Lab - Configuring NFS (Network File System) in Linux Fedora

Overview

In this lab, you will learn to configure NFS in Linux Fedora. NFS (Network File System) was developed for sharing files and folders between Linux/Unix systems by Sun Microsystems in 1980. It allows you to mount your local file systems on a network and allow remote hosts to interact with the files and folders as they if they were mounted locally on the same system.

Hardware Requirements

- One virtual install of Linux Fedora Server
- A second virtual install of either Fedora Server or Workstation.

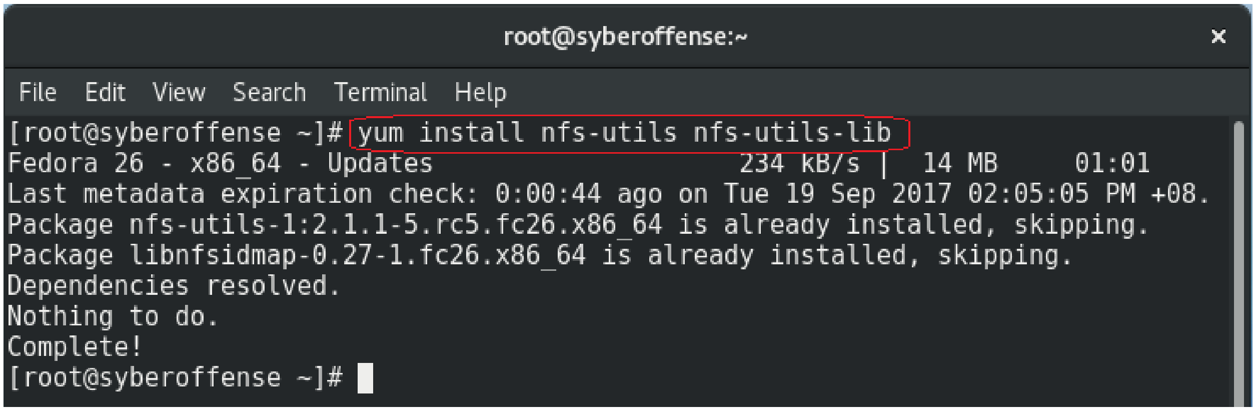
For this lab, I'll be using one Fedora Server and one Fedora workstation.

Begin the lab!

Logon to your VM for Fedora Server as root.
Logon to your VM for Fedora Workstation as root.
Open a terminal window on both.

Install the NFS packages on both our NFS Server and your NFS Client machine.

```
yum install nfs-utils nfs-utils-lib
```

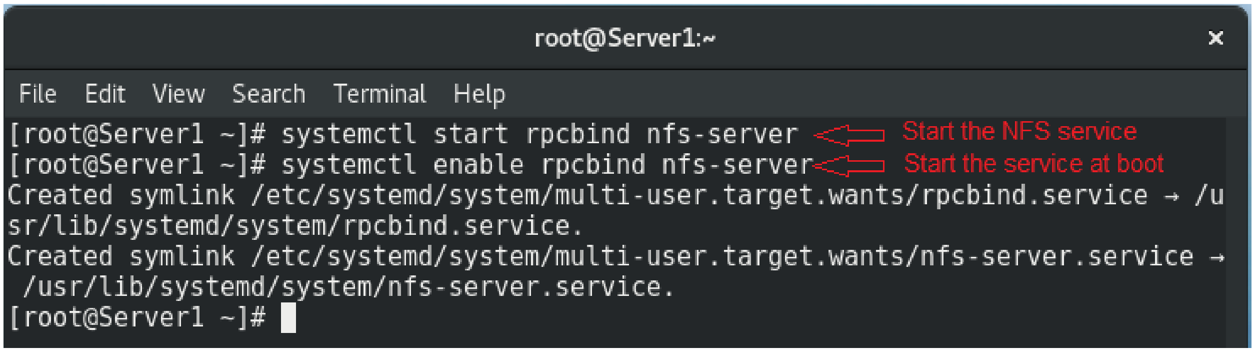


```
[root@syberoffense ~]# yum install nfs-utils nfs-utils-lib
Fedora 26 - x86_64 - Updates                    234 kB/s |  14 MB     01:01
Last metadata expiration check: 0:00:44 ago on Tue 19 Sep 2017 02:05:05 PM +08.
Package nfs-utils-1:2.1.1-5.rc5.fc26.x86_64 is already installed, skipping.
Package libnfsidmap-0.27-1.fc26.x86_64 is already installed, skipping.
Dependencies resolved.
Nothing to do.
Complete!
[root@syberoffense ~]# 
```

