

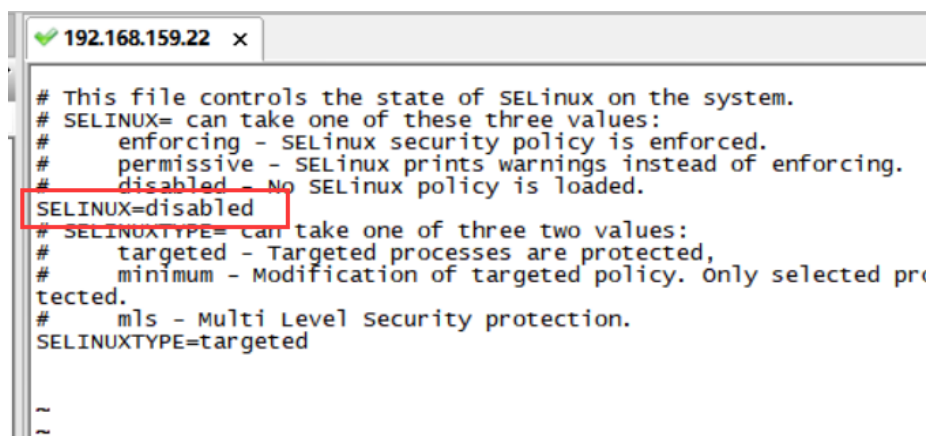
第 00 章作业 - CentOS 7.5 深度配置

1652817 钟钰琛 计算机科学与技术

2018 年 9 月 28 日

1 Linux 共享文件夹

首先关闭 selinux, 修改/etc/selinux/config 重启后生效.



```
192.168.159.22 x
# 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 is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected pro
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
~
~
```

图 1: 关闭 selinux

同时也要关闭防火墙,

```
systemctl stop firewalld.service
systemctl disable firewalld.service
```

安装 samba, 顺便把 samba 设置成开机启动

```
yum -y install samba samba-client samba-common
systemctl enable smb
```

建立一个新的用户 zyc, 并加入到 smb 数据库中

```
[root@vm-linux ~]# useradd zyc
[root@vm-linux ~]# smbpasswd -a zyc
New SMB password:
Retype new SMB password:
Added user zyc.
[root@vm-linux ~]#
```

图 2: add user

在根目录下建立 d1652817, 然后加权限

```
mkdir d1652817
chmod -R 777 d1652817
```

然后修改/etc/samba/smb.conf, 里面添加

```
[d1652817]
    path = /d1652817
    valid users = zyc
    browseable = yes
    writable = yes
    create mask = 0664
    directory mask = 0755
```

图 3: smb.conf

然后在 win+R 输入

```
\\192.168.159.22\d1652817
```

可以进入

1.1 a

右键该文件夹, 点“映射网络驱动器”, 映射为 Z 盘



图 4: 进入共享文件夹

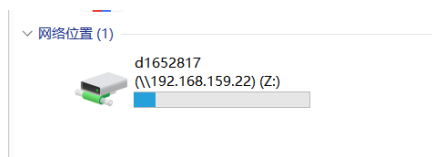


图 5: 映射

1.2 b

Windows 的访问权限可以通过修改刚刚在 `/etc/samba/smb.conf` 如果要求读写，就像刚刚那样设置即可。

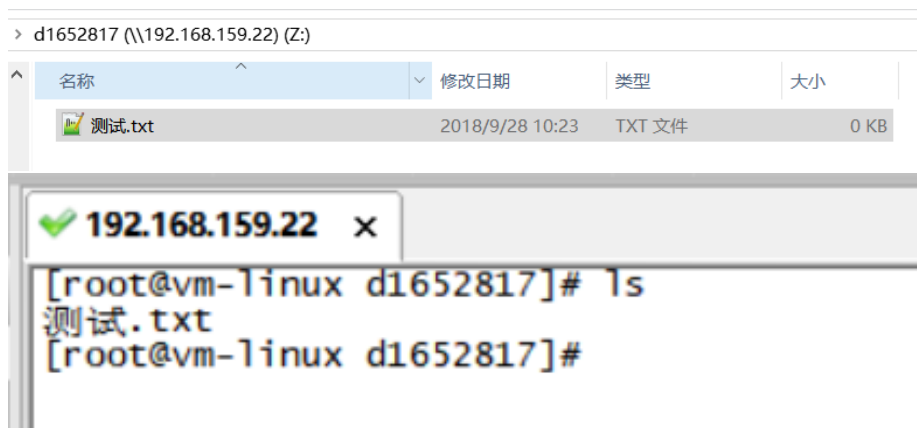


图 6: 读写

例如，只读可以设置成

```
[d1652817]
path = /root/d1652817
valid users = zyc
browseable = yes
read only = yes
```

