Exercise 1 –NFS Server Installation and configuration

Installation and configuration of NSF Server

Exercise 1.1: Tasks to perform on AlmaLinux:

Use the root account to complete this exercise

1. Install the **NFS server** package.

   dnf install nfs-utils -y

```
[root@server1 ~]#
[root@server1 ~]# dnf install -y nfs-utils
Last metadata expiration check: 2:45:09 ago on Tue 01 Apr 2025 07:49:51 AM.
Dependencies resolved.
================================================================================
 Package                 Architecture        Version                 Repository      Size
================================================================================
Installing:
 nfs-utils               x86_64              1:2.5.4-27.el9          baseos         431 k
Installing dependencies:
 gssproxy                x86_64              0.8.4-7.el9             baseos         108 k
 libev                   x86_64              4.33-5.el9              baseos          52 k
 libnfsidmap             x86_64              1:2.5.4-27.el9          baseos          59 k
 libverto-libev          x86_64              0.3.2-3.el9             baseos          13 k
 rpcbind                 x86_64              1.2.6-7.el9             baseos          56 k
 sssd-nfs-idmap          x86_64              2.9.5-4.el9_5.4         baseos          38 k

Transaction Summary
================================================================================
Install  7 Packages
```

2. Start and enable the **NFS service**.

   Systemctl enable –now nfs-server

```
[root@server1 ~]#
[root@server1 ~]# systemctl enable --now nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.
[root@server1 ~]# systemctl status nf
```

3. Verify that both the **NFS** and **rpcbind** services are **started** and **enabled**.

   systemctl status nf-server.service

```
[root@server1 ~]# systemctl status nf
nfs-blkmap.service  nfs-client.target   nfsdcld.service     nfs-idmapd.service  nfs-mountd.service  nfs-server.service  nfs-utils.service   nftables.service
[root@server1 ~]# systemctl status nfs-server.service
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
     Active: active (exited) since Tue 2025-04-01 10:42:54 EDT; 18s ago
       Docs: man:rpc.nfsd(8)
             man:exportfs(8)
    Process: 59003 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
    Process: 59004 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
    Process: 59019 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload gssproxy ; fi (code=exited, status=0/SUCCESS)
   Main PID: 59019 (code=exited, status=0/SUCCESS)
        CPU: 13ms

Apr 01 10:42:54 server1 systemd[1]: Starting NFS server and services...
Apr 01 10:42:54 server1 systemd[1]: Finished NFS server and services.
[root@server1 ~]# systemctl status r
```

systemctl status rpcbind

```
[root@server1 ~]# systemctl status rpcbind
● rpcbind.service - RPC Bind
     Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-04-01 10:42:53 EDT; 5min ago
TriggeredBy: ● rpcbind.socket
       Docs: man:rpcbind(8)
   Main PID: 58996 (rpcbind)
      Tasks: 1 (limit: 22829)
     Memory: 1.0M
        CPU: 10ms
     CGroup: /system.slice/rpcbind.service
             └─58996 /usr/bin/rpcbind -w -f

Apr 01 10:42:53 server1 systemd[1]: Starting RPC Bind...
Apr 01 10:42:53 server1 systemd[1]: Started RPC Bind.
[root@server1 ~]#
```

4. Authorize the necessary NFS services through the **firewall.**

firewall-cmd --permanent --add-service=nfs --zone=nm-shared

firewall-cmd --permanent --add-service=rpc-bind --zone=nm-shared

firewall-cmd --permanent --add-service=mountd --zone=nm-shared

```
[root@server1 ~]# firewall-cmd --permanent --add-service=nfs --zone=nm-shared
success
[root@server1 ~]# firewall-cmd --permanent --add-service=rpc-bind --zone=nm-shared
success
[root@server1 ~]# firewall-cmd --permanent --add-service=mountd --zone=nm-shared
success
[root@server1 ~]#
```

firewall-cmd - -reload

```
firewall-cmd            firewalld
[root@server1 ~]# firewall-cmd --reload
success
[root@server1 ~]#
```