The next step is to start the NFS service. Type the following command on both the server and the workstation to start the HFS service.

```
systemctl start rpcbind nfs-server
```

We next need to enable the NFS system to start at boot.

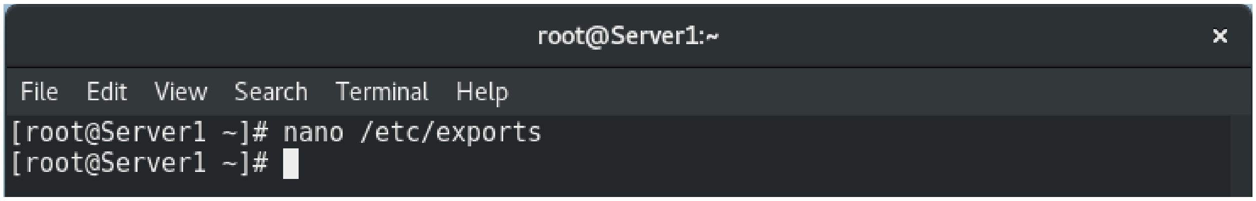


```
[root@Server1 ~]# systemctl start rpcbind nfs-server
[root@Server1 ~]# systemctl enable rpcbind nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /u
sr/lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
 /usr/lib/systemd/system/nfs-server.service.
[root@Server1 ~]# 
```

Start the NFS service

Start the service at boot

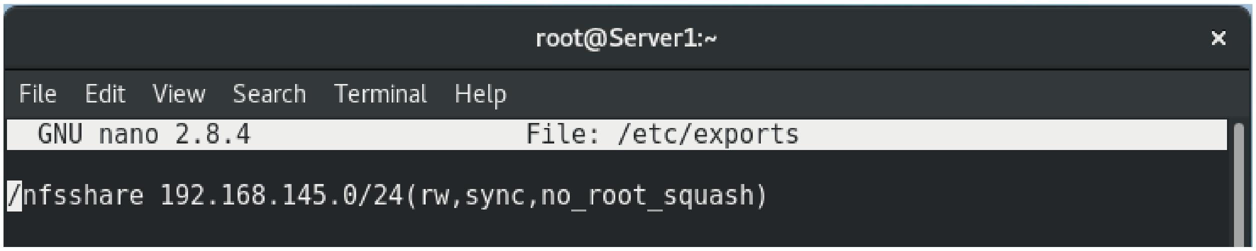
We are now ready to create a first NFS share. The main file that tracks the location of shares on the NFS server is exports lcated inside the ect directory. The path to this file is /etc.exports. We need to edit the file and add an entry into the file for our NFS shares. For this demonstration, I am using nano. You are free to use any text editor you want.



```
[root@Server1 ~]# nano /etc/exports
[root@Server1 ~]# 
```

Inside the text editor, you add a line for each NFS share you want advertised.

The name of my share is nfsshare and it is located at the root (/) of my server. I next add in the IP address of my network and subnet mask followed by the NFS file permissions I have granted to this directory.



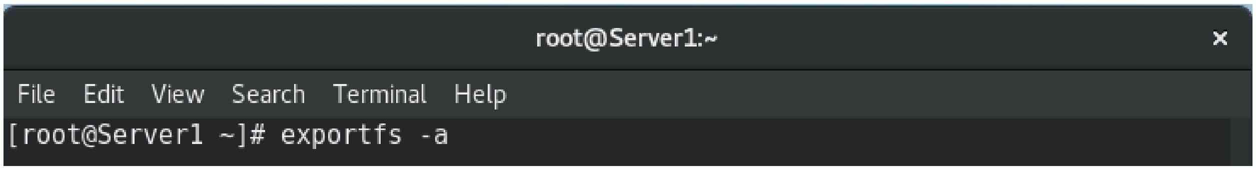
```
  GNU nano 2.8.4                    File: /etc/exports

/nfsshare 192.168.145.0/24(rw,sync,no_root_squash)
```

- rw – Stands for read and write access to the directory share.
- sync - Sync confirms requests to the shared directory only once the changes have been committed.
- no_root_squash: This phrase allows root to connect to the designated directory.