这时候如果建立一个文件，会出现



图 7: 只读

1.3 c

敏感

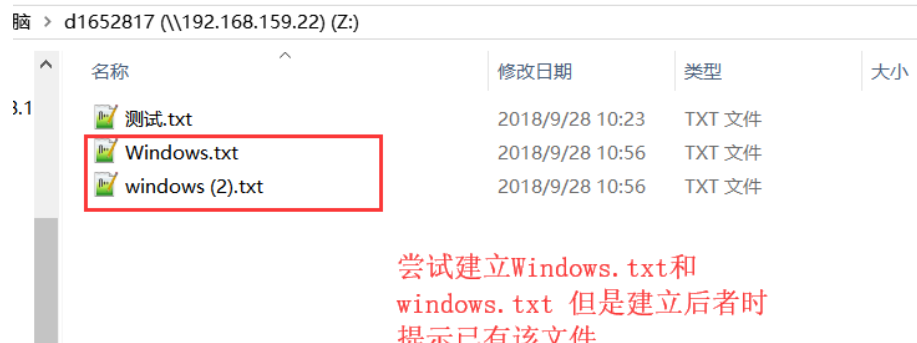


图 8: 只读

1.4 d

通过 smbclient

```
[root@vm-linux ~]# smbclient //192.168.159.22/d1652817 -u zyc
Enter SAMBA\zyc's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0  Fri Sep 28 10:56:47 2018
..               D          0  Fri Sep 28 11:29:17 2018
测试.txt         N          0  Fri Sep 28 10:23:28 2018
windows.txt      N          0  Fri Sep 28 10:56:21 2018
windows (2).txt  N          0  Fri Sep 28 10:56:28 2018

                    52403200 blocks of size 1024. 45988104 blocks available
smb: \>
```

图 9: 另一台 linux 访问共享文件夹

1.5 e

继续在/etc/samba/smb.conf 后面添加 hosts deny

```
[d1652817]
    path = /d1652817
    valid users = zyc
    browseable = yes
#    read only = yes
    writable = yes
    create mask = 0664
    directory mask = 0755
    hosts deny = 192.168.159.23
```

图 10: smb.conf

然后另一台 linux 再登陆时：会出现

```
[root@vm-linux ~]# smbclient //192.168.159.22/d1652817 -u zyc
Enter SAMBA\zyc's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
[root@vm-linux ~]#
```

图 11: linux 被拒绝登陆

1.6 f

在/etc/samba/smb.conf 的 [global] 加

```
unix charset = cp936
dos charset = cp936
```

```
> 192.168.159.22 > d1652817
```

| 名称 | 修改日期 | 类型 | 大小 |
|------------|-----------------|--------|------|
| 新建文本文档.txt | 2018/9/28 11:35 | TXT 文件 | 0 KB |

```
[root@vm-linux d1652817]# ls
新建文本文档.txt
[root@vm-linux d1652817]#
```

