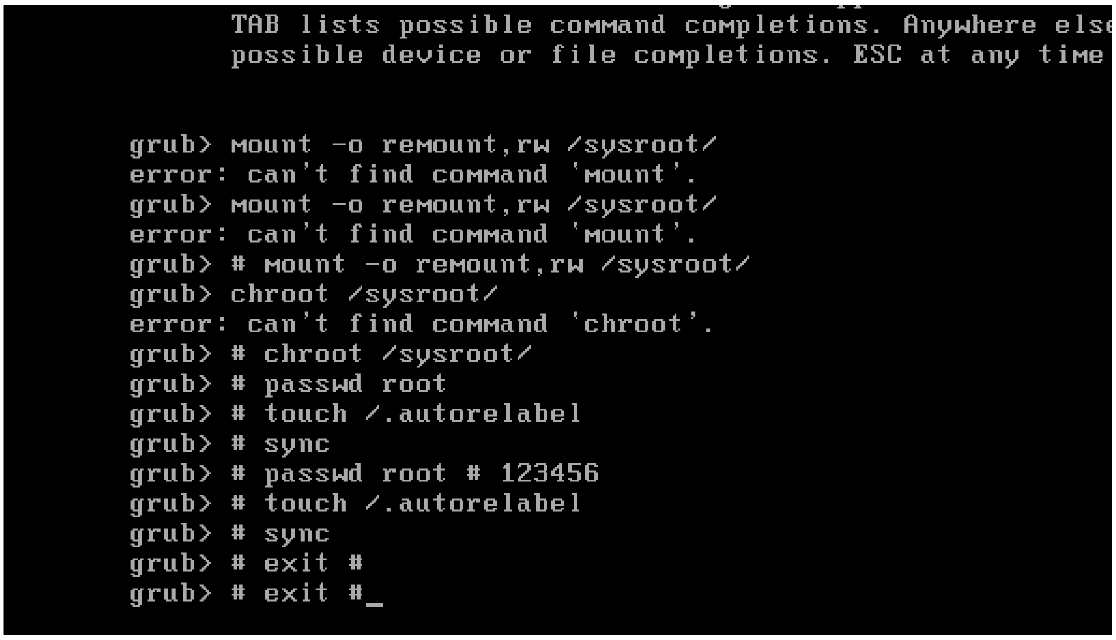


```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet net.ifnames=0
biosdevname=0"
GRUB_DISABLE_RECOVERY="true"
~
~
```



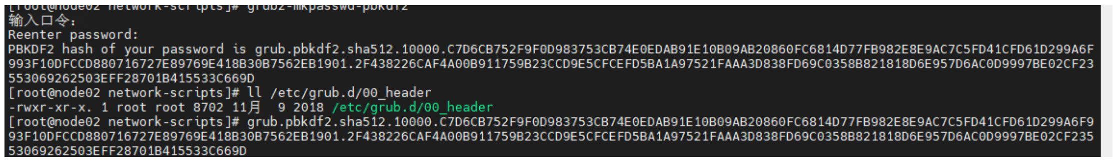
```
[root@node02 ~]# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:7a:c8:2b:b3  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.96.138  netmask 255.255.255.0  broadcast 192.168.96.255
        inet6 fe80::53e7:4326:7e19:cff2  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:da:ed:6e  txqueuelen 1000  (Ethernet)
        RX packets 89  bytes 7665 (7.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 248  bytes 16870 (16.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
```

5、使用 GRUB2 破解 root 用户的密码。



```
           TAB lists possible command completions. Anywhere else
           possible device or file completions. ESC at any time


      grub> mount -o remount,rw /sysroot/
      error: can't find command 'mount'.
      grub> mount -o remount,rw /sysroot/
      error: can't find command 'mount'.
      grub> # mount -o remount,rw /sysroot/
      grub> chroot /sysroot/
      error: can't find command 'chroot'.
      grub> # chroot /sysroot/
      grub> # passwd root
      grub> # touch /.autorelabel
      grub> # sync
      grub> # passwd root # 123456
      grub> # touch /.autorelabel
      grub> # sync
      grub> # exit #
      grub> # exit #_
```

