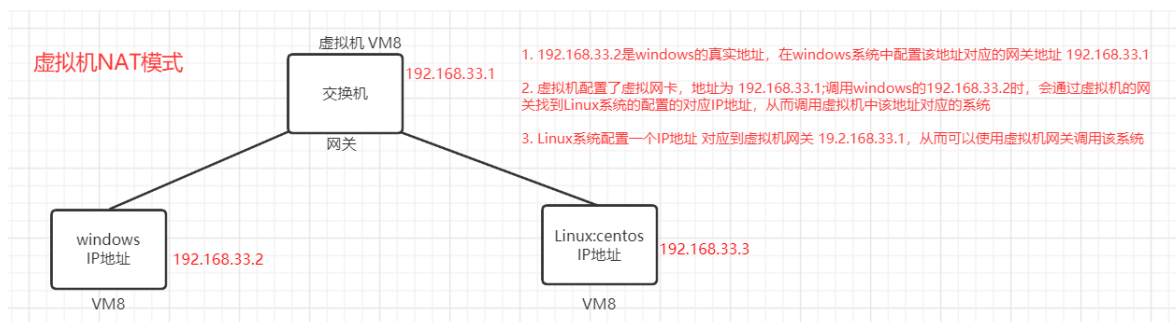
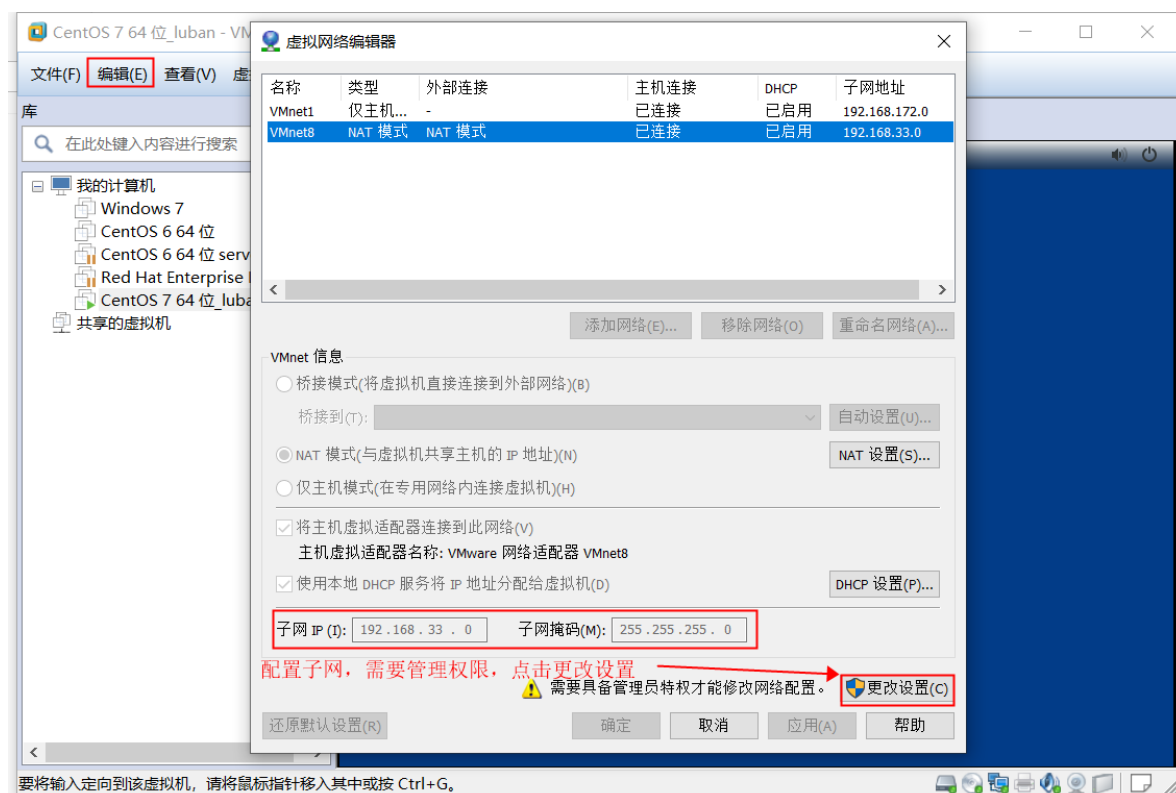


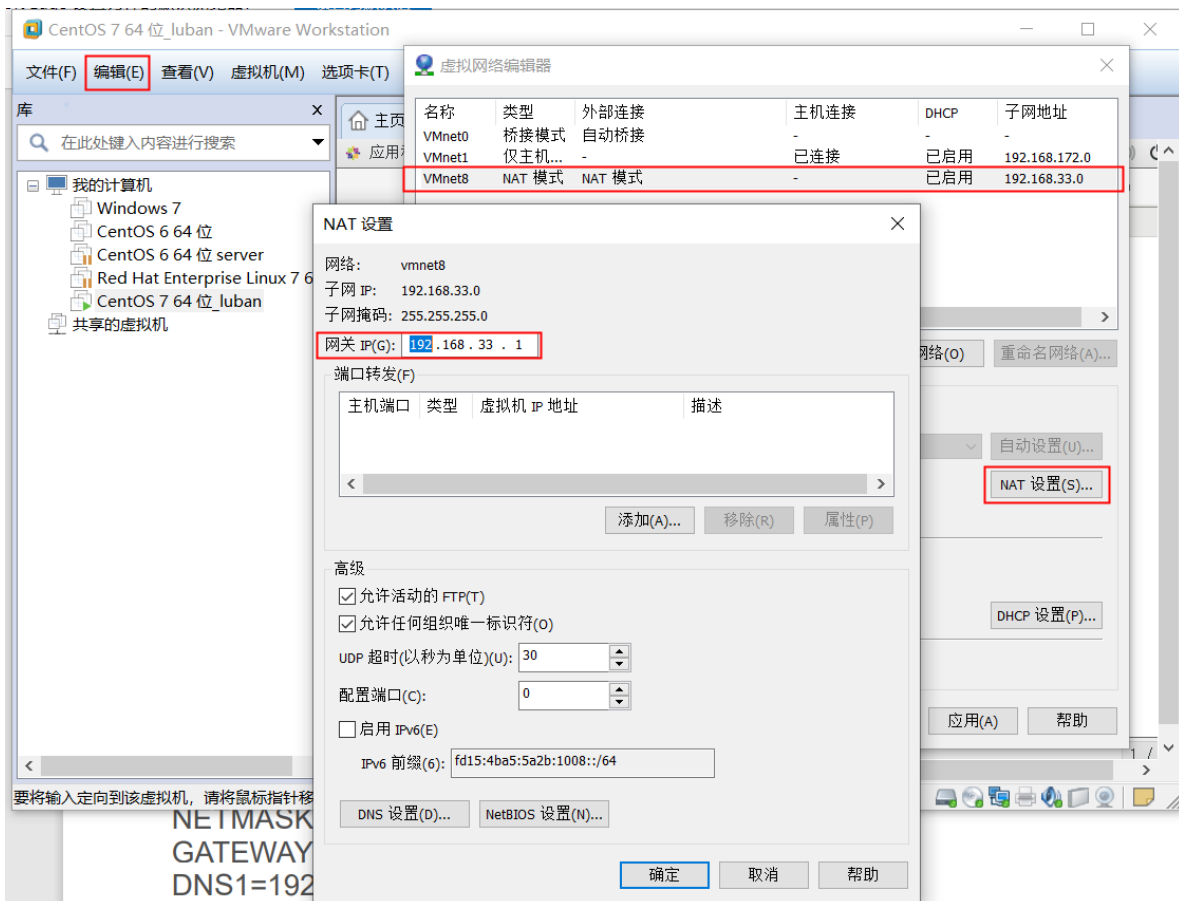
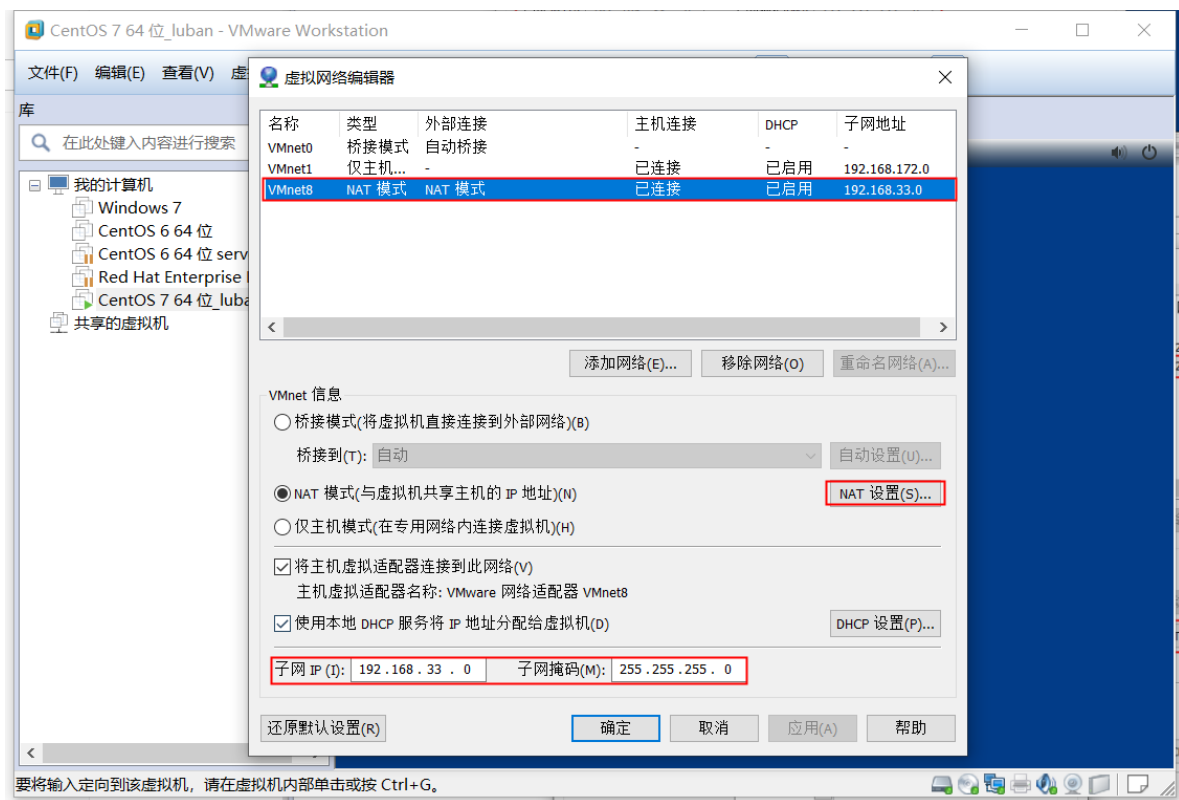
1.虚拟机网路配置



