Network File System (NFS)

NFS allows local partitions to be shared **over the network.** Clients can access these shared directories as if they are local partitions.

NFS Key Details

Component	Description
Package	Nfs-utils (server & Client)
Default Port	2049
Service	nfs-server
Daemons	nfsd, mountd, nfslogd, nfsstatd, nfslockd
Config Files	/etc/nfs.conf, /etc/nfsmount.conf

Daemon Responsibilities

- **nfsd** – Handles exports, data locking/unlocking.
- **mountd** – Processes mount requests.
- **nfslogd** – Logs NFS activities to **/var/log/messages.**
- **nfsstatd** – Monitors network connectivity between NFS server and clients.
- **nfslockd** – Locks/unlocks data transfers based on statd signals to prevent corruption.

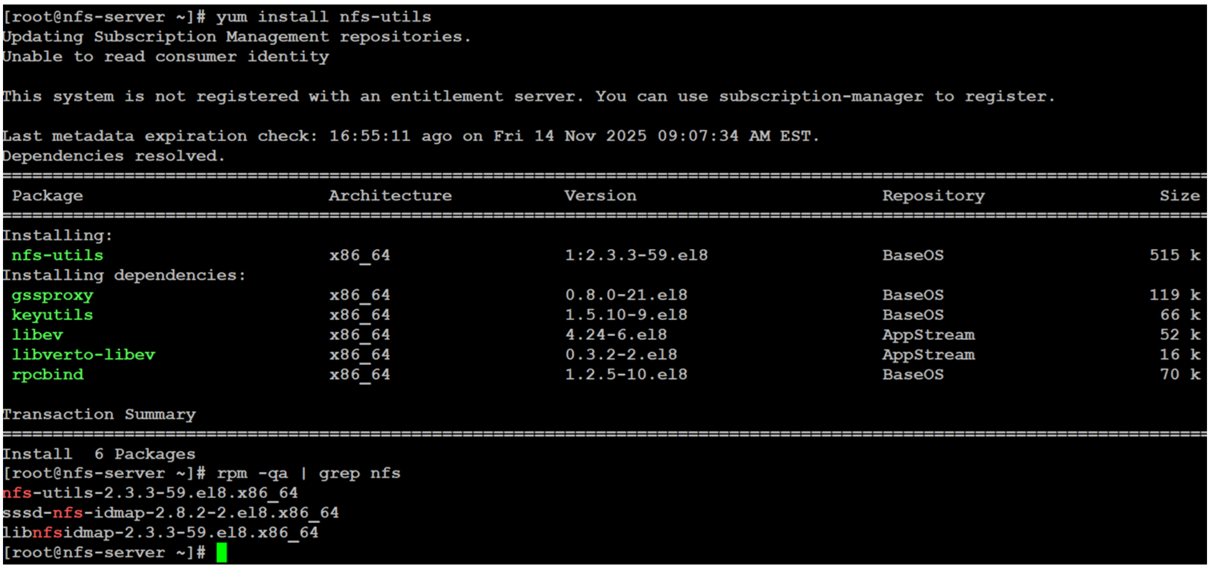
NFS Server Configuration

Server 1 (NFS Server): 192.168.23.145
Server 2 (NFS Client): 192.168.23.146

Steps on NFS Server

Step 1: Install NFS package

Command: **yum install nfs-utils**



```
[root@nfs-server ~]# yum install nfs-utils
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 16:55:11 ago on Fri 14 Nov 2025 09:07:34 AM EST.
Dependencies resolved.
=======================================================================================================================
 Package                        Architecture          Version                        Repository                  Size
=======================================================================================================================
Installing:
 nfs-utils                      x86_64                1:2.3.3-59.el8                 BaseOS                     515 k
Installing dependencies:
 gssproxy                       x86_64                0.8.0-21.el8                   BaseOS                     119 k
 keyutils                       x86_64                1.5.10-9.el8                   BaseOS                      66 k
 libev                          x86_64                4.24-6.el8                     AppStream                   52 k
 libverto-libev                 x86_64                0.3.2-2.el8                    AppStream                   16 k
 rpcbind                        x86_64                1.2.5-10.el8                   BaseOS                      70 k

Transaction Summary
=======================================================================================================================
Install  6 Packages
[root@nfs-server ~]# rpm -qa | grep nfs
nfs-utils-2.3.3-59.el8.x86_64
sssd-nfs-idmap-2.8.2-2.el8.x86_64
libnfsidmap-2.3.3-59.el8.x86_64
[root@nfs-server ~]#
```