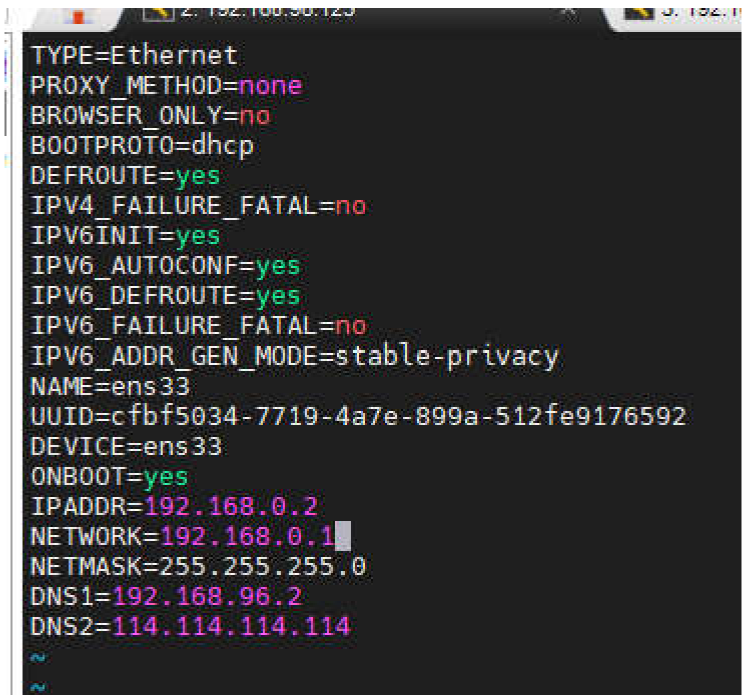
6、设置 GRUB2 PBKDF2 加密口令。



```
[root@node02 network-scripts]# grub2-mkpasswd-pbkdf2
输入口令:
Reenter password:
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.C7D6CB752F9F0D983753CB74E0EDAB91E10B09AB20860FC6814D77FB982E8E9AC7C5FD41CFD61D299A6F
993F10DFCCD880716727E89769E418B30B7562EB1901.2F438226CAF4A00B911759B23CCD9E5CFCEFD5BA1A97521FAAA3D838FD69C0358B821818D6E957D6AC0D9997BE02CF23
553069262503EFF28701B415533C669D
[root@node02 network-scripts]# ll /etc/grub.d/00_header
-rwxr-xr-x. 1 root root 8702 11月  9 2018 /etc/grub.d/00_header
[root@node02 network-scripts]# grub.pbkdf2.sha512.10000.C7D6CB752F9F0D983753CB74E0EDAB91E10B09AB20860FC6814D77FB982E8E9AC7C5FD41CFD61D299A6F9
93F10DFCCD880716727E89769E418B30B7562EB1901.2F438226CAF4A00B911759B23CCD9E5CFCEFD5BA1A97521FAAA3D838FD69C0358B821818D6E957D6AC0D9997BE02CF235
53069262503EFF28701B415533C669D
```

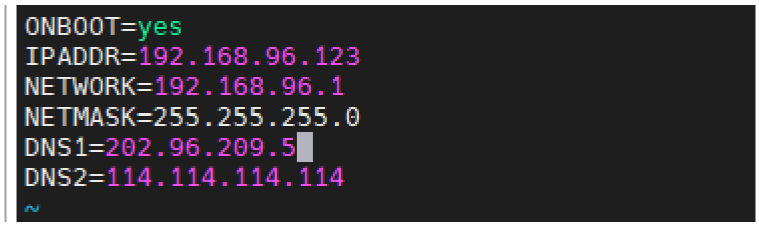
第十二章（如果没有 eno16777736，可使用本机第一块网卡操作，完成 1-3 题实验后恢复原配置）