1. 192.168.33.2是windows的真实地址，在windows系统中配置该地址对应的网关地址 192.168.33.1
2. 虚拟机配置了虚拟网卡，地址为 192.168.33.1;调用windows的192.168.33.2时，会通过虚拟机的网关找到Linux系统的配置的对IP地址，从而调用虚拟机中该地址对应的系统
3. Linux系统配置一个IP地址 对应到虚拟机网关 19.2.168.33.1，从而可以使用虚拟机网关调用该系统

1.1 配置虚拟机的网关设置





- 虚拟机点击 编辑 --> 虚拟网络编辑器 --> 选择对应的模式 --> Net设置 --> 编辑网关地址

1.2 修改网络配置文件

```

DEVICE=eth0
HWADDR=00:0C:29:80:77:D7
TYPE=Ethernet
UUID=0025ae9f-2469-4e8a-9d68-537a5f46b9bb
ONBOOT=no
NM_CONTROLLED=yes
#BOOTPROTO=dhcp
BOOTPROTO=static
IPADDR=192.168.33.4
NETMASK=255.255.255.0
GATEWAY=192.168.33.1
DNS1=192.168.33.1
~
~
~
~
~
~
-- INSERT --

```

动态分配ip, 使用#将这行注释掉了

静态的ip

IP地址为: 192.168.33.4

子网掩码为: 255.255.255.0

网关地址为: 192.168.33.1 这是虚拟机配置的

域名解析地址: 192.168.33.1 等价于windows的localhost

- 设置 对应服务器中的配置信息
- 配置文件地址 `/etc/sysconfig/network-scripts/ifcfg-ens33`

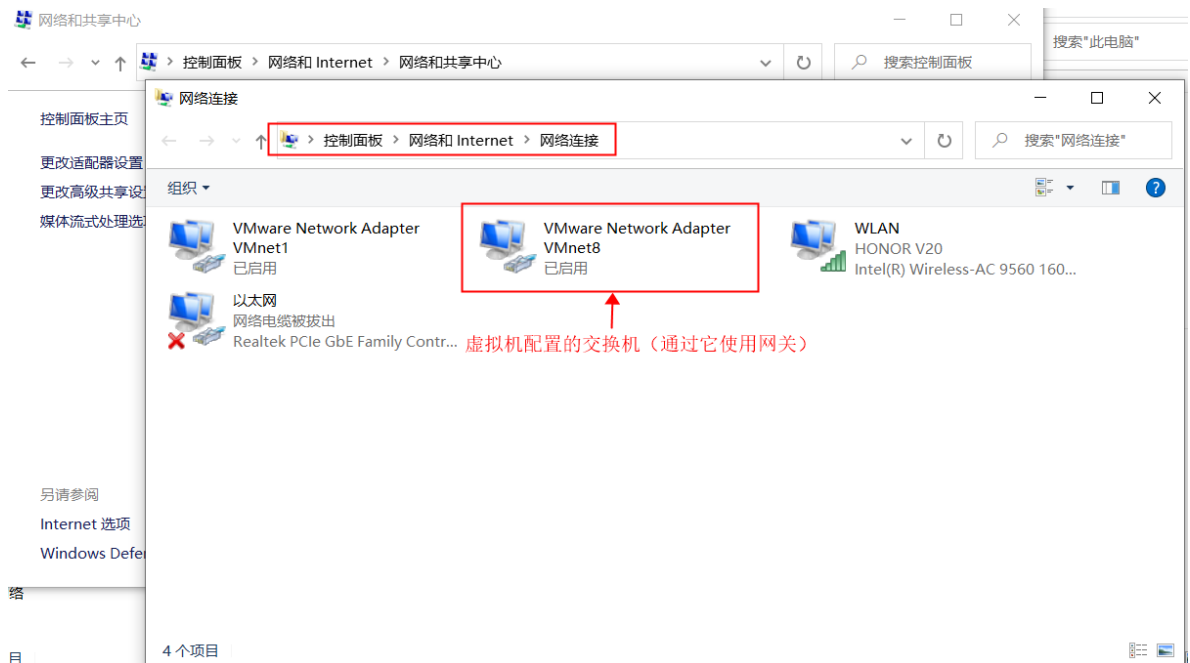
1.3 重启网络服务

- 执行 `service network restart` 命令

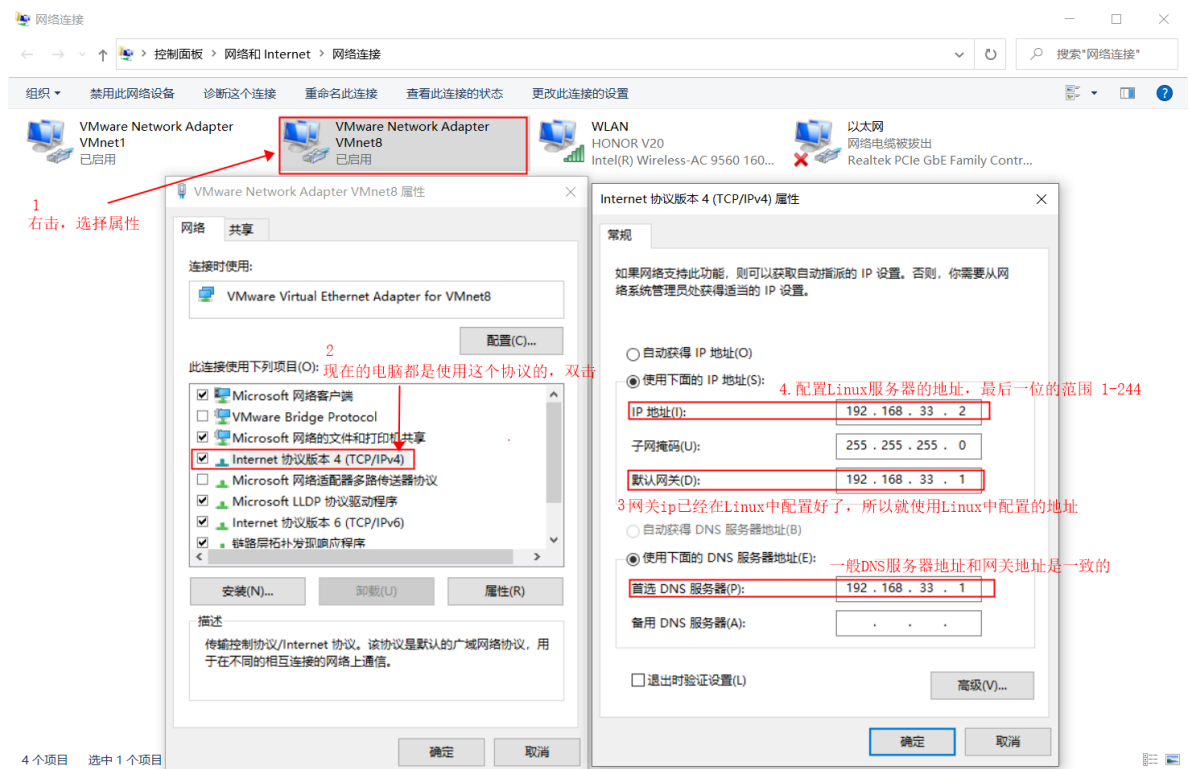
2. 配置windows连接的网关

2.1 配置windows到虚拟机网关的链接





- VMware Network Adapter VMnet1 和 VMware Network Adapter VMnet8 就是虚拟机设置的虚拟网卡
- 因为虚拟机中使用使用 VMnet8 进行配置网关，所以windows中也选择它配置网关



2.2 检查连接状况

```
命令提示符
子网掩码 . . . . . : 255.255.255.0
获得租约的时间 . . . . . : 2020年12月12日 11:44:12
租约过期的时间 . . . . . : 2020年12月12日 14:14:07
默认网关. . . . . :
DHCP 服务器 . . . . . : 192.168.172.254
