Aim: Configure NFS Server to share directories on your Network, Configure NFS Client

NFS (Network File System):

* Network File System is a distributed file system protocol originally developed only by Sun Microsystems. Through NFS, you can allow a system to share directories and files with others over a network
* . By sharing NFS files, users and even programs can access information on remote systems almost as if they were on a local machine.
* NFS operates in a client-server environment where the server is responsible for managing the authentication, authorization, and administration of clients, as well as for all data shared within a particular file system.

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**

To install the NFS Kernel Server on your system:

**$ sudo apt install nfs-kernel-server**

Step 2: Create the Export Directory

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

**$ sudo mkdir -p /mnt/sharedfolder**

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**

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.

**$ sudo nano /etc/exports**

Editing this file needs root access; therefore you will need to use sudo with your command.

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

**/mnt/sharedfolder clientIP(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**

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

**$ 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.

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

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

**$ sudo ufw status**

Configuring the NFS Client Machine:

Step 1: Install NFS Common

**$ sudo apt-get update**

**$ 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

Mount the shared folder from the host to a mount folder on the client:

**$ sudo mount serverIP:/exportFolder\_server /mnt/mountfolder\_client**

**$ 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.