图 12: 中文字符集

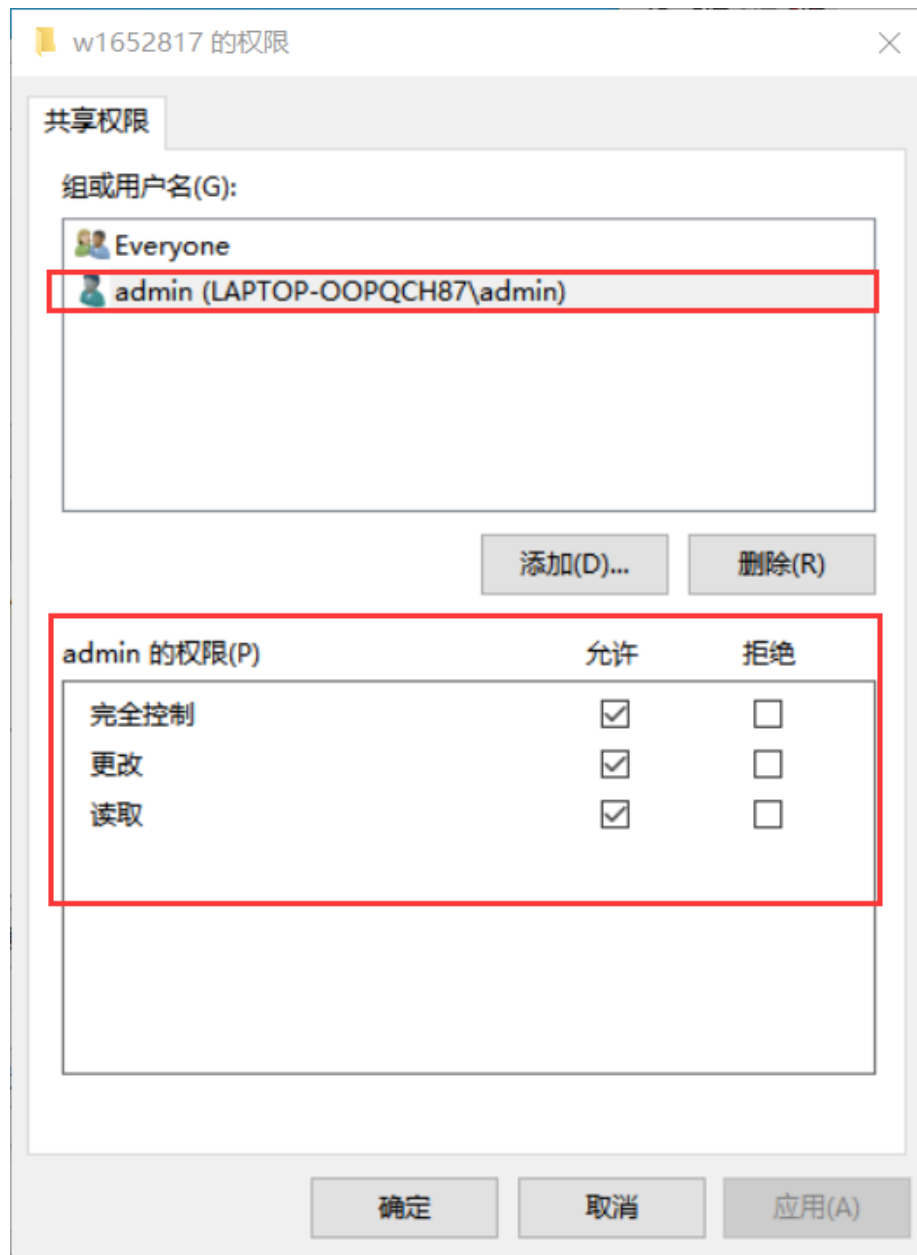


图 14: 建立共享文件夹

2.1 a

然后在 linux 中输入

```
mount -t cifs //192.168.1.101/w1652817 /mnt/w1652817 -o
username=admin,password=zyc759631647,iocharset=gb2312
```

就可以成功挂载

```
[root@vm-linux ~]# df -h
文件系统 容量 已用 可用 已用% 挂载点
/dev/mapper/centos-root 50G 6.2G 44G 13% /
devtmpfs 899M 0 899M 0% /dev
tmpfs 911M 0 911M 0% /dev/shm
tmpfs 911M 9.6M 902M 2% /run
tmpfs 911M 0 911M 0% /sys/fs/cgroup
/dev/sda1 1014M 145M 870M 15% /boot
/dev/mapper/centos-home 203G 33M 203G 1% /home
tmpfs 183M 0 183M 0% /run/user/0
//192.168.1.101/w1652817 466G 380G 87G 82% /mnt/w1652817
[root@vm-linux ~]#
```

图 15: 挂载成功

2.2 b

权限直接在图 14 的下方红框中可以修改

2.3 c

大小写敏感, 例如

```
[root@vm-linux w1652817]# mkdir zyc
[root@vm-linux w1652817]# mkdir ZYC
mkdir: 无法创建目录"ZYC": 文件已存在
```

