NFSv4 Server Export Setup

⚠ **Work in progress**

This site is in the process of being reviewed and updated.

Setting Up The Server

For an NFSv4 server, all of the exports are handled through one export point (the pseudofilesystem), with all other exports grouped underneath a root export.

1. Create a root export directory.
2. Bind all of the shares we intend to export into the root export directory.
3. Export everything.

Create Server Export Directories

The "/exports" directory will hold all of our local filesystem resources that will be made available as exports. The subdirectories are the actual exported resources which need to be mapped back to the real resources.

```
[DIRxSRVx10:root@nfs1 ~]# mkdir /exports
[DIRxSRVx10:root@nfs1 ~]# mkdir /exports/distros
[DIRxSRVx10:root@nfs1 ~]# mkdir /exports/sge
[DIRxSRVx10:root@nfs1 ~]# mkdir -m 1777 /exports/users
```

Make sure all of the directories have at least one test file in them, so we can more easily verify when a mount has worked.

```
[DIRxSRVx10:root@nfs1 ~]# touch /exports/distros/distros.tmp
[DIRxSRVx10:root@nfs1 ~]# touch /exports/sge/sge.tmp
[DIRxSRVx10:root@nfs1 ~]# touch /exports/users/users.tmp
```

Read-Only NFSv4 Mount

Read-Only (RO) mounts are useful for distributing public files, installation software, etc. In this example, we export a 'distros' directory as read-only.

Read/Write NFSv4 Mount

The most common type of NFS mount is the Read/Write (RW) mount. Since we created the /exports/users directory with the sticky bit set, any remote user will be able to read and write to the directory, but only the owner of the files will be able to modify or delete them. If the "root" user on the NFSv4 client writes a file to the directory, the ownership will be changed to "nobody". Root squashing is on by default, which means that the "root" user on remote NFS clients does not have root privileges on the server.

The only difference between read-only and read-write exports is the 'ro' vs. 'rw' in the export statement.

The following configuration is bind mounting the original directories (left) into the main "/exports" directory. Note the filesystem type is defined as "none" and the only option defined is the "bind" method.

```
[DIRxSRVx10:root@nfs1 ~]# vi /etc/fstab

/opt/distros    /exports/distros        none    bind 0 0
/opt/sge        /exports/sge            none    bind 0 0
/home/REALM     /exports/users          none    bind 0 0
```

After the /etc/fstab file has been configured, the mounts can be bound using the following command.

```
[DIRxSRVx10:root@nfs1 ~]# mount -a -t none
```

For testing, you can use mount --bind and exportfs to set this up temporarily:

```
[DIRxSRVx10:root@nfs1 ~]# mount --bind /opt/distros /exports/distros
[DIRxSRVx10:root@nfs1 ~]# mount --bind /opt/sge /exports/sge
[DIRxSRVx10:root@nfs1 ~]# mount --bind /home/REALM /exports/users
[DIRxSRVx10:root@nfs1 ~]# exportfs -ofsid=0,insecure,no_subtree_check *:/exports
[DIRxSRVx10:root@nfs1 ~]# exportfs -oro,nohide,insecure,no_subtree_check *:/exports/distros
[DIRxSRVx10:root@nfs1 ~]# exportfs -orw,nohide,insecure,no_subtree_check *:/exports/sge
[DIRxSRVx10:root@nfs1 ~]# exportfs -orw,nohide,insecure,no_subtree_check *:/export/users
```

Check the active mounts to confirm the successful mounting of the bound directories.

```
[DIRxSRVx10:root@nfs1 ~]# mount -l | grep bind
/opt/distros on /exports/distros type none (ro,bind)
/opt/sge on /exports/sge type none (rw,bind)
/home/REALM on /exports/users type none (rw,bind)
```

Doing a directory listing of the "/exports/sge" directory displays the directories that are actually located in "/opt/sge".

```
[DIRxSRVx10:root@nfs1 ~]# ls -l /exports/sge
drwxr-xr-x  3 root root 4096 Dec  9 15:49 pub
```

The exports can now be defined for the server. Instead of exporting a number of distinct exports, an NFSv4 client sees the NFSv4 server's exports as existing inside a single filesystem, called the nfsv4 "pseudofilesystem". The most important configuration setting here is the "fsid=0" option which tells the server that this is the pseudofilesystem and that all other directories are contained within this one. Another important setting here is the anonuid and anongid values, they are set to 65534 which is the nobody account. Be sure to check the nfs man page for NFSv4 specific export options.

To export using Kerberos v5, use the special client named "gss/krb5". You can use "krb5i" for integrity or "krb5p" for privacy.

```
[DIRxSRVx10:root@nfs1 ~]# vi /etc/exports

/exports             gss/krb5p(rw,insecure,sync,wdelay,no_subtree_check,fsid=0,no_root_squash,anonuid=65534,
anongid=65534)
/exports/distros     gss/krb5p(ro,insecure,sync,wdelay,no_subtree_check,nohide,no_root_squash,anonuid=65534,
anongid=65534)
/exports/sge         gss/krb5p(rw,insecure,sync,wdelay,no_subtree_check,nohide,no_root_squash,anonuid=65534,
anongid=65534)
/exports/users       gss/krb5p(rw,insecure,sync,wdelay,no_subtree_check,nohide,no_root_squash,anonuid=65534,
anongid=65534)
```

Testing

The exports that are available from the server can be checked with the following commands.

```
[DIRxSRVx10:root@nfs1 ~]# exportfs -v
/exports/distros gss/krb5p(ro,wdelay,insecure,root_squash,no_subtree_check,fsid=0,anonuid=65534,anongid=65534)
/exports/sge gss/krb5p(rw,wdelay,insecure,root_squash,no_subtree_check,fsid=0,anonuid=65534,anongid=65534)
/exports/users gss/krb5p(rw,wdelay,insecure,root_squash,no_subtree_check,fsid=0,anonuid=65534,anongid=65534)

[DIRxSRVx10:root@nfs1 ~]# showmount -e
Export list for nfs1.example.com:
/exports/distros gss/krb5p
/exports/sge gss/krb5p
/exports/users gss/krb5p
```

If you have changed the export configuration and need to re-export, you can use:

```
[DIRxSRVx10:root@nfs1 ~]# exportfs -rv
exporting gss/krb5p:/exports/distros
exporting gss/krb5p:/exports/sge
exporting gss/krb5p:/exports/users
```