

NFS Server Setup

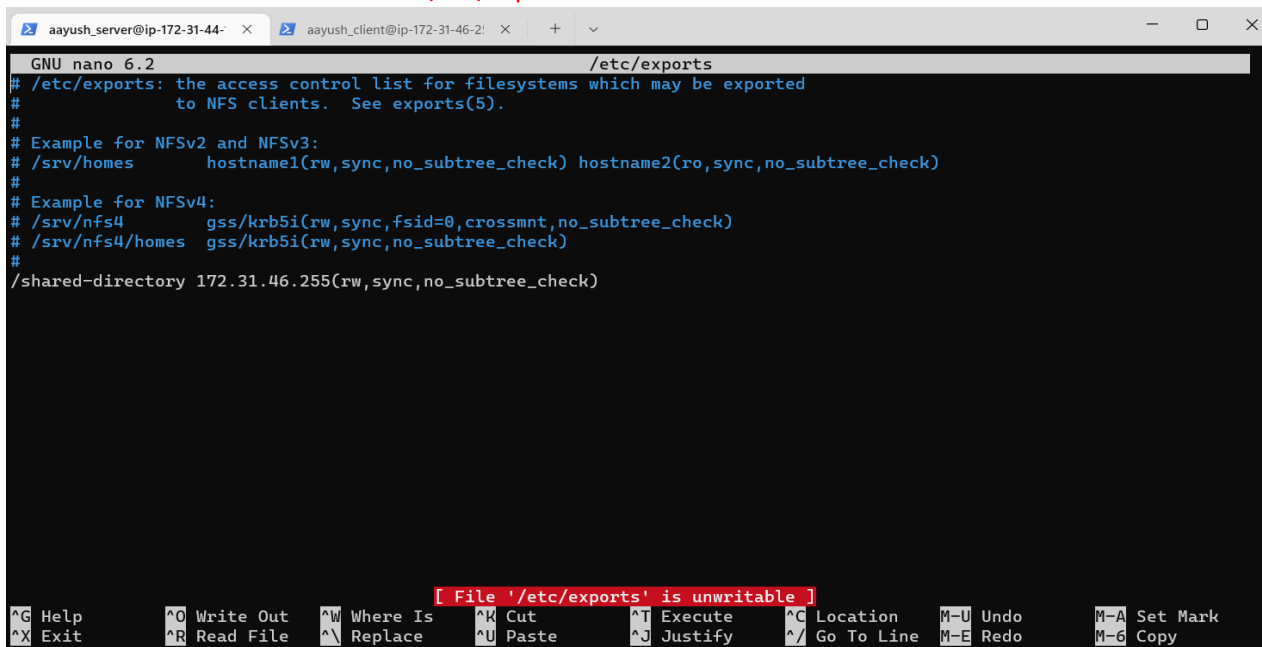
1. **Install NFS Server Packages:** On the server (172.31.44.132), install the NFS server package.

`sudo apt update`

`sudo apt install nfs-kernel-server`

2. **Configure Shared Directories:** Edit the `/etc/exports` file to define which directories to share and the permissions.

`sudo nano /etc/exports`



```
GNU nano 6.2 /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/srv/nfs4 /shared-directory 172.31.46.255(rw,sync,no_subtree_check)

[ File '/etc/exports' is unwritable ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^/_ Go To Line  M-E Redo     M-G Copy
```

Add a line for the directory you want to share, specifying the client IP address and permissions. For example:

`/shared-directory 172.31.46.255(rw,sync,no_subtree_check)`

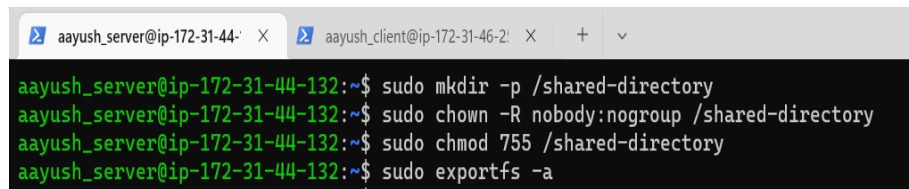
- `/shared-directory` is the directory you want to share.
- `172.31.46.255` is the client IP address.
- `rw` allows read and write access.
- `sync` ensures changes are written synchronously.
- `no_subtree_check` prevents checking of the directory subtree.