图 16: 大小写敏感

2.4 d

取消映射输入

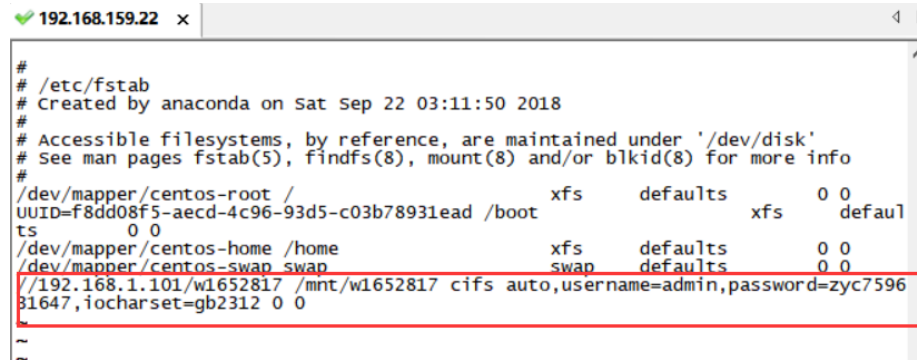
```
umount /mnt/w1652817
```

如果出现“目标忙”的报错,则 kill 掉相关进程

```
fuser -m /mnt/w1652817  
kill -9 xxx
```

2.5 e

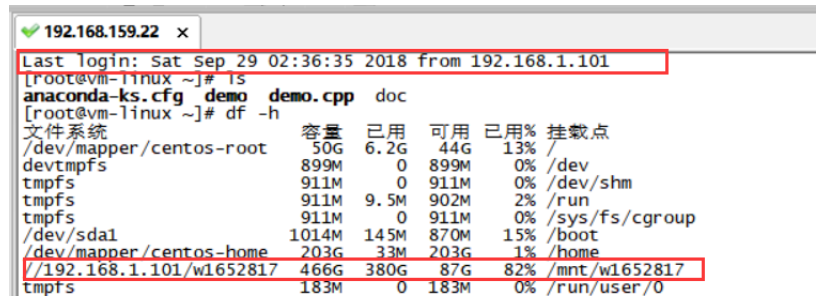
在/etc/fstab 加入以下内容



```
#  
# /etc/fstab  
# Created by anaconda on Sat Sep 22 03:11:50 2018  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk'  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info  
#  
/dev/mapper/centos-root / xfs defaults 0 0  
UUID=f8dd08f5-aecd-4c96-93d5-c03b78931ead /boot xfs default  
ts 0 0  
/dev/mapper/centos-home /home xfs defaults 0 0  
/dev/mapper/centos-swap swap swap defaults 0 0  
//192.168.1.101/w1652817 /mnt/w1652817 cifs auto,username=admin,password=zyc7596  
81647,iocharset=gb2312 0 0  
~  
~
```

图 17: /etc/fstab

reboot 后,发现自动挂载成功



```
192.168.159.22 x  
Last login: Sat Sep 29 02:36:35 2018 from 192.168.1.101  
[root@vm-linux ~]# ls  
anaconda-ks.cfg demo demo.cpp doc  
[root@vm-linux ~]# df -h  
文件系统 容量 已用 可用 已用% 挂载点  
/dev/mapper/centos-root 50G 6.2G 44G 13% /  
devtmpfs 899M 0 899M 0% /dev  
tmpfs 911M 0 911M 0% /dev/shm  
tmpfs 911M 9.5M 902M 2% /run  
tmpfs 911M 0 911M 0% /sys/fs/cgroup  
/dev/sda1 1014M 145M 870M 15% /boot  
/dev/mapper/centos-home 203G 33M 203G 1% /home  
//192.168.1.101/w1652817 466G 380G 87G 82% /mnt/w1652817  
tmpfs 183M 0 183M 0% /run/user/0
```

图 18: /etc/fstab

2.6 f



图 19: 中文正常

3 新增硬盘

右键虚拟机，点击设置
进入虚拟机，输入 `fdisk -l`
然后进行分区

```
parted /dev/sdb
(parted) mklabel gpt
(parted) mkpart primary ext4 0% 50%
(parted) mkpart primary ext4 50% 100%
(parted) quit
```