1、通过修改/etc/sysconfig/network-scripts/ifcfg-eno16777736 文件，设置计算机 IP 地址为 192.168.0.2，子网掩码为 255.255.255.0，网关 IP 地址为 192.168.0.1。



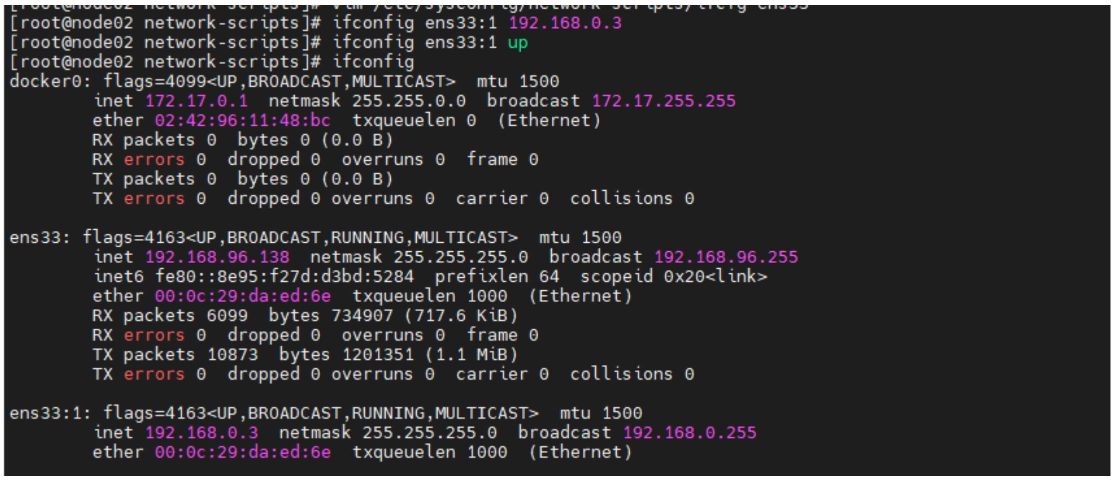
```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=cfbf5034-7719-4a7e-899a-512fe9176592
DEVICE=ens33
ONBOOT=yes
IPADDR=192.168.0.2
NETWORK=192.168.0.1
NETMASK=255.255.255.0
DNS1=192.168.96.2
DNS2=114.114.114.114
~
~
```

2、设置计算机解析域名时所指向的主 DNS 服务器 IP 地址为 202.96.209.5。



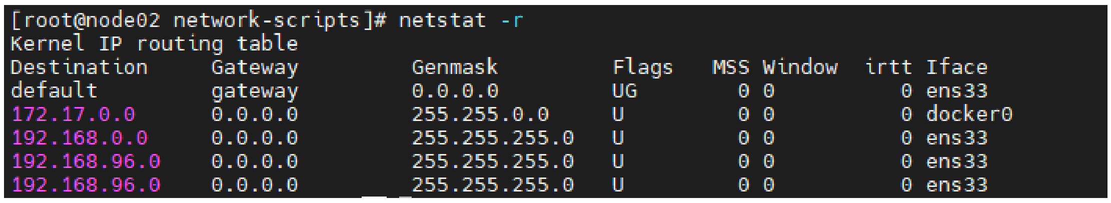
```
ONBOOT=yes
IPADDR=192.168.96.123
NETWORK=192.168.96.1
NETMASK=255.255.255.0
DNS1=202.96.209.5
DNS2=114.114.114.114
~
```

3、配置网卡 eno16777736 别名设备 eno16777736:1 的 IP 地址为 192.168.0.3，并且激活网卡 eno16777736:1 设备。



