Install NFS Server and Client on Ubuntu 18.04 LTS

NFS or Network File System is a distributed file system protocol, originally built but the Sun Microsystems. Through NFS, you can allow a system to share directories and files with others over a network. In NFS file sharing, users and even programs can access information on remote systems almost as if they were residing on a local machine.

NFS is operated in a client-server environment where the server is responsible for managing the authentication, authorization, and management of clients, as well as all the data shared within a specific file system. Upon authorization, any number of clients can access the shared data as if it was present in their internal storage. Setting up an NFS server on your Ubuntu system is very simple. All you need to do is make some necessary installations and configurations, both on the server and client machines and you are good to go.

In this article, we will explain step by step how to set up an NFS server and client which will enable you to share files from one Ubuntu system to the other.

We have run the commands and procedures described in this article on a Ubuntu 18.04 LTS system. Since we are using the Ubuntu command line, the Terminal, to perform all the operations; you can open it either through the system dash or the Ctrl+Alt+T shortcut.

Setting up the host server

In order to set up the host system to share directories, we will need to install the NFS Kernel server on it, and then create and export the directories that we want the client systems to access. Please follow these steps in order to smoothly set up the host side:

Step 1: Install NFS Kernel Server

Before installing the NFS Kernel server, we need to update our system's repository index with that of the Internet through the following apt command as sudo:

$ sudo apt-get update

The above command lets us install the latest available version of a software through the Ubuntu repositories.

Now, run the following command in order to install the NFS Kernel Server on your system:

$ sudo apt install nfs-kernel-server

```
File  Edit  View  Search  Terminal  Help
sana@Linux:~$ sudo apt install nfs-kernel-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer
  dkms libgsoap-2.8.60 libqt5opengl5 libqt5positioning5 libqt5printsu
  libqt5qml5 libqt5quick5 libqt5sensors5 libqt5webchannel5 libqt5webk
  libqt5x11extras5 libvncserver1 python-notify qml-module-qtgraphical
  qml-module-qtquick-controls qml-module-qtquick-dialogs
  qml-module-qtquick-layouts qml-module-qtquick-privatewidgets
  qml-module-qtquick-window2 qml-module-qtquick2 simplescreenrecorder
  virtualbox-dkms
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  keyutils libnfsidmap2 libtirpc1 nfs-common rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server rpcbin
0 upgraded, 6 newly installed, 0 to remove and 161 not upgraded.
Need to get 490 kB of archives.
After this operation, 1,700 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

The system will prompt you with a Y/n option to confirm if you want to continue with the installation. Please enter Y and then hit Enter to continue, after which the software will be successfully installed on your system.

Step 2: Create the Export Directory

The directory that we want to share with the client system is called an export directory. You can name it according to your choice; here, we are creating an export directory by the name of “sharedfolder” in our system’s mnt(mount) directory.

Use the following command, by specifying a mount folder name according to your need, through the following command as root:

$ sudo mkdir -p /mnt/sharedfolder

```
File  Edit  View  Search  Terminal  Help
sana@Linux:~$ sudo mkdir -p /mnt/sharedfolder
[sudo] password for sana:
```

As we want all clients to access the directory, we will remove restrictive permissions of the export folder through the following commands:

$ sudo chown nobody:nogroup /mnt/sharedfolder

$ sudo chmod 777 /mnt/sharedfolder

Now all users from all groups on the client system will be able to access our “sharedfolder”.

