NFS [Network File System in Linux]

Learning via Demonstration in Linux

Asst. Prof. Ashwini Mathur

Source : https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/storage_administration_guide/ch09s02

Demonstration reference : www.techmint.com

What is Network File System “NFS” ?

A *Network File System* (*NFS*) allows remote hosts to mount file systems over a network and interact with those file systems as though they are mounted locally.

This enables system administrators to consolidate resources onto centralized servers on the network.

Little History !!

NFS (**Network File System**) is basically developed for sharing of **files** and **folders** between **Linux**/**Unix** systems by **Sun Microsystems** in **1980**. It allows you to mount your local file systems over a network and remote hosts to interact with them as they are mounted locally on the same system

Benefits of NFS

1. **NFS** allows local access to remote files.
2. It uses standard **client**/**server** architecture for file sharing between all ***nix** based machines.
3. With **NFS** it is not necessary that both machines run on the same **OS**.
4. With the help of **NFS** we can configure **centralized storage** solutions.
5. Users get their **data** irrespective of physical location.
6. No manual **refresh** needed for new files.
7. Newer version of **NFS** also supports **acl**, **pseudo** root mounts.
8. Can be secured with **Firewalls** and **Kerberos**.

How NFS Works ?

Overview

Currently, there are three versions of NFS. NFS version 2 (NFSv2) is older and widely supported. NFS version 3 (NFSv3) supports safe asynchronous writes and is more robust at error handling than NFSv2; it also supports 64-bit file sizes and offsets, allowing clients to access more than 2Gb of file data

All versions of NFS can use *Transmission Control Protocol* (*TCP*) running over an IP network, with NFSv4 requiring it. NFSv2 and NFSv3 can use the *User Datagram Protocol* (UDP) running over an IP network to provide a stateless network connection between the client and server.

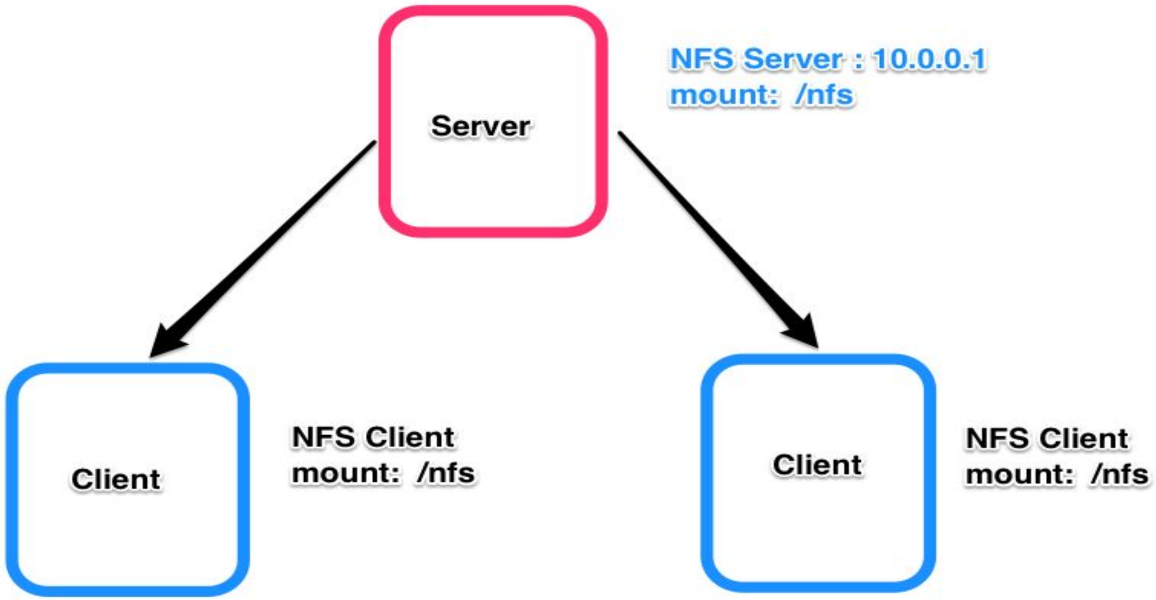
Important Files for NFS Configuration

1. **/etc/exports** : Its a main configuration file of **NFS**, all exported **files** and **directories** are defined in this file at the **NFS Server** end.
2. **/etc/fstab** : To mount a **NFS directory** on your system across the **reboots**, we need to make an entry in **/etc/fstab**.
3. **/etc/sysconfig/nfs** : Configuration file of **NFS** to control on which port **rpc** and other services are **listening**.

