



# Linux操作系统

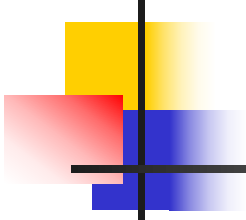
---

单位：杭州电子科技大学  
通信工程学院



# 第十章 服务器的配置

---

- 
- 
- 要完成网络配置工作，可以修改相应的配置文件、**使用网络命令或通过图形界面进行配置**。要管理好网络服务，可以使用服务器配置工具以及相应的命令启动和停止服务。



## 10.1 查看网络配置

---

linux系统中网络信息包括网络接口信息、路由信息、主机名、网络连接状态等等。

### 10.1.1 ifconfig

使用ifconfig命令查看和更改网络接口的地址和参数，格式：

```
ifconfig -interface [options] address
```

说明：

（1）interface是指定的网络接口名，如eth0,eth1,ens\*\*等；

（2）options指代如下：

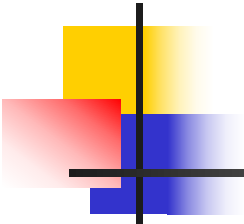
up：激活指定的网络接口卡。

down：关闭指定的网络接口。

broadcast address：设置接口的广播地址。

pointopoint：启用点对点方式。

netmask address：设置接口的子网掩码



(3) address是设置指定接口设备的IP地址。

例：显示当前系统中ens33接口的参数，如图所示。

```
lex@lex-vm:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.101  netmask 255.255.255.0  broadcast 192.168.1.255
    inet6 fe80::81b6:9e2f:5899:3edf  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:ba:2c:d1  txqueuelen 1000  (Ethernet)
    RX packets 148  bytes 53780 (53.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 167  bytes 36520 (36.5 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
lex@lex-vm:~/桌面$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.101  netmask 255.255.255.0  broadcast 192.168.1.255
    inet6 fe80::81b6:9e2f:5899:3edf  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:ba:2c:d1  txqueuelen 1000  (Ethernet)
    RX packets 488  bytes 112765 (112.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 305  bytes 49956 (49.9 KB)
    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 172  bytes 14630 (14.6 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 172  bytes 14630 (14.6 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

ens33:网卡名称

flags=4163 <UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

//UP:表示“接口已启用”。

//BROADCAST:表示“主机支持广播”。

//RUNNING:表示“接口在工作中”。

//MULTICAST:表示“主机支持多播”。

//MTU 1500(最大传输单元)为1500字节。

- `inet 192.168.1.101 netmask 255.255.255.0 broadcast 192.168.1.255`  
//IP地址,子网掩码和广播地址;
- `inet6 fe80::81b6:9e2f:5899:3edf prefixlen 64 scopeid 0x20<link>`  
//网卡的IPv6地址;
- `ether 00:0c:29:ba:2c:d1 txqueuelen 1000 (Ethernet)`  
//连接类型: Ethernet(以太网) 和硬件mac地址  
//txqueuelen:网卡设置的传送队列长度,即用来传输数据的缓冲区的存储长度。
- `RX packets 488 bytes 112765 (112.7 KB)`  
//接收正确的数据包数和数据bytes数。
- `RX errors ; RX dropped; overruns; frame`  
//分别对应着接收时错误的、丢弃的、由于速度过快而丢失的、发生frame错误而丢失的数据包数。
- TX 对应着发射时的情况, `carrier`为发生carrier错误而丢失的数据包数,collisions冲突信息包的数目。
- `lo`为网络回环或者回环接口,为虚拟网络接口,回环地址为127.0.0.1,并非真实存在,并不真实地从外界接收和发送数据包,而是在系统内部接收和发送数据包,因此虚拟网络接口不需要驱动程序。为什么会有该接口?如果包是由一个本地进程为另一个本地进程产生的,它们将通过外出链的lo接口,然后返回进入链的lo接口。



## 10.1.2 route

使用route命令查看主机路由表，如图10.3所示

说明：网关地址为\*，表示目标是本主机所属的网络，不需要路由。

```
lex@lex-vm:/etc/network$ route
Kernel IP routing table
Destination        Gateway            Genmask           Flags Metric Ref    Use Iface
default            192.168.0.1       0.0.0.0           UG      100    0      0 ens33
link-local         0.0.0.0           255.255.0.0       U        1000    0      0 ens33
192.168.0.0        0.0.0.0           255.255.255.0     U        100    0      0 ens33
```

## 10.1.3 hostname

查看系统和主机名

```
lex@lex-vm:/etc/network$ hostname
lex-vm
```



## 10.1. 4 netstat

通过netstat查看网络连接状态，显示网络连接、路由表和网络接口信息。

netstat -[选项]

-s 显示各个协议的网络统计数据。

-c 显示连续列出的网络状态

-i 显示网络接口信息表单。

-r 显示关于路由表的信息，类似于route命令。

-a 显示所有的有效连接信息

-n 显示所有已建立的有效连接

-t 显示TCP协议的连接

-u 显示UDP协议的连接

-p 显示正在使用的进程ID

例：查看当前系统所有的监听端口，如图所示。

```
maflifei@maflifei:~$ netstat -natu
激活Internet连接（服务器和已建立连接的）
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:2049             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:39143            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:39496            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:3306             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:38219            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:139              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111              0.0.0.0:*               LISTEN
tcp        0      0 127.0.1.1:53             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:46101            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:48699            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:445              0.0.0.0:*               LISTEN
tcp6       0      0 :::2049                  :::*                     LISTEN
```

## 10.2 修改网络配置

### 10.2.1 使用命令修改

#### 1、修改ens33接口的ip地址、子网掩码

```
$sudo ifconfig ens33 192.168.30.129 netmask 255.255.255.0
```

#### 2、修改默认网关

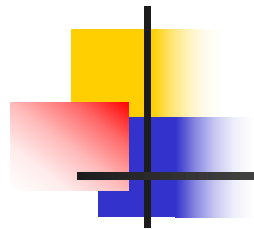
```
$sudo route add default gw 192.168.0.1
```

#### 3、修改主机名

```
$sudo hostname lex-vm1
```

```
lex@lex-vm:/etc/network$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.103 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::81b6:9e2f:5899:3edf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ba:2c:d1 txqueuelen 1000 (Ethernet)
    RX packets 7528 bytes 9618210 (9.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3302 bytes 245054 (245.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lex@lex-vm:/etc/network$ sudo ifconfig ens33 192.168.0.101 netmask 255.255.255.0
[sudo] password for lex:
lex@lex-vm:/etc/network$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.101 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::81b6:9e2f:5899:3edf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ba:2c:d1 txqueuelen 1000 (Ethernet)
    RX packets 7539 bytes 9619166 (9.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3313 bytes 246881 (246.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



### 10.2.2使用配置文件修改

使用命令的方式修改网络参数，在系统重启后会失效，要想重新启动系统后能够生效，就要修改配置文件。

#### 1、/etc/network/interfaces

修改/etc/network/interfaces配置文件，可以修改网络接口的ip地址、子网掩码、默认网关。

使用命令#sudo nano /etc/network/interfaces打开文件，并按照以下格式修改后保存：

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.30.129
```

```
netmask 255.255.255.0
```

```
gateway 192.168.0.1
```

保存后重启：

```
# sudo /etc/init.d/networking restart
```

注意：不建议使用这种方法配置ip，很容易使网卡掉线，得手动重启，然后每次都需要重新手动配置网卡。



## 2、/etc/hostname文件中

修改/etc/hostname文件中保存的主机名，系统重启后，会从此文件中读出主机名，如图所示。

```
lex@lex-vm:/etc/network$ cat /etc/hostname
lex-vm
```

## 3、/etc/resolv.conf文件

- (1) 修改/etc/resolv.conf 配置文件指定lo对应的DNS服务器，保存其域名和ip地址，文件每行以一个关键字开头，后接配置参数。（/etc/resolv.conf是/run/systemd/resolve/下面某个文件的链接，可以使用ls -l查看）
- (2) 实际的网卡ens33的域名设置是在/run/systemd/resolve/\*resolv.conf文件里。