5. Verify that the required services are added and allowed in the firewall.

   firewall-cmd --list-services --zone=nm-shared

```
[root@server1 ~]# firewall-cmd --list-services --zone=nm-shared
dhcp dns mountd nfs rpc-bind ssh
[root@server1 ~]#
```

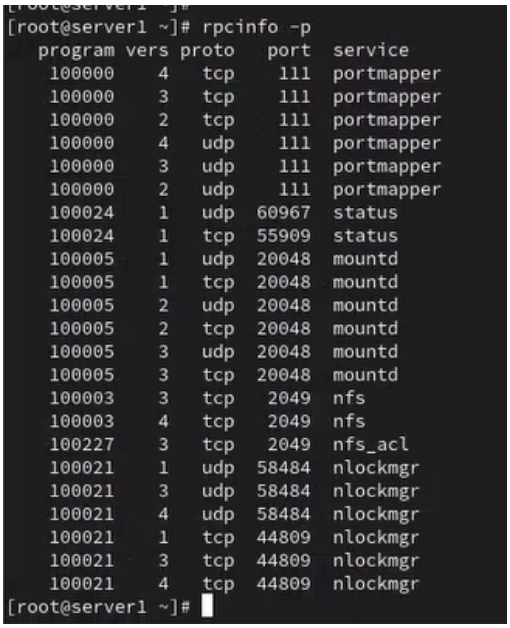
6. List all **TCP** and **UDP** ports currently listening on the server.

   netstat -tunap

```
[root@server1 ~]#
[root@server1 ~]# netstat -tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:20048           0.0.0.0:*               LISTEN      59002/rpc.mountd
tcp        0      0 0.0.0.0:44809           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:55909           0.0.0.0:*               LISTEN      58998/rpc.statd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      995/cupsd
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      996/sshd: /usr/sbin
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN      -
tcp6       0      0 :::20048                :::*                    LISTEN      59002/rpc.mountd
tcp6       0      0 :::49985                :::*                    LISTEN      58998/rpc.statd
tcp6       0      0 ::1:631                 :::*                    LISTEN      995/cupsd
tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
tcp6       0      0 :::22                   :::*                    LISTEN      996/sshd: /usr/sbin
tcp6       0      0 :::2049                 :::*                    LISTEN      -
tcp6       0      0 :::45501                :::*                    LISTEN      -
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1/systemd
udp        0      0 127.0.0.1:323           0.0.0.0:*                           787/chronyd
udp        0      0 127.0.0.1:700           0.0.0.0:*                           58998/rpc.statd
udp        0      0 0.0.0.0:51924           0.0.0.0:*                           766/avahi-daemon: r
udp        0      0 0.0.0.0:58484           0.0.0.0:*                           -
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           766/avahi-daemon: r
udp        0      0 0.0.0.0:60967           0.0.0.0:*                           58998/rpc.statd
udp        0      0 0.0.0.0:20048           0.0.0.0:*                           59002/rpc.mountd
udp        0      0 192.168.204.128:68      192.168.204.254:67      ESTABLISHED 972/NetworkManager
udp6       0      0 :::111                  :::*                                1/systemd
udp6       0      0 :::39221                :::*                                58998/rpc.statd
udp6       0      0 ::1:323                 :::*                                787/chronyd
udp6       0      0 :::45789                :::*                                766/avahi-daemon: r
udp6       0      0 :::52147                :::*                                -
udp6       0      0 :::5353                 :::*                                766/avahi-daemon: r
udp6       0      0 :::20048                :::*                                59002/rpc.mountd
[root@server1 ~]#
```