3. **Create and Set Permissions for Shared Directory:**

`sudo mkdir -p /shared-directory`

`sudo chown -R nobody:nogroup /shared-directory`

`sudo chmod 755 /shared-directory`



```
aayush_server@ip-172-31-44-~$ sudo mkdir -p /shared-directory
aayush_server@ip-172-31-44-132:~$ sudo chown -R nobody:nogroup /shared-directory
aayush_server@ip-172-31-44-132:~$ sudo chmod 755 /shared-directory
aayush_server@ip-172-31-44-132:~$ sudo exportfs -a
```

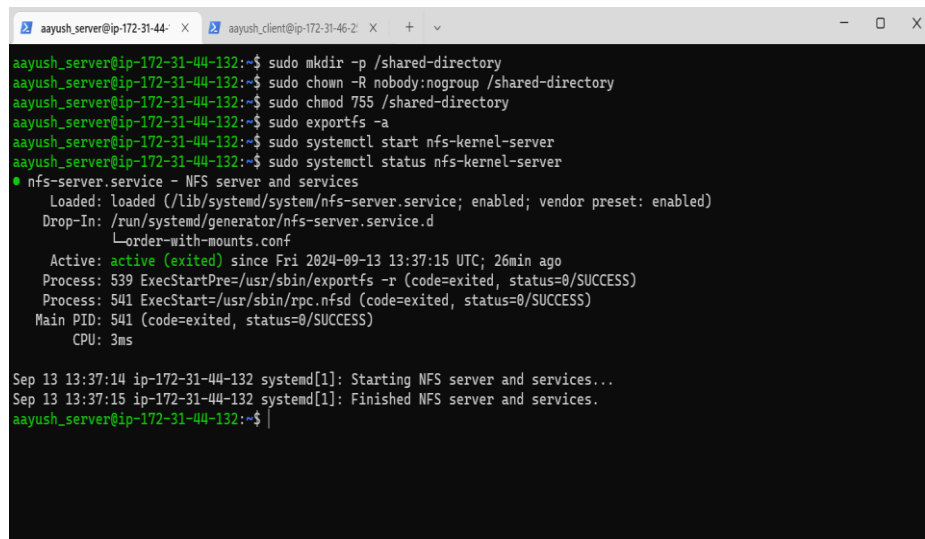
4. Export the Shared Directory:

`sudo exportfs -a`

5. Start and Enable the NFS Server:

`sudo systemctl start nfs-kernel-server`

`sudo systemctl enable nfs-kernel-server`



```
aayush_server@ip-172-31-44-~$ sudo mkdir -p /shared-directory
aayush_server@ip-172-31-44-132:~$ sudo chown -R nobody:nogroup /shared-directory
aayush_server@ip-172-31-44-132:~$ sudo chmod 755 /shared-directory
aayush_server@ip-172-31-44-132:~$ sudo exportfs -a
aayush_server@ip-172-31-44-132:~$ sudo systemctl start nfs-kernel-server
aayush_server@ip-172-31-44-132:~$ sudo systemctl status nfs-kernel-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Drop-In: /run/systemd/generator/nfs-server.service.d
            └─order-with-mounts.conf
   Active: active (exited) since Fri 2024-09-13 13:37:15 UTC; 26min ago
     Process: 539 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
     Process: 541 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
    Main PID: 541 (code=exited, status=0/SUCCESS)
       CPU: 3ms

Sep 13 13:37:14 ip-172-31-44-132 systemd[1]: Starting NFS server and services...
Sep 13 13:37:15 ip-172-31-44-132 systemd[1]: Finished NFS server and services.
aayush_server@ip-172-31-44-132:~$
```

6. Open Firewall for NFS (if applicable):

`sudo ufw allow from 172.31.46.255 to any port nfs`

NFS Client Setup

1. **Install NFS Client Packages:** On the client (172.31.46.255), install the NFS client package.

```
sudo apt update
```

```
sudo apt install nfs-common
```