```
lrwxrwxrwx 1 root root 39 3月 20 15:08 resolv.conf -> ../run/systemd/resolve/stub-resolv.conf
```

```
lex@lex-vm:/run/systemd/resolve$ cat resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search
# domains.
#
# Third party programs should typically not access this file directly,
# but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5)
# in a
# different way, replace this symlink by a static file or a different
# symlink.
#
# See man:systemd-resolved.service(8) for details about the supported
# modes of
# operation for /etc/resolv.conf.

nameserver 192.168.1.1
nameserver 192.168.0.1
```



#### 4、IP和主机名的映射

在Ubuntu系统中，主机名是在etc/hostname文件中的，主机名与IP的映射关系是在etc/hosts文件中的，格式如下：

```
127.0.0.1 localhost.localdomain localhost
```

```
lex@lex-vm:/etc$ cat hosts
127.0.0.1      localhost
127.0.1.1      lex-vm

# The following lines are desirable for IPv6 capable hosts
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```



## 10.3 samba服务器

---

### 10.3.1 samba服务器简介

- (1)Linux下进行资源共享有很多方式，Samba服务器就是常见的一种。主要实现在Windows 下访问linux。
- (2)Samba是在Linux和Unix系统上实现SMB协议的一个免费软件，由服务器和客户端程序构成。
- (3)SMB(Server Messages Block,信息服务块)是一种在局域网上共享文件和打印机的一种通信协议，为局域网内不同计算机之间提供文件及打印机等资源的共享服务。
- (4)SMB协议是客户机/服务器型协议，客户机通过该协议可以访问服务器上的共享文件系统、打印机及其他资源。
- (5)通过设置NetBIOS可以使samba与全世界电脑共享资源。



## 10.3 samba服务器

### 10.3.2 安装samba服务器

#### 1. 在命令行下安装samba服务器

在命令行中直接用Ubuntu提供的apt-get软件包管理工具安装samba，命令为：

\$sudo apt-get update(先更新)

\$sudo apt-get install samba cifs-utils

```
lex@lex-vm:~$ sudo apt-get update
[sudo] password for lex:
命中:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy InRelease
获取:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates InRelease [115 kB]
获取:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-backports InRelease [101 kB]
获取:4 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-security InRelease [110 kB]
获取:5 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/main Sources [157 kB]
获取:6 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/universe Sources [46.0 kB]
获取:7 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/restricted Sources [14.2 kB]
获取:8 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/main amd64 Packages [468 kB]
获取:9 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/main i386 Packages [205 kB]
获取:10 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/main Translation-en [1
获取:11 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/main amd64 DEP-11 Meta
获取:12 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/main amd64 c-n-f Meta
获取:13 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/universe i386 Packages
获取:14 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/universe amd64 Package
获取:15 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/universe Translation-e
获取:16 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/universe amd64 DEP-11
获取:17 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/universe amd64 c-n-f M
获取:18 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-updates/multiverse amd64 DEP-1
获取:19 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-backports/universe amd64 DEP-1
获取:20 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy-security/universe Sources [20.

lex@lex-vm:~$ sudo apt-get install samba cifs-utils
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
libboost-program-options1.71.0
使用'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件:
attr ibverbs-providers libcephfs2 libibverbs1 librados2 librdmacm1
liburing1 python3-crypto python3-dnspython python3-ecdsa
python3-gpg python3-markdown python3-pycryptodome python3-pygments
python3-samba python3-tdb samba-common samba-common-bin
samba-dsdb-modules samba-vfs-modules tdb-tools
建议安装:
keyutils smbclient winbind python-markdown-doc python-pygments-doc
ttf-bitstream-vera bind9 bind9utils ctdb ldb-tools ntp | chrony
smbldap-tools heimdal-clients
下列【新】软件包将被安装:
attr cifs-utils ibverbs-providers libcephfs2 libibverbs1 librados2
librdmacm1 liburing1 python3-crypto python3-dnspython
python3-ecdsa python3-gpg python3-markdown python3-pycryptodome
python3-pygments python3-samba python3-tdb samba samba-common
samba-common-bin samba-dsdb-modules samba-vfs-modules tdb-tools
升级了 0 个软件包，新安装了 23 个软件包，要卸载 0 个软件包，有 208 个软件包未被升级。
```

## 10.3.3 配置samba服务器

### 1. 建立samba共享文件夹

为samba服务器创建共享文件夹：/home/lex/share，且该文件夹的权限为对所有用户可读可写可运行。

```
lex@lex-vm:~$ ls -l
total 64
drwxr-xr-x 2 lex lex 4096 5月 12 09:14 公共的
drwxr-xr-x 2 lex lex 4096 5月 12 09:14 模板
drwxr-xr-x 2 lex lex 4096 5月 12 09:14 视频
drwxr-xr-x 2 lex lex 4096 5月 12 09:14 图片
drwxr-xr-x 2 lex lex 4096 5月 12 09:14 文档
drwxr-xr-x 3 lex lex 4096 5月 13 10:22 下载
drwxr-xr-x 2 lex lex 4096 5月 12 09:14 音乐
drwxr-xr-x 3 lex lex 4096 5月 13 10:30 桌面
drwxrwxr-x 3 lex lex 4096 5月 13 10:24 eclipse
drwxrwxr-x 4 lex lex 4096 5月 13 12:00 eclipse-workspace
drwxrwxr-x 7 lex lex 4096 5月 13 09:13 lesson
-rw-rw-r-- 1 lex lex 31 5月 12 09:13 lessonbk
drwxrwxr-x 4 lex lex 4096 5月 19 20:46 prac
drwxrwxr-x 2 lex lex 4096 5月 12 08:56 prac7_1_2008
drwxrwxrwx 2 lex lex 4096 5月 23 08:10 share
-rw-rw-r-- 1 lex lex 138 5月 23 08:12 tmp
```





## 2. 创建一个samba专用账户

为了samba服务器的安全，需要建立一个专用账户，使用命令smbpasswd，

smbpasswd主要作用是为用户创建samba用户，格式如下：

smbpasswd -a 新建用户

创建的samba用户必须在系统用户中存在。否则samba找不到系统用户则创建失败。

-d 冻结用户，就是这个用户不能在登录了

-e 恢复用户，解冻用户，让冻结的用户可以在使用

-n 把用户的密码设置成空。

## 2. 创建一个samba专用账户

### 创建samba账户步骤

(1) 利用useradd/adduser先创建系统用户，**没有这步后面直接生成smb用户时会出错。**

(2) 利用smbpasswd -a 生成smb用户。

```
lex@lex-vm:~$ sudo adduser samba_lex
Adding user `samba_lex' ...
Adding new group `samba_lex' (1005) ...
Adding new user `samba_lex' (1017) with group `samba_lex' ...
Creating home directory `/home/samba_lex' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for samba_lex
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
```

```
lex@lex-vm:~$ sudo smbpasswd -a samba2
New SMB password:
Retype new SMB password:
Failed to add entry for user samba2.
lex@lex-vm:~$ tail -n 3 /etc/passwd
lex102:x:1015:1015::/home/lex102:/bin/sh
lex103:x:1016:1016::/home/lex103:/bin/sh
samba_lex:x:1017:1005:,,,:/home/samba_lex:/bin/bash
```

```
lex@lex-vm:~$ sudo pdbedit -L
samba_lex:1017:
```

```
lex@lex-vm:~$ tail -n 3 /etc/passwd
lex102:x:1015:1015::/home/lex102:/bin/sh
lex103:x:1016:1016::/home/lex103:/bin/sh
samba_lex:x:1017:1005:,,,:/home/samba_lex:/bin/bash
```