DHCPv6 IAID . . . . . : 687886422
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-25-42-46-D5-F8-75-A4-05-1B-8E
DNS 服务器 . . . . . : fec0:0:0:ffff::1%1
                        fec0:0:0:ffff::2%1
                        fec0:0:0:ffff::3%1
TCP/IP 上的 NetBIOS . . . . . : 已启用

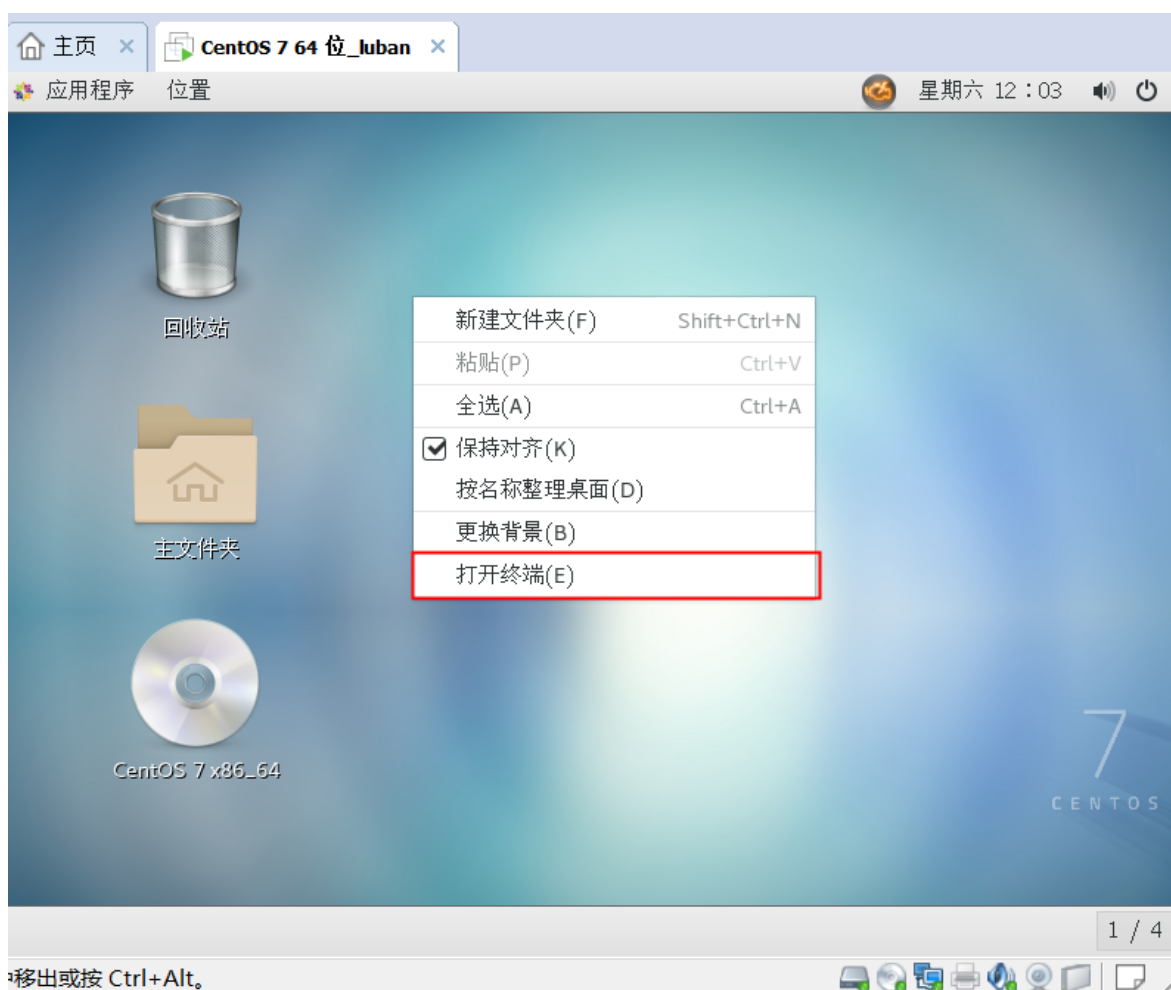
以太网适配器 VMware Network Adapter VMnet8:

连接特定的 DNS 后缀 . . . . . :
描述. . . . . : VMware Virtual Ethernet Adapter for VMnet8
物理地址. . . . . : 00-50-56-C0-00-08
DHCP 已启用 . . . . . : 否
自动配置已启用. . . . . : 是
本地链接 IPv6 地址. . . . . : fe80::74b0:c56c:19d4:9024%9(首选)
IPv4 地址 . . . . . : 192.168.33.2(首选)
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . : 192.168.33.1
DHCPv6 IAID . . . . . : 704663638
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-25-42-46-D5-F8-75-A4-05-1B-8E
DNS 服务器 . . . . . : 192.168.33.1
TCP/IP 上的 NetBIOS . . . . . : 已启用

无线局域网适配器 WLAN:

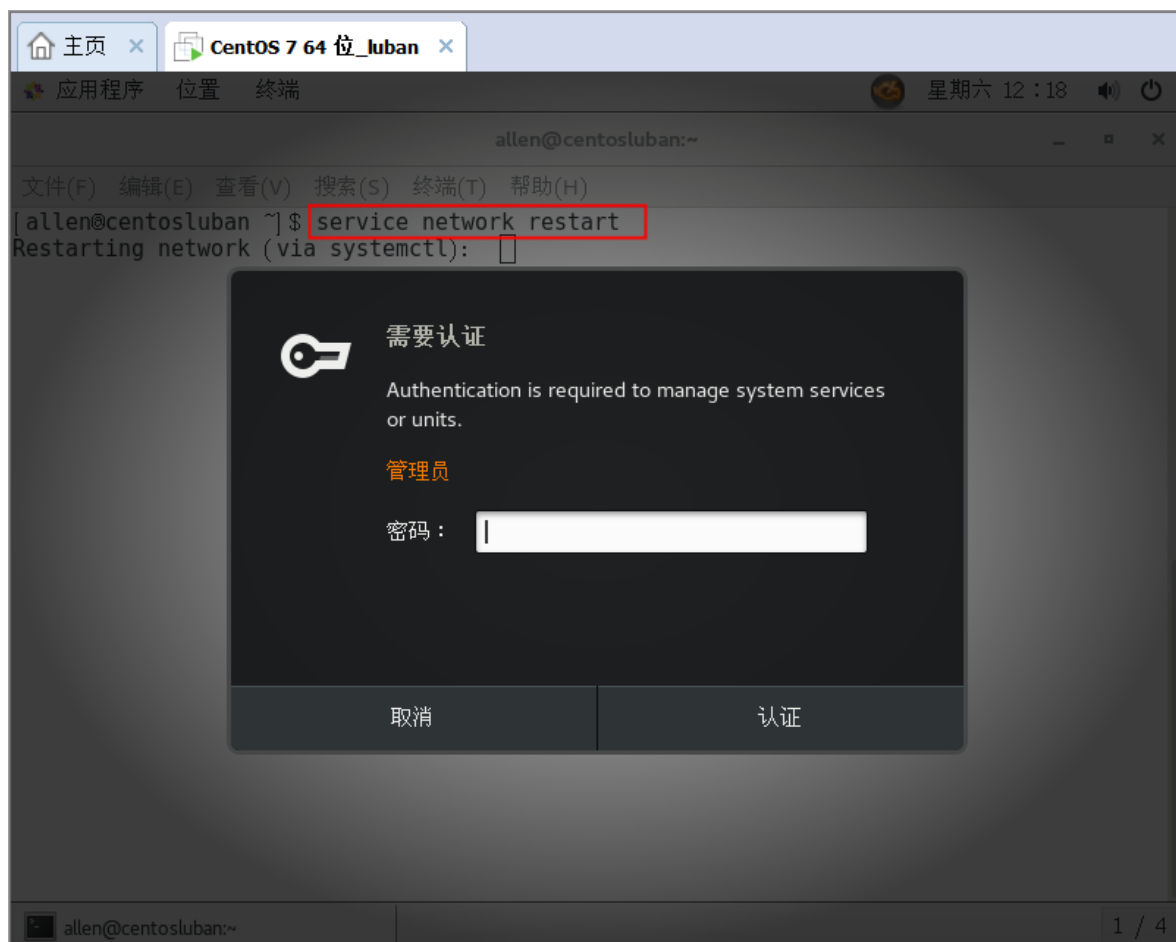
连接特定的 DNS 后缀 . . . . . :
描述. . . . . : Intel(R) Wireless-AC 9560 160MHz
物理地址. . . . . : 7C-B2-7D-55-A8-4D
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
IPv6 地址 . . . . . : 2409:8900:1240:1d3e:c164:597a:62b5:5b40(首选)
临时 IPv6 地址. . . . . : 2409:8900:1240:1d3e:a0d4:ad64:7d97:869e(首选)
本地链接 IPv6 地址. . . . . : fe80::c164:597a:62b5:5b40%20(首选)
```