再次输入 `fdisk -l`，发现有了两个分区。

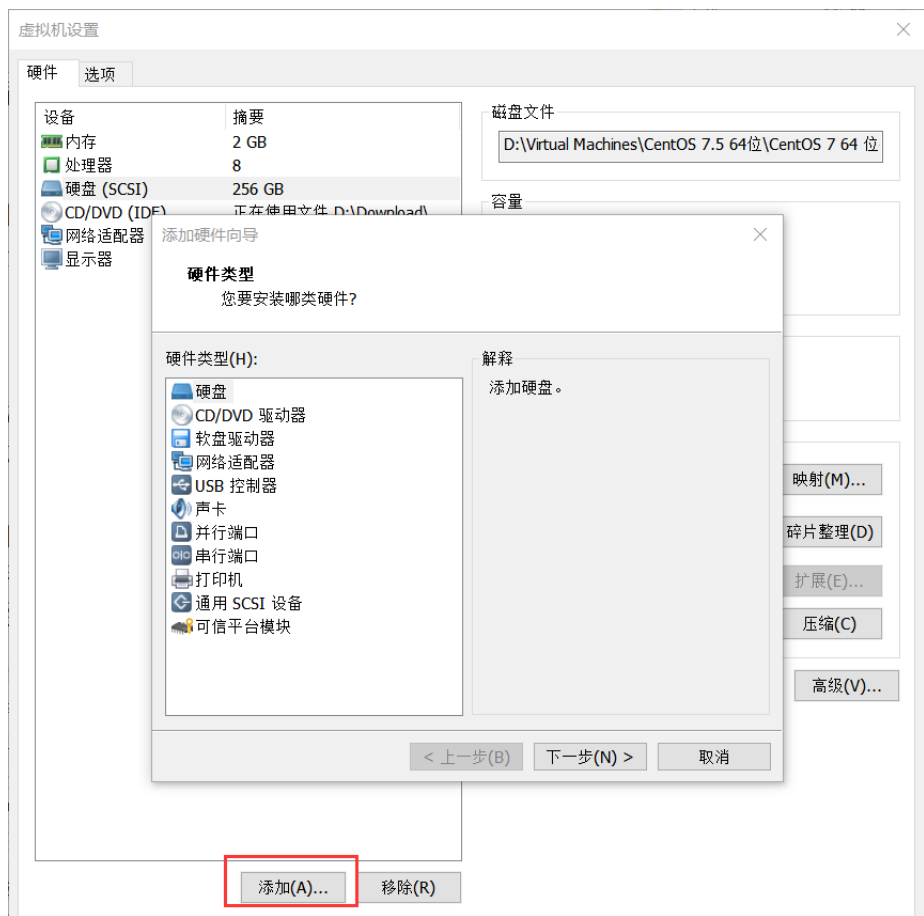


图 20: 新建硬盘



图 21: 新建硬盘

```
磁盘 /dev/sdb: 53.7 GB, 53687091200 字节, 104857600 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
```

图 22: /dev/sdb

```
磁盘 /dev/sdb: 53.7 GB, 53687091200 字节, 104857600 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: gpt
Disk identifier: 6EEF1E54-DAD0-486D-A5B7-298C198217AF

#          Start      End      Size Type      Name
1          2048      52428799    25G Microsoft basic primary
2      52428800  104855551    25G Microsoft basic primary
```

图 23: /dev/sdb

然后进行格式化并挂载分区

```
mkfs.ext4 /dev/sdb1
mkfs.ext4 /dev/sdb2
mount /dev/sdb1 /disk-1652817-1
mount /dev/sdb2 /disk-1652817-2
```

用 df -h 查看

```
[root@vm-linux ~]# df -h
文件系统 容量 已用 可用 已用% 挂载点
/dev/mapper/centos-root 50G 6.2G 44G 13% /
devtmpfs 899M 0 899M 0% /dev
tmpfs 911M 0 911M 0% /dev/shm
tmpfs 911M 9.5M 902M 2% /run
tmpfs 911M 0 911M 0% /sys/fs/cgroup
/dev/sda1 1014M 145M 870M 15% /boot
/dev/mapper/centos-home 203G 33M 203G 1% /home
//192.168.1.101/w1652817 466G 378G 89G 82% /mnt/w1652817
tmpfs 183M 0 183M 0% /run/user/0
/dev/sdb1 25G 45M 24G 1% /disk-1652817-1
/dev/sdb2 25G 45M 24G 1% /disk-1652817-2
[root@vm-linux ~]#
```

