How to Install and Configure an NFS Server on Ubuntu 18.04

James Kiarie | Last Updated: April 12, 2021 | Ubuntu | 13 Comments

NFS (**Network File Share**) is a protocol that allows you to share directories and files with other Linux clients in a network. The directory to be shared is usually created on the **NFS** server and files added to it.

The client systems mount the directory residing on the **NFS** server, which grants them access to the files created. NFS comes in handy when you need to share common data among client systems especially when they are running out of space.

This guide will comprise 2 main sections: Installing and configuring **NFS Server** on **Ubuntu 18.04/20.04** and Installing the **NFS client** on the client Linux system.

Installing and configuring NFS Server on Ubuntu

To install and configure the **NFS** server, follow the steps outlined below.

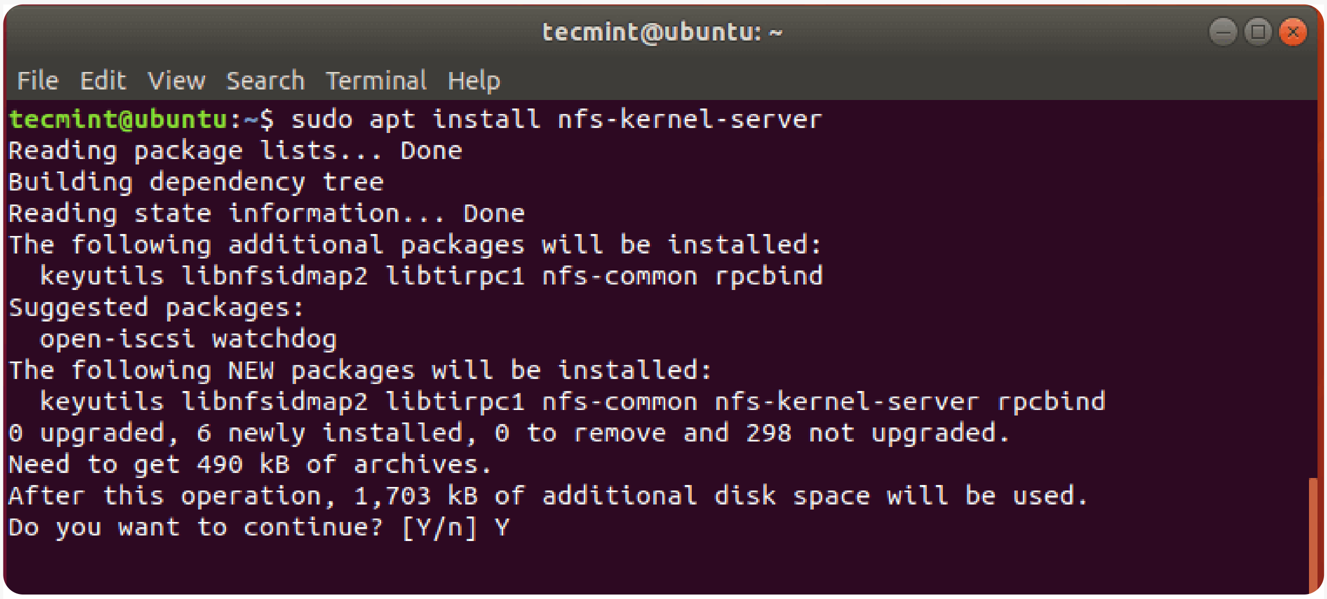
Step 1: Install NFS Kernel Server in Ubuntu

The first step is to install the **nfs-kernel-server** package on the server. But before we do this, let's first update the system packages using the following apt command.

```
$ sudo apt update
```

