

31338 Network Servers

Lab 8a: NFS

Aims

1. To configure a Linux machine to act as an NFS server and NFS client
2. To understand the use of the `exportfs` command to list available exports

Task 1: Basic NFS client and server functionality

1. In this lab, we will setup NFSv4. The steps to setup NFSv4 are different to other NFS versions, e.g. v3 or v2.
 - On the Linux virtual machine, ensure the `ens37` interface uses a static network address, e.g. `10.0.2.1/24`.
 - Ensure the required packages - `nfs-utils`, `rpcbind`, and `nfs4-acl-tools` - are installed.
 - If they are not installed, you can install them by using

```
dnf install packagename
```
 - In your virtual machine, create a directory called `/share/IT_Projects`. Export the `/share/IT_Projects` directory as **read-only** by modifying the `/etc/exports` file. Only hosts on your local subnet (`10.0.2.0/24`) should be able to mount this file system using NFS.
 - Turn on the `rpcbind` and NFS server service (daemons) using `systemctl`.

```
systemctl start rpcbind
systemctl start nfs-server
```
 - Also use the `systemctl` command to make sure these service(s) for NFS are **enabled** (start every time the machine boots in future).
2. Use the command `exportfs -v` to verify the `/share/IT_Projects` directory is exported. Also verify that it is read-only (ro).

Test 1 – local mount

As a first test, use the same virtual machine as an NFS client to access the exported directory. In this case, we are using the same machine to act as both client and server (it is a client to itself). When you do this task it might help if you have two shell windows open and think of one of them as the server and the other as the client.

Create a directory called `/mnt/projects` that we will use as a mount point (in your “client” shell window). Then, in your “client” shell window, try mounting your exported `/share/IT_Projects` directory onto the `/mnt/projects` mount point using the command below. This reflects what a client machine would do.

```
mount -t nfs4 10.0.2.1:/share/IT_Projects /mnt/projects
```

For testing:

1. Create a file in the `/share/IT_Projects` directory (in your “server” shell window). Just using `touch /share/IT_Projects/file1` is enough.
Verify that you can see “file1” (or whatever file you created) inside the `/mnt/projects` directory too.
2. Now try creating a file in the `/mnt/projects` directory (in your “client” shell window), e.g. `touch /mnt/projects/file2`
(it shouldn’t allow you, as you exported the filesystem read-only). Document the commands, testing process, and error message in your journal.

Finally, to finish the test, unmount the directory. Remember again that the command is “umount”, not “unmount”.

```
umount /mnt/projects
```

If any of your shell windows is using the directory (e.g. because you used “cd” to change into the directory), the umount command won’t allow you to unmount it. Just change directory to somewhere else. Document what happens (if you get an error message) and why.

Task 2: Advanced Task – Setup a separate Linux virtual machine to access the remote NFS share

Setup a second Linux virtual machine by either:

- creating a second “linked clone” of the Linux VM, if you are working in the labs at UTS (see Lab 1); or
- making a complete copy of your VM folder, if you are working on your own laptop or at home.

This second machine will be configured to act as an NFS client. When you start the second VM and are asked whether you moved or copied it, make sure you say you copied it.

1. When you start the second VM, set the ens37 IP address to 10.0.2.2/24.
2. Install the required NFS packages as in Task 1.
3. Start up the NFS related services using `systemctl`. Also make sure the NFS related services are enabled so they will start every time the machine boots in future.
4. Create the `/mnt/projects` directory to use as a mount point. Mount the remote NFS exported directory `/share/IT_Projects` from server 10.0.2.1 onto `/mnt/projects`. Verify you can access the remote directory from the client machine.
5. To automatically have the client machine mount the filesystem from the server during startup, append something like the following line to `/etc/fstab`. Explore the fields in the `/etc/fstab` file on the client.

```
10.0.2.1/share/IT_Projects      /mnt/projects      nfs      default      1 1
```

Reboot your second (client) virtual machine and verify that it automatically mounts the remote NFS filesystem on boot and that you can access the remote NFS exported directory. The access should be read-only as before.

6. As a final test, on the NFS server, change the filesystem to read-write rather than read-only by editing `/etc/exports`. Tell the server to reread the exports file (re-export) with the `-r` option. Also make sure you change the UNIX filesystem permissions on the `/share/IT_Projects` directory so it is writable.

```
exportfs -r                      (on the NFS server)
```

```
chmod 777 /share/IT_Projects     (on the NFS server)
```

On the client, ask it to remount the filesystem from the server, using the `remount` option to mount:

```
mount -o remount /mnt/projects  (on the NFS client)
```

Verify that the client machine can now create and edit files in `/mnt/projects`. If you are logged in as the root user, and create a file, what happens? Make sure you use “`ls -l`” to see the long directory listing. Who owns the new file you created? (Hint: read about “root squash”). Note that if for whatever reason you can’t create a second VM