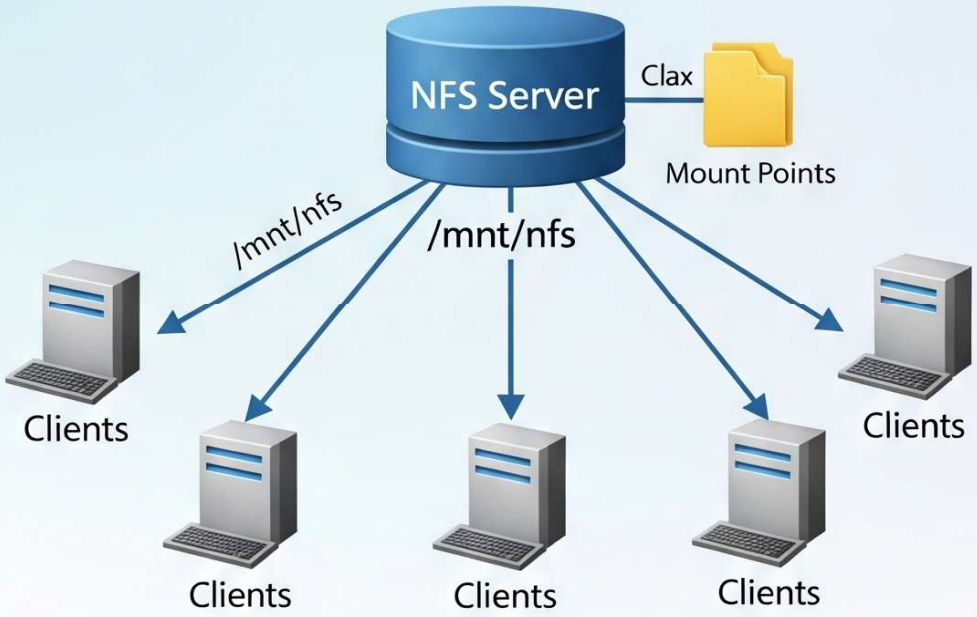
NFS-SERVER

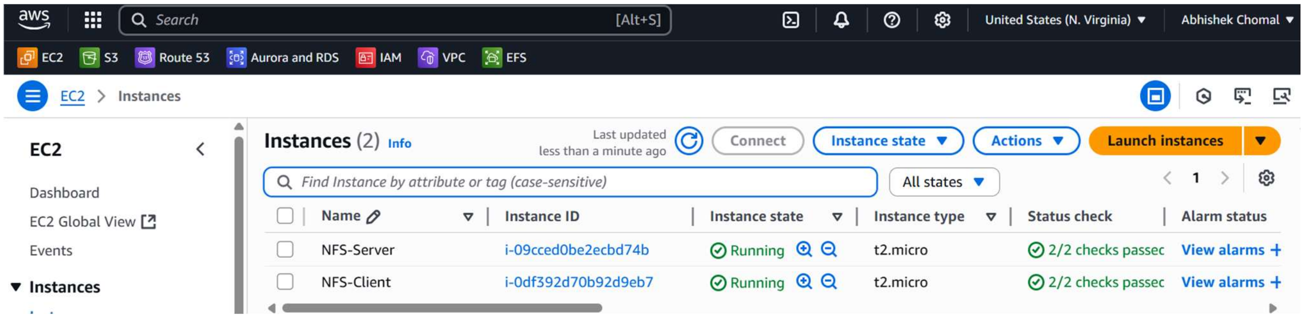
Setup Guide



By Abhishek Chomal

- In this project, I used **two AWS instances** (one as **NFS server** and one as **client**) to configure file sharing using NFS on RHEL 9. I created a **shared directory** on the server, **set proper permissions**, and **mounted** it on the **client system** for seamless remote access.