Once the update is complete, proceed and install the **nfs-kernel-server** package as shown below. This will store additional packages such as **nfs-common** and **rpcbind** which are equally crucial to the setup of the file share.

```
$ sudo apt install nfs-kernel-server
```



```
tecmint@ubuntu:~$ sudo apt install nfs-kernel-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  keyutils libnfsidmap2 libtirpc1 nfs-common rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server rpcbind
0 upgraded, 6 newly installed, 0 to remove and 298 not upgraded.
Need to get 490 kB of archives.
After this operation, 1,703 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Install NFS Server on Ubuntu

Step 2: Create an NFS Export Directory

The second step will be creating a directory that will be shared among client systems. This is also referred to as the export directory and it's in this directory that we shall later create files that will be accessible by client systems.

Run the command below by specifying the NFS mount directory name.

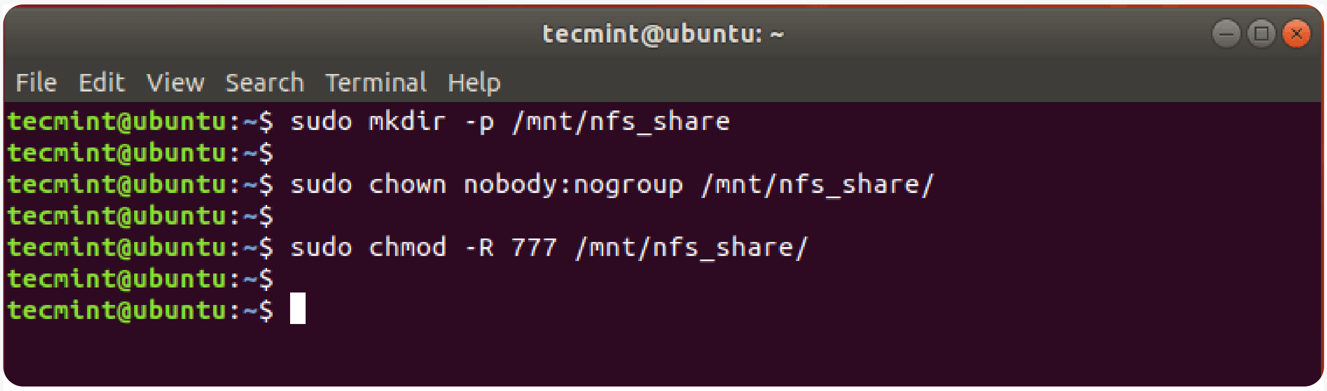
```
$ sudo mkdir -p /mnt/nfs_share
```

Since we want all the client machines to access the shared directory, remove any restrictions in the directory permissions.

```
$ sudo chown -R nobody:nogroup /mnt/nfs_share/
```

You can also tweak the file permissions to your preference. Here's we have given the read, write and execute privileges to all the contents inside the directory.

```
$ sudo chmod 777 /mnt/nfs_share/
```



```
tecmint@ubuntu:~$ sudo mkdir -p /mnt/nfs_share
tecmint@ubuntu:~$
tecmint@ubuntu:~$ sudo chown nobody:nogroup /mnt/nfs_share/
tecmint@ubuntu:~$
tecmint@ubuntu:~$ sudo chmod -R 777 /mnt/nfs_share/
tecmint@ubuntu:~$
tecmint@ubuntu:~$
```

Create NFS Share in Ubuntu

Step 3: Grant NFS Share Access to Client Systems

Permissions for accessing the NFS server are defined in the **/etc/exports** file. So open the file using your favorite text editor:

```
$ sudo vim /etc/exports
```

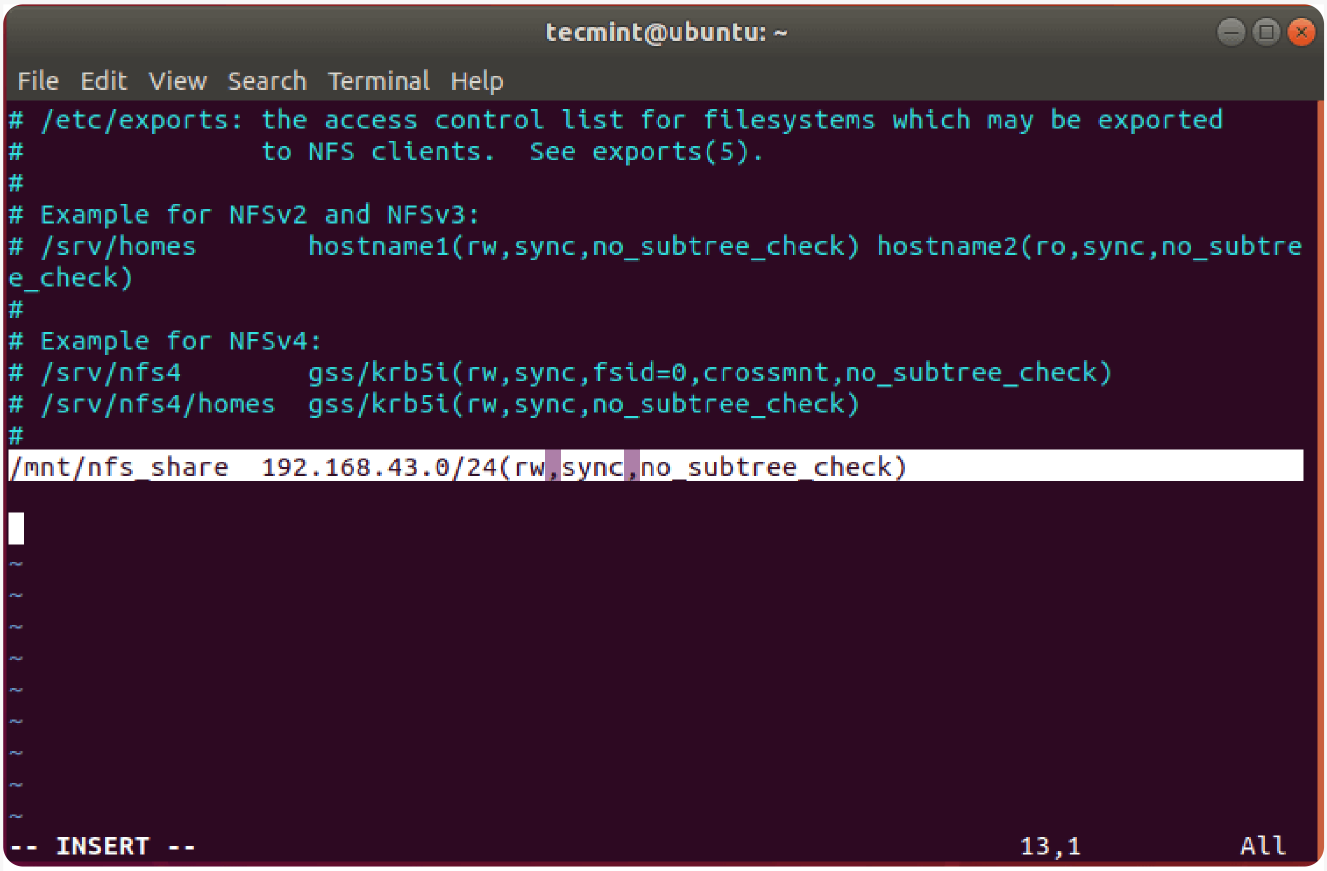
You can provide access to a single client, multiple clients, or specify an entire subnet.

In this guide, we have allowed an entire subnet to have access to the NFS share.

```
/mnt/nfs_share  192.168.43.0/24(rw,sync,no_subtree_check)
```

Explanation about the options used in the above command.

- **rw**: Stands for Read/Write.
- **sync**: Requires changes to be written to the disk before they are applied.
- **No_subtree_check**: Eliminates subtree checking.



```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/mnt/nfs_share  192.168.43.0/24(rw,sync,no_subtree_check)

~
~
~
~
~
~
~
~
~
-- INSERT --                                              13,1          All
```

Set NFS Share Access

To grant access to a single client, use the syntax:

```
/mnt/nfs_share  client_IP_1 (re,sync,no_subtree_check)
```

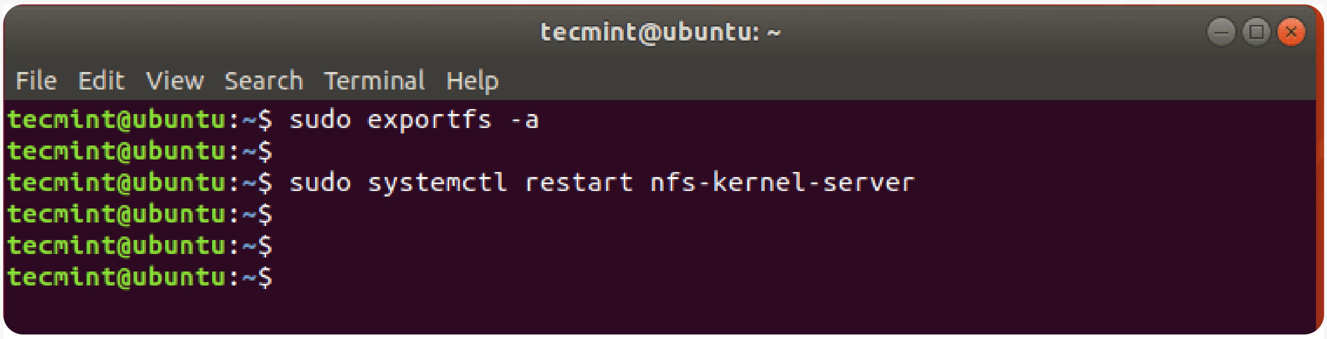
For multiple clients, specify each client on a separate file:

```
/mnt/nfs share  client IP 1 (re,sync,no subtree check)
```

Step 4: Export the NFS Share Directory

After granting access to the preferred client systems, export the NFS share directory and restart the NFS kernel server for the changes to come into effect.

```
$ sudo exportfs -a
$ sudo systemctl restart nfs-kernel-server
```



```
tecmint@ubuntu:~$ sudo exportfs -a
tecmint@ubuntu:~$
tecmint@ubuntu:~$ sudo systemctl restart nfs-kernel-server
tecmint@ubuntu:~$
tecmint@ubuntu:~$
tecmint@ubuntu:~$
```

Export NFS Share Directory

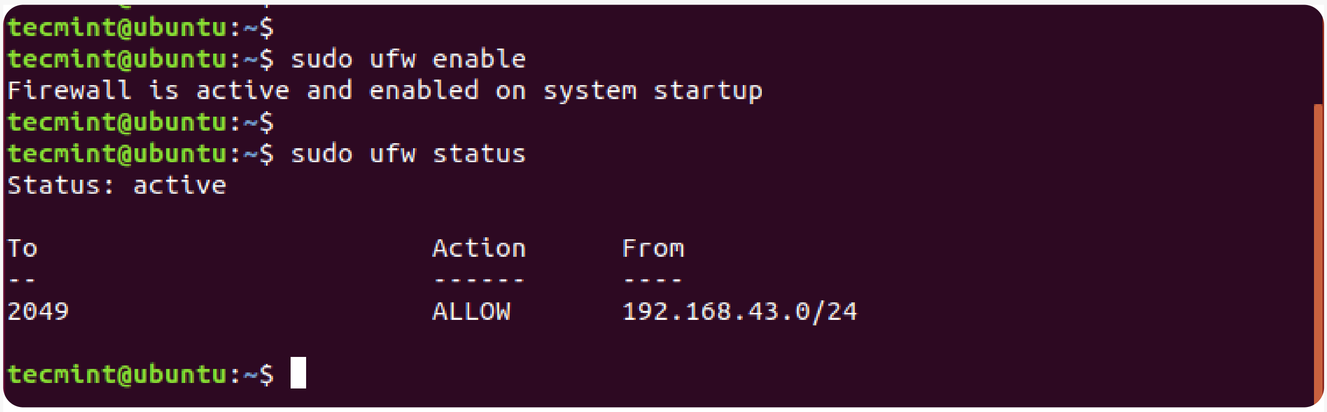
Step 5: Allow NFS Access through the Firewall

For the client to access the NFS share, you need to allow access through the firewall otherwise, accessing and mounting the shared directory will be impossible. To achieve this run the command:

```
$ sudo ufw allow from 192.168.43.0/24 to any port nfs
```

Reload or enable the firewall (if it was turned off) and check the status of the firewall. Port **2049**, which is the default file share, should be opened.

```
$ sudo ufw enable
$ sudo ufw status
```



```
tecmint@ubuntu:~$
tecmint@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
tecmint@ubuntu:~$
tecmint@ubuntu:~$ sudo ufw status
Status: active

To                           Action      From
--                           ------      ----
2049                         ALLOW       192.168.43.0/24