```
sana@Linux:~$ sudo chown nobody:nogroup /mnt/sharedfolder
sana@Linux:~$ sudo chmod 777 /mnt/sharedfolder
```

You can create as many sub-folders in the export folder as you want, for the client to access.

Step 3: Assign server access to client(s) through NFS export file

After creating the export folder, we will need to provide the clients the permission to access the host server machine. This permission is defined through the exports file located in your system’s /etc folder. Please use the following command in order to open this file through the Nano editor:

$ sudo nano /etc/exports

Editing this file needs root access; therefore you will need to use sudo with your command. You can also open the file in any of your personal favorite text editors.

Once you have opened the file, you can allow access to:

- A single client by adding the following line in the file:

/mnt/sharedfolder clientIP(rw,sync,no_subtree_check)

- Multiple clients by adding the following lines in the file:

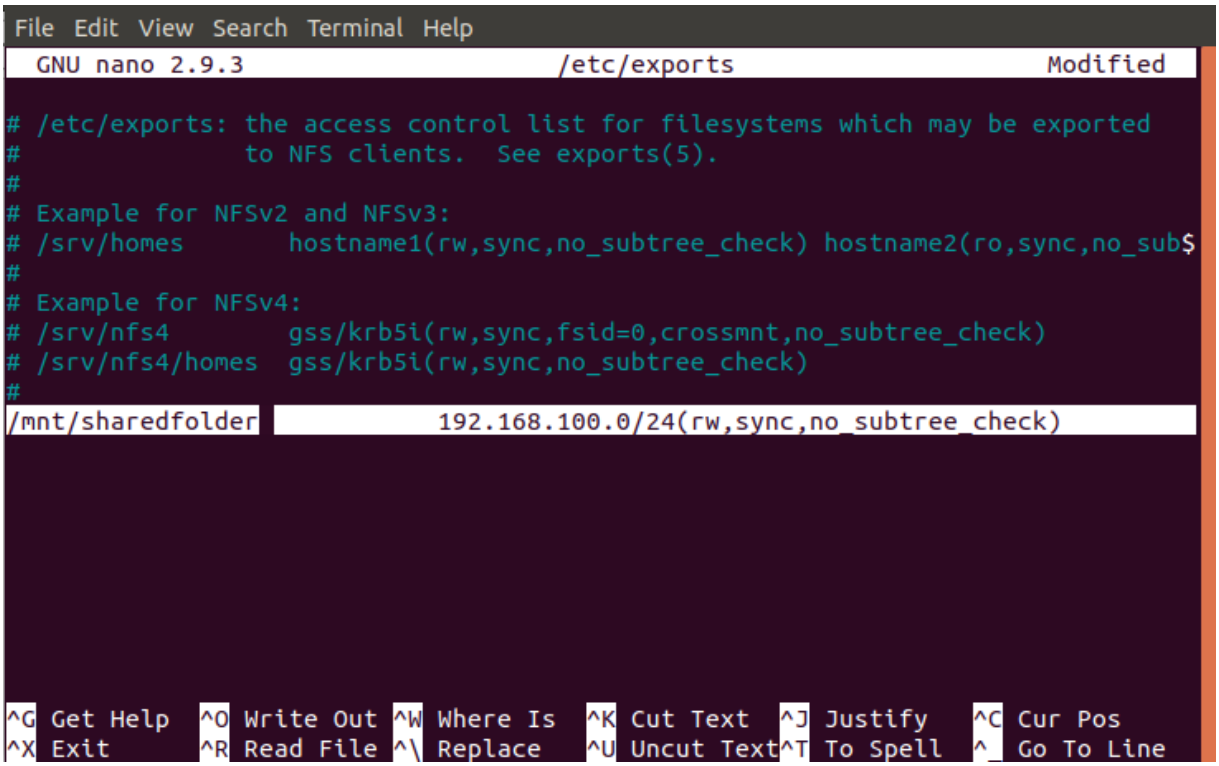
/mnt/sharedfolder client1IP(rw,sync,no_subtree_check)

/mnt/sharedfolder client2IP(rw,sync,no_subtree_check)

- Multiple clients, by specifying an entire subnet that the clients belong to:

/mnt/sharedfolder subnetIP/24(rw,sync,no_subtree_check)

In this example, we are specifying an entire subnet of all the clients we want to grant access to our export folder (sharedfolder):



Add the required line(s) to your exports file and then save it by hitting Ctrl+X, entering Y, and then hitting Enter.

The permissions “rw,sync,no_subtree_check” permissions defined in this file mean that the client(s) can perform:

- **rw**: read and write operations
- **sync**: write any change to the disc before applying it
- **no_subtree_check**: prevent subtree checking

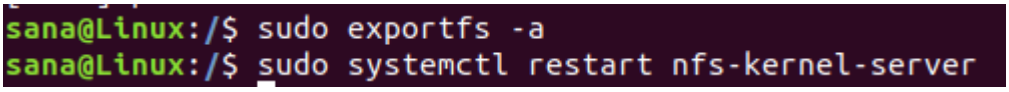
Step 4: Export the shared directory

After making all the above configurations in the host system, now is the time to export the shared directory through the following command as sudo:

$ sudo exportfs -a

Finally, in order to make all the configurations take effect, restart the NFS Kernel server as follows:

$sudo systemctl restart nfs-kernel-server



Step 5: Open firewall for the client (s)

An important step is to verify that the server's firewall is open to the clients so that they can access the shared content. The following command will configure the firewall to give access to clients through NFS:

$ sudo ufw allow from [clientIP or clientSubnetIP] to any port nfs

In our example, we are giving access to an entire subnet of clients machines through the following command:

$ sudo ufw allow from 192.168.100/24 to any port nfs

```
sana@Linux:/$ sudo ufw allow from 192.168.100.0/24 to any port nfs
Rule added
```

Now when you check the status of your Ubuntu firewall through the following command, you will be able to view the Action status as "Allow" for the client's IP.

$ sudo ufw status

```
sana@Linux:/$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache Full                ALLOW       Anywhere
10000/tcp                  ALLOW       Anywhere
20/tcp                     ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
Apache                     ALLOW       Anywhere
2049                       ALLOW       192.168.100.5
2049                       ALLOW       192.168.100.0/24
Apache Full (v6)           ALLOW       Anywhere (v6)
10000/tcp (v6)             ALLOW       Anywhere (v6)
20/tcp (v6)                ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

Your host server is now ready to export the shared folder to the specified client(s) through the NFS Kernel Server.