使用pdbedit命令来查看系统里的samba账户。

### 3.配置Samba服务器

单击samba的图标，启动samba的配置界面，如果不能启动，安装图形界面。

#### (1) 图形界面配置Samba服务器

①安装Ubuntu samba图形管理界面，命令为：

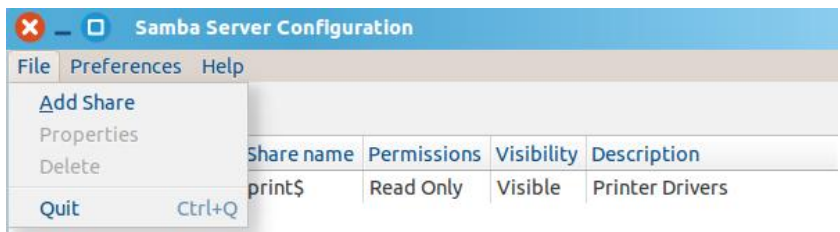
`#sudo apt-get install system-config-samba`

②启动samba图形管理界面，命令为：`#sudo system-config-samba`。

启动图形界面后可见如图10.11所示内容：



③在配置界面的菜单栏中，选择File→Add Share。如图所示：



④弹出Create Samba Share的对话框，浏览及填写相关内容。如图所示：

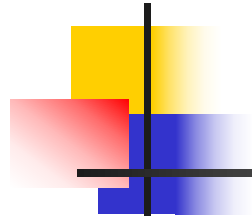


⑤选择共享用户，如图所示，选择在前面新建的专用账户samba，点击OK完成配置。



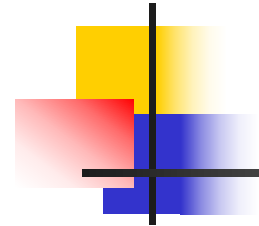
配置完成后，会出现一个新的共享信息。





如果没有出现图形界面，可以直接编辑/etc/samba/smb.conf 文件，内容如下：

```
[share]
comment = samba with web static server
path = /home/lex/share
writeable = yes
available = yes
browseable = yes
public = yes
valid users = samba_lex
```



**browseable = yes/no**

说明：browseable用来指定该共享是否可以浏览。

**writable = yes/no**

说明：writable用来指定该共享路径是否可写。

**available = yes/no**

说明：available用来指定该共享资源是否可用。

**valid users = 允许访问该共享的用户**

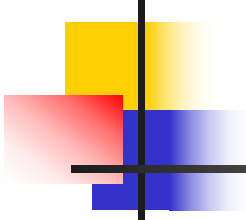
说明：valid users用来指定允许访问该共享资源的用户。

例如：valid users = bobyuan, @bob, @tech（多个用户或者组中间用逗号隔开，如果要加入一个组就用“@+组名”表示。）

**public = yes/no**

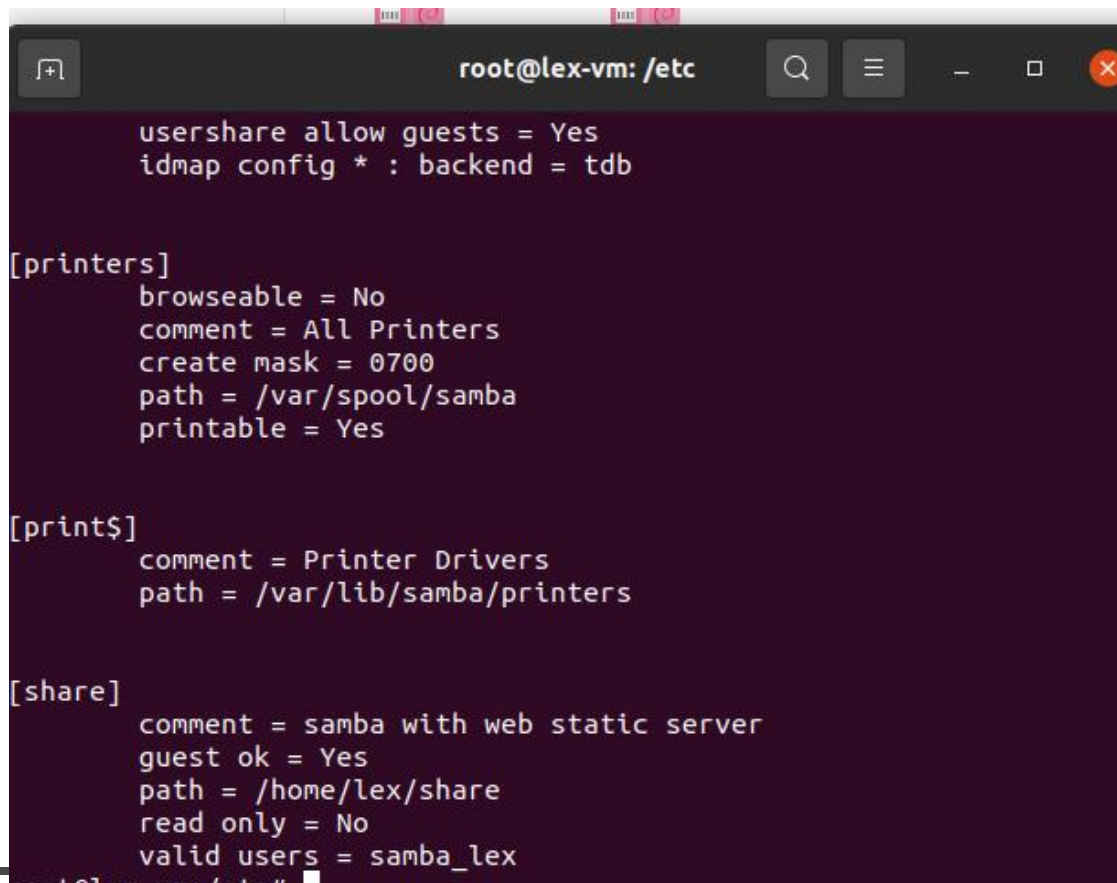
说明：public用来指定该共享是否允许guest账户访问。

---



测试samba服务是否配置完成

1、testparm(test parameter)指令简单测试samba的配置文件。



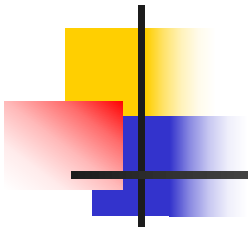
```
root@lex-vm: /etc
usershare allow guests = Yes
idmap config * : backend = tdb

[printers]
  browseable = No
  comment = All Printers
  create mask = 0700
  path = /var/spool/samba
  printable = Yes

[print$]
  comment = Printer Drivers
  path = /var/lib/samba/printers

[share]
  comment = samba with web static server
  guest ok = Yes
  path = /home/lex/share
  read only = No
  valid users = samba_lex
```





