

## [Configure linux nfs server step by step guide example and implementation](#)

NFS, or **Network File System**, is a server-client protocol for sharing files between computers on a common network. NFS enables you to mount a file system on a remote computer as if it were local to your own system. You can then directly access any of the files on that remote file system. The server and client do not have to use the same operating system. The client system just needs to be running an **NFS client** compatible with the **NFS server**.

For example **NFS server** could be a Linux system and Unix could be a client. But it can't be a window system because window is not NFS compatible. The NFS server exports one or more directories to the client systems, and the client systems mount one or more of the shared directories to local directories called mount points. After the share is mounted, all I/O operations are written back to the server, and all clients notice the change as if it occurred on the local filesystem.

A manual refresh is not needed because the client accesses the remote filesystem as if it were local. because access is granted by IP address, a username and password are not required. However, there are security risks to consider because the **NFS server** knows nothing about the users on the client system.

**Exam question 1** Some users home directory is shared from your system. Using **showmount -e localhost** command, the shared directory is not shown. Make access the shared users home directory

**Exam question 2** The System you are using is for **NFS (Network File Services)**. Some important data are shared from your system. Make automatically start the nfs and portmap services at boot time

**Exam question 3** Share **/data** directory using NFS only to **192.168.0.0/24** members. These hosts should get read and write access on shared directory.

### **Configure nfs server**

In this example we will configure a nfs server and will mount shared directory from client side.

For this example we are using two systems one linux server one linux clients . To complete these per quest of nfs server Follow this link

[basic network configurations Example and Implementations ip configurations hosts files portmap xinetd services](#)

- A linux server with ip address 192.168.0.254 and hostname Server
- A linux client with ip address 192.168.0.1 and hostname Client1
- Updated /etc/hosts file on both linux system
- Running portmap and xinetd services
- Firewall should be off on server

We suggest you to review that article before start configuration of nfs server. Once you have completed the necessary steps follow this guide.

Three rpm are required to configure nfs server. **nfs**, **portmap**, **xinetd** check them if not found then install

```
[root@Server ~]# rpm -qa nfs*
nfs-utils-1.0.9-24.el5
nfs-utils-lib-1.0.8-7.2.z2
[root@Server ~]# rpm -qa portmap*
portmap-4.8-65.2.2.1
[root@Server ~]# rpm -qa xinetd*
xinetd-2.3.14-10.el5
[root@Server ~]# _
```

Now check **nfs**, **portmap**, **xinetd** service in system service it should be on

```
#setup
Select System service
from list
[*]portmap [*]xinetd [*]nfs
```

Now restart **xinetd** and **portmap** service

```
[root@Server ~]# service portmap restart
Stopping portmap: [ OK ]
Starting portmap: [ OK ]
[root@Server ~]# service xinetd restart
Stopping xinetd: [ OK ]
Starting xinetd: [ OK ]
[root@Server ~]# _
```

To keep on these services after reboot on then via **chkconfig** command

```
[root@Server ~]# chkconfig portmap on
[root@Server ~]# chkconfig xinetd on
[root@Server ~]# _
```

After **reboot** verify their status. It must be in **running** condition

```
[root@Server ~]# service portmap status
portmap (pid 3430) is running...
[root@Server ~]# service xinetd status
xinetd (pid 3462) is running...
[root@Server ~]# _
```

now create a **/data** directory and grant full permission to it

```
[root@Server ~]# mkdir /data
[root@Server ~]# chmod 777 /data
[root@Server ~]# _
```

now open **/etc/exports** file

```
[root@Server ~]# vi /etc/exports _
```

share **data** folder for the network of **192.168.0.254/24** with read and write access

```
/data 192.168.0.0/24(rw,sync)_
```

save file with **:wq** and exit

now restart the **nfs service** and also on it with **chkconfig**

```
[root@Server ~]# service nfs restart
Shutting down NFS mountd: [ OK ]
Shutting down NFS daemon: nfsd: last server has exited
nfsd: unexporting all filesystems [ OK ]
Shutting down NFS quotas: [ OK ]
Shutting down NFS services: [ OK ]
Starting NFS services: [ OK ]
Starting NFS quotas: [ OK ]
Starting NFS daemon: NFSD: Using /var/lib/nfs/v4recovery as the NFSv4
very directory
NFSD: starting 90-second grace period [ OK ]
Starting NFS mountd: [ OK ]
[root@Server ~]# chkconfig nfs on
[root@Server ~]# _
```

also restart **nfs daemons** with **expotfs**

```
[root@Server ~]# exportfs -r
[root@Server ~]# _
```

verify with **showmount** command that you have successfully shared **data folder**

```
[root@Server ~]# showmount -e
Export list for Server:
/data 192.168.0.0/24
[root@Server ~]# _
```

## configure client system

ping from **nfs server** and check the **share folder**

```
[root@Client1 ~]# showmount -e 192.168.0.254
Export list for 192.168.0.254:
/data 192.168.0.0/24
[root@Client1 ~]# _
```

now **mount** this share folder on **mnt** mount point. To test this share folder change directory to **mnt** and create a **test file**

```
[root@Client1 ~]# mount -t nfs 192.168.0.254:/data /mnt
[root@Client1 ~]# cd /mnt
[root@Client1 mnt]# cat > test
this is test file created on client side
[root@Client1 mnt]# _
```

After use you should always **unmount** from **mnt** mount point

```
[root@Client1 mnt]# cd
[root@Client1 ~]# umount /mnt
[root@Client1 ~]# _
```

In this way you can use **shared folder**. But this share folder will be available till system is **up**. It will not be available after **reboot**. To keep it available after reboot make its entry in **fstab**

create a **mount point**, by making a directory

```
[root@Client1 ~]# mkdir /temp_
```

now open **/etc/fstab** file

```
[root@Client1 ~]# vi /etc/fstab _
```

make entry for **nfs** shared directory and define **/temp** to mount point

```
LABEL=/                /                ext3    defaults    1 1
LABEL=/home            /home           ext3    defaults    1 2
LABEL=/boot            /boot          ext3    defaults    1 2
tmpfs                  /dev/shm        tmpfs    defaults    0 0
devpts                 /dev/pts        devpts   gid=5,mode=620 0 0
sysfs                  /sys            sysfs    defaults    0 0
proc                   /proc           proc     defaults    0 0
LABEL=SWAP-sda3        swap            swap     defaults    0 0
192.168.0.254:/data    /temp           nfs      defaults    0 0
```

save the with **:wq** and exit reboot the system with **reboot -f** command

```
#reboot -f
```

after reboot check **/temp** directory it should show all the **shared data**

```
[root@Client1 ~]# cd /temp
[root@Client1 temp]# ls
test
[root@Client1 temp]# _
```