Back at the terminal type:

```
exportfs -a
```



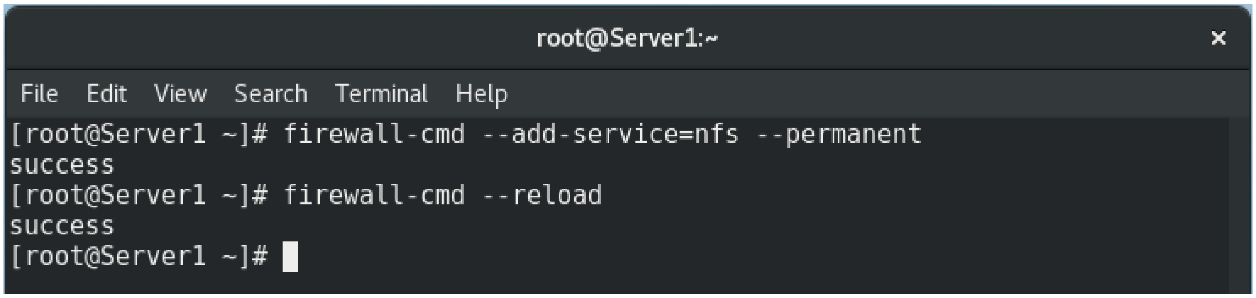
```
[root@Server1 ~]# exportfs -a
```

Before we move over to client, we next need to add an exception for the NFS protocol to our server's firewall. To do this, from the terminal, type in the following command:

```
firewall-cmd --add-service=nfs –permanent
```

Reload the firewall:

```
firewall-cmd --reload
```



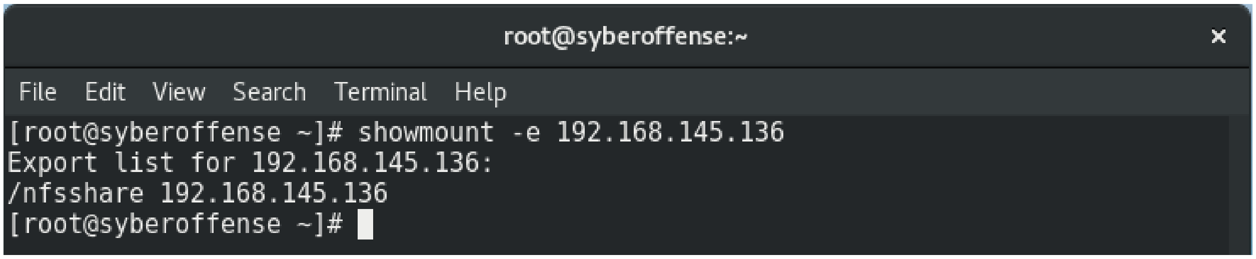
```
[root@Server1 ~]# firewall-cmd --add-service=nfs --permanent
success
[root@Server1 ~]# firewall-cmd --reload
success
[root@Server1 ~]# 
```

Switch terminal over to your Fedora Workstation or NFS client.

To find any NFS shares on my NFS server for the client, I can use the showmount -e command followed by the IP address of the NFS server.

```
showmount -e 192.168.145.136
```

This is my IP address not yours! Yours will differ!