Step 2: Create shared directory

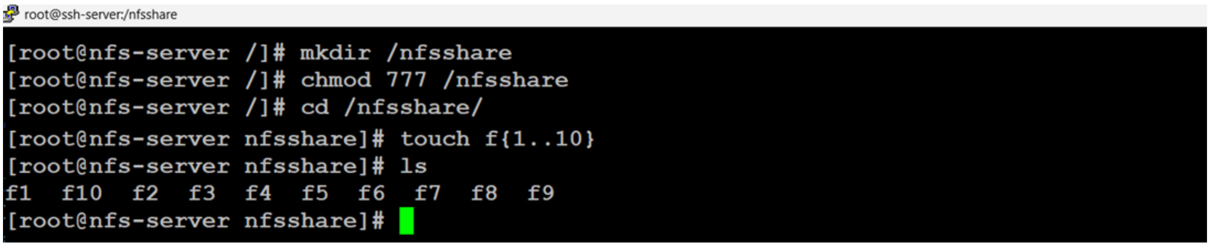
Commands:

mkdir /nfsshare

chmod 777 /nfsshare

cd /nfsshare

touch f{1..20}



```
[root@nfs-server /]# mkdir /nfsshare
[root@nfs-server /]# chmod 777 /nfsshare
[root@nfs-server /]# cd /nfsshare/
[root@nfs-server nfsshare]# touch f{1..10}
[root@nfs-server nfsshare]# ls
f1  f10  f2  f3  f4  f5  f6  f7  f8  f9
[root@nfs-server nfsshare]#
```

Step 3: Export the directory

Commands:

vi /etc/exports

        /nfsshare *(rw,sync)

Step 4: Start and enable NFS services

Commands:

systemctl start nfs-server

systemctl enable nfs-server

systemctl status nfs-server

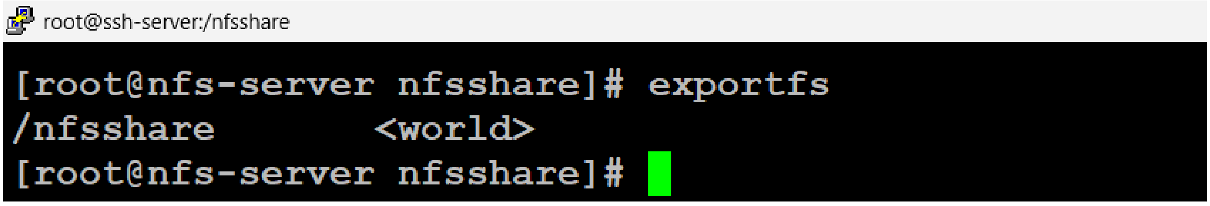
Step 5: Allow NFS Through Firewall and SELinux

Commands:

```
firewall-cmd --permanent --add-service=nfs
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --permanent --add-service=mountd
firewall-cmd --reload
firewall-cmd --list-all

setsebool -P nfs_export_all_rw on
setsebool -P nfs_export_all_ro on
setsebool -P use_nfs_home_dirs on
semanage port -l | grep nfs
```

Step 6: Verify exported shares
Command: exportfs


root@ssh-server:/nfsshare

```
[root@nfs-server nfsshare]# exportfs
/nfsshare       <world>
[root@nfs-server nfsshare]# 
```

NFS Client Configuration

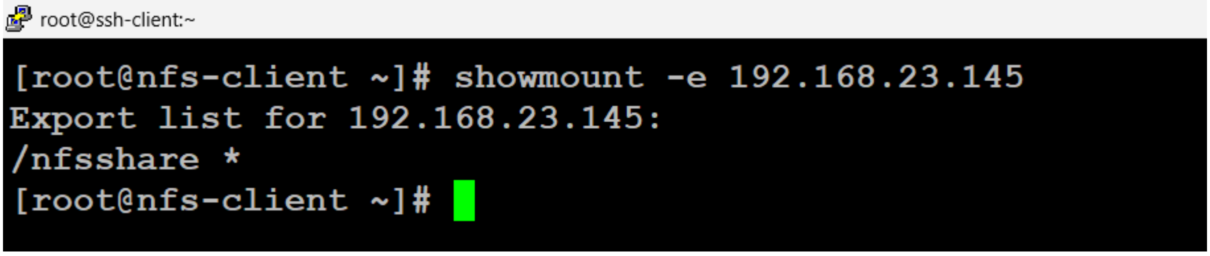
Step 1: Install NFS package

Command: yum install nfs-utils

Step 2: Check connectivity
Commands:

```
ping 192.168.23.145
nc -vz 192.168.23.145 2049
```