Aws Cloud Instance	Private IP	Public IP	Configured For
NFS-Server	172.31.95.180	34.201.59.134	NFS Server Setup
Client Server	172.31.84.57	100.24.34.169	NFS Client Configuration



- **Setup of NFS Server**

Step 1: Install the Packages using yum

- Package name: **nfs-utils**
- Command: **yum install nfs-utils**

```
[root@NFS-SERVER ~]# yum install nfs-utils
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Red Hat Enterprise Linux 10 for x86_64 - AppStream from RHUI (RPMs)          18 MB/s | 2.8 MB     00:00
Red Hat Enterprise Linux 10 for x86_64 - BaseOS from RHUI (RPMs)             42 MB/s | 9.4 MB     00:00
Red Hat Enterprise Linux 10 Client Configuration                             22 kB/s | 1.7 kB     00:00
Dependencies resolved.
================================================================================================================
 Package              Architecture     Version                  Repository                          Size
================================================================================================================
Installing:
 nfs-utils            x86_64           1:2.8.2-3.el10           rhel-10-baseos-rhui-rpms            487 k
Installing dependencies:
 gssproxy             x86_64           0.9.2-10.el10            rhel-10-baseos-rhui-rpms            118 k
 libev                x86_64           4.33-14.el10             rhel-10-baseos-rhui-rpms             56 k
 libnfsidmap          x86_64           1:2.8.2-3.el10           rhel-10-baseos-rhui-rpms             67 k
 libtirpc             x86_64           1.3.5-1.el10             rhel-10-baseos-rhui-rpms             98 k
 libverto-libev       x86_64           0.3.2-10.el10            rhel-10-baseos-rhui-rpms             15 k
 quota                x86_64           1:4.09-9.el10            rhel-10-baseos-rhui-rpms            201 k
 quota-nls            noarch           1:4.09-9.el10            rhel-10-baseos-rhui-rpms             79 k
 rpcbind              x86_64           1.2.7-3.el10             rhel-10-baseos-rhui-rpms             63 k
 sssd-nfs-idmap       x86_64           2.10.2-3.el10_0.2        rhel-10-baseos-rhui-rpms             37 k

Transaction Summary
================================================================================================================
Install  10 Packages
```

Step 2: Verify that the packages were installed correctly

- Command: **rpm -q nfs-utils**

```
[root@NFS-SERVER ~]# rpm -q nfs-utils
nfs-utils-2.8.2-3.el10.x86_64
```

Step 3: Start and enable the NFS Server

- Service name: **nfs-server**
- Command: **systemctl enable --now nfs-server**

```
[root@NFS-SERVER ~]# systemctl enable --now nfs-server
Created symlink '/etc/systemd/system/multi-user.target.wants/nfs-server.service' → '/usr/lib/systemd/system/nfs-server.service'.
[root@NFS-SERVER ~]#
```

Step 4: Check the status of NFS server

- Command: **systemctl status nfs-server**

```
[root@NFS-SERVER ~]# systemctl status nfs-server
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
    Drop-In: /run/systemd/generator/nfs-server.service.d
             └─order-with-mounts.conf
     Active: active (exited) since Sun 2025-06-29 10:17:33 UTC; 2min 13s ago
 Invocation: 09d548a8b66947e7b712a32e762f95c9
       Docs: man:rpc.nfsd(8)
             man:exportfs(8)
    Process: 3371 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
    Process: 3373 ExecStart=/bin/sh -c /usr/sbin/nfsdctl autostart || /usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
    Process: 3401 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload gssproxy ; fi (code=exited, status=0
   Main PID: 3401 (code=exited, status=0/SUCCESS)
   Mem peak: 1.6M
        CPU: 29ms

Jun 29 10:17:32 NFS-SERVER systemd[1]: Starting nfs-server.service - NFS server and services...
Jun 29 10:17:33 NFS-SERVER systemd[1]: Finished nfs-server.service - NFS server and services.
lines 1-17/17 (END)
```

Step 5: Make a directory named /nfs-share

- Command: **mkdir /nfs-share**

```
[root@NFS-SERVER ~]# mkdir /nfs-share
[root@NFS-SERVER ~]# ls -ld /nfs-share/
drwxr-xr-x. 2 root root 6 Jun 29 07:41 /nfs-share/
```

- We will **keep** all the **files** in this **folder** that we want to **share** with the **client** over the **network**.