7. Identify the **TCP port numbers** used by the NFS services.

   rpcinfo -p



```
[root@server1 ~]# rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  60967  status
    100024    1   tcp  55909  status
    100005    1   udp  20048  mountd
    100005    1   tcp  20048  mountd
    100005    2   udp  20048  mountd
    100005    2   tcp  20048  mountd
    100005    3   udp  20048  mountd
    100005    3   tcp  20048  mountd
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
    100021    1   udp  58484  nlockmgr
    100021    3   udp  58484  nlockmgr
    100021    4   udp  58484  nlockmgr
    100021    1   tcp  44809  nlockmgr
    100021    3   tcp  44809  nlockmgr
    100021    4   tcp  44809  nlockmgr
[root@server1 ~]#
```

8. What is the name of the **main configuration file** used by the NFS server?

   /etc/nfs.conf

```
[root@server1 ~]# cat /etc/nfs.conf
#
# This is a general configuration for the
# NFS daemons and tools
#
[general]
# pipefs-directory=/var/lib/nfs/rpc_pipefs
#
[nfsrahead]
# nfs=15000
# nfs4=16000
#
[exports]
# rootdir=/export
#
[exportfs]
# debug=0
#
[gssd]
# verbosity=0
# rpc-verbosity=0
# use-memcache=0
# use-machine-creds=1
use-gss-proxy=1
# avoid-dns=1
# limit-to-legacy-enctypes=0
# allowed-enctypes=aes256-cts-hmac-sha384-192,aes128-cts-hmac-sha256-128,camellia256-cts-
cmac,camellia128-cts-cmac,aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96
# context-timeout=0
# rpc-timeout=5
# keytab-file=/etc/krb5.keytab
# cred-cache-directory=
# preferred-realm=
# set-home=1
```

```
# upcall-timeout=30
# cancel-timed-out-upcalls=0
#
[lockd]
# port=0
# udp-port=0
#
[exportd]
# debug="all|auth|call|general|parse"
# manage-gids=n
# state-directory-path=/var/lib/nfs
# threads=1
# cache-use-ipaddr=n
# ttl=1800
[mountd]
# debug="all|auth|call|general|parse"
# manage-gids=n
# descriptors=0
# port=0
# threads=1
# reverse-lookup=n
# state-directory-path=/var/lib/nfs
# ha-callout=
# cache-use-ipaddr=n
# ttl=1800
#
[nfsdcld]
# debug=0
# storagedir=/var/lib/nfs/nfsdcld
#
[nfsdcltrack]
# debug=0
# storagedir=/var/lib/nfs/nfsdcltrack
#
[nfsd]
# debug=0
# threads=8
# host=
# port=0
# grace-time=90
# lease-time=90
# udp=n
# tcp=y
# vers3=y
# vers4=y
# vers4.0=y
# vers4.1=y
# vers4.2=y
rdma=y
rdma-port=20049

[statd]
# debug=0
# port=0
# outgoing-port=0
# name=
# state-directory-path=/var/lib/nfs/statd
# ha-callout=
# no-notify=0
#
[sm-notify]
# debug=0
# force=0
# retry-time=900
# outgoing-port=
# outgoing-addr=
# lift-grace=y

#tag1234 - Used for install purposes only

[root@server1 ~]#
```

Network share creation

Exercise 1.2: Tasks to perform on AlmaLinux:

Use the root account to complete this exercise

1. Create a user named **teacher1** with UID **1500**.

   useradd -u 1500 teacher1

2. Create a group named **teachers** with GID **1700**.

   groupadd -g 1700 teachers

3. Set the primary group of **teacher1** to **teachers**.

   usermod -g teachers teacher1

4. Using a single command, create the **/mnt/share/IT** directory.

   mkdir -p /mnt/share/IT

5. Change the owner and group of the **/mnt/share/IT** directory to **teacher1** and **teachers.**

   chown teacher1:teachers /mnt/share/IT

6. Set the directory permissions of **/mnt/share/IT** to **770**.

   chmod 770 /mnt/share/IT

7. List the contents of **/mnt/share/IT** to verify the configuration.

   ls -ld /mnt/share/IT

```
[root@server1 ~]#
[root@server1 ~]# useradd -u 1500 teacher1
[root@server1 ~]# groupadd -g 1700 teachers
[root@server1 ~]# usermod -g teachers teacher1
[root@server1 ~]# mkdir -p /mnt/share/IT
[root@server1 ~]# chown teacher1:teachers /mnt/share/IT
[root@server1 ~]# chmod 770 /mnt/share/IT
[root@server1 ~]# ls -lda /mnt/share/IT
drwxrwx---. 2 teacher1 teachers 6 Apr  1 17:48 /mnt/share/IT
[root@server1 ~]#
```