tecmint@ubuntu:~$
```

Open NFS Port on Firewall

Install the NFS Client on the Client Systems

We're done installing and configuring the NFS service on the Server, let's now install NFS on the client system.

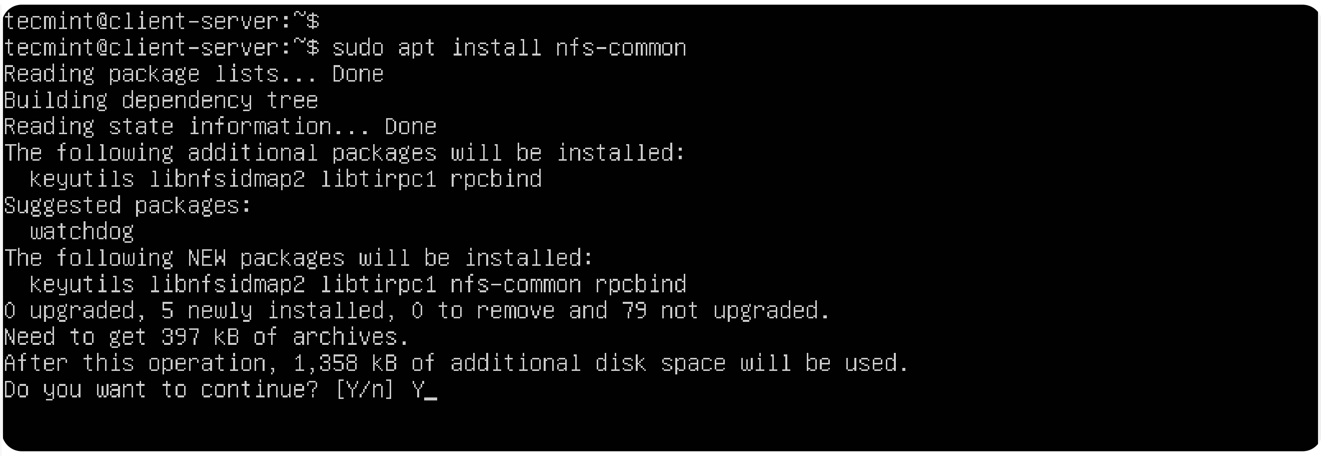
Step 1: Install the NFS-Common Package

As is the norm, begin by updating the system packages and repositories before anything else.

```
$ sudo apt update
```

Next, install **nfs-common** packages as shown.

```
$ sudo apt install nfs-common
```



```
tecmint@client-server:~$
tecmint@client-server:~$ sudo apt install nfs-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  keyutils libnfsidmap2 libtirpc1 rpcbind
Suggested packages:
  watchdog
The following NEW packages will be installed:
  keyutils libnfsidmap2 libtirpc1 nfs-common rpcbind
0 upgraded, 5 newly installed, 0 to remove and 79 not upgraded.
Need to get 397 kB of archives.
After this operation, 1,358 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y_
```

Install NFS on Client System

Step 2: Create an NFS Mount Point on Client

Next, you need to create a mount point on which you will mount the nfs share from the NFS server. To do this, run the command:

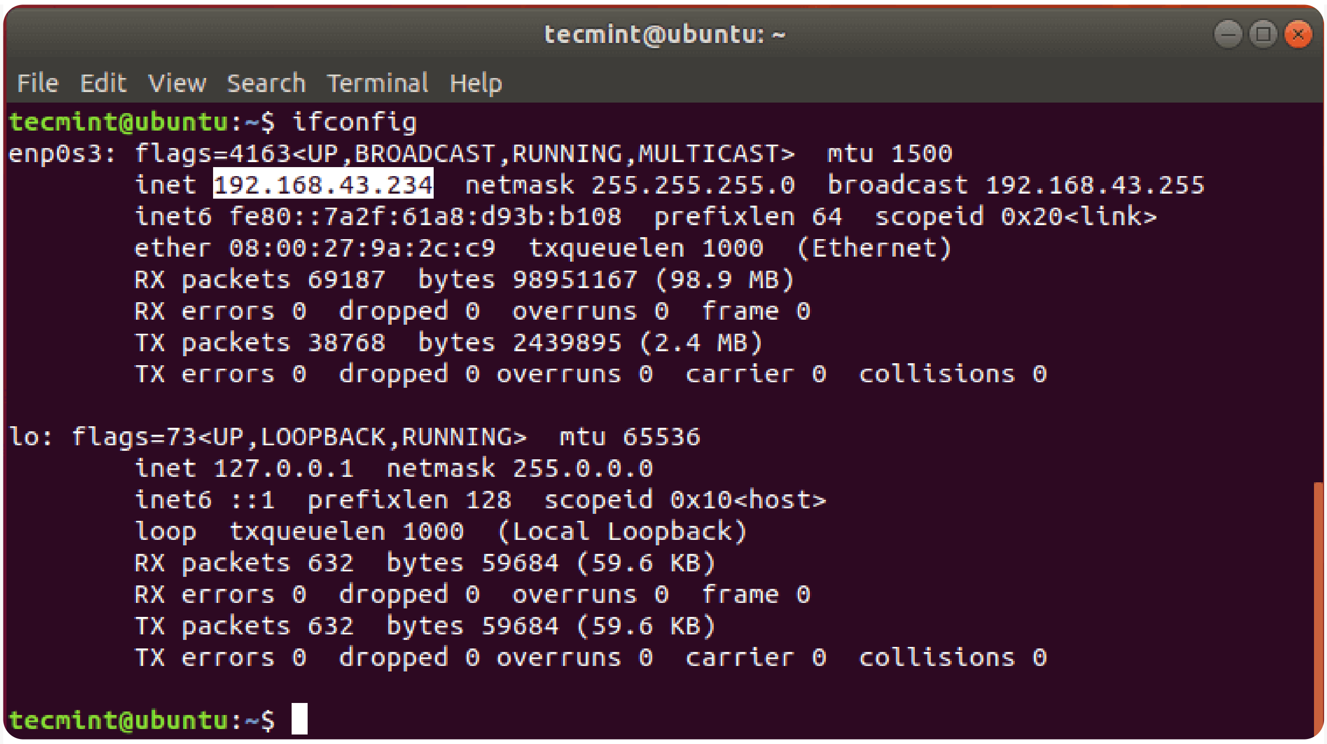
```
$ sudo mkdir -p /mnt/nfs_clientshare
```

Step 3: Mount NFS Share on Client System

The last step remaining is mounting the NFS share that is shared by the NFS server. This will enable the client system to access the shared directory.

Let’s check the NFS Server’s IP address using the ifconfig command.

