Search

...s through server and client ...

...s of network file systems in Linux.

...d | Tyler Carrigan (Editorial Team, Red Hat)

Image by riyan hidayat from Pixabay

Great Linux resources

- Advanced Linux commands cheat sheet

- Download RHEL 9 at no charge through the Red Hat Developer program

- A guide to installing applications on Linux

- Linux system administration skills assessment

- Cheat sheet: Old Linux commands and their modern replacements

I have used NFS in a limited capacity for years. I was familiar with the concept and had been accessing NFS shares, however, I had never actually configured one. Chances are, I am not alone in this. I thought I would learn how to set it all up and even bring you all along with me. Let's get started on the beginner's guide to learning NFS.

a distributed file system that allows various remote systems to access a file share. We red on a central server for security and ease of backup. NFS provides us with a file anaged and controls client access to resources.

to nail down a few prerequisites. First up, we need two different systems that are r via the network. Since NFS uses a server to client(s) relationship, we will use the

ple.com - 172.25.1.5
- NFS client - client.example.com - 172.25.1.4

You can use the `ping` command to confirm communications between the two systems. I have these two machines on a NAT network and have tested the connections both ways.

After that, let's ensure that both of our systems are up to date. As these systems are RHEL 8.2 and Fedora 32, we will use the following command on both the server and client:

```
[root@rhel tcarrigan]# sudo yum -y update
```

And finally, we need to install the `nfs-utils` package to both our systems.

```
[root@rhel tcarrigan]# sudo yum -y install nfs-utils (must do on both servers)
Updating Subscription Management repositories.
Last metadata expiration check: 0:55:54 ago on Wed 24 Jun 2020 11:53:45 AM EDT.
Package nfs-utils-1:2.3.3-31.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

*Note: the package is already installed in the above example.

Now, let's jump into configuring the server.

Configure the server

Step 1: Start and enable the newly-installed `nfs-utils` service.

```
[tcarrigan@rhel ~]$ sudo systemctl start nfs-server.service
[tcarrigan@rhel ~]$ sudo systemctl enable nfs-server.service
```

Step 2: Confirm the `nfs-server` service is up and running.

```
[tcarrigan@rhel ~]$ sudo systemctl status nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabl>
                          ) since Wed 2020-06-24 12:50:23 EDT; 18min >
                     ited, status=0/SUCCESS)
                  )

                nfs-server.service

              ample.com systemd[1]: Starting NFS server a>
              ample.com systemd[1]: Started NFS server an>
```

Subscribe now   x

Get the highlights in your inbox
every week.

Work email *

Country *

(you can see this information in column two).

```
                   :o -p | grep nfs
100003    3    tcp    2049  nfs
100003    4    tcp    2049  nfs
100227    3    tcp    2049  nfs_acl
```

*Note that you can find the NFS daemon configuration files at `/etc/nfs.conf`. You can also find the config file for the mount at `/etc/nfsmount.conf`.

The NFS service is now up and running on your server. Next, let's create an NFS share.

Create and export the share

First, we need to designate a folder for sharing. Since one doesn't already exist on my system, I will create a directory to share.

```
[tcarrigan@rhel ~]$ sudo mkdir -p /test/nfs_share/docs
```

Now, I learned from some trial and error and then well-written resources that you can avoid a lot of headache by changing the permissions and ownership to match the following:

```
[tcarrigan@rhel ~]$ sudo chown -R nobody: /test/nfs_share/docs/
[tcarrigan@rhel ~]$ sudo chmod -R 777 /test/nfs_share/docs/
```

*Note: you may not be able to do this in a production environment due to security considerations. Be sure that you know what you are doing before you remove all restrictions from a file or directory.

Next, we need to create an `/etc/exports` file.

```
[root@server docs]# vi /etc/exports
```

Make the following entry in the new file:

```
/test/nfs_share    172.25.1.0/24(rw,sync,no_all_squash,root_squash)
```

To better understand the parameters used here, let's break them down one by one.

- rw - Allows us to read and write to the NFS share.
- sync - Requires writing of changes to the disk before any other operations are completed.
- no_all_squash - Maps all UIDs and GIDs from the client request to the identical UIDs and GIDs on the NFS server.
- root_squash - Maps requests from the client-side root user to an anonymous UID/GID.

```
[root@server docs]# exportfs -rav
exporting 172.25.1.0/24:/test/nfs_share
```

Subscribe now   X e subnet here. You can include only a single IP or hostname here if you prefer.

Get the highlights in your inbox
every week.

Work email *

en created and exported the share. Next, we configure a tunnel through the firewall. We
:-bind, and `mountd`. Don't forget to reload the firewall config when completed.

Country *

```
d --permanent --add-service=nfs

d --permanent --add-service=rpc-bind
```

[Privacy Statement](#)

```
[                    ]# ...... ...d --permanent --add-service=mountd
success
[root@server]# firewall-cmd --reload
success
```

With the server-side completed, we can now focus our attention on the client machine.

Configure the client

*Note: all future steps are carried out on the client machine.

Since we already updated our system and installed the `nfs-utils` package, this should be pretty straightforward.

We start by creating an entry in `/etc/hosts` for the NFS server. It should look similar to this:

```
[root@client]# cat /etc/hosts
127.0.0.1  localhost
::1        localhost
172.25.1.5 localhost
```

Now, let's see if anything is shared from the NFS server. If you followed along in the earlier sections, you should see `/test/nfs_share/docs` as a shared directory.