Step 6: Configure NFS Server

- **Edit the /etc/exports file.**
  - In this file, we define the **folder** we want to share along with the **client details**, so the server **knows** which **folder** to share with which **client**.
  - Open the **/etc/exports** configuration file using **vim**
  - Command: **vim /etc/exports**

```
[root@NFS-SERVER ~]# vim /etc/exports
[root@NFS-SERVER ~]#
```

  - **Add** the following **entries** in this file.

```
/nfs-share      100.24.34.169(rw,sync,no_root_squash)
~
```

  - **Explanation:**
    - **/nfs-share** – folder path we want to share
    - **100.24.34.169** – client IP that can access it
    - **rw –** Client can read and write
    - **sync –** Saves changes immediately
    - **no_root_squash** – Allows client root user to act as root

Step 7: Apply the Export configuration

- Command: **exportfs -arv**

```
[root@NFS-SERVER ~]# exportfs -arv
exporting 100.24.34.169:/nfs-share
```

Step 8: Restart NFS Service to apply changes

- Command: **systemctl restart nfs-server**

```
[root@NFS-SERVER ~]# systemctl restart nfs-server
[root@NFS-SERVER ~]#
```

Step 9: Configure SELinux for NFS

- By default, **SELinux blocks** NFS shares. So, we need to set the correct **SELinux context** on the **shared folder**.
- Command: **semanage fcontext -a -t nfs_t /nfs-share**

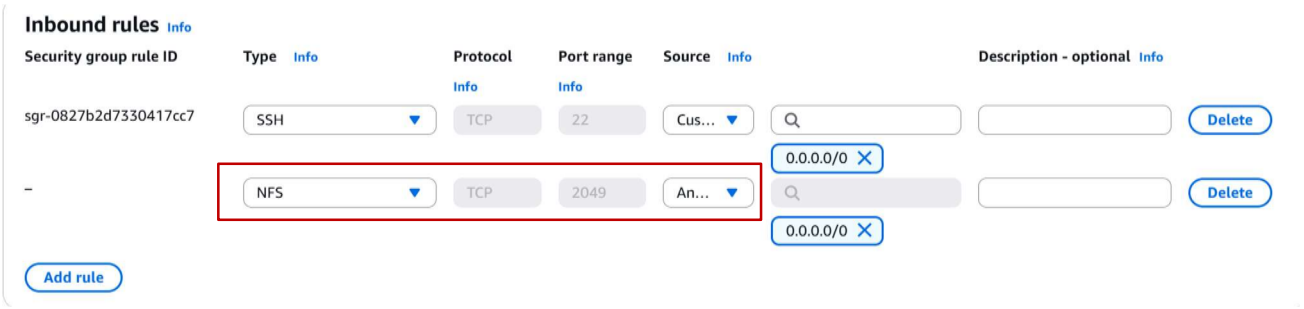
```
[root@NFS-SERVER ~]# semanage fcontext -a -t nfs_t /nfs-share
[root@NFS-SERVER ~]#
```

- **Relabel** the **folder** so **SELinux accepts** the **change**
- Command: **restorecon -Rv /nfs-share**

```
[root@NFS-SERVER ~]# restorecon -Rv /nfs-share
Relabeled /nfs-share from unconfined_u:object_r:default_t:s0 to unconfined_u:object_r:nfs_t:s0
[root@NFS-SERVER ~]#
```

Step 10: Allow Port 2049 in security group of NFS Server

- To allow **NFS traffic**, we need to **open port 2049** in the **security group** of the **NFS server instance** **(in AWS).**