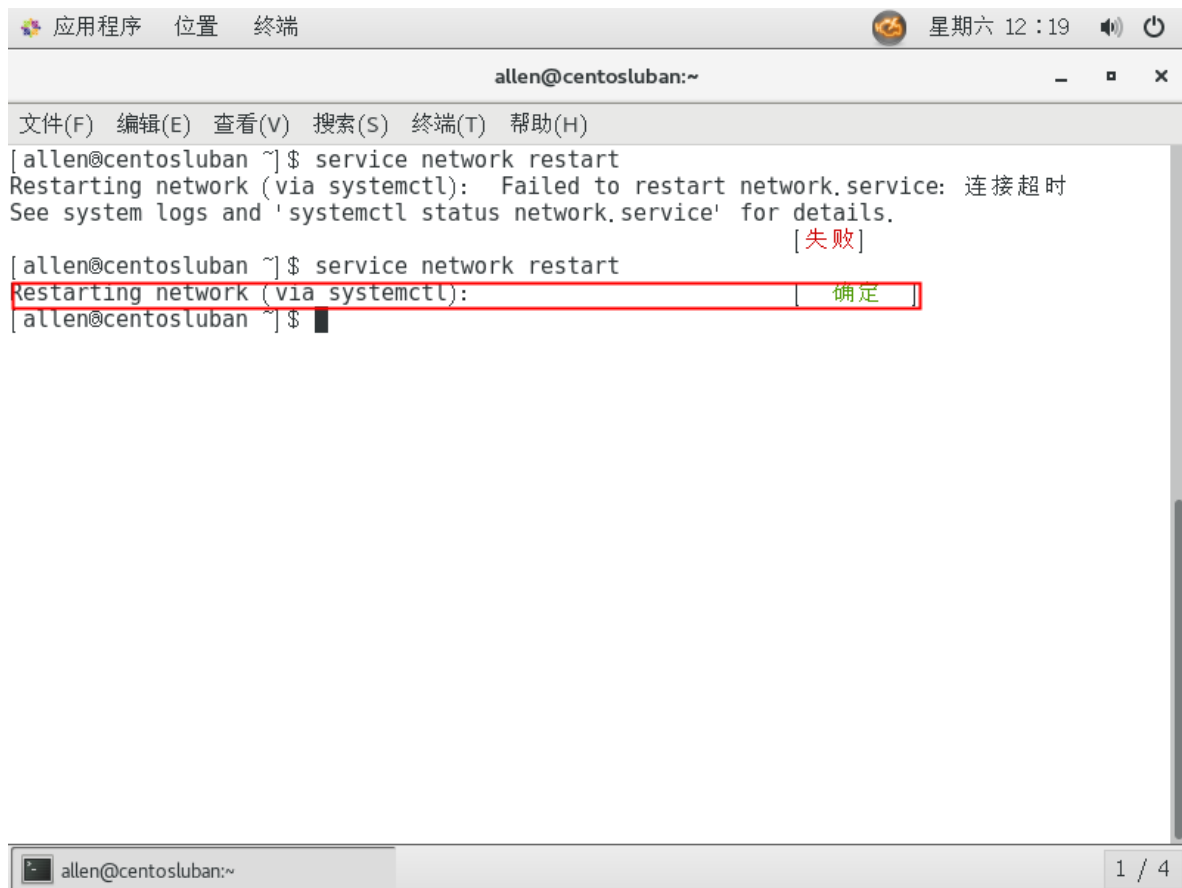
3.配置Linux连接的网关



```
应用程序 位置 终端 星期六 12:07
allen@centosluban:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[allen@centosluban ~]$ cat /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
#BOOTPROTO=static
BOOTPROTO=static
IPADDR=192.168.33.3
NETMASK=255.255.255.0
GATEWAY=192.168.33.1
DNS1=192.168.33.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6_INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=0f758049-a273-4450-9178-8fd32889161f
DEVICE=ens33
ONBOOT=no
[allen@centosluban ~]$ ll /etc/sysconfig/network-scripts/ifcfg-ens33
-rw-r--r-- 1 root root 381 12月 12 13:19 /etc/sysconfig/network-scripts/ifcfg-ens33
[allen@centosluban ~]$
```

- 在 `/etc/sysconfig/network-scripts/ifcfg-ens33` 文件中修改网络ip设置
- 配置服务器的IP地址: `IPADDR=192.168.33.3`
- 配置子网掩码: `NETMASK=192.168.33.1`
- 配置网关地址: `GATEWAY=192.168.33.1`
- DNS需要配置, 如果不配置, Linux只能链接内部网络, 不能连接外部网络