图 24: 结果

4 新增网卡

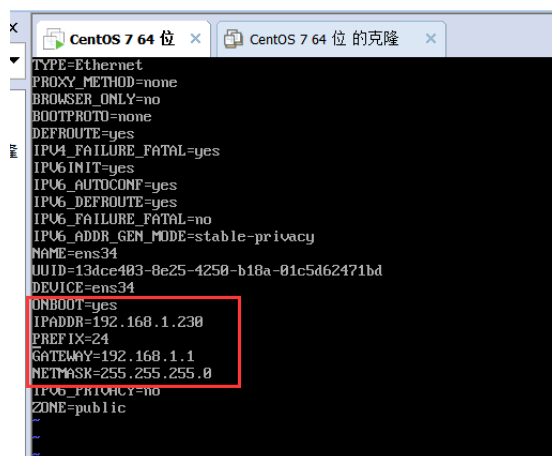
在 vmware 下添加网卡。设置成 Bridge 模式

进入 linux, 输入 ip link, 发现多出来的是 ens34

```
[root@om-linux network-scripts]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
fault qlen 1000
    link/ether 00:0c:29:99:df:a5 brd ff:ff:ff:ff:ff:ff
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
fault qlen 1000
    link/ether 00:0c:29:99:df:af brd ff:ff:ff:ff:ff:ff
[root@om-linux network-scripts]#
```

图 25: 新增 ens34

然后在/etc/sysconfig/network-scripts 目录下建立 ifcfg-ens34, 做如下配置



```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6_INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens34
UUID=13dce403-8e25-4250-b18a-01c5d62471bd
DEVICE=ens34
ONBOOT=yes
IPADDR=192.168.1.230
PREFIX=24
GATEWAY=192.168.1.1
NETMASK=255.255.255.0
IPV6_PRIVACY=no
ZONE=public
```

图 26: ifcfg-ens34

然后输入 systemctl restart network, 查看 ifconfig 均能 ping 通

```

[root@vm-linux network-scripts]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.159.22 netmask 255.255.255.0 broadcast 192.168.159.255
    inet6 fe80::f11c:34ba:b012:8508 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:99:df:a5 txqueuelen 1000 (Ethernet)
    RX packets 177 bytes 20326 (19.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 322 bytes 25642 (25.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens34: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.230 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::20c:29ff:fe99:daf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:99:df:af txqueuelen 1000 (Ethernet)
    RX packets 451 bytes 34357 (33.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 204 bytes 17178 (16.7 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
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 232 bytes 21408 (20.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 232 bytes 21408 (20.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

图 27: ifconfig

```

[root@vm-linux network-scripts]# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data:
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.18 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.963 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=5.04 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=2.77 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.963/2.491/5.044/1.632 ms
[root@vm-linux network-scripts]# ping 192.168.159.1
PING 192.168.159.1 (192.168.159.1) 56(84) bytes of data:
64 bytes from 192.168.159.1: icmp_seq=1 ttl=128 time=0.373 ms
64 bytes from 192.168.159.1: icmp_seq=2 ttl=128 time=0.389 ms
64 bytes from 192.168.159.1: icmp_seq=3 ttl=128 time=0.447 ms
64 bytes from 192.168.159.1: icmp_seq=4 ttl=128 time=0.350 ms
^C
--- 192.168.159.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.350/0.389/0.447/0.043 ms
[root@vm-linux network-scripts]#

```

图 28: ping

5 创建普通用户

输入

```

groupadd stu
useradd -d /home/u1652817 -m u1652817 -G stu
useradd -d /home/u1652816 -m u1652816 -G stu

```

查看 stu 下所有用户

```
[root@vm-linux ~]# grep "stu" /etc/group
stu:x:1001:u1652817,u1652816
[root@vm-linux ~]#
```

图 29: stu 下用户

查看/home 下

```
[root@vm-linux home]# ls
u1652816 u1652817 zyc
[root@vm-linux home]#
```

图 30: 用户目录

设置密码:

```
192.168.159.22 x
[root@vm-linux home]# passwd u1652817
更改用户 u1652817 的密码。
新的 密码:
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@vm-linux home]# passwd u1652816
更改用户 u1652816 的密码。
新的 密码:
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@vm-linux home]#
```

图 31: 设置密码

禁用/启用某个普通用户:

```
[root@vm-linux home]# grep u1652817 /etc/shadow
u1652817:!:5$ksHNCx8w$wo200Zw72cMjg5wkbN97oP4s6QZEF9DK5TC5/gyQt1xuy75.Y2xu04D0ErS
VYS2ALnhsvH.0PbdaORDlv6Pnw/:17802:0:99999:7:::
[root@vm-linux home]# usermod -L u1652817
[root@vm-linux home]# grep u1652817 /etc/shadow
u1652817:!:5$ksHNCx8w$wo200Zw72cMjg5wkbN97oP4s6QZEF9DK5TC5/gyQt1xuy75.Y2xu04D0ErS
SVYS2ALnhsvH.0PbdaORDlv6Pnw/:17802:0:99999:7:::
[root@vm-linux home]# usermod -u u1652817
[root@vm-linux home]# grep u1652817 /etc/shadow
u1652817:!:5$ksHNCx8w$wo200Zw72cMjg5wkbN97oP4s6QZEF9DK5TC5/gyQt1xuy75.Y2xu04D0ErS
VYS2ALnhsvH.0PbdaORDlv6Pnw/:17802:0:99999:7:::
[root@vm-linux home]#
```

感叹号表示被禁用

图 32: 禁用/启用普通用户

删除普通用户:

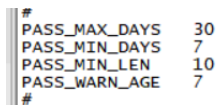
```
userdel u1652816
```

建立一个超级用户：在/etc/passwd 里面修改第三列 uid 和第四列 gid 为 0 即可

```
vim /etc/passwd  
u1652817:x:0:0:/home/u1652817:/bin/bash
```

6 用户密码复杂性

在/etc/login.defs 里面可以修改密码的有效期例如：



```
#  
PASS_MAX_DAYS 30  
PASS_MIN_DAYS 7  
PASS_MIN_LEN 10  
PASS_WARN_AGE 7  
#
```

图 33: /etc/login.defs

pan_cracklib.so 是控制密码复杂度的，有这么几种参数

- minlen 密码最短长度
- dcredit 至少多少个数字
- ucredit 至少多少个大写字母
- lcredit 至少多少个小写字母
- ocredit 至少多下个其他字符
- difok 新密码与旧密码至少多少个不同字符
- remember 不能与前多少次旧密码相同

修改/etc/pam.d/system-auth 文件，
如果要第一次登陆后强制改密，可以在第一次登陆后强制其密码立刻过期

```
chage -d 0 u1652817
```

```
password requisite pam_cracklib.so retry=5 difok=3 minlen=10 ucredit=-1 1
credit=-1 dcredit=-1 remember=10
```

图 34: /etc/pam.d/system-auth

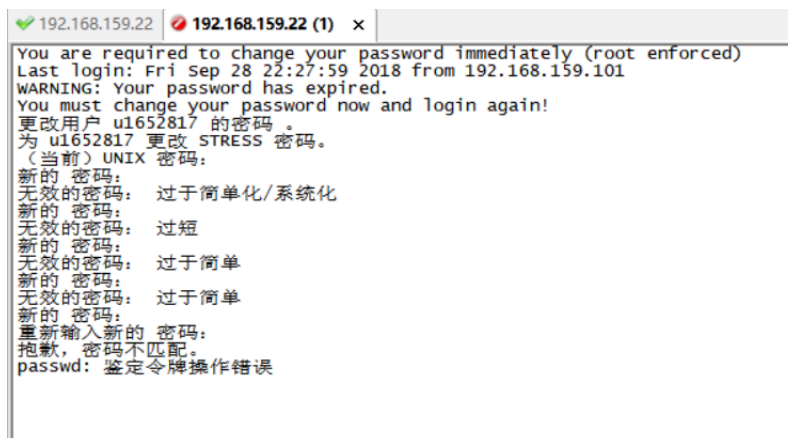
如果已经设置了用户的密码，先禁用该用户，再执行 chage