## 5. 启动与关闭Samba服务器

### (1) 重启samba服务

```
#sudo /etc/init.d/smbd restart
```

### (2) 关闭samba服务

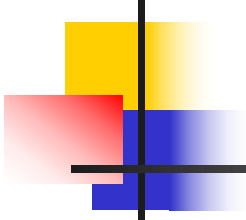
```
#sudo /etc/init.d/smbd stop
```

### (3) 启动samba服务

```
#sudo /etc/init.d/smbd start
```

### (4) 显示samba服务是否启动

```
#sudo /etc/init.d/smbd status
```

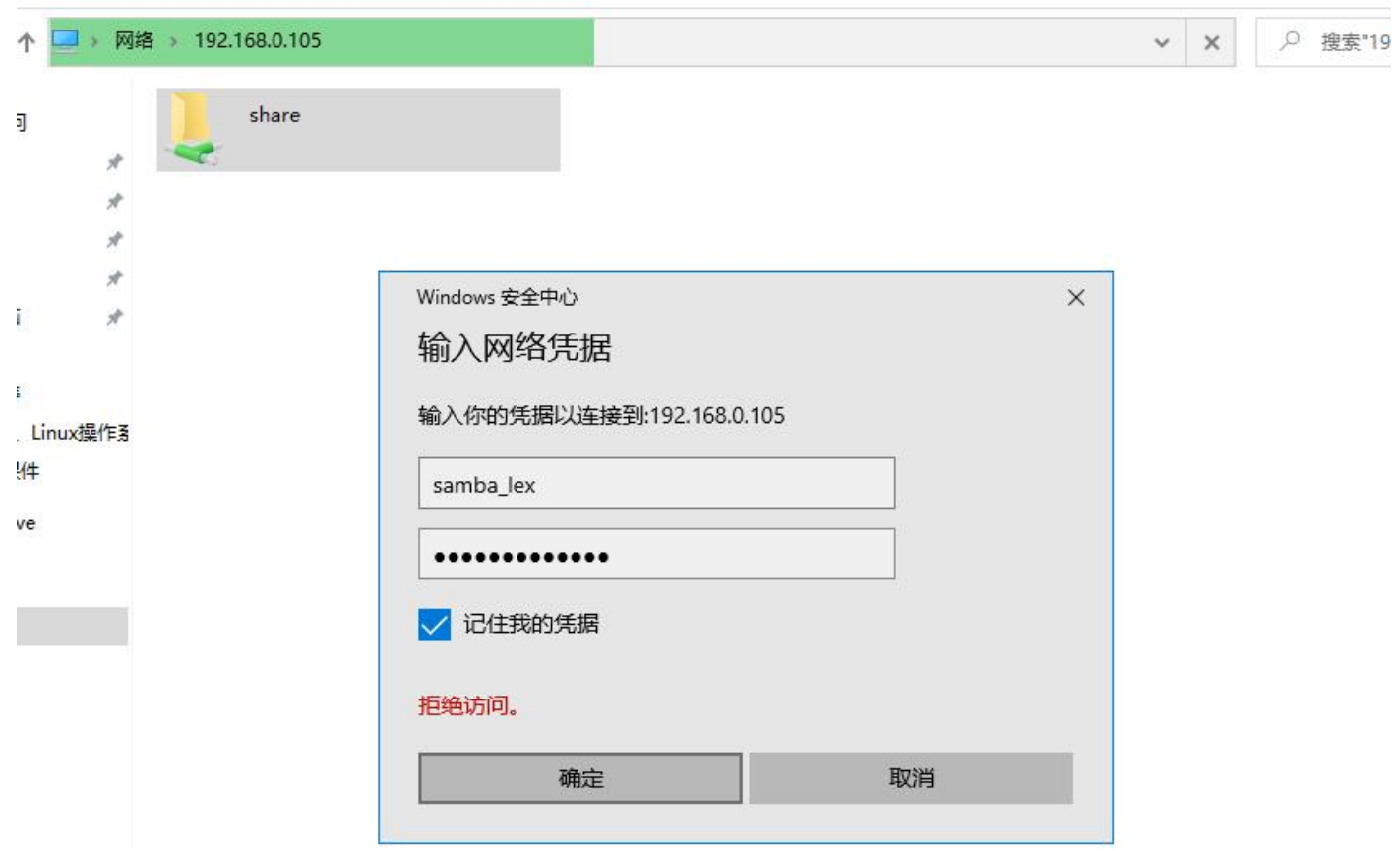


```
root@lex-vm:/etc/init.d# sudo /etc/init.d/smbd restart
Restarting smbd (via systemctl): smbd.service.
root@lex-vm:/etc/init.d# sudo /etc/init.d/smbd stop
Stopping smbd (via systemctl): smbd.service.
root@lex-vm:/etc/init.d# sudo /etc/init.d/smbd start
Starting smbd (via systemctl): smbd.service.
root@lex-vm:/etc/init.d# sudo /etc/init.d/smbd status
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-05-23 11:19:47 CST; 5s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Process: 16559 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
 Main PID: 16563 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 4 (limit: 2275)
    Memory: 7.9M
    CGroup: /system.slice/smbd.service
            └─16563 /usr/sbin/smbd --foreground --no-process-group
              └─16565 /usr/sbin/smbd --foreground --no-process-group
                └─16566 /usr/sbin/smbd --foreground --no-process-group
                  └─16567 /usr/sbin/smbd --foreground --no-process-group

5月 23 11:19:47 lex-vm systemd[1]: Starting Samba SMB Daemon...
5月 23 11:19:47 lex-vm systemd[1]: Started Samba SMB Daemon.
```

## 6. 登录Samba服务器

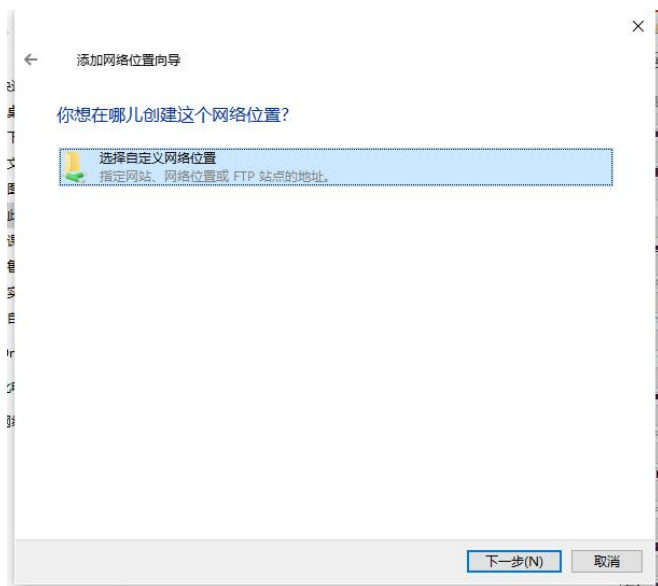
/home/lex/share 文件夹下,使用Windows 10 作为客户端，访问samba服务器：开始→运行，输入虚拟机的 IP地址，登录之后，可看到共享文件夹share，



## 6. 登录Samba服务器

或者按照如下步骤进行访问:

此电脑-添加一个网络位置-选择自定义-输入IP地址和共享文件夹名即可。



Internet 地址或网络地址(A):

\\192.168.0.105\share

浏览(R)...

第三步 2/3

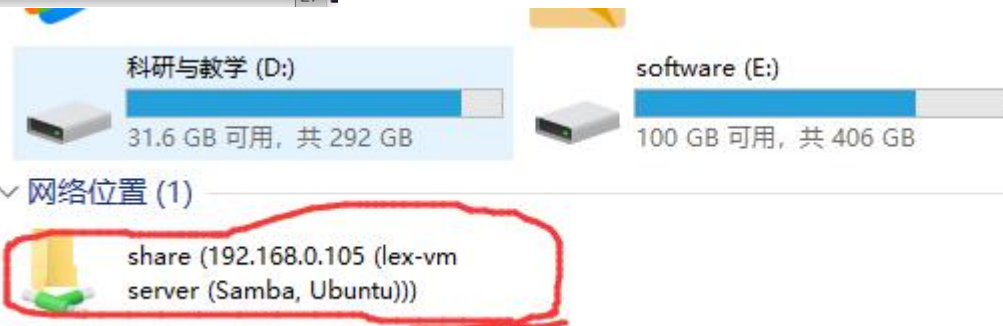
这个网络位置的名称是什么?