- 配置好ip地址，需要重启服务，命令：`service network restart`



- 服务器连接网络

```
命令提示符
Microsoft Windows [版本 10.0.18363.1379]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\Allen>ping 192.168.33.3

正在 Ping 192.168.33.3 具有 32 字节的数据:
来自 192.168.33.3 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.33.3 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.33.3 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.33.3 的回复: 字节=32 时间<1ms TTL=64

192.168.33.3 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\Allen>
```



- Linux 连接外网测试

4. 知识点

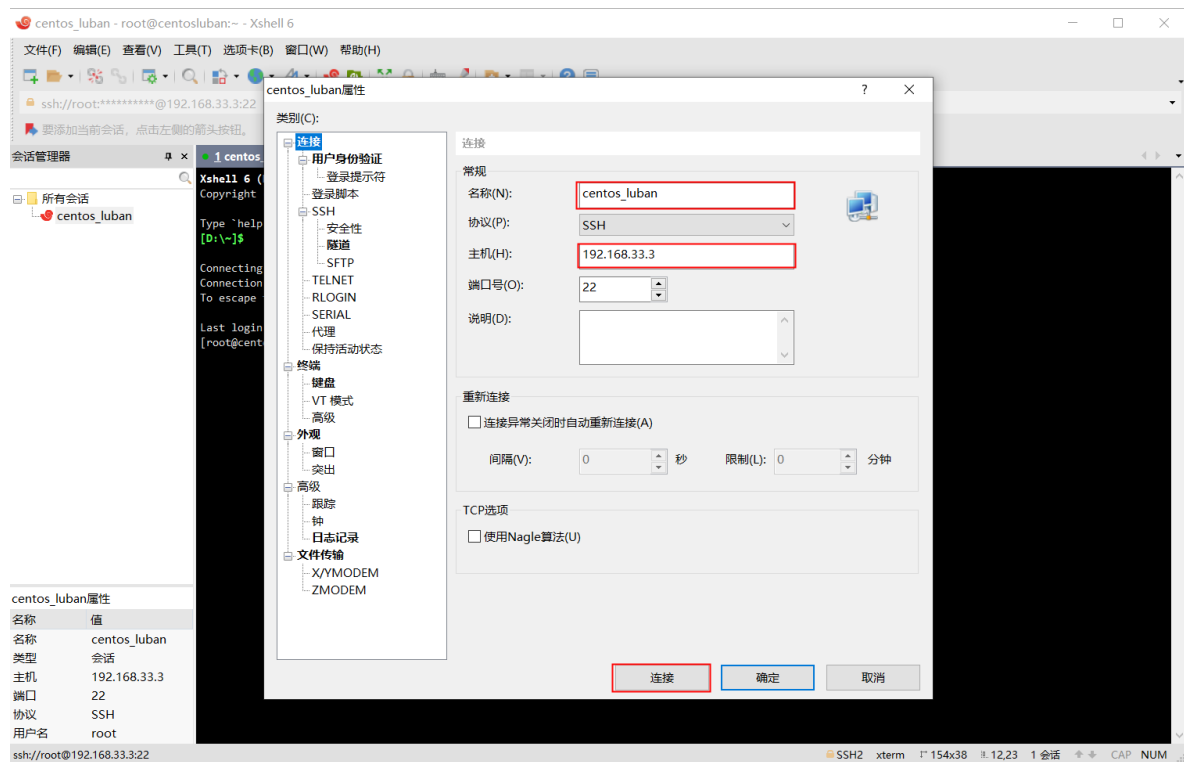
4.1 虚拟机

1. NAT 网络模式：虚拟机给Linux系统虚拟出一个网卡，进行网络连接
2. SATA 机械硬盘作为的磁盘
NVMe 固态硬盘作为磁盘

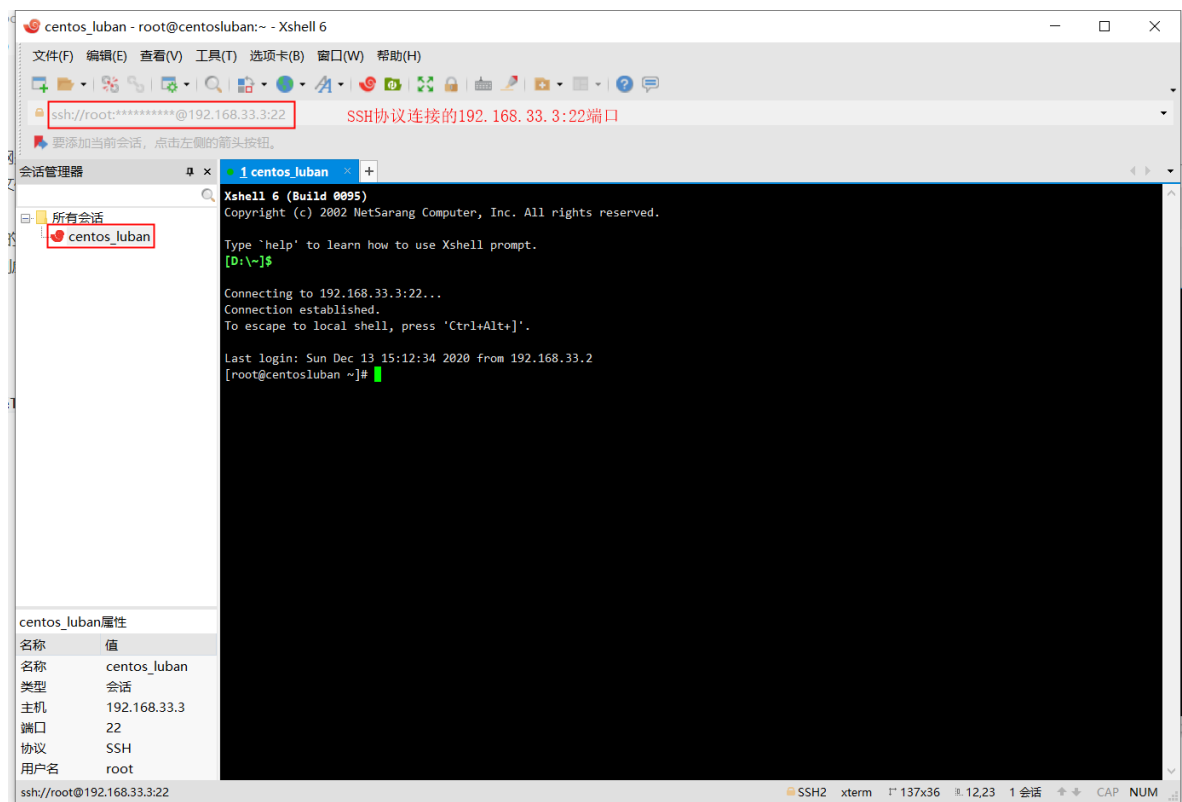
4.2 Linux

4.2.1 通过 XShell 连接 Linux 服务器

1. 连接服务器



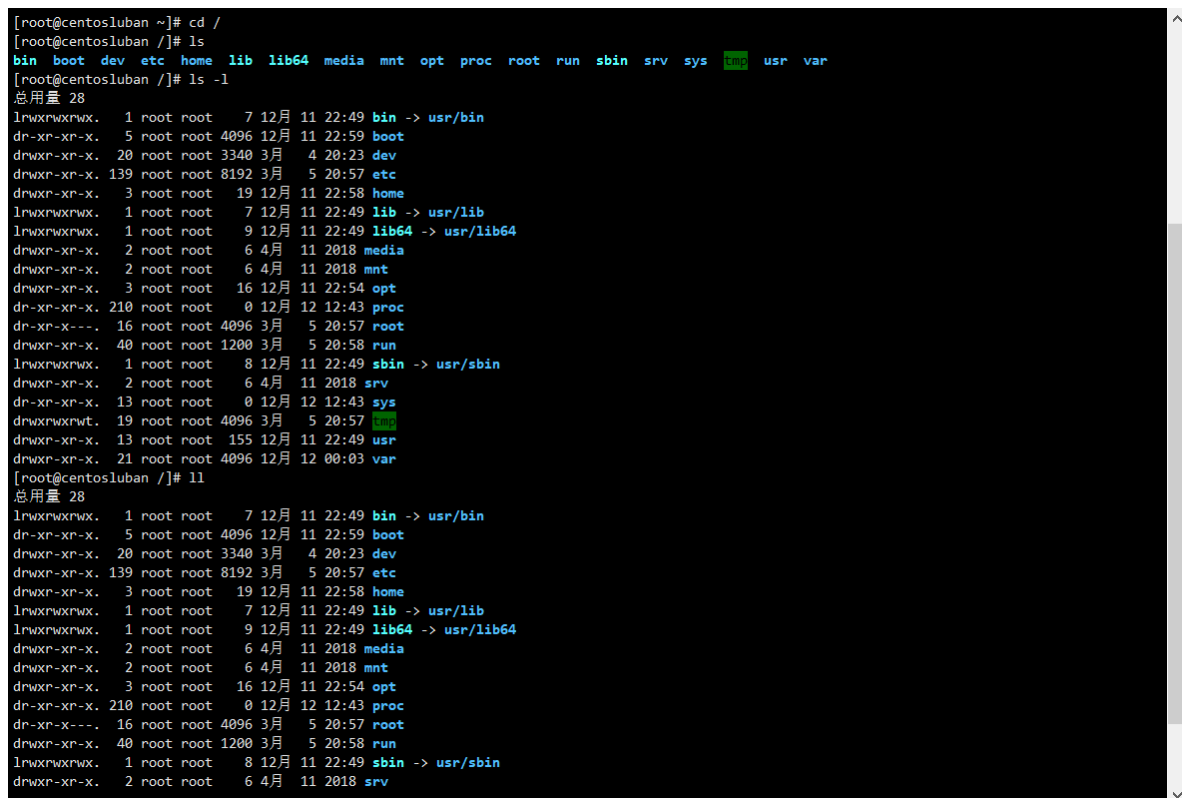
- 配置 连接的主机 点击连接
2. 连接成功



- SSH协议连接到 192.168.33.3:22 成功

4.2.2 常用命令

1. 进入根目录: `cd /`
2. 查看目录: `ls`
3. 带参数详细查看目录: `ls -l`
4. 简化命令详细查看目录: `ll`



5. 进入目录: `cd 目录名称`
6. 回到上级目录: `cd ..`

7. 创建文件夹: `mkdir` 文件夹名称

8. 创建多级文件夹: `mkdir -p a/b/c` 直接创建了3级的文件夹 a/b/c

```
[root@centosluban ~]# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban ~]# mkdir a/b/c
mkdir: 无法创建目录'a/b/c': 没有那个文件或目录
[root@centosluban ~]# mkdir -p a/b/c
[root@centosluban ~]# ls
a bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban ~]# ll
总用量 28
drwxr-xr-x. 3 root root 15 3月 5 21:16 a
lrwxrwxrwx. 1 root root 7 12月 11 22:49 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 12月 11 22:59 boot
drwxr-xr-x. 20 root root 3340 3月 4 20:23 dev
drwxr-xr-x. 139 root root 8192 3月 5 20:57 etc
drwxr-xr-x. 3 root root 19 12月 11 22:58 home
lrwxrwxrwx. 1 root root 7 12月 11 22:49 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 12月 11 22:49 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 11 2018 media
drwxr-xr-x. 2 root root 6 4月 11 2018 mnt
drwxr-xr-x. 3 root root 16 12月 11 22:54 opt