```
[root@client ~]# showmount --exports nfs-server
Export list for nfs-server:
/test/nfs_share/docs 172.25.1.0/24
```

Next, create a directory on the client machine to mount the remote share.

```
[tcarrigan@client ~]$ sudo  mkdir p /test/client_share
```

Now that we have created a mount directory, let's mount the share.

```
[tcarrigan@client ~]$ sudo mount -t nfs 172.25.1.5:/test/nfs_share/docs
/home/tcarrigan/test/client_share
```

Run the following command to verify the share:

```
[tcarrigan@client ~]$ sudo mount | grep -i nfs
```

```
172.25.0.5:/test/nfs_share/docs /home/tcarrigan/test/client_share  nfs defaults 0 0
```

let's test the configured share. Create a file on the server in `/test/nfs_share/docs`

```
ls -lrt

) Jun 25 16:21 test_doc
```

orted to our client machine via NFS.

On the client machine:

```
[tcarrigan@client ~]$ cd test/client_share/
[tcarrigan@client client_share]$ ls
docs
[tcarrigan@client client_share]$ ls docs/
test_doc
[tcarrigan@client client_share]$
```

Here we see the `test_doc` exists on the NFS Server.

To test in the other direction, I create a file on the client named `client-test-doc`.

```
[tcarrigan@client docs]$ vi client-test-doc
```

Let's jump over to the server and see if we can view the newly created file.

NFS server:

```
[tcarrigan@server docs]$ ls -lrt
   total 8
   -rw-r--r--. 1 root        root        39 Jun 25 16:21 test_doc
   -rw-rw-r--. 1 tcarrigan tcarrigan   5 Jul  6 13:25 client-test-doc
```

We can see both the original file `test_doc` as well as the newly-created file `client-test-doc`.

Congratulations on setting up a functioning NFS server/client pair.

[Free online course: [Red Hat Enterprise Linux technical overview]().]

Topics: Linux   Storage

Tyler Carrigan

Tyler is the Sr. Community Manager at Enable Sysadmin, a submarine veteran, and an all-round tech enthusiast! He was first introduced to Red Hat in 2012 by way of a Red Hat Enterprise Linux-based combat system inside the USS Georgia Missile Control Center. [More about me]()
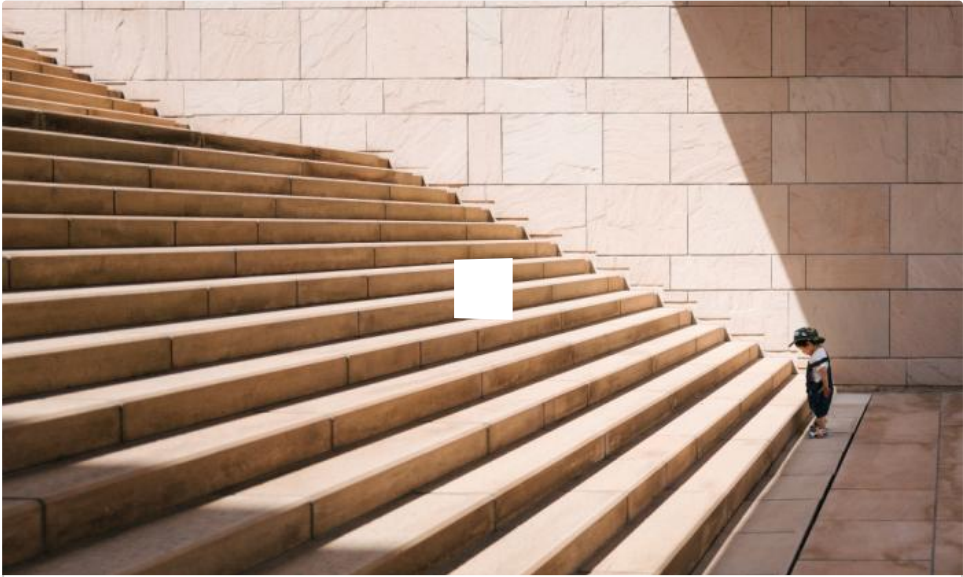
R

Understanding Linux file permissions, freeing up storage, and more tips for sysadmins

Check out Enable Sysadmin's top 10 articles from January 2023

Posted: February 3, 2023
Author: Vicki Walker (Editorial Team, Red Hat)

7 pro tips for using the GDB step command

There are several ways to step into even complicated functions, so give these GDB techniques a try next time you're troubleshooting your code.

Posted: January 20, 2023
Author: Alexandra Petlanova Hajkova (Red Hat)

How to install containerized applications on Fedora Silverblue

Silverblue is a Fedora Workstation variant that focuses on immutability and containers, so installing software is a little different from the usual Linux process.

Posted: January 26, 2023
Author: Don Watkins

Subscribe now X

Get the highlights in your inbox every week.

BEST CONTENT, DELIVERED TO YOUR INBOX

Work email *

Work email *

Country *

Country *

-- Please Select --

Submit

Privacy Statement

The opinions expressed on this website are those of each author, not of the author's employer or of Red Hat. The content published on this site are community contributions and are for informational purpose only AND ARE NOT, AND ARE NOT INTENDED TO BE, RED HAT DOCUMENTATION, SUPPORT, OR ADVICE.

Red Hat and the Red Hat logo are trademarks of Red Hat, Inc., registered in the United States and other countries.

About Red Hat

Jobs

Events

Locations

Contact Red Hat

Red Hat Blog

Diversity, equity, and inclusion

Cool Stuff Store

Red Hat Summit