为这个快捷方式创建一个帮助你识别这个网络位置的名称:

\\192.168.0.105\share。

请键入该网络位置的名称(T):

share (192.168.0.105 (lex-vm server (Samba, Ubuntu)))



在Linux下的测试验证 samba 先在Linux下安装客户端  
# smbclient //服务器地址/共享文件夹 -U 访问共享的用户名  
# smbclient //192.168.0.105/share -U samba\_lex

```
root@lex-vm:/home/lex/share# apt-get install smbclient
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
  libboost-program-options1.71.0
使用 'apt autoremove' 来卸载它(它们)。
建议安装:
  heimdal-clients
下列【新】软件包将被安装:
  smbclient
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级。
需要下载 405 kB 的归档。
解压缩后会消耗 2,102 kB 的额外空间。
获取:1 http://mirrors.aliyun.com/ubuntu groovy-updates/main amd64 smbclient amd64 2:4.12.5+dfsg-3ub
untu4.3 [405 kB]
已下载 405 kB，耗时 1秒 (776 kB/s)
正在选中未选择的软件包 smbclient。
(正在读取数据库 ... 系统当前共安装有 205891 个文件和目录。)
准备解压 .../smbclient_2%3a4.12.5+dfsg-3ubuntu4.3_amd64.deb ...
正在解压 smbclient (2:4.12.5+dfsg-3ubuntu4.3) ...
正在设置 smbclient (2:4.12.5+dfsg-3ubuntu4.3) ...
正在处理用于 man-db (2.9.3-2) 的触发器 ...
```

```
root@lex-vm:/home/lex/share# smbclient //192.168.0.105/share -U samba_lex
Enter WORKGROUP\samba_lex's password:
Try "help" to get a list of possible commands.
smb: \> ?
?
```

allinfo	altname	archive	backup
cancel	case_sensitive	cd	chmod
close	del	deltree	dir
echo	exit	get	getfacl
hardlink	help	history	iosize
link	lock	lowercase	ls
mask	md	mget	mkdir
mput	newer	notify	open
posix_encrypt	posix_open	posix_mkdir	posix_rmdir
posix_unlink	posix_whoami	prompt	put
q	queue	quit	readlink
recurse	reget	rename	reput
rmdir	showacls	setea	setmode
stat	symlink	tar	tarmode
translate	unlock	volume	vuid
logon	listconnect	showconnect	tccon
tid	utimes	logoff	..



## 10.4 NFS服务器

---

### 10.4.1 NFS简介

- NFS是Network File System 的简写，即网络文件系统，由SUN公司开发，目前已经成为文件服务的一种标准（RFC1904，RFC1813）。它允许网络中的计算机之间通过TCP/IP网络共享资源。在NFS的应用中，本地NFS的客户端应用可以透明地读写位于远端NFS服务器上的文件，就像访问本地文件一样。NFS允许一个系统在网络上与他人共享目录和文件，文件就像位于本地硬盘一样，操作方便，**主要用在linux 之间的文件共享。**





## 10.4.2 NFS 应用

NFS 有很多实际应用，例如：

- 1. 多个机器共享一台CDROM或者其他设备。这对于在多台机器中安装软件来说更加方便。
- 2. 在大型网络中，配置一台中心 NFS 服务器用来放置所有用户的home目录，用户不管在哪台工作站上登录，总能得到相同的home目录。
- 3. 不同客户端可在NFS上观看影视文件，节省本地空间。
- 4. 在客户端完成的工作数据，可以备份保存到NFS服务器上用户自己的路径下。

NFS是运行在应用层的协议。随着NFS多年的发展和改进，NFS既可以用于局域网也可以用于广域网，且与操作系统和硬件无关，可以在不同的计算机或系统上运行。

## 10.4.3 NFS服务器的安装与配置

### 1. NFS的安装前准备

新建用于nfs文件共享的文件夹/home/lex/nfs，并修改权限，以便让其他用户访问，命令如下：

```
#sudo mkdir /home/lex/nfs
```

```
#sudo chmod 777 /home/lex/nfs
```

### 2. NFS的安装

ubuntu中默认没有安装nfs，nfs有客户端和服务端，只安装NFS服务端就可以，命令如下：

```
#sudo apt-get install nfs-kernel-server
```

```
lex@lex-vm:~$ sudo apt-get install nfs-kernel-server
[sudo] password for lex:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  libboost-program-options1.71.0
使用'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件：
  keyutils nfs-common rpcbind
建议安装：
  open-iscsi watchdog
下列【新】软件包将被安装：
  keyutils nfs-common nfs-kernel-server rpcbind
升级了 0 个软件包，新安装了 4 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级。
需要下载 394 kB 的归档。
解压后会消耗 1,571 kB 的额外空间。
您希望继续执行吗？ [Y/n] y
获取:1 http://mirrors.aliyun.com/ubuntu groovy/main amd64 rpcbind amd64 1.2.5-9 [45.0 kB]
获取:2 http://mirrors.aliyun.com/ubuntu groovy/main amd64 keyutils amd64 1.6.1-2ubuntu1 [45.7
kB]
获取:3 http://mirrors.aliyun.com/ubuntu groovy/main amd64 nfs-common amd64 1:1.3.4-2.5ubuntu6
[204 kB]
获取:4 http://mirrors.aliyun.com/ubuntu groovy/main amd64 nfs-kernel-server amd64 1:1.3.4-2.5u
buntu6 [99.2 kB]
已下载 394 kB，耗时 1秒 (647 kB/s)
```





### 3. 配置exports文件

编辑/etc/exports文件，添加共享的目录及权限：

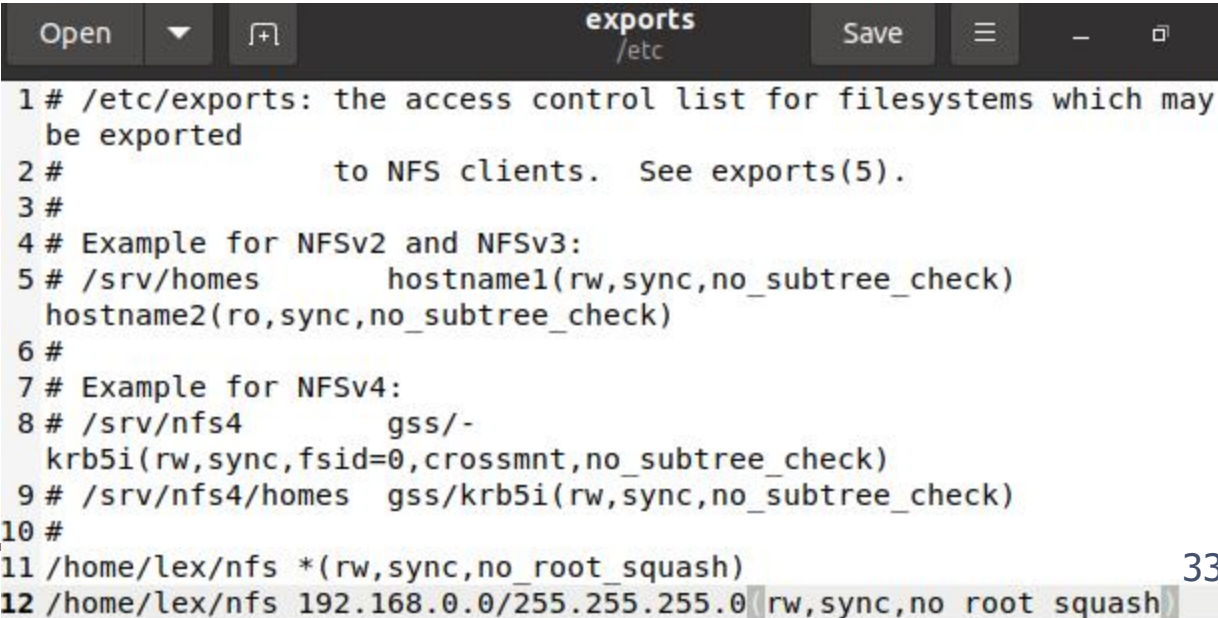
(1) 打开/etc/exports: `#sudo gedit /etc/exports`

(2) 在该文件中添加如下两行：

`/home/lex/nfs *(rw,sync,no_root_squash)` 定义要共享的目录及访问目录的权限

`/home/lex/nfs 192.168.0.0/255.255.255.0(rw,sync,no_root_squash)` 定义对目录访问的机器的限制。

例：添加/home/lex/nfs目录，并指定可以访问的网段



```
Open ▼ [icon] exports /etc Save [icon] [icon]
1 # /etc/exports: the access control list for filesystems which may
  be exported
2 #
  to NFS clients.  See exports(5).
3 #
4 # Example for NFSv2 and NFSv3:
5 # /srv/homes      hostname1(rw,sync,no_subtree_check)
  hostname2(ro,sync,no_subtree_check)
6 #
7 # Example for NFSv4:
8 # /srv/nfs4      gss/-
  krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
9 # /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
10 #
11 /home/lex/nfs *(rw,sync,no_root_squash)
12 /home/lex/nfs 192.168.0.0/255.255.255.0(rw,sync,no_root_squash)
```