dr-xr-xr-x. 210 root root 0 12月 12 12:43 proc
dr-xr-xr-x. 16 root root 4096 3月 5 20:57 root
drwxr-xr-x. 40 root root 1200 3月 5 20:58 run
lrwxrwxrwx. 1 root root 8 12月 11 22:49 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 11 2018 srv
dr-xr-xr-x. 13 root root 0 12月 12 12:43 sys
drwxrwxrwt. 19 root root 4096 3月 5 20:57 tmp
drwxr-xr-x. 13 root root 155 12月 11 22:49 usr
drwxr-xr-x. 21 root root 4096 12月 12 00:03 var
[root@centosluban ~]#
```

9. 删除文件/目录: `rm -r` 目录 可以删除有多个级别的文件夹

```
[root@centosluban ~]# ls
a bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban ~]# rm a
rm: 无法删除'a': 是一个目录
[root@centosluban ~]# rm -r a
rm: 是否进入目录'a'? y
rm: 是否进入目录'a/b'? y
rm: 是否删除目录 'a/b/c'? y
rm: 是否删除目录 'a/b'? y
rm: 是否删除目录 'a'? y
[root@centosluban ~]# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban ~]# mkdir -p a/b/c
[root@centosluban ~]# ls
a bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban ~]# rm -rf a
[root@centosluban ~]# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban ~]#
```

- 使用 `rm` 只能删除一级目录, 不能删除多级目录
- 使用 `rm -r a/b/c` 删除多级目录时, 需要不断验证
- 使用 `rm -rf a/b/c` 直接删除多级目录, 不需要验证

10. 输出到控制台: `echo` 内容

11. 查看文件内容: `cat` 文件

```
[root@centosluban /]# echo hello world
hello world
[root@centosluban /]# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban /]# ls > first.txt
[root@centosluban /]# ls
bin boot dev etc first.txt home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban /]# cat first.txt
bin
boot
dev
etc
first.txt
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
[root@centosluban /]#
```

- `ls > first.txt` 将 `ls` 命令查询出来的结果写入到 `first.txt` 文件中
- `cat first.txt`：查看 `first.txt` 的内容

12. 使用编辑器修改文件：vi 文件

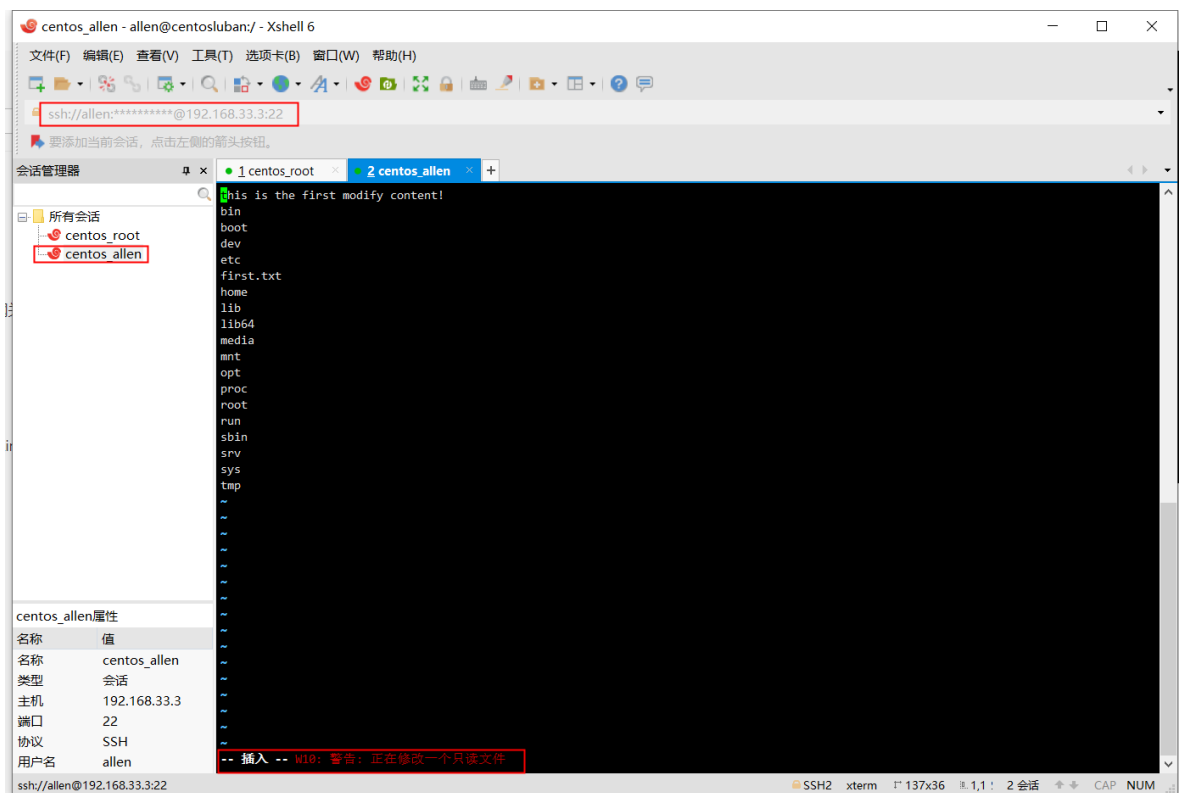
```
[root@centosluban /]# ls
bin boot dev etc first.txt home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
[root@centosluban /]# vi first.txt 进入first.txt文件进行修改
```