```
usermod -L u1652817
```

最后再启用该用户

```
usermod -U u1652817
```

再次登陆时出现



```
192.168.159.22 192.168.159.22 (1) x
You are required to change your password immediately (root enforced)
Last login: Fri Sep 28 22:27:59 2018 from 192.168.159.101
WARNING: Your password has expired.
You must change your password now and login again!
更改用户 u1652817 的密码。
为 u1652817 更改 STRESS 密码。
(当前) UNIX 密码:
新的 密码:
无效的密码: 过于简单化/系统化
新的 密码:
无效的密码: 过短
新的 密码:
无效的密码: 过于简单
新的 密码:
无效的密码: 过于简单
新的 密码:
重新输入新的 密码:
抱歉, 密码不匹配。
passwd: 鉴定令牌操作错误
```

图 35: 设置个密码真难

一段时间强制改密可以用设置改密间隔

```
chage -m days
```

也可以修改/etc/login.defs 中的 PASS_MAX_DAYS、PASS_MIN_DAYS

试验证明跟修改 date 没关系，照样登陆先设置一天就密码过期，然后设置 date 到 2019 年

```

Last login: Sun Sep 30 00:00:07 2018 from 192.168.159.101
[u1652817@vm-linux ~]$ date
2019年 09月 30日 星期一 00:01:22 CST
[u1652817@vm-linux ~]$

```

图 36: 时间飞逝到 2019 年, 照常登陆

7 普通用户磁盘配额控制

首先在/etc/fstab 中的/home 中增加 usrquota, grpquota

```

#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=F8dd08f5-aecd-4c96-93d5-c03b78931ead /boot xfs defaults 0 0
/dev/mapper/centos-home /home xfs defaults,usrquota,grpquota 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
//192.168.1.101/w1652817 /mnt/w1652817 cifs auto,username=admin,password=zyc7596 0 0

```

图 37: /etc/fstab

然后生成配置, 启动 quota 服务

```

quotacheck -cum /home
quotaon -vug /home

```

设置用户 quota 容量

```
edquota u1652817
```

```

192.168.159.22 x
Disk quotas for user u1652817 (uid 1001):
Filesystem blocks soft hard inodes soft
/dev/mapper/centos-home 20 0 0 10 0

```

图 38: 配置容量

软硬设置区别: 一旦用户超过软限制, 配额计时器便开始计时。在配额计时器计时期间, 用户可以使用高于软限制的配额, 但不能超过硬限制。

```
edquota -t
```

```
192.168.159.22 x
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem      Block grace period   Inode grace period
/dev/mapper/centos-home      7days                7days
}
}
```

图 39: period 设置

可以指定 grace period, 给超过限额的用户一段时间来调整容量。block grace period 指定宽限时间

用户组配额: 跟用户设置磁盘配额限制类似, 加参数-g 即可
查看用户当前磁盘配额

```
quota -u u1652817
```

8 shell

```
if [ $3 ]
then
    echo "多余参数"
    echo -e "\t$0 student.conf initpwd.dat"
    echo
    exit
fi
```

```
if [ -z $2 ]
then
    echo "缺少参数"
    echo -e "\t$0 student.conf initpwd.dat"
    echo
    exit
fi
```

```
password=$(cat $2)
while read id name extras
do
```

```

if [[ $id =~ \#.* ]] || [ -z $name ]
then
    continue
fi

username = useradd u$id

echo "$password" | passwd --stdin u$id > /dev/null

passwd -e u$id > /dev/null

echo "u$id[$name] 已建立"
done < $1

```

这一段是如果有第三个参数, 说明参数多余, 报错, 退出

```

if [ $3 ]
then
    echo "多余参数"
    echo -e "\t$0 student.conf initpwd.dat"
    echo
    exit
fi

```

这一段是如果少于两个参数, 报错, 退出

```

if [ -z $2 ]
then
    echo "缺少参数"
    echo -e "\t$0 student.conf initpwd.dat"
    echo
    exit
fi

```

先取密码, 只读前两列, 后面的忽略 (extras). 如果含有 #, 那么是注释, 跳过, 如果是空行也跳过. 然后取用户名和密码, 用 useradd 和 passwd 来实现

```
password=$(cat $2)
```

```
while read id name extras
do
    if [[ $id =~ \#.* ]] || [ -z $name ]
    then
        continue
    fi

    username = useradd u$id

    echo "$password" | passwd --stdin u$id > /dev/null

    passwd -e u$id > /dev/null

    echo "u$id[$name] 已建立"
done < $1
```
