Configuring NFS server

1. Install the required nfs packages if not already installed on the server :

# rpm -qa | grep nfs-utils

# yum install nfs-utils rpcbind

2. Enable the services at boot time:

# systemctl enable nfs-server

# systemctl enable rpcbind

# systemctl enable nfs-lock

In RHEL7.1 (nfs-utils-1.3.0-8.el7) enabling nfs-lock does not work (No such file or directory). it does not need to be enabled since rpc-statd.service is static.

# systemctl enable nfs-idmap

In RHEL7.1 (nfs-utils-1.3.0-8.el7) this does not work (No such file or directory). it does not need to be enabled since nfs-idmapd.service is static.

3. Start the NFS services:

# systemctl start rpcbind

# systemctl start nfs-server

# systemctl start nfs-lock

# systemctl start nfs-idmap

4. Check the status of NFS service:

# systemctl status nfs

5. Create a shared directory:

# mkdir /test

6. Export the directory. The format of the /etc/exports file is :

dir client1 (options) [client2(options)...]

Client options include (defaults are listed first) :  
**ro / rw** :  
a) ro : allow clients read only access to the share.  
b) rw : allow clients read write access to the share.  
**sync / async** :  
a) sync : NFS server replies to request only after changes made by previous request are written to disk.  
b) async : specifies that the server does not have to wait.  
**wdelay / no\_wdelay**  
a) wdelay : NFS server delays committing write requests when it suspects another write request is imminent.  
b) no\_wdelay : use this option to disable to the delay. no\_wdelay option can only be enabled if default **sync** option is enabled.  
**no\_all\_squash / all\_squash** :  
a) no\_all\_squash : does not change the mapping of remote users.  
b) all\_squash : to squash all remote users including root.  
**root\_squash / no\_root\_squash** :  
a) root\_squash : prevent root users connected remotely from having root access. Effectively squashing remote root privileges.  
b) no\_root\_squash : disable root squashing.

Example :

# vi /etc/exports

/test \*(rw)

7. Exporting the share :

# exportfs -r

-r re-exports entries in /etc/exports and sync /var/lib/nfs/etab with /etc/exports. The /var/lib/nfs/etab is the master export table. Other options that can be used with exportfs command are :

-a : exports entries in /etc/exports but do not synchronize with /var/lib/nfs/etab

-i : ignore entries in /etc/exports and uses command line arguments.

-u : un-export one or more directories

-o : specify client options on command line

8. Restart the NFS service:

# systemctl restart nfs-server

Configuring NFS client

1. Install the required nfs packages if not already installed on the server :

# rpm -qa | grep nfs-utils

# yum install nfs-utils

2. Use the mount command to mount exported file systems. Syntax for the command:

mount -t nfs -o options host:/remote/export /local/directory

Eample :

# mount -t nfs -o ro,nosuid remote\_host:/home /remote\_home

This example does the following:  
– It mounts /home from remote host (remote\_host) on local mount point /remote\_home.  
– File system is mounted read-only and users are prevented from running a setuid program (-o ro,nosuid options).

3. Update /etc/fstab to mount NFS shares at boot time.

# vi /etc/fstab

remote\_host:/home /remote\_home nfs ro,nosuid 0 0

Firewalld services to be active on NFS server

For the NFS server to work, enable the nfs, mountd, and rpc-bind services in the relevant zone in the firewall-config application or using firewall-cmd :

# firewall-cmd --add-service=nfs --zone=internal --permanent

# firewall-cmd --add-service=mountd --zone=internal --permanent

# firewall-cmd --add-service=rpc-bind --zone=internal --permanent