NFS(NETWORK FILE SYSTEM/SHARING)

NFS stands for Network File System, and is a way to share files between machines as if they were on your local hard drive. Linux can be both an NFS server and an NFS client, which means that it can export filesystems to other systems, and mount filesystems exported from other machines.

Profile for NFS:

Package : nfs-utils

Daemons : rpc.nfsd, rpc.mountd, rpc.statd, rpc.lockd, rpc.rquotad

Script :/etc/init.d/nfs

Port number : 2049

Configuration File: /etc/exports

Other imp files:/var/lib/nfs/etab,/var/lib/nfs/rmtab

Steps to configure NFS server:

Step1: Install the NFS package using yum or rpm.

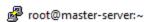
Step2: Create a dir or directory on partition and add some data in it.

Step3: Export the directory by editing /etc/exports file and using exportfs command

Step4: Restart the services and make it permanent.

Step1: Install the NFS package

- → Check whether the package is installed
- → #rpm –q nfs-utils



```
[root@master-server ~]# rpm -qa | grep nfs-utils
nfs-utils-lib-1.1.5-3.el6.x86_64
nfs-utils-1.2.3-7.el6.x86_64
[root@master-server ~]#
```

- → If it is not installed use following command to install it
- → #yum install nfs-utils* -y

Step2: Create a directory or create a partition and mount it and make a mount point and add data to it.

→ Create a partition, format it and mount it, access the mount point and add data to it

```
    root@master-server:~
```

- Update the partition table and format it
- → #partx –a /dev/vda

root@master-server:~

```
[root@master-server ~]# partx -a /dev/sdb1
last arg is not the whole disk
call: partx -opts device wholedisk
[root@master-server ~]#
```

mkfs.ext4 /dev/sdb1

```
[root@master-server ~]# mkfs.ext4 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
66384 inodes, 265064 blocks
13253 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=272629760
9 block groups
32768 blocks per group, 32768 fragments per group
7376 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 21 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[root@master-server ~]#
```

- → Create a directory and mount the partition over it and also make it permanent in /etc/fstab
- → # mkdir /nfs

```
🧬 root@master-server:∼
```

```
[root@master-server ~]# mkdir /nfs
[root@master-server ~]#
```

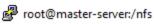
→ #mount /dev/sdb1 /nfs

```
root@master-server:~
```

→ Make entry in /etc/fstab

```
@ root@master-server ~] # tail -1 /etc/fstab
/dev/sdb1 /nfs ext4 defaults 0 0
[root@master-server ~] #
```

→ Access the mount point and add some data in it



```
[root@master-server ~]# cd /nfs/
[root@master-server nfs]# ls
lost+found yalla.txt
[root@master-server nfs]# cat yalla.txt
addedd nfs first lline in server side
[root@master-server nfs]#
```

Step3: Export the directory by editing /etc/exports file and using exportfs command

→ Edit the /etc/exports file

```
root@master-server:/nfs
```

```
[root@master-server nfs]# cat /etc/exports
/nfs 192.168.111.0/24(rw,sync)
[root@master-server nfs]#
```

/nfs : Name of the directory to be exported

192.168.111.0/24 : Range of network where directory can be mounted

To give permission to only one node, just give the IP ADDR

Of that node (ex: 192.168.111.0)

(rw,sync) : Mount options

The Mount options which can be used

Rw : Sets read/write permissions

Ro : Sets read-only permissions

Sync: Specifies that all changes must be written to disk before a Command completes no_wdelay: Forces the writing of changes immediately (useful for logs if Something crashes) root_squash: Prevent root users

- → Now run the exportfs command to export the directory
- → #exportfs –avr

```
root@master-server:/nfs
```

Options:

- -a: Exports or un-exports all directories
- -r: Reexport all directories
- -U: Unexports one or more directories
- -v: Provides verbose output

Step4: Restart the services and make it permanent.

#service nfs restart

```
root@master-server:/nfs
[root@master-server nfs]# service nfs restart
Shutting down NFS mountd:
Shutting down NFS daemon:
Shutting down NFS quotas:
Shutting down NFS services:
Starting NFS services:
Starting NFS quotas:
Starting NFS daemon:
Starting NFS mountd:
[root@master-server nfs]# service nfs status
rpc.svcqssd is stopped
rpc.mountd (pid 26436) is running...
nfsd (pid 26433 26432 26431 26430 26429 26428 26427 26426) is running...
rpc.rquotad (pid 26420) is running...
[root@master-server nfs]#
```

Check the directories which is exported in /var/lib/nfs/etab and /var/lib/nfs/rmtab

Note: stop the iptables services by using #service iptables stop and chkconfig iptables off.

Client side configuration for NFS mounting

Step1: Check and Install the NFS package if not installed

Step2: Start the NFS services

Step3: Check which directory is exported for this machine using showmount command

Step4: Make a directory and mount the NFS dir over it.

Step5: Add some data to it and check the same is updated on server side.

Step1: Check and Install the package for NFS

#rpm -q nfs-utils

```
    # root@master-server:~
```

```
[root@master-server ~]# rpm -qa | grep nfs-utils nfs-utils-lib-1.1.5-3.el6.x86_64 nfs-utils-1.2.3-7.el6.x86_64 [root@master-server ~]#
```

→ It is already installed, if it is not installed use yum install nfs-utils* -y

Step2: check and start the NFS services and make it permanent.

#service nfs start

#chkconfig nfs on

Step3: Check which directory is exported for this machine using showmount command

- → To check the exported directories from server the syntax is
- → #showmount -e <server IP address>

[root@master-server ~]#

```
[root@master-server ~] # showmount -e 192.168.111.135

Export list for 192.168.111.135:

/nfs 192.168.111.0/24

[root@master-server ~] # ______
```

Step4: Make a directory and mount NFS over it.

#mkdir /clientnfs

#mount -t nfs 192.168.111.135:/nfs /clientnfs

Step5: Add some data to it and check the same is updated on server side.

```
root@master-server:/clientnfs
```

```
[root@master-server clientnfs]# ls
lost+found yalla.txt
[root@master-server clientnfs]# pwd
/clientnfs
[root@master-server clientnfs]# touch newfile-clientside
touch: cannot touch `newfile-clientside': Permission denied
[root@master-server clientnfs]# sudo touch newfile-clientside
touch: cannot touch `newfile-clientside': Permission denied
[root@master-server clientnfs]#
```

- → Note that it is showing permission error because on server side the directory does not have write permissions neither for group nor for others.
- → Log into server and add write permission to NFS directory
- → Note: login to the nfs server. Give permission to /nfs directory

```
🧬 root@master-server:~
```

```
[root@master-server ~]# ls -ld /nfs/
drwxr-xr-x. 3 root root 4096 May 9 21:45 /nfs/
[root@master-server ~]# chmod 777 /nfs/
[root@master-server ~]# ls -ld /nfs/
drwxrwxrwx. 3 root root 4096 May 9 21:45 /nfs/
[root@master-server ~]#
```

→ Now , Again move back to client machine and try uploading some files

```
[root@master-server clientnfs]# touch newfile-clientside
[root@master-server clientnfs]# ls
lost+found newfile-clientside yalla.txt
[root@master-server clientnfs]#
```

→ To make it permanent mount edit /etc/fstab file as follows

###################END NFS