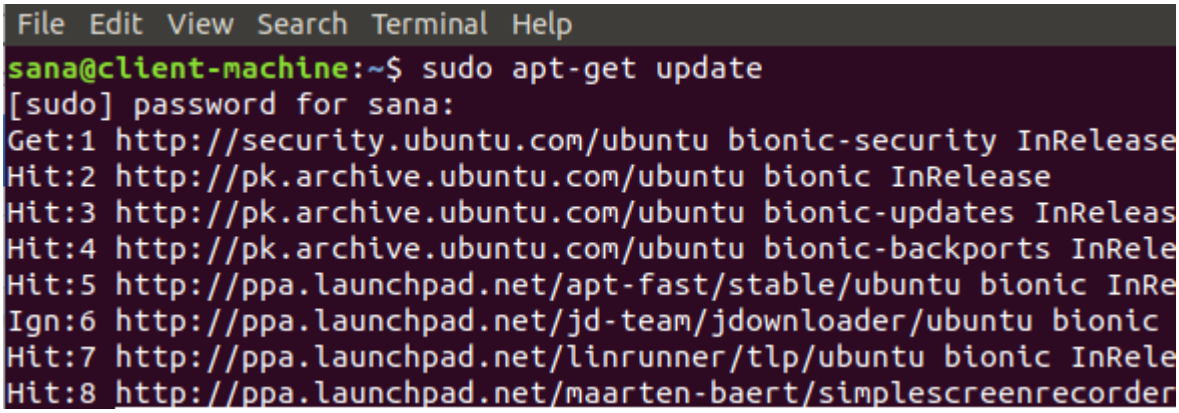
Configuring the Client Machine

Now is the time to make some simple configurations to the client machine, so that the shared folder from the host can be mounted to the client and then accessed smoothly.

Step 1: Install NFS Common

Before installing the NFS Common application, we need to update our system's repository index with that of the Internet through the following apt command as sudo:

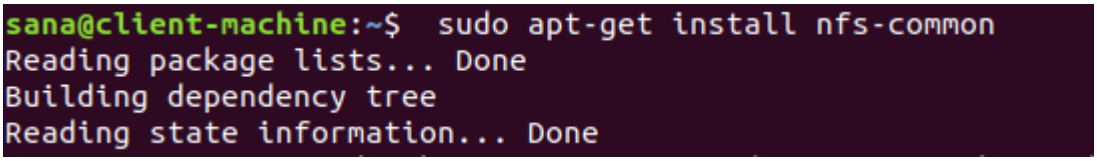
$ sudo apt-get update



The above command lets us install the latest available version of a software through the Ubuntu repositories.

Now, run the following command in order to install the NFS Common client on your system:

$ sudo apt-get install nfs-common

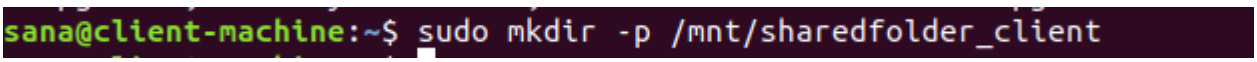


The system will prompt you with a Y/n option to confirm if you want to continue with the installation. Please enter Y and then hit Enter to continue, after which the software will be successfully installed on your system.

Step 2: Create a mount point for the NFS host's shared folder

Your client's system needs a directory where all the content shared by the host server in the export folder can be accessed. You can create this folder anywhere on your system. We are creating a mount folder in the mnt directory of our client's machine:

$ sudo mkdir -p /mnt/sharedfolder_client



Step 3: Mount the shared directory on the client

The folder that you created in the above step is like any other folder on your system unless you mount the shared directory from your host to this newly created folder.

Use the following command in order to mount the shared folder from the host to a mount folder on the client:

$ sudo mount serverIP:/exportFolder_server /mnt/mountfolder_client

In our example, we are running the following command to export our “sharedfolder” from the server to the mount folder “sharedfolder_client” on the client machine:

$ sudo mount 192.168.100.5:/mnt/sharedfolder /mnt/sharedfolder_client

Step 4: Test the connection

Please create or save a file in the export folder of the NFS host server. Now, open the mount folder on the client machine; you should be able to view the same file shared and accessible in this folder.

Conclusion

Setting up an NFS client-server environment on Ubuntu systems is an easy task. Through this article, you learned how to install the required NFS packages on both the server and the clients. You also learned how to configure the NFS server and client machines so that folders can be shared and then accessed smoothly without any firewall or permissions-related glitch. Now you can easily share content from one Ubuntu system to the other using the NFS protocol.