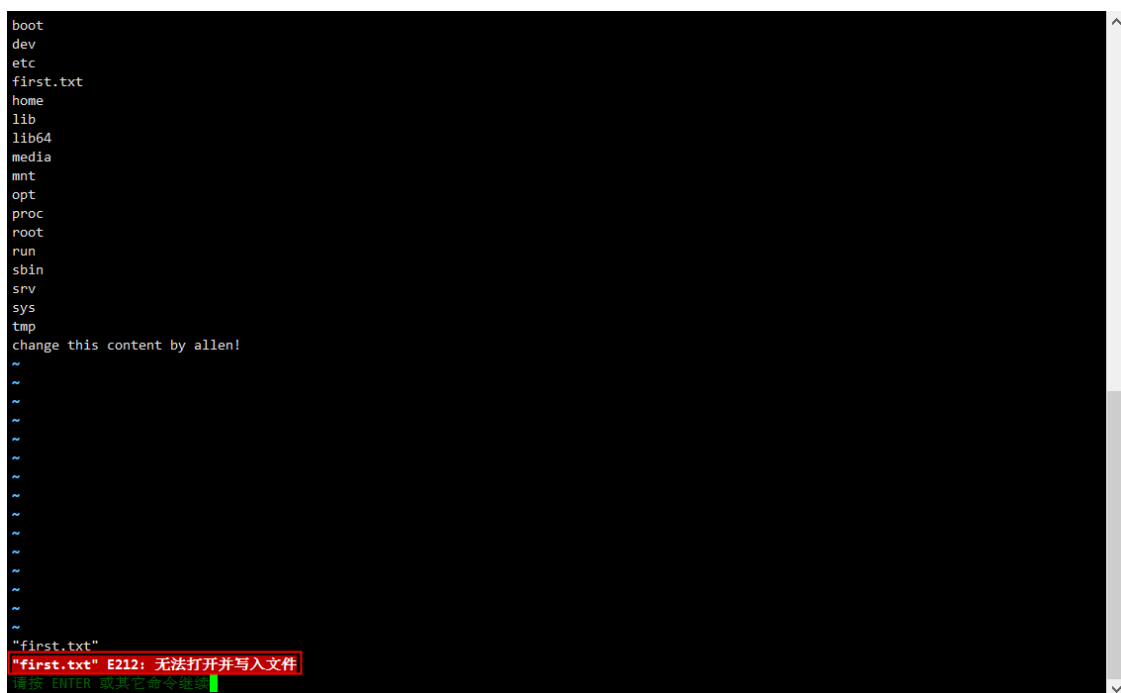
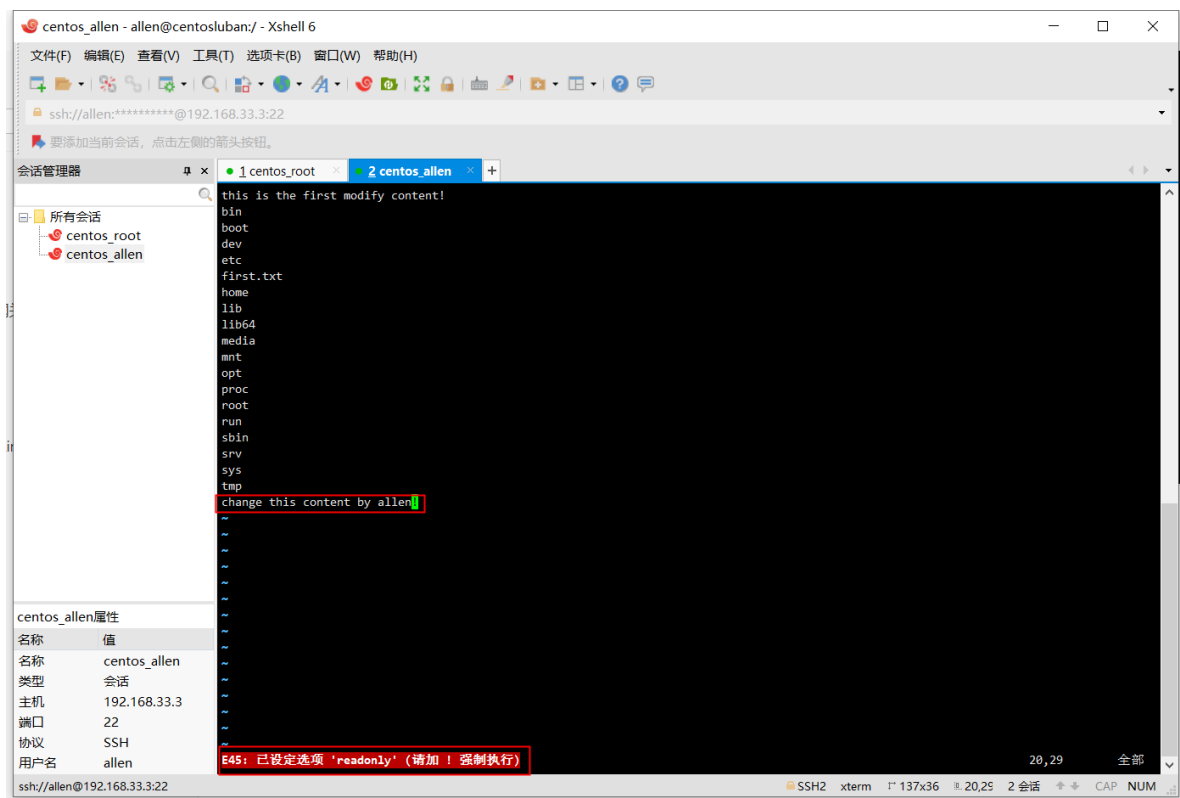


```
[root@centosluban /]# ll
总用量 32
lrwxrwxrwx. 1 root root 7 12月 11 22:49 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 12月 11 22:59 boot
drwxr-xr-x. 20 root root 3340 3月 4 20:23 dev
drwxr-xr-x. 139 root root 8192 3月 6 08:59 etc
-rw-r--r--. 1 root root 122 3月 5 21:29 first.txt
drwxr-xr-x. 4 root root 37 3月 6 08:54 home
lrwxrwxrwx. 1 root root 7 12月 11 22:49 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 12月 11 22:49 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 4月 11 2018 media
drwxr-xr-x. 2 root root 6 4月 11 2018 mnt
drwxr-xr-x. 3 root root 16 12月 11 22:54 opt
dr-xr-xr-x. 217 root root 0 12月 12 12:43 proc
dr-xr-xr-x. 16 root root 4096 3月 6 08:58 root
drwxr-xr-x. 40 root root 1200 3月 6 08:45 run
lrwxrwxrwx. 1 root root 8 12月 11 22:49 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 4月 11 2018 srv
dr-xr-xr-x. 13 root root 0 12月 12 12:43 sys
drwxrwxrwt. 19 root root 4096 3月 6 08:54 tmp
drwxr-xr-x. 13 root root 155 12月 11 22:49 usr
drwxr-xr-x. 21 root root 4096 12月 12 00:03 var
[root@centosluban /]#
```

- first.txt 文件是用 root 账号创建的，从查看的结果：
 - 该文件对于 root 账户 具有**可读、可写、不可执行** 的权限
 - 该文件对于 root 账户所属组的其余账户 具有**可读、不可写、不可执行** 的权限
 - 该文件对于 上述两种以外的其余账户 具有**可读、不可写、不可执行** 的权限