说明：

- (1) /home/lex/nfs 是要共享的目录。
- (2) \* 允许所有的网段访问
- (3) ro: 共享目录只读; rw: 共享目录可读可写;
- (4) Sync:同步写入数据到内存与硬盘中。
- (5) no\_root\_squash: 如果登录NFS主机使用共享目录的使用者是root, 那么对于这个共享的目录来说, 它具有root的权限。
- (6) root\_squash: 当登录NFS主机使用共享目录的使用者是root时, 其权限将被转换成为匿名使用者, 通常它的UID与GID都会变成nobody身份。
- (7) 指定范围的主机192.168.0.0/255.255.255.0 这个网段的主机, 才能访问服务器的共享文件, 也可以指定单个主机或使用通配符\*和? 指定满足条件的主机。

#### 4. 启动rpcbind服务

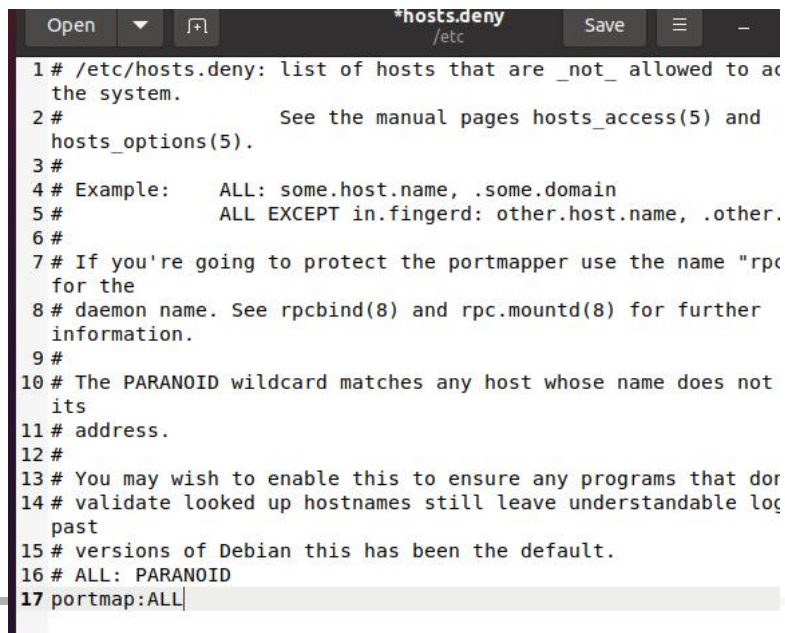
nfs是一个RPC程序，使用它前，需要映射好端口，这里只需要启动该服务就可以，命令如下：

#sudo /etc/init.d/rpcbind start

```
lex@lex-vm:/etc$ sudo /etc/init.d/rpcbind start
Starting rpcbind (via systemctl): rpcbind.service.
```

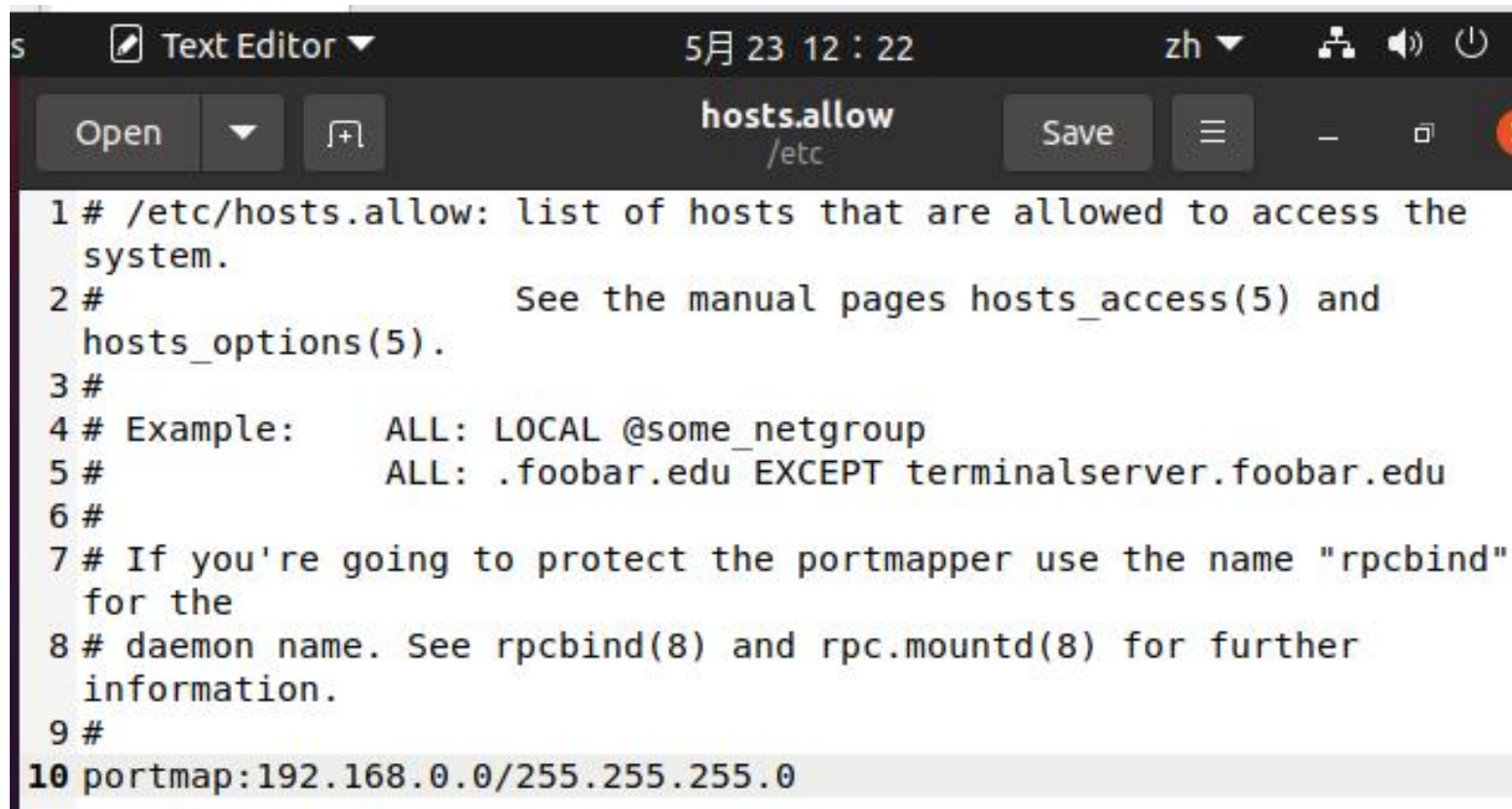
#### 5. 配置host.allow和host.deny文件

(1) 首先使用/etc/hosts.deny配置文件禁止任何主机能够和NFS服务器建立连接，在文件中添加：portmap:ALL。

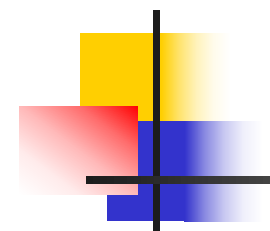


```
*hosts.deny
/etc
Open Save
1 # /etc/hosts.deny: list of hosts that are _not_ allowed to ac
the system.
2 # See the manual pages hosts_access(5) and
hosts_options(5).
3 #
4 # Example: ALL: some.host.name, .some.domain
5 # ALL EXCEPT in.fingerd: other.host.name, .other.
6 #
7 # If you're going to protect the portmapper use the name "rpc
for the
8 # daemon name. See rpcbind(8) and rpc.mountd(8) for further
information.
9 #
10 # The PARANOID wildcard matches any host whose name does not
its
11 # address.
12 #
13 # You may wish to enable this to ensure any programs that dor
14 # validate looked up hostnames still leave understandable log
past
15 # versions of Debian this has been the default.
16 # ALL: PARANOID
17 portmap:ALL
```