8. Configure NFS to make the **/mnt/share/IT** directory accessible to the **192.168.50.0/24** network with **read** and **write** permissions.

   Edit the /etc/exports file and add the following line:

   /mnt/share/IT 192.168.50.0/24(rw,sync,no_all_squash)

This line in the `/etc/exports` file configures the Network File System (NFS) to share the directory `/mnt/share/IT` with devices on the `192.168.50.0/24` network.

vim /mnt/share/IT

```
                                                    root@server1:~
/mnt/share/IT   192.168.50.0/24(rw,sync,no_all_squash)
~
~
~
```

9. Export the directory.

exportfs -arv

-a Export all directories listed in /etc/exports.

-r Re-export all entries (useful after editing /etc/exports).

-v Enable verbose mode to display detailed information about what is being exported.

```
[root@server1 ~]# exportfs -arv
exporting 192.168.50.0/24:/mnt/share/IT
[root@server1 ~]#
```

10. View the current list of exported directories.

exportfs -v

```
[root@server1 ~]# exportfs -v
/mnt/share/IT   192.168.50.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
[root@server1 ~]#
```

or

exportfs -s

```
[root@server1 ~]#
[root@server1 ~]# exportfs -s
/mnt/share/IT  192.168.50.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
[root@server1 ~]#
```

Export Details:

- Directory: `/mnt/share/IT`

- Allowed Clients: `192.168.50.0/24`

- Options: sync, wdelay, hide, no_subtree_check, sec=sys, rw, secure, root_squash, no_all_squash

1. **sync: Write changes to disk before acknowledging requests (safer but slower).**
2. **wdelay: Delay writes to improve performance (enabled by default).**
3. **hide: Hide subdirectories if the parent is exported elsewhere.**
4. **no_subtree_check: Disable subtree checking (improves reliability for frequently renamed directories).**
5. **sec=sys: Use UNIX user/group IDs for authentication.**
6. **rw: Allow read and write access.**
7. **secure: Restrict exports to ports below 1024 (default).**
8. **root_squash: Map `root` (UID 0) to an anonymous user for security.**
9. **no_all_squash: Do not map all users to anonymous (preserve client UIDs/GIDs).**

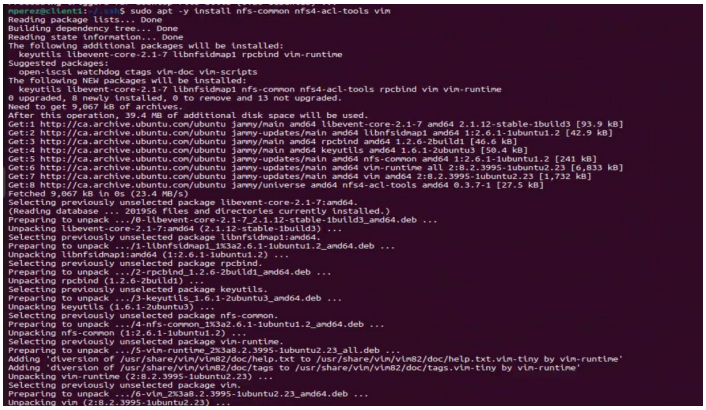
Mounting shared directories on the client

