Exercise 1 –NFS Server Installation and configuration

Installation and configuration of NSF Server

Exercise 1.1: Tasks to perform on AlmaLinux:

Use the root account to complete this exercise

1. Install the **NFS server** package.

```
root@server07 ~ $ dnf install -y nfs-utils
Last metadata expiration check: 0:04:18 ago on Tue 01 Apr 2025 11:42:19 AM.
Dependencies resolved.
================================================================================
 Package                 Architecture          Version                 Repository              Size
================================================================================
Installing:
 nfs-utils               x86_64                1:2.5.4-27.el9          baseos                 431 k
Installing dependencies:
 gssproxy                x86_64                0.8.4-7.el9             baseos                 108 k
 libev                   x86_64                4.33-5.el9              baseos                  52 k
 libnfsidmap             x86_64                1:2.5.4-27.el9          baseos                  59 k
 libverto-libev          x86_64                0.3.2-3.el9             baseos                  13 k
 rpcbind                 x86_64                1.2.6-7.el9             baseos                  56 k
 sssd-nfs-idmap          x86_64                2.9.5-4.el9_5.4         baseos                  38 k

Transaction Summary
================================================================================
Install  7 Packages

Total download size: 757 k
Installed size: 1.9 M
Downloading Packages:
(1/7): gssproxy-0.8.4-7.el9.x86_64.rpm                  1.5 MB/s | 108 kB     00:00
(2/7): libev-4.33-5.el9.x86_64.rpm                      747 kB/s |  52 kB     00:00
(3/7): libverto-libev-0.3.2-3.el9.x86_64.rpm            726 kB/s |  13 kB     00:00
(4/7): libnfsidmap-2.5.4-27.el9.x86_64.rpm              403 kB/s |  59 kB     00:00
(5/7): sssd-nfs-idmap-2.9.5-4.el9_5.4.x86_64.rpm        502 kB/s |  38 kB     00:00
```

2. Start and enable the **NFS service**.

```
root@server07 ~ $ systemctl enable --now nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.se
rvice.
root@server07 ~ $
```

3. Verify that both the **NFS** and **rpcbind** services are **started** and **enabled**.

```
root@server07 ~ $ systemctl status nfs-server
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
     Active: active (exited) since Tue 2025-04-01 11:47:38 EDT; 36s ago
       Docs: man:rpc.nfsd(8)
             man:exportfs(8)
```

```
root@server07 ~ $ systemctl status nfs-idmapd.service
● nfs-idmapd.service - NFSv4 ID-name mapping service
     Loaded: loaded (/usr/lib/systemd/system/nfs-idmapd.service; static)
     Active: active (running) since Tue 2025-04-01 11:47:37 EDT; 1min 38s ago
       Docs: man:idmapd(8)
    Process: 9342 ExecStart=/usr/sbin/rpc.idmapd (code=exited, status=0/SUCCESS)
```

```
root@server07 ~ $ systemctl status rpcbind
● rpcbind.service - RPC Bind
     Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-04-01 11:47:36 EDT; 2min 14s ago
TriggeredBy: ● rpcbind.socket
       Docs: man:rpcbind(8)
   Main PID: 9345 (rpcbind)
```

4. Authorize the necessary NFS services through the **firewall.**

```
root@server07 ~ $ firewall-cmd --permanent --add-service=nfs --zone=nm-shared
success
root@server07 ~ $ firewall-cmd --permanent --add-service=mountd --zone=nm-shared
success
root@server07 ~ $ firewall-cmd --permanent --add-service=rpc-bind --zone=nm-shared
success
root@server07 ~ $
```

5. Verify that the required services are added and allowed in the firewall.

```
root@server07 ~ $ firewall-cmd --reload
success
root@server07 ~ $ firewall-cmd --list-services --zone=nm-shared
dhcp dns mountd nfs rpc-bind ssh
```

6. List all **TCP** and **UDP** ports currently listening on the server.

```
root@server07 ~ $ rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  36447  status
    100024    1   tcp  46593  status
    100005    1   udp  20048  mountd
    100005    1   tcp  20048  mountd
    100005    2   udp  20048  mountd
    100005    2   tcp  20048  mountd
    100005    3   udp  20048  mountd
    100005    3   tcp  20048  mountd
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
    100021    1   udp  43950  nlockmgr
    100021    3   udp  43950  nlockmgr
    100021    4   udp  43950  nlockmgr
    100021    1   tcp  39825  nlockmgr
    100021    3   tcp  39825  nlockmgr
    100021    4   tcp  39825  nlockmgr
root@server07 ~ $
```