Setup and Configure NFS Mounts on Linux Server

To setup **NFS** mounts, we'll be needing at least two **Linux/Unix** machines. Here in this tutorial, I'll be using two servers.

1. **NFS Server**: nfsserver.example.com with IP**-192.168.0.100**
2. **NFS Client** : nfsclient.example.com with IP**-192.168.0.101**


Server
NFS Server : 10.0.0.1
mount: /nfs
Client
NFS Client
mount: /nfs
Client
NFS Client
mount: /nfs

NFS SETUP

NFS mount needed at least two machines. The machine hosting the shared folders is called as server and which connects is called as clients.

Step 1: Configure a server

install `nfs-kernel-server` packages.

Command : `sudo apt install nfs-kernel-server`

Configure the exports file ..

Then edit the exports file with nano text editor. The exports files tell NFS server which directories or file systems will be shared to client.

```
sudo nano /etc/exports
```

The syntax is as follows

```
/path/to/directory     client-IP address(options)
```

For example, you want to share you home directory to the second Ubuntu computer with IP address `192.168.1.101`, then put the following line at the end of the file. Replace `username` with your actual username. Delimit the two columns with Tab key.

```
/home/username      192.168.1.101(rw,sync,root_squash,subtree_check)
```

/etc/export configurations ---

- **rw**: This option gives the client computer both read and write access to the volume.
- **sync**: This option forces NFS to write changes to disk before replying. This results in a more stable and consistent environment since the reply reflects the actual state of the remote volume. However, it also reduces the speed of file operations.
- **no_subtree_check**: This option prevents subtree checking, which is a process where the host must check whether the file is actually still available in the exported tree for every request. This can cause many problems when a file is renamed while the client has it opened. In almost all cases, it is better to disable subtree checking.
- **no_root_squash**: By default, NFS translates requests from a root user remotely into a non-privileged user on the server. This was intended as security feature to prevent a root account on the client from using the file system of the host as root. `no_root_squash`disables this behavior for certain shares.

Step 2: Configure the Client

install `nfs-common` package.

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

Then edit `/etc/fstab` file.

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

Add the following line in the file. Replace `nfs-server-ip` with the IP address of the first Ubuntu computer.

```
nfs-server-ip:/home/username    /mnt/nfs-share      nfs
rw,soft,intr,noatime,x-gvfs-show
```

Cont ..

The above line will mount the home directory under `/mnt/nfs-share` directory in read and write mode. `x-gvfs-show` option will let you see the shared directory in your file manager.

Save and close the file. Then create the mount point.

```
sudo mkdir /mnt/nfs-share
```

Next, run the below command

```
sudo mount -a
```

Now you can access the server’s home directory in your file manager.

If you want to share your home directory with all computers in your home network, then add this line in `/etc/exports` file, supposing `192.168.1.1` is the IP address of your router.

```
/home/username      192.168.1.0/24(rw,sync,root_squash,subtree_check)
```

If you make any changes to `/etc/exports` file on the server side, then run the below command to tell NFS server to re-read `/etc/exports` file.

```
sudo exportfs -ra
```

Important commands for NFS

Some more important commands for **NFS**.

1. **showmount -e** : Shows the available **shares** on your local machine
2. **showmount -e <server-ip or hostname>**: Lists the available **shares** at the **remote** server
3. **showmount -d** : Lists all the **sub directories**
4. **exportfs -v** : Displays a list of shares **files** and **options** on a server
5. **exportfs -a** : Exports all shares listed in **/etc/exports**, or given name
6. **exportfs -u** : Unexports all shares listed in **/etc/exports**, or given name