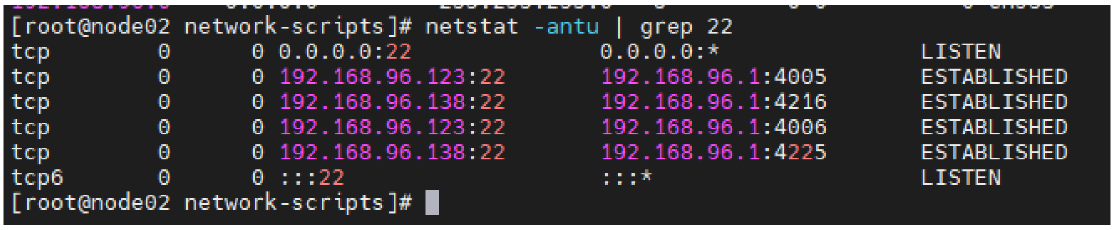
```
[root@node02 network-scripts]# ifconfig ens33:1 192.168.0.3
[root@node02 network-scripts]# ifconfig ens33:1 up
[root@node02 network-scripts]# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:96:11:48:bc  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.96.138  netmask 255.255.255.0  broadcast 192.168.96.255
        inet6 fe80::8e95:f27d:d3bd:5284  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:da:ed:6e  txqueuelen 1000  (Ethernet)
        RX packets 6099  bytes 734907 (717.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10873  bytes 1201351 (1.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.3  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 00:0c:29:da:ed:6e  txqueuelen 1000  (Ethernet)
```

4、使用命令显示当前计算机系统的内核路由表信息。



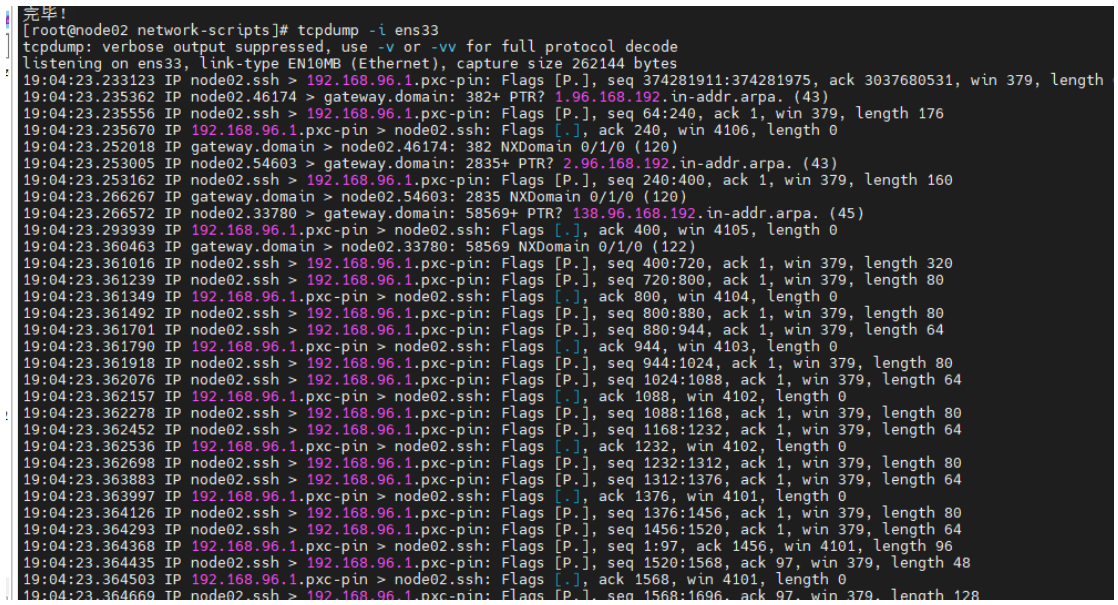
```
[root@node02 network-scripts]# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         gateway         0.0.0.0         UG        0 0          0 ens33
172.17.0.0      0.0.0.0         255.255.0.0     U         0 0          0 docker0
192.168.0.0     0.0.0.0         255.255.255.0   U         0 0          0 ens33
192.168.96.0    0.0.0.0         255.255.255.0   U         0 0          0 ens33
192.168.96.0    0.0.0.0         255.255.255.0   U         0 0          0 ens33
```