2. **Create a Mount Point:**

```
sudo mkdir -p /mnt/shared-directory
```

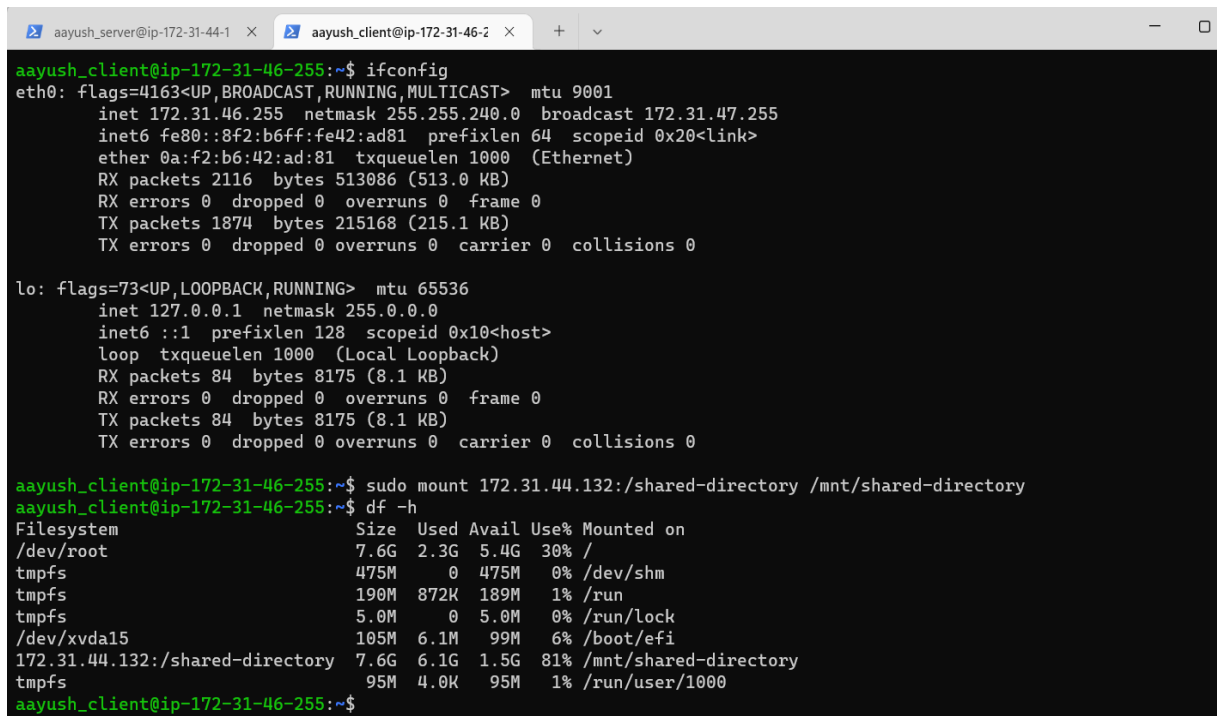
3. **Mount the NFS Share:**

```
sudo mount 172.31.44.132:/shared-directory /mnt/shared-directory
```

4. **Verify the Mount:**

```
df -h
```

Check if /mnt/shared-directory is listed as mounted.



```
aayush_client@ip-172-31-46-255:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.46.255 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::8f2:b6ff:fe42:ad81 prefixlen 64 scopeid 0x20<link>
    ether 0a:f2:b6:42:ad:81 txqueuelen 1000 (Ethernet)
    RX packets 2116 bytes 513086 (513.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1874 bytes 215168 (215.1 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 84 bytes 8175 (8.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 84 bytes 8175 (8.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

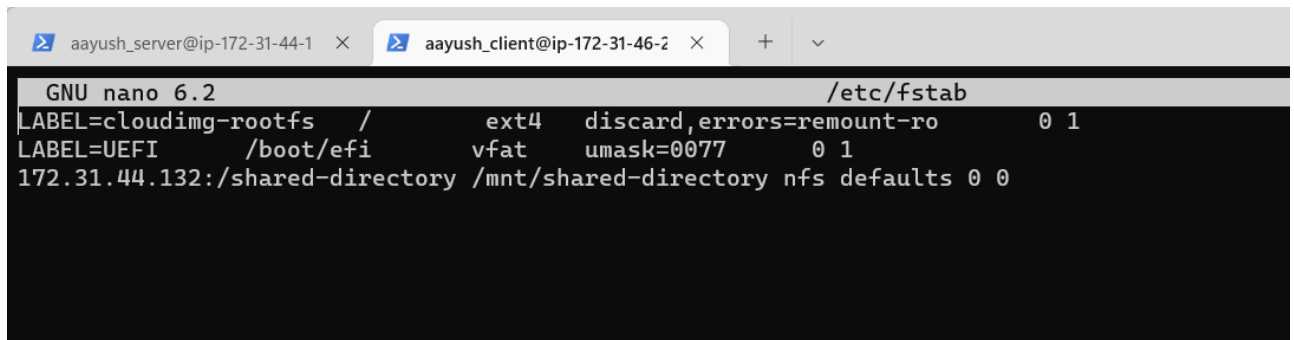
aayush_client@ip-172-31-46-255:~$ sudo mount 172.31.44.132:/shared-directory /mnt/shared-directory
aayush_client@ip-172-31-46-255:~$ df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/root                  7.6G      2.3G   5.4G  30% /
tmpfs                      475M        0   475M   0% /dev/shm
tmpfs                     190M      872K   189M   1% /run
tmpfs                      5.0M        0    5.0M   0% /run/lock
/dev/xvda15                105M       6.1M    99M   6% /boot/efi
172.31.44.132:/shared-directory 7.6G      6.1G   1.5G  81% /mnt/shared-directory
tmpfs                      95M       4.0K    95M   1% /run/user/1000
aayush_client@ip-172-31-46-255:~$
```

5. **Configure Automatic Mount at Boot:** To automatically mount the NFS share on boot, add an entry to /etc/fstab.

```
sudo nano /etc/fstab
```

Add the following line:

172.31.44.132:/shared-directory /mnt/shared-directory nfs defaults 0 0



```
GNU nano 6.2 /etc/fstab
LABEL=cloudimg-rootfs / ext4 discard,errors=remount-ro 0 1
LABEL=UEFI /boot/efi vfat umask=0077 0 1
172.31.44.132:/shared-directory /mnt/shared-directory nfs defaults 0 0
```

6. Test the Configuration:

Reboot the client machine or use mount -a to test the automatic mounting.

