# Build NFS Server

This is a total rewrite since FreeBSD and ZFS have the ability to export nfs directly. You can always not use the sharenfs function in zfs (ie, sharenfs=off) and use the standard way of exporting. ZFS running nfs actually is not as pretty as the old way, but it is very efficient for maintenance.

The following assumes you have a zpool named *storage* and you will create a tree of nfs exportable directories under that.

For our needs, we want some exports that will go to our Linux Xen machines, which will have common data between them. For example, our xen configuration files should be available across all of our DOM0's. That allows us to migrate from one DOM0 to another. Additionally, we need some space to store installer images (iso's), and another one where we just put some stuff that is handy to have on DOM0's.

We also want to export to a couple of running virtuals to store large amounts of data. Since they have public interfaces, we need to set up our nfs server to only allow limited access to our nfs server, both by limiting through nfs and also with firewalls.

## Devuan (Debian)

Linux does not, by default, install an nfs server. To install (Devuan, Debian derivative)

```
apt install -y nfs-kernel-server
```

Also, the syntax for /etc/exports is slightly different for nfs v2 and 3

```
/srv/nfs 192.0.2.10(rw,sync,no_subtree_check,no_root_squash)
```

And the way to start/restart nfsd is

```
exportfs -ra # re-export filesystems
exportfs -v # view exports
/etc/init.d/nfs-kernel-server restart
```

## Set up storage space

First, we need some directories to export. Since we want to configure space, etc..., we will do this by creating zfs entries.

```
zfs create -o atime=off -o dedup=off -o mountpoint=/media/nfs_root
storage/nfs_root

zfs create -o quota=100G storage/nfs_root/dom0
mkdir -p /media/nfs_root/dom0/xen-configs
```

```
mkdir -p /media/nfs_root/dom0/xen-store
mkdir -p /media/nfs_root/dom0/xen-images
chmod 777 /media/nfs_root/dom0/xen*

zfs create storage/nfs_root/simon
zfs create storage/nfs_root/strax
```

Note that we created storage/nfs_root so we could set some options that will be inherited by subdirectories, in this case, the mount point, atime=off and dedup=off (that is the default anyway).

We then created storage/nfs_root/dom0 under it, and gave it a quota of 100G. The dom0 stuff should be fairly secure since their firewalls limit the access, so we just put the alldirs option in to allow the dom0's to mount subdirectories wherever they want.

Finally, we create two stores for some servers to put their stuff in, and we limit access to them to only the server itself.

## Configure and start NFS

NFS can be exported via ZFS, but that appears to be a little flaky still, so I went back to the old, tried and true (v1 was built by Sun Microsystems in 1984). On FreeBSD (and everything else I've worked on), this is done via the */etc/exports* file. Following will export our stuff:

exports

```
/media/nfs_root/dom0  -alldirs  10.81.210.37 10.81.210.32
/media/nfs_root/simon  -maproot=root  192.168.150.68
/media/nfs_root/strax  -maproot=root  strax.example.com
```

**Note:**

```
service mountd reload
```

must be run any time the exports file is modified.

This allows servers at IP addresses 10.81.210.37 and 10.81.210.32 to access /media/nfsroot/dom0, and mount specific directories under it independantly. We then set up simon and strax to mount the other two, limiting simon by IP address and strax by it's DNS name. In this case, we allow the root user to mark files/directories as the root user (vs nobody).

There are tons of other options. See https://www.freebsd.org/doc/handbook/network-nfs.html for a brief discussion and/or *man 5 exports* on any Unix system for details.

You can now start nfs by

```
service nfsd onestart
```

which will start rpcbind and nfsd. If you have an error, you can shut it down with two commands:

```
service nfsd onestop
service rpcbind onestop
```

When you have no errors, you can set this up to be permenant by adding the following to /etc/rc.conf

rc.conf.additional

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_enable="YES"

# the -h below binds nfsd to a specific IP, so a machine with
# multiple IP's will only respond if this one is used for the
# request
mountd_flags="-r -h 192.168.0.1"
# uncomment the following lines if server side file locks are
# needed. Note you must replicate this on the clients
#rpc_lockd_enable="YES"
#rpc_statd_enable="YES"
```

nfs will now be enabled at boot time. Note the two commented lines beginning with rpc_. Enabling these if you ever need server side locking on your nfs mounts. You must also enable them on each client. Locks will then be honored for all clients which access this mount.

Note the mountd_flags entry. mountd is what actually handles the mount request from clients. The -r flag allows individual files to be mounted, such as swap a swap file, or a virtual image. Additional flags may be specified. See *man 5 mountd* for more information. Two nice ones are the -h and -p flags which allow you to specify the IP and Port mountd will listen on.

# Maintenance

If you need to add/remove/modify the NFS shares, create/modify any file systems on the server, then add/edit the appropriate stanzas to /etc/exports. Once you have done that, tell mountd to re-read the configuration:

```
service mountd reload
```

At this point, the modifications will be in effect. NOTE, if you have modified a share which was mounted, you should umount it, make the changes, then remount.

# Links

- https://www.freebsd.org/doc/handbook/network-nfs.html
- https://www.freebsd.org/cgi/man.cgi?query=exports&sektion=5&manpath=freebsd-release-ports
- http://serverfault.com/questions/451287/how-to-nfsv4-share-a-zfs-file-system-on-freebsd#4516

63

- [https://forums.freebsd.org/threads/rpcprog_mnt-rpc-authentication-error-why-client-credential-too-weak.22937/]
- https://www.freebsd.org/doc/handbook/zfs-zpool.html
- https://serverfault.com/questions/347163/mounting-nfsv4-share-from-debian-linux-6-to-freebsd-9-rc3-server-requires-stron?rq=1
- https://forums.freebsd.org/threads/zfs-on-nfs-permissions-and-export-values.9570/
- https://forums.freebsd.org/threads/creating-a-zfs-network-share-over-nfs.34828/

From:
https://wiki.linuxservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
**https://wiki.linuxservertech.com/unix/freebsd/system_builds/nfsserver**

Last update: **2026/04/14 23:29**