Setup and Configure NFS Mounts on Linux Server

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.

To setup we will need 2 servers. Master and Slave.

```
nfsmaster.server.com    192.168.33.135  # Hosts the NFS shared drive.
nfsslave.server.com     192.168.33.132  # Client to use the master shared drive.
```

NOTE: You can add the hostnames in the `/etc/hosts` file and user the hostnames in the configuration rather than IP addresses.

Steps to setup NFS.

1. Install NFS and rpcbind on master and slave servers. `nfs-utils nfs-utils-lib rpcbind`
2. Configure NFS on master server.
3. Configure mount points on slave server.
4. Mount NFS on slave server.

Installing NFS Server and NFS Slave

We need to install NFS packages using `yum`.

```
[root@nfsmaster ~]# yum install nfs-utils nfs-utils-lib
[root@nfsmaster ~]# yum install rpcbind
```

Make sure to install rpcbind and start it first. Now start the services on both machines.

```
[root@nfsmaster ~]# /etc/init.d/rpcbind start
```

NOTE: Start rpcbind first else you will get the below error.

```
[root@nfsmaster /]# service nfs start
Starting NFS services: [ OK ]
Starting NFS quotas: Cannot register service: RPC: Unable to receive; errno = Connection refused
rpc.rquotad: unable to register (RQUOTAPROG, RQUOTAVERS, udp).
[FAILED]
Starting NFS daemon: [FAILED]
```

After starting rpcbind check the rpcinfo. should look similar as below.

```
[root@nfsmaster /]# rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  rpcbindper
    100000    3   tcp    111  rpcbindper
    100000    2   tcp    111  rpcbindper
    100000    4   udp    111  rpcbindper
    100000    3   udp    111  rpcbindper
    100000    2   udp    111  rpcbindper
    100024    1   udp  33472  status
    100024    1   tcp  40795  status
```

Now we start the NFS service.

```
[root@nfsmaster ~]# service nfs start
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
Starting RPC idmapd:                                       [  OK  ]
```

Setting services to start on reboot.

```
[root@nfsmaster ~]# chkconfig rpcbind on
[root@nfsmaster ~]# chkconfig nfs on
```

NOTE : Starting services on both the machines

Setting Up the NFS Server

First we will be configuring the NFS server.

Configure Export directory

For sharing a directory with NFS, we need to make an entry in `/etc/exports` configuration file. Lets create a directory to be shared across the network. We are creating a common shared directory for all the `login` users.

```
[root@nfsmaster ~]# mkdir /export/home
```

Now we need to make an entry in `/etc/exports` and restart the services to make our directory shared in the network.

```
[root@nfsmaster ~]# vi /etc/exports
# directory  Slave-IP  (permissions on the directory)
/export/home 192.168.33.132(rw,sync,no_root_squash)
/export/home 192.168.33.135(rw,sync,no_root_squash)
/export/home 192.168.33.131(rw,sync,no_root_squash)
```

In the above example, there is a directory as `/export/home`. This is shared with Slaves with IP `192.168.33.132` with read and write (rw) privilege, you can also use `hostname` of the Slave in the place of IP.

In the above `/etc/exports` we have 3 clients which are allowed to access the shared mount.

```
1. 192.168.33.132 - nfsslave    (slave)
2. 192.168.33.135 - nfsmaster   (master)
3. 192.168.33.131 - nfsslave2   (another slave)
```

Now restart `NFS` service.

```
[root@nfsmaster home]# service nfs restart
Shutting down NFS daemon:                                  [  OK  ]
Shutting down NFS mountd:                                  [  OK  ]
Shutting down NFS quotas:                                  [  OK  ]
Shutting down NFS services:                                [  OK  ]
Shutting down RPC idmapd:                                  [  OK  ]
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
Starting RPC idmapd:                                       [  OK  ]
```

NFS Options

Some other options we can use in `/etc/exports` file for file sharing is as follows.

1. **`ro`**: With the help of this option we can provide read only access to the shared files i.e Slave will only be able to read.
2. **`rw`**: This option allows the Slave server to both read and write access within the shared directory.
3. **`sync`**: Sync confirms requests to the shared directory only once the changes have been committed.
4. **`no_subtree_check`**: This option prevents the subtree checking. When a shared directory is the subdirectory of a larger file system, nfs performs scans of every directory above it, in order to verify its permissions and details. Disabling the subtree check may increase the reliability of NFS, but reduce security.
5. **`no_root_squash`**: This phrase allows root to connect to the designated directory.

For more options with `/etc/exports`, you are recommended to read the man pages for export.

Configuring and Setup of NFS Slave.

After the server is configured we need to mount the shared drive on the slave.

1. Create a directory in slave to mount the shared drive.
2. Mount the drive to the newly created mount point.
3. update `/etc/fstab` to make changes permanent.

First lets check the mount points on the server.

```
[root@nfsslave ~]# showmount -e 192.168.33.135

Export list for 192.168.33.135:
/export/home 192.168.33.132,192.168.33.135,192.168.33.131
```

Command above shows that the directory `/export/home` is available at `192.168.33.135` and ready to be shared with `132`,`131` and `135` servers.

Create Mount Point and Mount Shared NFS Directory.

Creating a new mount point.

```
[root@nfsslave ~]# mkdir -p /nfs_client_mount
```

To mount that shared NFS directory we can use following mount command.

```
[root@nfsslave ~]# mount -t nfs 192.168.33.135:/export/home /nfs_client_mount
```

The above command will mount that shared directory in `/nfs_client_mount` on the slave server.

Update `/etc/fstab` file to automount on reboot.

```
[root@nfsslave ~]# vi /etc/fstab
```

Add the following new line as shown below.

```
192.168.33.135:/export/home /nfs_client_mount  nfs defaults 0 0
```

Checking mount points.

```
[root@nfsslave home]# mount -a
[root@nfsslave home]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
vmware-vmblock on /var/run/vmblock-fuse type fuse.vmware-vmblock
                                              (rw,nosuid,nodev,default_permissions,allow_other)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
192.168.33.135:/export/home on /nfs_client_mount type nfs
                                        (rw,vers=4,addr=192.168.33.135,clientaddr=192.168.33.132)
```

Testing by creating a TEST directory on slave which will be created in `/export/home` on master, which hosts the shared NFS drive.

```
[root@nfsslave ~]# df -h -F nfs
Filesystem            Size  Used Avail Use% Mounted on
192.168.33.135:/export/home
                       18G  5.1G   12G  31% /nfs_client_mount
[root@nfsslave ~]#
[root@nfsslave ~]# cd /nfs_client_mount/
[root@nfsslave home]# mkdir TEST
```

Check directory creation on Master.

```
[root@nfsmaster ~]# cd /export/home/
[root@nfsmaster home]# ls
TEST
```

We can see the newly created directory. We are good. :)

Removing the NFS Mount

If we need to unmount the NFS directory on slave.

```
[root@nfsslave ~]# umount /nfs_client_mount
```

Check if the NFS is unmounted.

```
[root@nfsslave ~]# df -h -F nfs
df: no file systems processed
```

No NFS available.

Important commands for NFS

Some more important commands for NFS.

```
showmount -e : Shows the available shares on your local machine
showmount -e <server-ip or hostname>: Lists the available shares at the remote server
showmount -d : Lists all the sub directories
exportfs -v : Displays a list of shares files and options on a server
exportfs -a : Exports all shares listed in /etc/exports, or given name
exportfs -u : Unexports all shares listed in /etc/exports, or given name
exportfs -r : Refresh the server's list after modifying /etc/exports
```

Important Links.

Techmint

Redhat

MaazanJum