`sudo mount -a`

Create files

On the NFS Server

1. Navigate to the Shared Directory:

`cd /shared-directory`

2. Create a File:

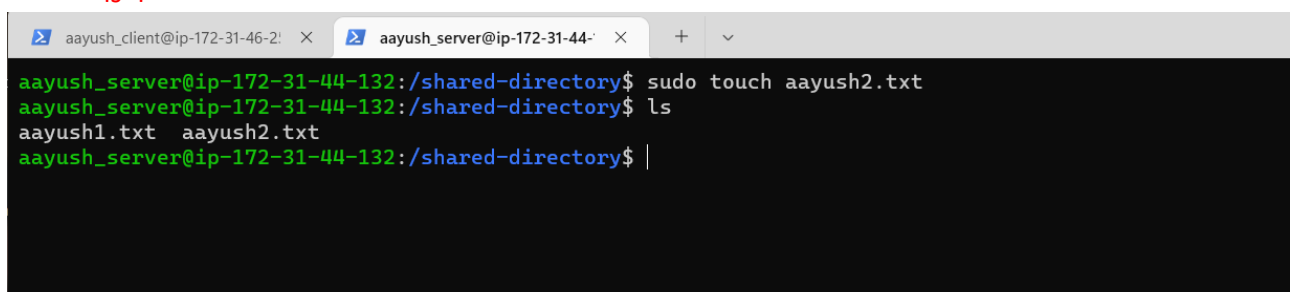
You can create a file using various commands, such as touch, echo, or using a text editor.

`sudo touch aayush2.txt`

3. Verify Permissions:

Ensure that the permissions are set correctly so that the client can access the file.

`ls -l`



```
aayush_server@ip-172-31-44-132:/shared-directory$ sudo touch aayush2.txt
aayush_server@ip-172-31-44-132:/shared-directory$ ls
aayush1.txt  aayush2.txt
aayush_server@ip-172-31-44-132:/shared-directory$
```

On the NFS Client

1. **Navigate to the Mounted Directory:**

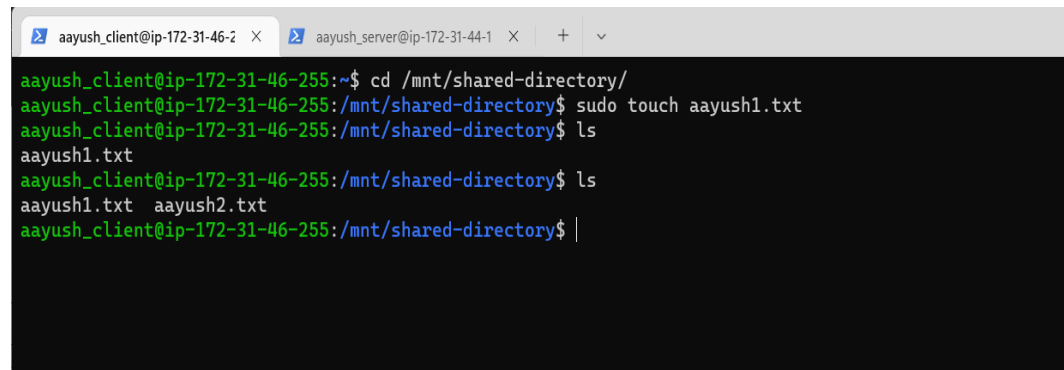
```
cd /mnt/shared-directory
```

2. **Create a File:** Similar to the server, you can create a file on the client using:

```
touch aayush1.txt
```

3. **Verify File Creation:** Check the directory to ensure the file has been created.

```
ls -l
```

A terminal window with two tabs: 'aayush_client@ip-172-31-46-2' and 'aayush_server@ip-172-31-44-1'. The active tab is the client. The terminal shows the following commands and output:

```
aayush_client@ip-172-31-46-255:~$ cd /mnt/shared-directory/
aayush_client@ip-172-31-46-255:/mnt/shared-directory$ sudo touch aayush1.txt
aayush_client@ip-172-31-46-255:/mnt/shared-directory$ ls
aayush1.txt
aayush_client@ip-172-31-46-255:/mnt/shared-directory$ ls
aayush1.txt  aayush2.txt
aayush_client@ip-172-31-46-255:/mnt/shared-directory$ |
```