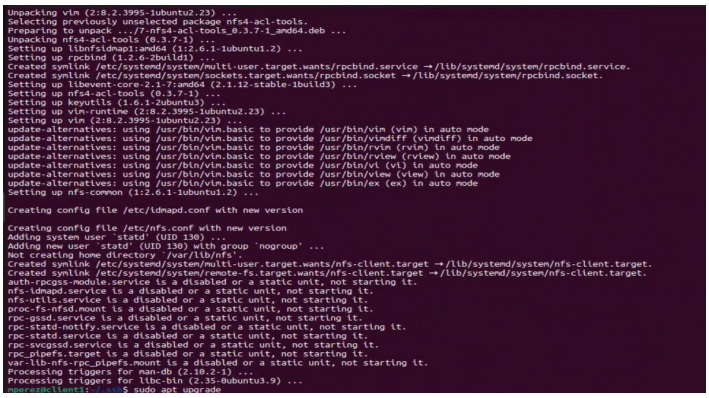
Exercise 1.3: Tasks to perform on Ubuntu:

Use your Ubuntu user account to complete this exercise on Ubuntu

1. Install the **NFS client** on **Ubuntu**.

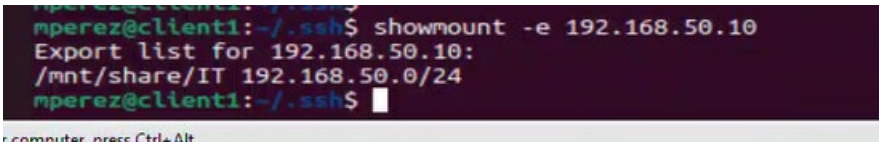
   sudo apt -y install nfs-common nfs4-acl-tools vim





2. Run a command to list the directories exported by the NFS server.

   showmount -e 192.168.50.10



3. Create the user **teacher1** and the group **teachers** using the **same UID** and **GID** as in the previous exercise. Assign the password alma to the teacher1 user.

sudo groupadd -g 1700 teachers

sudo adduser -- ingroup teachers -u 1500 teacher1

```
mperez@client1:~/.ssh$ sudo groupadd -g 1700 teachers
[sudo] password for mperez:
mperez@client1:~/.ssh$ sudo adduser --ingroup teachers -u 1500 teacher1
Adding user `teacher1' ...
Adding new user `teacher1' (1500) with group `teachers' ...
Creating home directory `/home/teacher1' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for teacher1
Enter the new value, or press ENTER for the default
        Full Name []: Nikola Tesla
        Room Number []: 111
        Work Phone []: 555-555-5555
        Home Phone []: 444-444-4444
        Other []: Inventor
Is the information correct? [Y/n] y
mperez@client1:~/.ssh$
```

4. Create the local directory: **/share/tech**.

   sudo mkdir -p /share/tech

```
mperez@client1:~/.ssh$ sudo mkdir -p /share/tech
mperez@client1:~/.ssh$
```

5. Mount the **/mnt/share/IT** directory exported by the **AlmaLinux** server to the local **/share/tech** directory on **Ubuntu**.

   sudo mount 192.168.50.10:/mnt/share/IT /share/tech

```
mperez@client1:~/.ssh$
mperez@client1:~/.ssh$ sudo mount -t nfs 192.168.50.10:/mnt/share/IT /share/tech/
mperez@client1:~/.ssh$
```

6. Run a command to confirm that the NFS share has been successfully mounted.

   sudo mount | grep nfs

```
mperez@client1:~/.ssh$
mperez@client1:~/.ssh$ sudo mount | grep nfs
192.168.50.10:/mnt/share/IT on /share/tech type nfs4 (rw,relatime,vers=4.2,rsize=524288,wsize=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.50.20,local_lock=none,addr=192.168.50.10)
mperez@client1:~/.ssh$
mperez@client1:~/.ssh$ df -h | grep '/share/tech'
192.168.50.10:/mnt/share/IT   40G  5.3G   35G  14% /share/tech
mperez@client1:~/.ssh$
```

   df -h

```
mperez@client1:~/.ssh$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
tmpfs                       387M  2.0M  385M   1% /run
/dev/sda6                    22G  8.6G   12G  43% /
tmpfs                       1.9G   14M  1.9G   1% /dev/shm
tmpfs                       5.0M  4.0K  5.0M   1% /run/lock
/dev/sda4                   944M  198M  682M  23% /boot
/dev/sda2                   976M  6.1M  969M   1% /boot/efi
/dev/sda5                   4.6G  254M  4.1G   6% /home
tmpfs                       387M  108K  387M   1% /run/user/1000
192.168.50.10:/mnt/share/IT  40G  5.3G   35G  14% /share/tech
mperez@client1:~/.ssh$
```