3. 通过 allen 账户尝试修改 first.txt 文件 内容





1. 添加用户对文件的修改权限: `chmod +w 文件`
2. 删除用户对文件的修改权限: `chmod -w 文件`

```
[root@centosluban /]# ll first.txt
-rw-r--r-- 1 root root 122 3月  5 21:29 first.txt
[root@centosluban /]# chmod -w first.txt
[root@centosluban /]# ll first.txt
-r--r--r-- 1 root root 122 3月  5 21:29 first.txt
[root@centosluban /]# chmod +w first.txt
[root@centosluban /]# ll first.txt
-rw-r--r-- 1 root root 122 3月  5 21:29 first.txt
[root@centosluban /]#
```

4.2.4 环境变量

1. **/etc/profile** , Linux中环境变量配置在 **profile 文件**中
2. 当使用 jdk 时, 需要在环境变量中配置 jdk 的 JAVA_HOME 等变量:

```
HOSTNAME=`/usr/bin/hostname 2>/dev/null`
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

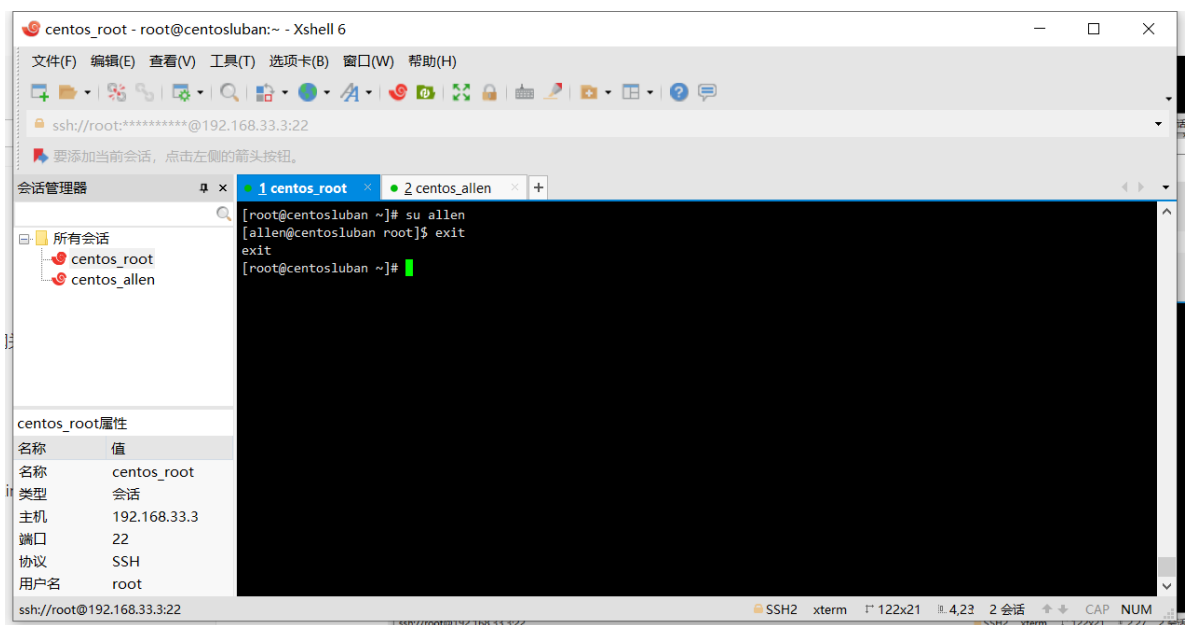
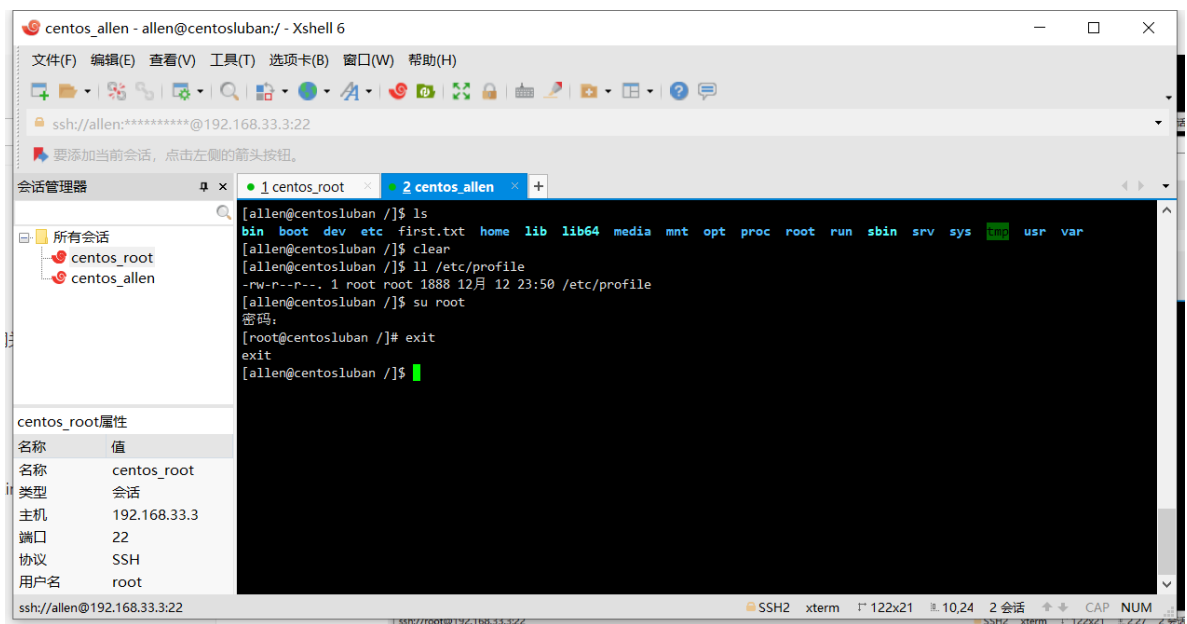
for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
export JAVA_HOME=/root/apps/jdk1.8/
export PATH=$PATH:JAVA_HOME/bin/
[root@centosluban etc]#
```

- 在 profile 文件中, 通过 export 添加变量, 中间冒号隔开

4.2.5 切换用户

1. 在使用 allen 账户时, 权限不够, 切换到 root 账户



- 通过 `su root` 命令，切换到 root 账号，要输入root账户的密码
 - 从root账户切换到普通用户时，使用 `su 用户名`，不需要使用密码
 - 使用 `exit` 命令，退出切换的用户
2. `su` 切换账号，需要知道账号密码，这不安全，所以使用 `sudo` 命令

```
[allen@centosluban ~]$ sudo vi /etc/profile

我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来不外乎这三点：

#1) 尊重别人的隐私。
#2) 输入前要先考虑(后果和风险)。
#3) 权力越大，责任越大。

[sudo] allen 的密码：
allen 不在 sudoers 文件中。此事将被报告。
[allen@centosluban ~]$ clear
[allen@centosluban ~]$ sudo vi /etc/profile
[sudo] allen 的密码：
allen 不在 sudoers 文件中。此事将被报告。
[allen@centosluban ~]$ sudo vi /etc/profile
[sudo] allen 的密码：

/etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case "${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}
```

```
# Defaults    env_keep += "HOME"

Defaults     secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##     user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
allen   ALL=(ALL)    ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL

## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users    localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
:wq
```

/etc/sudoers 配置文件中

root账户可以在任何地方执行任何命令
给 allen 账号配置同样的权限

- 使用 `sudo vi /etc/profile` 命令给 allen 账号赋予编辑 /etc/profile 文件的权限
- 使用该命令时，会有温馨提示，三句话
- 执行该命令过程中，发现 /etc/sudoers 配置文件中没有配置 allen 的 `sudo` 权限，所以执行失败，并且上报此次信息
- 通过root账号，修改 /etc/sudoers 配置文件，给 allen 添加 `sudo` 命令权限
- 使用 `sudo` 命令，只需要输入自己的账号密码