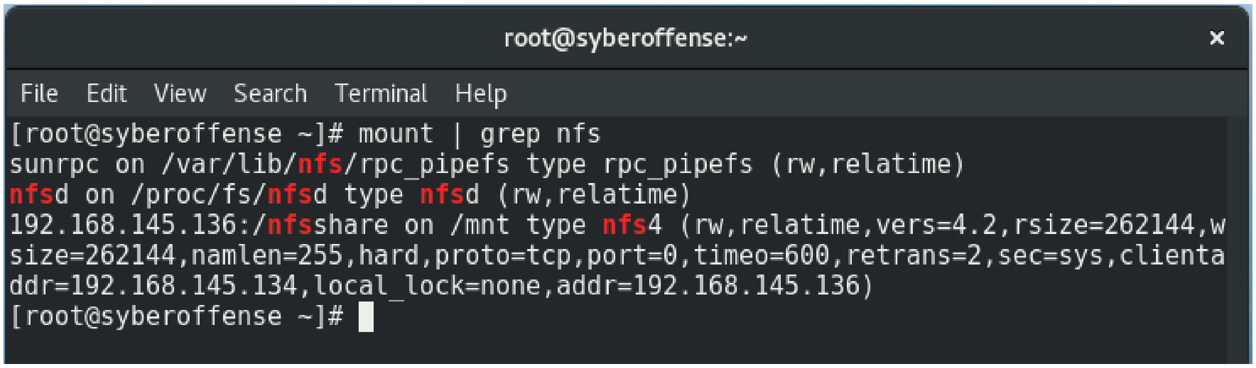
```
[root@syberoffense ~]# showmount -e 192.168.145.136
Export list for 192.168.145.136:
/nfsshare 192.168.145.136
[root@syberoffense ~]# 
```

Once you have confirmed the NFS mount does exist, you are ready to mount the the remote NFS share from your client. At the client terminal, type the following:

```
mount -t nfs 192.168.145.136:/nfsshare /mnt
```

Verify the mount is in plave using the following command form the terminal of the client:

```
mount | grep nfs
```

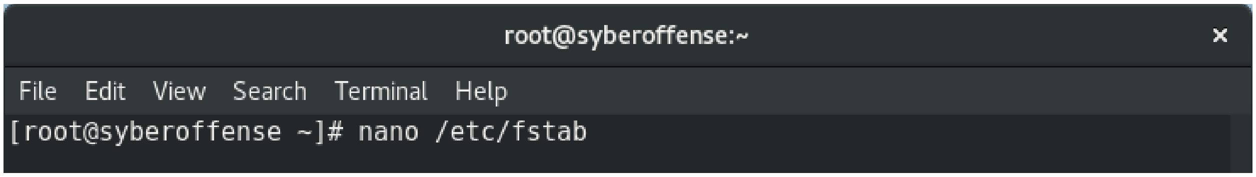


```
[root@syberoffense ~]# mount | grep nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)
192.168.145.136:/nfsshare on /mnt type nfs4 (rw,relatime,vers=4.2,rsize=262144,w
size=262144,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clienta
ddr=192.168.145.134,local_lock=none,addr=192.168.145.136)
[root@syberoffense ~]# 
```

The mount we created is only temporary and will be lost when the client is rebooted. To make the mount permanent on the client we need to add an entry to the fstap file located in the /etc directory.

In this example, I'll be using the nano text editor but you are free to use any text editor you choose. At the terminal type the following command:

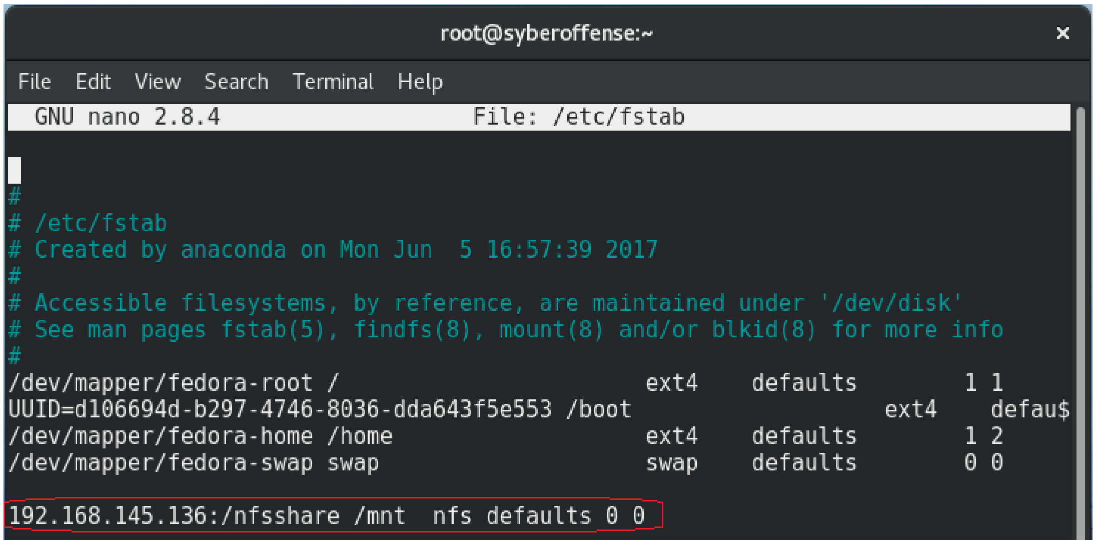
```
nano /etc/fstab
```



```
[root@syberoffense ~]# nano /etc/fstab
```