(2) 然后在etc/hosts.allow配置文件中允许哪些主机能够和NFS服务器建立连接，在文件中添加：portmap:192.168.0.0/255.255.255.0，如图10.23所示。



```
1 # /etc/hosts.allow: list of hosts that are allowed to access the
   system.
2 #
   See the manual pages hosts_access(5) and
   hosts_options(5).
3 #
4 # Example:      ALL: LOCAL @some_netgroup
5 #
   ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
6 #
7 # If you're going to protect the portmapper use the name "rpcbind"
   for the
8 # daemon name. See rpcbind(8) and rpc.mountd(8) for further
   information.
9 #
10 portmap:192.168.0.0/255.255.255.0
```



## 6. 重启 NFS、rpcbind服务.

在配置完成后需要重启NFS和rpcbind服务，重启的命令如下：

```
#sudo /etc/init.d/nfs-kernel-server restart
```

```
#sudo /etc/init.d/rpcbind restart
```

显示NFS服务是否运行，命令如下：

```
# #sudo /etc/init.d/nfs-kernel-server status
```

如果执行结果为nfs running ,表示nfs在运行，否则说明nfs有问题，没有启动。

注意：如果nfs没有启动，重新启动机器。

```
lex@lex-vm:/etc$ sudo /etc/init.d/nfs-kernel-server restart
Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
lex@lex-vm:/etc$ sudo /etc/init.d/rpcbind restart
Restarting rpcbind (via systemctl): rpcbind.service.
lex@lex-vm:/etc$ sudo /etc/init.d/nfs-kernel-server status
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sun 2021-05-23 12:23:09 CST; 33s ago
     Process: 4959 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
     Process: 4960 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
    Main PID: 4960 (code=exited, status=0/SUCCESS)

5月 23 12:23:08 lex-vm systemd[1]: Starting NFS server and servic....
5月 23 12:23:08 lex-vm exportfs[4959]: exportfs: /etc/exports [1]...s".
5月 23 12:23:08 lex-vm exportfs[4959]: Assuming default behavio...').
5月 23 12:23:08 lex-vm exportfs[4959]: NOTE: this default has c...0.x
5月 23 12:23:08 lex-vm exportfs[4959]: exportfs: /etc/exports [2]...s".
5月 23 12:23:08 lex-vm exportfs[4959]: Assuming default behavio...').
5月 23 12:23:08 lex-vm exportfs[4959]: NOTE: this default has c...0.x
5月 23 12:23:09 lex-vm systemd[1]: Finished NFS server and services.
Hint: Some lines were ellipsized, use -l to show in full.
```

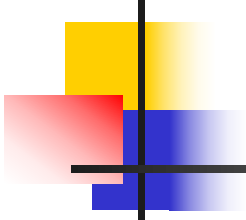




## 7. showmount命令

在NFS服务器上使用showmount命令显示NFS服务器的输出清单（也称为共享目录列表）如果配置正确，则执行结果，如图所示。

```
lex@lex-vm:/etc$ showmount -e  
Export list for lex-vm:  
/home/lex/nfs (everyone)
```



## 10.4.4客户端访问NFS服务

客户端在访问共享目录前，需要将共享目录挂载到本地目录上，挂载命令的格式：

#sudo mount -t nfs NFS服务器的ip地址:共享目录 本地目录

### 1、本地挂载共享目录

将共享目录/home/lex/nfs挂载到 /mnt下，

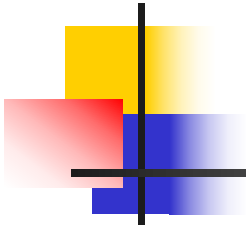
mount -t nfs 192.168.146.128（虚拟机的IP）:/home/lex/nfs /mnt/nfsdir 如下图所示。

```
root@lex-vm:/mnt# mount -t nfs 192.168.0.105:/home/lex/nfs /mnt/nfsdir
root@lex-vm:/mnt# ls
nfsdir  sdb1  sdb2  sdb5  usb
root@lex-vm:/mnt# cd nfsdir
root@lex-vm:/mnt/nfsdir# touch /home/lex/nfs/test1.lex
root@lex-vm:/mnt/nfsdir# ls
test1.lex
root@lex-vm:/mnt/nfsdir#
```

运行 df 查看结果，如图所示：

```
test1.lex
root@lex-vm:/mnt/nfsdir# df
df: /root/.cache/doc: Operation not permitted
Filesystem                1K-blocks      Used Available Use% Mounted on
tmpfs                      200196        3680    196516   2% /run
/dev/sda3                 19991152 10873212    8079400  58% /
tmpfs                     1000980          0    1000980   0% /dev/shm
tmpfs                      5120           4     5116   1% /run/lock
tmpfs                      4096           0     4096   0% /sys/fs/cgroup
/dev/sdb1                 4062912      8260    3844940   1% /myquota
/dev/sda2                 524272       8020    516252   2% /boot/efi
tmpfs                      200196        108    200088   1% /run/user/1000
192.168.0.105:/home/lex/nfs 19991296 10873344    8079616  58% /mnt/nfsdir
```





---

关闭NFS，不能挂载

```
malimei@malimei-virtual-machine:~$ sudo /etc/init.d/nfs-kernel-server stop
[sudo] malimei 的密码:
[ ok ] Stopping nfs-kernel-server (via systemctl): nfs-kernel-server.service.
malimei@malimei-virtual-machine:~$ sudo mount -t nfs 192.168.157.128:/home/malim
ei/nfs /mnt
```



## 2. 其它主机挂载共享目录

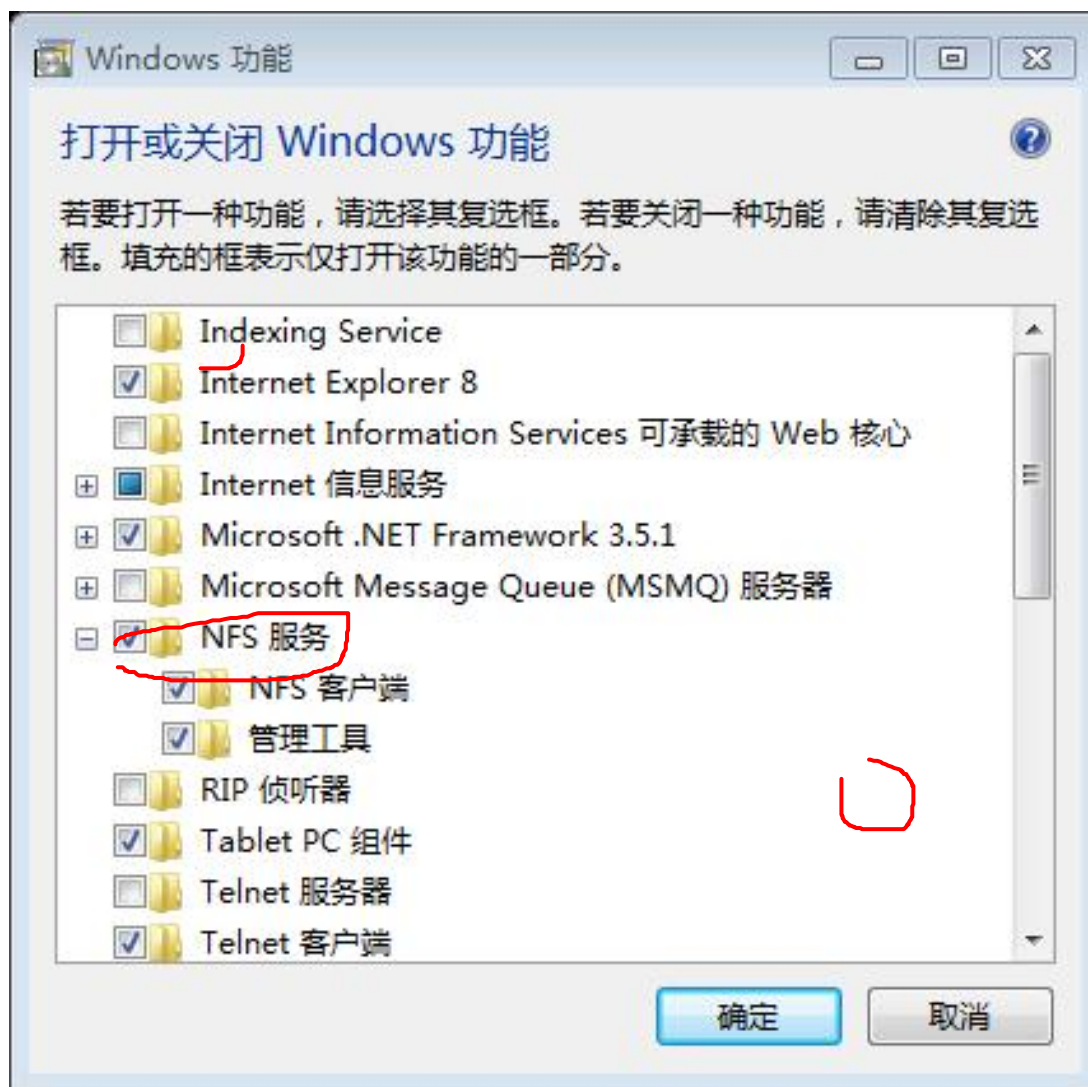
将共享目录挂载到其他主机中的一个目录上，比如与NFS服务器在同一局域网中的一台主机，在这台主机上挂载共享目录。挂载后查看该目录的内容与NFS服务器上共享目录一致。