7. Use the **su -** command to switch to the **teacher1** user.

   su – teacher1

```
mperez@client1:~/.ssh$
mperez@client1:~/.ssh$ su - teacher1
Password:
teacher1@client1:~$
teacher1@client1:~$
teacher1@client1:~$
```

8. Try to create a text file in the **/share/tech** directory. Are you able to create the file? Why or why not?

   Yes I am able to create the file, as we can see the permissions for folder /share/tech the user teacher1 is able to read, write an execute.

   touch /share/tech/testfile.txt

   ll -a /share/tech/

```
teacher1@client1:~$ touch /share/tech/testfile.txt
teacher1@client1:~$ ll -a /share/tech/
total 4
drwxrwx--- 2 teacher1 teachers   26 Apr  1 21:05 ./
drwxr-xr-x 3 root     root     4096 Apr  1 20:50 ../
-rw-r--r-- 1 teacher1 teachers    0 Apr  1 21:05 testfile.txt
teacher1@client1:~$
```

```
teacher1@client1:~$ ll -a /share/
total 8
drwxr-xr-x  3 root     root     4096 Apr  1 20:50 ./
drwxr-xr-x 22 root     root     4096 Apr  1 20:50 ../
drwxrwx---  2 teacher1 teachers   26 Apr  1 21:05 tech/
teacher1@client1:~$
```

9. Return to the **AlmaLinux** server and check the contents of the **/mnt/share/IT** directory. What do you observe?

   ll -a /mnt/share/IT/

   The directory includes testfile.txt file created by teacher1 in Ubuntu client.

```
[root@server1 ~]# ll -a /mnt/share/IT/
total 0
drwxrwx---. 2 teacher1 teachers 26 Apr  1 21:05 .
drwxr-xr-x. 3 root     root     16 Apr  1 17:48 ..
-rw-r--r--. 1 teacher1 teachers  0 Apr  1 21:05 testfile.txt
```

10. Go back to **Ubuntu** and **log out** from the **teacher1** session.

    exit

```
teacher1@client1:~$ exit
logout
mperez@client1:~/.ssh$
mperez@client1:~/.ssh$
```

11. Unmount the **/share/tech** directory.

    sudo umont /share/tech

```
mperez@client1:~/.ssh$
mperez@client1:~/.ssh$ sudo umount /share/tech/
mperez@client1:~/.ssh$
```

Check is not mounted

sudo mount | grep nfs

df -h

```
mperez@client1:~/.ssh$ sudo mount | grep nfs
mperez@client1:~/.ssh$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           387M  2.0M  385M   1% /run
/dev/sda6        22G  8.6G   12G  43% /
tmpfs           1.9G   14M  1.9G   1% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda4       944M  198M  682M  23% /boot
/dev/sda2       976M  6.1M  969M   1% /boot/efi
/dev/sda5       4.6G  254M  4.1G   6% /home
tmpfs           387M  108K  387M   1% /run/user/1000
mperez@client1:~/.ssh$
```

12. Ensure that the **/share/tech** directory is now empty.

sudo ls -la /share/tech

```
sudo: ll: command not found
mperez@client1:~/.ssh$ sudo ls -la /share/tech/
total 8
drwxr-xr-x 2 root root 4096 Apr  1 20:50 .
drwxr-xr-x 3 root root 4096 Apr  1 20:50 ..
mperez@client1:~/.ssh$
```