At the bottom of the text file, add the following line of text:

```
192.168.145.136:/nfsshare /mnt  nfs defaults 0 0
```



```
  GNU nano 2.8.4                   File: /etc/fstab


#
# /etc/fstab
# Created by anaconda on Mon Jun  5 16:57:39 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/fedora-root /                       ext4    defaults        1 1
UUID=d106694d-b297-4746-8036-dda643f5e553 /boot                   ext4    defau$
/dev/mapper/fedora-home /home                   ext4    defaults        1 2
/dev/mapper/fedora-swap swap                    swap    defaults        0 0

192.168.145.136:/nfsshare /mnt  nfs defaults 0 0
```

Save the file and come back to the terminal.

Return to the NFS server.

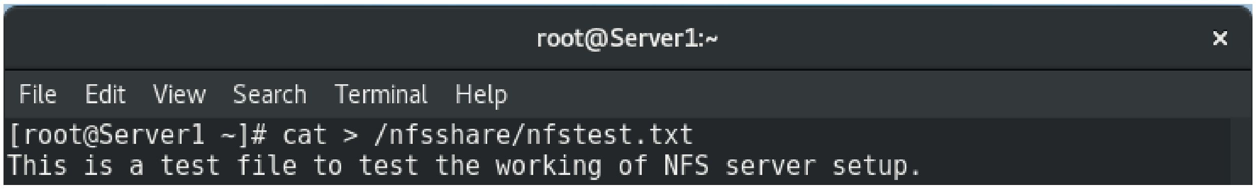
We next need to add a text file to the NFS share we created on the server. We will use this text file to text and see if our NFS server communicating with the NFS client and updating the NFS share.

At the server terminal, use the cat command to create and edit a new nfstest.txt file. At the prompt. type the following;

```
cat > /nfsshare/nfstest.txt
```

Add the following text:

```
This is a test file to test the working of NFS server setup.
```

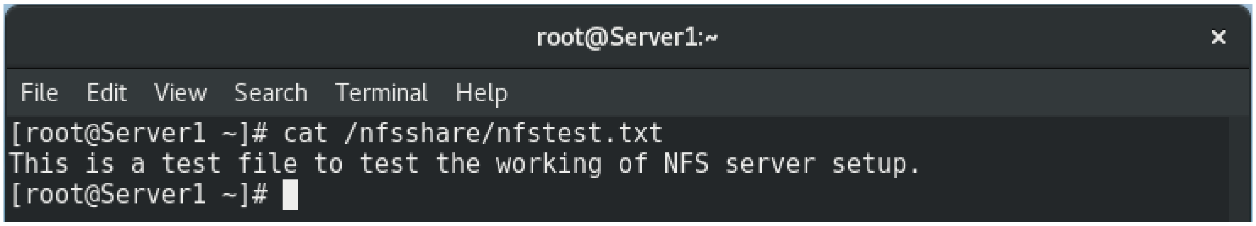


```
[root@Server1 ~]# cat > /nfsshare/nfstest.txt
This is a test file to test the working of NFS server setup.
```