下面以主机Win 10 系统(需企业版本，家庭版需升级或第三方软件)下挂载共享目录。

### (1) 主机win10系统开启NFS服务

主机win10系统，系统内装虚拟机ubuntu20\*\*\*\*，开启NFS客户端程序；开启方式如下：在设置-应用-找到“程序和功能”->打开或关闭windows 功能，选择NFS服务，确定。

选择“NFS服务”，然后确定。



## (2) 虚拟机下Ubuntu 的网络适配器采用“NAT模式”



(3) Ubuntu NFS配置如前所述，不再重复。启动NFS服务器：

```
#sudo /etc/init.d/nfs-kernel-server restart
```

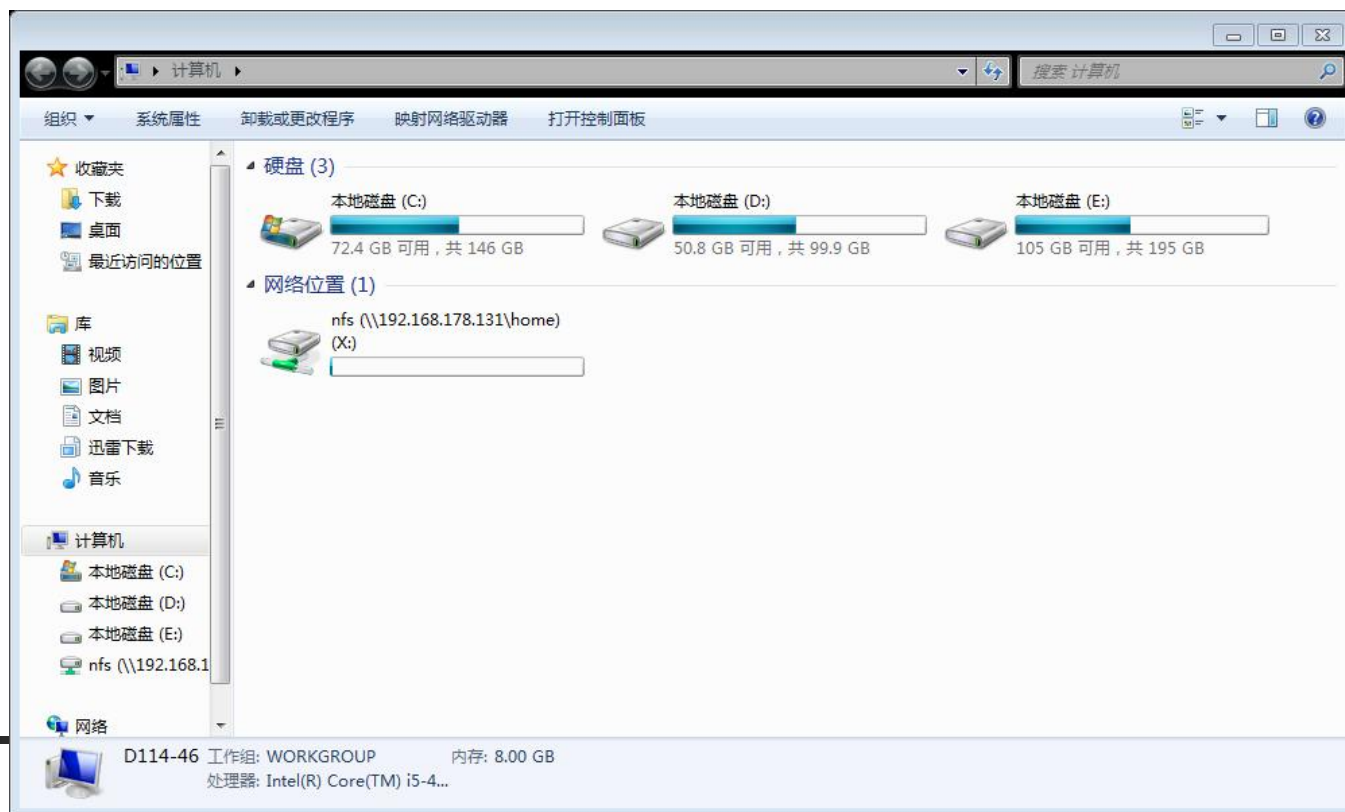
```
#sudo /etc/init.d/rpcbind restart
```

用 netstat -lt 来查看NFS服务器的启动情况：如图所示

```
Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
lex@lex-vm:~/桌面$ sudo /etc/init.d/rpcbind restart
Restarting rpcbind (via systemctl): rpcbind.service.
lex@lex-vm:~/桌面$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:58443           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:40619           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:netbios-ssn     0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:sunrpc           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:51573              0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh              0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:microsoft-ds       0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:nfs                  0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:44519                 0.0.0.0:*               LISTEN
tcp6       0      0 [::]:netbios-ssn       [::]:*                  LISTEN
tcp6       0      0 [::]:45453              [::]:*                  LISTEN
tcp6       0      0 [::]:sunrpc             [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:ipp      [::]:*                  LISTEN
tcp6       0      0 [::]:58233              [::]:*                  LISTEN
tcp6       0      0 [::]:32989              [::]:*                  LISTEN
tcp6       0      0 [::]:50941              [::]:*                  LISTEN
tcp6       0      0 [::]:microsoft-ds       [::]:*                  LISTEN
tcp6       0      0 [::]:nfs                 [::]:*                  LISTEN
lex@lex-vm:~/桌面$
```

同时，在主机win7系统的“计算机”中，可看到如下图所示：  
这样就可以通过主机NFS客户端来同步虚拟机ubuntu中的NFS服务器。

(4) 在主机win10上，打开命令提示符，用  
`mount 192.168.0.105:/home/malimei/nfs X:`  
来挂载ubuntu 14.04的NFS服务器，192.168.0.105（虚拟机的IP）是Ubuntu IP  
地址。





1、用途方面的区别:samba是DEC开始开发的，用于不同的Unix机器间进行资源共享。后来基于tcp重新开发后，效率有了很大提高。它使用的协议是SMB。

而NFS由SUN开发，用于UNIX机器之间的资源共享，其设置方便一些。

samba主要用于在windows和unix之间共享资源。资源包括文件、打印机等等。

NFS主要用于在UNIX/LINUX上。

从配置来看，samba比较复杂，nfs比较简单。

2、实用性方面的区别：samba能解决win和linux,linux和linux之间的共享，但是SMB协议是MS的，SMB的高级特性是和windows的特性联系紧密的。

但NFS更简洁，方便，更原生，兼容性更好。