7. Identify the **TCP port numbers** used by the NFS services.

111 > portmapper (rpcbind)

20048 > mountd

2049 > nfs & nfs_acl

39825 & 43950 > nlockmgr

46593 & 36447 > status

8. What is the name of the **main configuration file** used by the NFS server?

```
root@server07 ~ $ dnf repoquery -l nfs-utils
Last metadata expiration check: 0:17:00 ago on Tue 01 Apr 2025 11:42:19 AM.
/etc/exports.d
/etc/gssproxy/24-nfs-server.conf
/etc/modprobe.d/lockd.conf
/etc/nfs.conf
/etc/nfsmount.conf
/etc/request-key.d/id_resolver.conf
/sbin/mount.nfs
/sbin/mount.nfs4
/sbin/nfsdcltrack
/sbin/rpc.statd
/sbin/umount.nfs
/sbin/umount.nfs4
/usr/lib/.build-id
/usr/lib/.build-id/0b
/usr/lib/.build-id/0b/15f7ce6503d9718a16d7c5046d248cb1d9ca0b
/usr/lib/.build-id/23
/usr/lib/.build-id/23/1f8fc4edf973a42dd0eabda9f8302cca60f339
/usr/lib/.build-id/62
/usr/lib/.build-id/62/fc16fee633e7775182f9329df0e302a1a5babf
/usr/lib/.build-id/6a
/usr/lib/.build-id/6a/7c1612243d6e7be216957e2f9bdb59e666d646
/usr/lib/.build-id/6b
/usr/lib/.build-id/6b/c318b41c7c1425d7a6a60a4062462b968aca8b
/usr/lib/.build-id/6f
/usr/lib/.build-id/6f/20fe2def2d5d5e87e30356887e26a32c53f216
/usr/lib/.build-id/7d
/usr/lib/.build-id/7d/7810466fc882f4f3bc323a4252d81bd0ad74c3
```

Main Configuration files:

/etc/nfs.conf: this is the primary configuration file for NFS server settings like threads, ports, and general options.

/etc/exports: That's the file where you define which directories are shared via NFS and who can access them.

Note. File "exports" is not created by default

Network share creation

Exercise 1.2: Tasks to perform on AlmaLinux:

Use the root account to complete this exercise

1. Create a user named **teacher1** with UID **1500**.
2. Create a group named **teachers** with GID **1700**.
3. Set the primary group of **teacher1** to **teachers**.

```
root@server07 ~ $ useradd -u 1500 teacher1
root@server07 ~ $ groupadd -g 1700 teachers
root@server07 ~ $ usermod -g teachers teacher1
root@server07 ~ $
```

4. Using a single command, create the **/mnt/share/IT** directory.
5. Change the owner and group of the **/mnt/share/IT** directory to **teacher1** and **teachers.**
6. Set the directory permissions of **/mnt/share/IT** to **770**.

```
root@server07 ~ $ mkdir -p /mnt/share/IT
root@server07 ~ $ chown -R teacher1:teachers /mnt/share/IT
root@server07 ~ $ chmod -R 770 /mnt/share/IT
root@server07 ~ $
```

7. List the contents of **/mnt/share/IT** to verify the configuration.

```
root@server07 ~ $ ls -ld /mnt/share/IT
drwxrwx---. 2 teacher1 teachers 6 Apr  1 12:26 /mnt/share/IT
root@server07 ~ $
```

8. Configure NFS to make the **/mnt/share/IT** directory accessible to the **192.168.50.0/24** network with **read** and **write** permissions.

```
root@server07 ~ $ echo "/mnt/share/IT 192.168.50.0/24(rw,sync,no_all_squash)" >> /etc/exports
root@server07 ~ $
```

9. Export the directory.

```
root@server07 ~ $ exportfs -arv
exporting 192.168.50.0/24:/mnt/share/IT
root@server07 ~ $
```

-a: export all entries from /etc/exports

-r: re-export all (useful after edits)

-v: verbose output (shows what's being exported)

10. View the current list of exported directories.

```
root@server07 ~ $ exportfs -s
/mnt/share/IT  192.168.50.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
root@server07 ~ $
```

Mounting shared directories on the client

Exercise 1.3: Tasks to perform on Ubuntu:

Use your Ubuntu user account to complete this exercise on Ubuntu

1. Install the **NFS client** on **Ubuntu**.

```
gkeymole@client07: $ sudo apt update
[sudo] password for gkeymole:
Hit:1 http://ca.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:3 http://ca.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:4 http://ca.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,207 B]
Fetched 260 kB in 1s (345 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
14 packages can be upgraded. Run 'apt list --upgradable' to see them.
gkeymole@client07: $ sudo apt -y install nfs-common nfs4-acl-tools vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 rpcbind vim-runtime
Suggested packages:
  open-iscsi watchdog ctags vim-doc vim-scripts
The following NEW packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 nfs-common nfs4-acl-tools rpcbind vim vim-runtime
0 upgraded, 8 newly installed, 0 to remove and 14 not upgraded.
Need to get 9,067 kB of archives.
After this operation, 39.4 MB of additional disk space will be used.
Get:1 http://ca.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-core-2.1-7 amd64 2.1.12-stable-1build3 [93.9 kB]
Get:2 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnfsidmap1 amd64 1:2.6.1-1ubuntu1.2 [42.9 kB]
Get:3 http://ca.archive.ubuntu.com/ubuntu jammy/main amd64 rpcbind amd64 1.2.6-2build1 [46.6 kB]
Get:4 http://ca.archive.ubuntu.com/ubuntu jammy/main amd64 keyutils amd64 1.6.1-2ubuntu3 [50.4 kB]
Get:5 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nfs-common amd64 1:2.6.1-1ubuntu1.2 [241 kB]
```

2. Run a command to list the directories exported by the NFS server.

```
gkeymole@client07:~$ showmount -e 192.168.50.10
Export list for 192.168.50.10:
/mnt/share/IT 192.168.50.0/24
gkeymole@client07:~$
```

3. Create the user **teacher1** and the group **teachers** using the **same UID** and **GID** as in the previous exercise. Assign the password alma to the teacher1 user.

```
gkeymole@client07:~$ sudo adduser --ingroup teachers -u 1500 teacher1
[sudo] password for gkeymole:
Adding user `teacher1' ...
Adding new user `teacher1' (1500) with group `teachers' ...
Creating home directory `/home/teacher1' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for teacher1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
gkeymole@client07:~$
```

4. Create the local directory: **/share/tech**.

```
gkeymole@client07:~$ sudo mkdir -p /share/tech
gkeymole@client07:~$
```

5. Mount the **/mnt/share/IT** directory exported by the **AlmaLinux** server to the local **/share/tech** directory on **Ubuntu**.

```
gkeymole@client07:~$ sudo mount -t nfs 192.168.50.10:/mnt/share/IT /share/tech
gkeymole@client07:~$
```

6. Run a command to confirm that the NFS share has been successfully mounted.

```
gkeymole@client07:~$ sudo mount | grep nfs
192.168.50.10:/mnt/share/IT on /share/tech type nfs4 (rw,relatime,vers=4.2,rsize=524288,wsize=524288,namlen=255,hard,proto=tcp,timeo=
600,retrans=2,sec=sys,clientaddr=192.168.50.20,local_lock=none,addr=192.168.50.10)
gkeymole@client07:~$
```

7. Use the **su -** command to switch to the **teacher1** user.

8. Try to create a text file in the **/share/tech** directory. Are you able to create the file? Why or why not?

```
gkeymole@client07:~$ su - teacher1
Password:
teacher1@client07:~$ touch /share/tech/testfile.txt
teacher1@client07:~$
```

9. Return to the **AlmaLinux** server and check the contents of the **/mnt/share/IT** directory. What do you observe?

```
root@server07 ~ $ ls -la /mnt/share/IT
total 0
drwxrwx---. 2 teacher1 teachers 26 Apr  1 14:22 .
drwxr-xr-x. 3 root     root     16 Apr  1 12:26 ..
-rw-r--r--. 1 teacher1 teachers  0 Apr  1 14:22 testfile.txt
root@server07 ~ $
```

10. Go back to **Ubuntu** and **log out** from the **teacher1** session.

11. Unmount the **/share/tech** directory.

12. Ensure that the **/share/tech** directory is now empty.

```
teacher1@client07:~$ exit
logout
gkeymole@client07:~$ sudo umount /share/tech
gkeymole@client07:~$ ls -l /share/tech
total 0
gkeymole@client07:~$ sudo ls -l /share/tech
total 0
gkeymole@client07:~$
```