

NFS - Quick and Dirty Setup

Known to work on Slackware 14, 14.1 and 14.2

Assumptions

- 1) This HOWTO assumes that you are using a **vanilla install of Slackware** and have **not changed** the default HOSTS_ALLOW, HOSTS_DENY, or firewall rules.
- 2) For this example, the shared directory on the server will be /nfs_share
- 3) For this example, the mount point of the NFS share will be /mnt/nfs_share
- 4) We want anyone on our subnet (192.168.1.x) to have RW access the share

Overview

- 1) Setup the server
- 2) Setup the client
- 3) Mount the NFS directories on the client

Server Setup

Add the NFS shares to the /etc/exports file

```
vi /etc/exports
```

add:

[/etc/exports](#)

```
/nfs_share 192.168.1.1/24(rw,sync,no_subtree_check)
```

Start the NFS and RPC daemons

```
chmod 755 /etc/rc.d/rc.nfsd
chmod 755 /etc/rc.d/rc.rpc
/etc/rc.d/rc.nfsd start
/etc/rc.d/rc.rpc start
```

Export the shares

```
exportfs -a
```

Check to see if the shares are being shared

```
exportfs
```

Client Setup

Create the mount point

```
mkdir /mnt/nfs_share
```

Start the RPC daemon

```
chmod 755 /etc/rc.d/rc.rpc  
/etc/rc.d/rc.rpc start
```

Mounting

On the CLIENT machine, you have 3 options here: manually mounting, auto-mount at boot, or semi-auto mount

MANUALLY MOUNT

```
mount my.nfs.server:/nfs_share /mnt/nfs_share
```

AUTO-MOUNT AT BOOT

Add the mount command to `/etc/fstab`

```
vi /etc/fstab
```

add:

[/etc/fstab](#)

```
my.nfs.server:/nfs_share /mnt/nfs_share nfs rw,defaults 0 0
```

SEMI-AUTO-MOUNT

Add the mount command to `/etc/fstab`

```
vi /etc/fstab
```

add:

[/etc/fstab](#)

```
my.nfs.server:/nfs_share /mnt/nfs_share nfs rw,noauto 0 0
```

then when you want to mount, just run:

```
mount /mnt/nfs_share
```

NOTE ABOUT AUTO_MOUNTING

If you mount at boot and the server machine is unavailable, it will cause your client machine to take a long time to boot as the NFS client will make multiple attempts to connect and you will have to wait for it to time-out for each attempt.

For The Impatient

All the steps with no explanation:

SERVER:

```
echo "/SHARED_DIR YOUR_SUBNET/24(rw, sync, no_subtree_check)" >> /etc/exports
chmod 755 /etc/rc.d/rc.nfsd
chmod 755 /etc/rc.d/rc.rpc
/etc/rc.d/rc.nfsd start
/etc/rc.d/rc.rpc start
exportfs -a
```

CLIENT:

```
mkdir /mnt/nfs_share
chmod 755 /etc/rc.d/rc.rpc
/etc/rc.d/rc.rpc start
echo "YOUR_NFS_SERVER:/SHARED_DIR /CLIENT_MOUNT_POINT nfs rw,defaults 0 0"
>> /etc/fstab
mount /CLIENT_MOUNT_POINT
```

Problems/Solutions

PROBLEM: NFS shares won't mount on client box with a "Stale NFS file handle" error.

SOLUTION:

1) Forcibly unmount the directory on the CLIENT machine (if mounted):

```
umount -f /mnt/nfs_share
```

2) Flush the NFS registry on the SERVER machine:

```
exportfs -f
```

3) Remount the NFS share

4) No Root Squash: There are many options for NFS and I want to keep this article short but effective so I am leaving out many of the various configuration items that you could do. However there is one option that is worth mentioning, **no_root_squash**. By default NFS will downgrade any files created with the root permissions to the nobody user. This is a security feature that prevents privileges from being shared unless specifically requested. If I create a file as the root user on the client on the NFS share, by default that file is owned by the nobody user.

```
root@client:~# touch /shared/nfs1/file2
root@server:/nfs# ls -la file2
-rw-r--r-- 1 nobody nogroup 0 Nov 18 18:06 file2
```

Sometimes it is important to share files that are owned as root with the proper permissions, in these cases this can be done by simply adding the **no_root_squash** attribute to the **/etc/exports** configuration.

Edit the /etc/exports file:

```
/nfs_share 192.168.1.1/24(rw, sync, no_root_squash)
```

Sources

* Originally written by [arfon](#)

[howtos](#), [software](#), [nfs](#), [slackware 13.37](#), [slackware 14.0](#), [slackware 14.1](#), [author arfon](#)

From: <https://docs.slackware.com/> - **SlackDocs**

Permanent link: https://docs.slackware.com/howtos:network_services:nfs-quick_and_dirty_setup

Last update: **2019/10/08 04:38 (UTC)**