To save your changes, type `ctrl+c.`

To verify the files does exist, view the contents of the nfstest.txt file using the following command:

```
cat /nfsshare/nfstest.txt
```



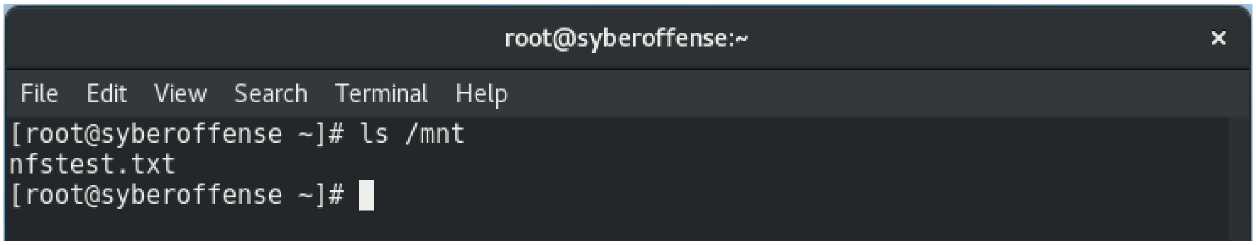
```
[root@Server1 ~]# cat /nfsshare/nfstest.txt
This is a test file to test the working of NFS server setup.
[root@Server1 ~]# 
```

To unmount a shared directory, you can use the “umount” command. Type the following command at the terminal:

Return to the client

If the NFS server share is properly configured and mounted on the client, we should be able to confirm the file we added is present on the client using the ls command to view the content of the mount. Type the following at the terminal of the client:

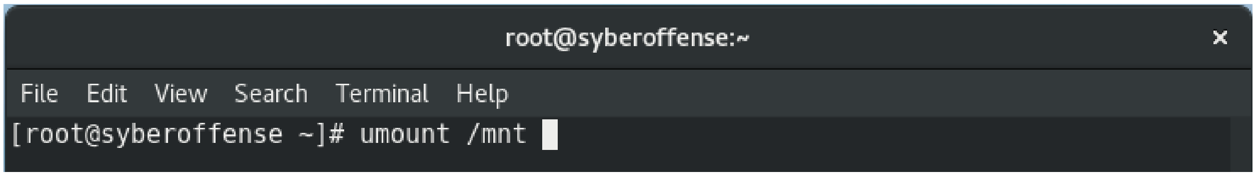
```
ls /mnt
```



```
[root@syberoffense ~]# ls /mnt
nfstest.txt
[root@syberoffense ~]#
```

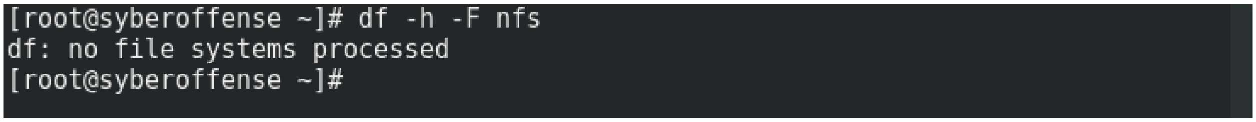
The NFS mount on the client shows that it has been updated with the new text file.

```
umount /mnt
```



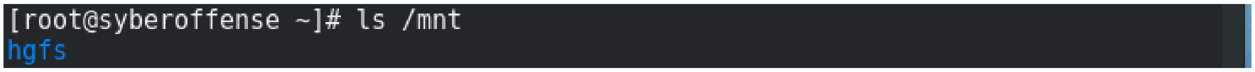
```
[root@syberoffense ~]# umount /mnt
```

To ensure all NFS mounts have been successfully removed form the client, type in the following command:



```
[root@syberoffense ~]# df -h -F nfs
df: no file systems processed
[root@syberoffense ~]#
```

All the NFS shares have been unmounted and the directories and the NFS files are no longer available.



```
[root@syberoffense ~]# ls /mnt
hgfs
```

Summary

In this lab, you learned how to configure NFs on the Linux server and access NFS shares from a Linux workstation. This is like creating an DFS server on a Windows network. By mounting the share on the client, the shares appear to be local which is all transparent to the to the user.