➢ Setup of NFS Client

Step 1: Install the Packages using yum

- Package name: **nfs-utils**
- Command: **yum install nfs-utils**

```
[root@NFS-Client ~]# yum install nfs-utils
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 0:15:54 ago on Sun Jun 29 11:24:11 2025.
Dependencies resolved.
=====================================================================================================================
 Package              Architecture      Version                        Repository                           Size
=====================================================================================================================
Installing:
 nfs-utils            x86_64            1:2.8.2-3.el10                 rhel-10-baseos-rhui-rpms            487 k
Installing dependencies:
 gssproxy             x86_64            0.9.2-10.el10                  rhel-10-baseos-rhui-rpms            118 k
 libev                x86_64            4.33-14.el10                   rhel-10-baseos-rhui-rpms             56 k
 libnfsidmap          x86_64            1:2.8.2-3.el10                 rhel-10-baseos-rhui-rpms             67 k
 libtirpc             x86_64            1.3.5-1.el10                   rhel-10-baseos-rhui-rpms             98 k
 libverto-libev       x86_64            0.3.2-10.el10                  rhel-10-baseos-rhui-rpms             15 k
 quota                x86_64            1:4.09-9.el10                  rhel-10-baseos-rhui-rpms            201 k
 quota-nls            noarch            1:4.09-9.el10                  rhel-10-baseos-rhui-rpms             79 k
 rpcbind              x86_64            1.2.7-3.el10                   rhel-10-baseos-rhui-rpms             63 k
 sssd-nfs-idmap       x86_64            2.10.2-3.el10_0.2              rhel-10-baseos-rhui-rpms             37 k

Transaction Summary
=====================================================================================================================
Install  10 Packages
```

Step 2: Create Mount Point

- We need a **folder** where we will **mount** the **shared NFS directory**.
- Command: **mkdir /mnt/share-data**

```
[root@NFS-Client ~]# mkdir  /mnt/share-data
[root@NFS-Client ~]# ls -ld /mnt/share-data/
drwxr-xr-x. 2 root root 6 Jun 29 12:19 /mnt/share-data/
```

Step 3: Mount the Directory

- Now we will **mount** the **shared folder** from the **NFS server** to the **client system**.
- Syntax: **mount -t nfs <NFS_Server_IP>:/nfs-share   /mnt/share-data**
- Command: **mount -t nfs 34.201.59.134:/nfs-share   /mnt/share-data**

```
[root@NFS-Client ~]# mount -t nfs 34.201.59.134:/nfs-share   /mnt/share-data
[root@NFS-Client ~]#
```

Step 4: Verify the Mount

- Command: **df -h | grep nfs**

```
[root@NFS-Client ~]# df -h | grep nfs
34.201.59.134:/nfs-share  9.8G  1.8G  8.1G  18% /mnt/share-data
```

Step 5: Make the Mount Permanent

- Open **/etc/fstab** file using **vim.**
- Command: **vim /etc/fstab**

```
[root@NFS-Client ~]# vim /etc/fstab
[root@NFS-Client ~]#
```

- **Add** the following entry in **/etc/fstab** file.
  <NFS_Server_IP>:/nfs-share /mnt/share-data nfs defaults 0 0

```
34.201.59.134:/nfs-share    /mnt/share-data    nfs    defaults    0 0
```

- Apply **mounting** point
- Command: **mount -a**

```
[root@NFS-Client ~]# mount -a
[root@NFS-Client ~]#
```

Step 6: Test the NFS Share

- Now we will **test** if the **NFS sharing** is working properly or not.
- Go to the **NFS server** and create a test file inside the **/nfs-share** directory:

```
[root@NFS-SERVER ~]# cd /nfs-share/
[root@NFS-SERVER nfs-share]# ls
[root@NFS-SERVER nfs-share]# touch abhishek.txt
[root@NFS-SERVER nfs-share]# touch file{1..10}.txt
[root@NFS-SERVER nfs-share]# ls
abhishek.txt  file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt  file8.txt  file9.txt
[root@NFS-SERVER nfs-share]#
```

- Now go to the **NFS client** and check if the **same file** is **visible** in the **mounted folder**.

```
[root@NFS-Client ~]# ls -l /mnt/share-data/
total 0
-rw-r--r--. 1 root root 0 Jun 29 12:14 abhishek.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file1.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file10.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file2.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file3.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file4.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file5.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file6.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file7.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file8.txt
-rw-r--r--. 1 root root 0 Jun 29 12:14 file9.txt
```

- Now you can see the **same files** on the **client**, which means the **NFS setup** is **working** perfectly.

✅ Conclusion

- NFS is a simple and reliable way to share files between systems on the same network.
  It helps in easy and fast access to shared data without using any third-party tools.
- If you're learning Linux or DevOps, setting up NFS is a great hands-on practice.
  It improves your understanding of system administration and real-time file sharing in Linux environments.