5、显示端口号为 22 的连接情况。



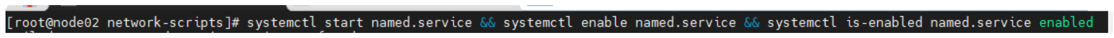
```
[root@node02 network-scripts]# netstat -antu | grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 192.168.96.123:22       192.168.96.1:4005       ESTABLISHED
tcp        0      0 192.168.96.138:22       192.168.96.1:4216       ESTABLISHED
tcp        0      0 192.168.96.123:22       192.168.96.1:4006       ESTABLISHED
tcp        0      0 192.168.96.138:22       192.168.96.1:4225       ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
[root@node02 network-scripts]#
```

6、捕获经过网络接口 eno16777736 的数据包。



```
完毕！
[root@node02 network-scripts]# tcpdump -i ens33
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
19:04:23.233123 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 374281911:374281975, ack 3037680531, win 379, length
19:04:23.235362 IP node02.46174 > gateway.domain: 382+ PTR? 1.96.168.192.in-addr.arpa. (43)
19:04:23.235556 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 64:240, ack 1, win 379, length 176
19:04:23.235670 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 240, win 4106, length 0
19:04:23.252018 IP gateway.domain > node02.46174: 382 NXDomain 0/1/0 (120)
19:04:23.253005 IP node02.54603 > gateway.domain: 2835+ PTR? 2.96.168.192.in-addr.arpa. (43)
19:04:23.253162 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 240:400, ack 1, win 379, length 160
19:04:23.266267 IP gateway.domain > node02.54603: 2835 NXDomain 0/1/0 (120)
19:04:23.266572 IP node02.33780 > gateway.domain: 58569+ PTR? 138.96.168.192.in-addr.arpa. (45)
19:04:23.293939 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 400, win 4105, length 0
19:04:23.360463 IP gateway.domain > node02.33780: 58569 NXDomain 0/1/0 (122)
19:04:23.361016 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 400:720, ack 1, win 379, length 320
19:04:23.361239 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 720:800, ack 1, win 379, length 80
19:04:23.361349 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 800, win 4104, length 0
19:04:23.361492 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 800:880, ack 1, win 379, length 80
19:04:23.361701 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 880:944, ack 1, win 379, length 64
19:04:23.361790 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 944, win 4103, length 0
19:04:23.361918 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 944:1024, ack 1, win 379, length 80
19:04:23.362076 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1024:1088, ack 1, win 379, length 64
19:04:23.362157 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 1088, win 4102, length 0
19:04:23.362278 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1088:1168, ack 1, win 379, length 80
19:04:23.362452 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1168:1232, ack 1, win 379, length 64
19:04:23.362536 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 1232, win 4102, length 0
19:04:23.362698 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1232:1312, ack 1, win 379, length 80
19:04:23.363883 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1312:1376, ack 1, win 379, length 64
19:04:23.363997 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 1376, win 4101, length 0
19:04:23.364126 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1376:1456, ack 1, win 379, length 80
19:04:23.364293 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1456:1520, ack 1, win 379, length 64
19:04:23.364368 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [P.], seq 1:97, ack 1456, win 4101, length 96
19:04:23.364435 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1520:1568, ack 97, win 379, length 48
19:04:23.364503 IP 192.168.96.1.pxc-pin > node02.ssh: Flags [.], ack 1568, win 4101, length 0
19:04:23.364669 IP node02.ssh > 192.168.96.1.pxc-pin: Flags [P.], seq 1568:1696, ack 97, win 379, length 128
```

7、使用命令启动 named 服务，并且设置该服务在计算机启动时一起启动。



```
[root@node02 network-scripts]# systemctl start named.service && systemctl enable named.service && systemctl is-enabled named.service enabled
```

总结和分析： 经过这次实验，我进一步熟悉了 linux 操作系统 bash 的常用基本内置命令，还学习了 linux 日常管理和维护的基本知识,了解了进程管理,查看以及终止等命令,也学会了任务计划的基本写法;后面也学习了 Linux 的网络配置熟悉了常见的网络配置文件，了解和使用了常用的网络命令，这是 linux 学习的基础，为后面的进一步学习打下了良好的基础。