By Falko Timme
Published: 2009-04-29 17:58

Setting Up An NFS Server And Client On Debian Lenny

Version 1.0
Author: Falko Timme <ft [at] falkotimme [dot] com>
Last edited 03/12/2009

This guide explains how to set up an NFS server and an NFS client on Debian Lenny. NFS stands for ***Network File System***; through NFS, a client can access (read, write) a remote share on an NFS server as if it was on the local hard disk.

I do not issue any guarantee that this will work for you!

1 Preliminary Note

I'm using two Debian systems here:

- NFS Server: *server.example.com*, IP address: *192.168.0.100*
- NFS Client: *client.example.com*, IP address: *192.168.0.101*

2 Installing NFS

server:

On the NFS server we run:

```
apt-get install nfs-kernel-server nfs-common portmap
```

client:

On the client we can install NFS as follows:

```
apt-get install nfs-common portmap
```

3 Exporting Directories On The Server

server:

I'd like to make the directories `/home` and `/var/nfs` accessible to the client; therefore we must "export" them on the server.

When a client accesses an NFS share, this normally happens as the user `nobody`. Usually the `/home` directory isn't owned by `nobody` (and I don't recommend to change its ownership to `nobody`!), and because we want to read and ***write*** on `/home`, we tell NFS that accesses should be made as root (if our `/home` share was read-only, this wouldn't be necessary). The `/var/nfs` directory doesn't exist, so we can create it and change its ownership to `nobody` and `nogroup`:

```
mkdir /var/nfs

chown nobody:nogroup /var/nfs
```

Now we must modify `/etc/exports` where we "export" our NFS shares. We specify `/home` and `/var/nfs` as NFS shares and tell NFS to make accesses to `/home` as root (to learn more about `/etc/exports`, its format and available options, take a look at

```
man 5 exports
```

)

```
vi /etc/exports
```

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home           192.168.0.101(rw,sync,no_root_squash,no_subtree_check)
/var/nfs        192.168.0.101(rw,sync,no_subtree_check)
```

(The `no_root_squash` option makes that `/home` will be accessed as root.)

Whenever we modify /etc/exports, we must run

```
exportfs -a
```

afterwards to make the changes effective.

4 Mounting The NFS Shares On The Client

client:

First we create the directories where we want to mount the NFS shares, e.g.:

```
mkdir -p /mnt/nfs/home

mkdir -p /mnt/nfs/var/nfs
```

Afterwards, we can mount them as follows:

```
mount 192.168.0.100:/home /mnt/nfs/home


mount 192.168.0.100:/var/nfs /mnt/nfs/var/nfs
```

You should now see the two NFS shares in the outputs of

```
df -h
```

```
client:~# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-root   19G  676M   17G   4% /
tmpfs                 253M     0  253M   0% /lib/init/rw
udev                   10M   80K   10M   1% /dev
tmpfs                 253M     0  253M   0% /dev/shm
/dev/sda1             471M   20M  427M   5% /boot
192.168.0.100:/home    29G  684M   27G   3% /mnt/nfs/home
192.168.0.100:/var/nfs
                       29G  684M   27G   3% /mnt/nfs/var/nfs
client:~#
```

and

```
mount
```

```
client:~# mount
  /dev/mapper/vg0-root on / type ext3 (rw,errors=remount-ro)
  tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
  proc on /proc type proc (rw,noexec,nosuid,nodev)
  sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
  udev on /dev type tmpfs (rw,mode=0755)
  tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
  devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
  /dev/sda1 on /boot type ext3 (rw)
  192.168.0.100:/home on /mnt/nfs/home type nfs (rw,addr=192.168.0.100)
  192.168.0.100:/var/nfs on /mnt/nfs/var/nfs type nfs (rw,addr=192.168.0.100)
client:~#
```

5 Testing

On the client, you can now try to create test files on the NFS shares:

client:

```
touch /mnt/nfs/home/test.txt

touch /mnt/nfs/var/nfs/test.txt
```

Now go to the server and check if you can see both test files:

server:

```
ls -l /home/
```

```
server:~# ls -l /home/
total 4
drwxr-xr-x 2 administrator administrator 4096 2009-02-16 13:18 administrator
-rw-r--r-- 1 root          root             0 2009-03-12 17:08 test.txt
server:~#
```

```
ls -l /var/nfs
```

```
server:~# ls -l /var/nfs
total 0
-rw-r--r-- 1 nobody nogroup 0 2009-03-12 17:08 test.txt
server:~#
```

(Please note the different ownerships of the test files: the `/home` NFS share gets accessed as root, therefore `/home/test.txt` is owned by root; the `/var/nfs` share gets accessed as *nobody*, therefore `/var/nfs/test.txt` is owned by *nobody*.)

6 Mounting NFS Shares At Boot Time

Instead of mounting the NFS shares manually on the client, you could modify `/etc/fstab` so that the NFS shares get mounted automatically when the client boots.

client:

Open `/etc/fstab` and append the following lines:

```
vi /etc/fstab
```

```
[...]
192.168.0.100:/home  /mnt/nfs/home   nfs      rw,sync,hard,intr  0     0
192.168.0.100:/var/nfs  /mnt/nfs/var/nfs   nfs      rw,sync,hard,intr  0     0
```

Instead of `rw,sync,hard,intr` you can use different mount options. To learn more about available options, take a look at

```
man nfs
```

To test if your modified `/etc/fstab` is working, reboot the client:

```
reboot
```

After the reboot, you should find the two NFS shares in the outputs of

```
df -h
```

```
client:~# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg0-root   19G  676M   17G   4% /
tmpfs                 253M     0  253M   0% /lib/init/rw
udev                   10M   80K   10M   1% /dev
tmpfs                 253M     0  253M   0% /dev/shm
/dev/sda1             471M   20M  427M   5% /boot
192.168.0.100:/home    29G  684M   27G   3% /mnt/nfs/home
192.168.0.100:/var/nfs
                       29G  684M   27G   3% /mnt/nfs/var/nfs
client:~#
```

and

```
mount
```

```
client:~# mount
  /dev/mapper/vg0-root on / type ext3 (rw,errors=remount-ro)
  tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
  proc on /proc type proc (rw,noexec,nosuid,nodev)
  sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
  udev on /dev type tmpfs (rw,mode=0755)
  tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
  devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
  /dev/sda1 on /boot type ext3 (rw)
  192.168.0.100:/home on /mnt/nfs/home type nfs (rw,sync,hard,intr,addr=192.168.0.100)
  192.168.0.100:/var/nfs on /mnt/nfs/var/nfs type nfs (rw,sync,hard,intr,addr=192.168.0.100)
client:~#
```

7 Links

- Linux NFS: ***http://nfs.sourceforge.net***
- Debian: ***http://www.debian.org***