Step 3: List exported file systems
Command: showmount -e 192.168.23.145


root@ssh-client:~

```
[root@nfs-client ~]# showmount -e 192.168.23.145
Export list for 192.168.23.145:
/nfsshare *
[root@nfs-client ~]# 
```

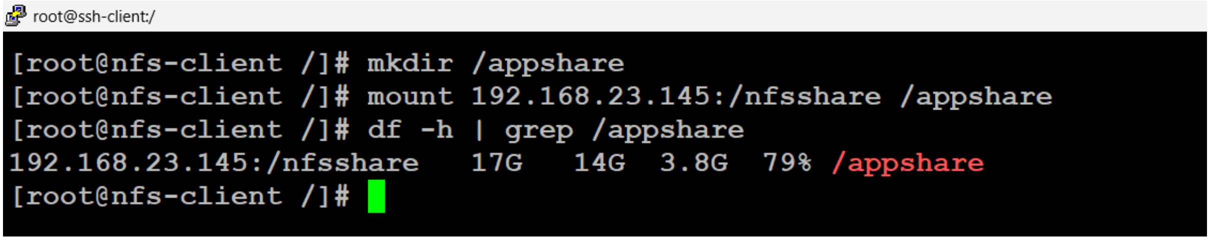
Step 4: Mount the NFS share

Commands:

mkdir /appshare

mount 192.168.23.145:/nfsshare /appshare

df -h



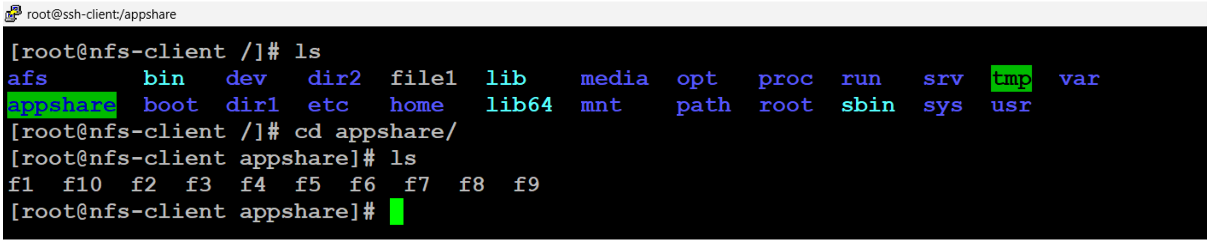
```
[root@nfs-client /]# mkdir /appshare
[root@nfs-client /]# mount 192.168.23.145:/nfsshare /appshare
[root@nfs-client /]# df -h | grep /appshare
192.168.23.145:/nfsshare   17G   14G  3.8G  79% /appshare
[root@nfs-client /]#
```

Step 5: Permanent mount entry

Command:

vi /etc/fstab

192.168.23.145:/nfsshare /appshare nfs defaults 1 1



```
[root@nfs-client /]# ls
afs       bin   dev   dir2  file1  lib    media  opt   proc  run   srv  tmp  var
appshare  boot  dir1  etc   home   lib64  mnt    path  root  sbin  sys  usr
[root@nfs-client /]# cd appshare/
[root@nfs-client appshare]# ls
f1  f10  f2  f3  f4  f5  f6  f7  f8  f9
[root@nfs-client appshare]#
```

NFS Export Options

Example entries in /etc/exports:

/sapdata *(rw,sync)

/sapdata 192.168.*.*(rw,sync)

/sapdata 192.168.23.146(rw,sync)

/sapdata 192.168.23.146(ro,sync)

Option meanings:

- **rw – Read and write.**
- **ro – Read-only.**
- **sync – Data confirmed only after full write (safer).**

- **async – Faster but less reliable.**
- **root_squash – Treat remote root users as normal users (recommended for security).**
- **no_root_squash – Allow remote root full access.**

Sync vs Async Behavior

- **sync –** Waits until full data copy completes before client acknowledgment.
- **async –** Immediately acknowledges client; may risk data loss if server fails mid-transfer.
- Recommended for safety, choose **sync**
- Recommended for performance, choose **async**