```
$ ifconfig
```



```
tecmint@ubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.234  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::7a2f:61a8:d93b:b108  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:9a:2c:c9  txqueuelen 1000  (Ethernet)
        RX packets 69187  bytes 98951167 (98.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 38768  bytes 2439895 (2.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 632  bytes 59684 (59.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 632  bytes 59684 (59.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tecmint@ubuntu:~$ 
```

Check Ubuntu Server IP Address

To achieve this run the command:

```
$ sudo mount 192.168.43.234:/mnt/nfs_share  /mnt/nfs_clientsh
```

Step 4: Testing the NFS Share on Client System

To verify that our NFS setup is working, we are going to create a few files in the NFS share directory located in the server.

```
$ cd /mnt/nfs_share/
$ touch file1.txt file2.txt file3.txt
```

Now head back to the NFS client system and check if the files exist.

```
$ ls -l /mnt/nfs_clientshare/
```

Test NFS Share on Client

Great! The output confirms that we can access the files we just created on the NFS server!

And that about it. In this guide, we walked you through the installation and configuration of the **NFS** server on **Ubuntu 18.04** and **Ubuntu 20.04**. **NFS** is rarely used nowadays and has been relegated in favor of the more robust and secure Samba share protocol.

Nfs , Ubuntu Tips

‹ LFCA: Learn Basics of Network IP Addressing – Part 9

LFCA: Learn Binary and De Numbers in Network – Part 10 ›

> If you liked this article, then do subscribe to email alerts for Linux tutorials. If you have any questions or doubts? do ask for help in the comments section.

Related Posts

How to Create eLearning Platform with Moodle and ONLYOFFICE

How to Install KDE Plasma in Linux Desktop

How to Install Ubuntu Alongside With Windows in Dual-Boot

How to Install Akaunting Accounting Software on Ubuntu 20.04

How to Install ONLYOFFICE Desktop Editors in Linux

How to Install and Use ProtonVPN on Desktop Linux

13 thoughts on “How to Install and Configure an NFS Server on Ubuntu 18.04”

Myunggi Yi

July 2, 2021 at 6:43 am

I’m adding a new cluster, and I want to share **/data** of an old cluster with the new cluster too.

I’m trying to set up NFS on two computer clusters. Let’s say A and B (new) for each head node in the cluster. Compute nodes A1, A2 shares **/home** and **/data** from A, and compute nodes B1, B2 shares **/home** from B but **/data** from “A”.

In another word, cluster A, B shares its own **/home**, but **/data** of A is shared through the whole cluster A and B. In order to do this, I did set up NFS and IP/Mask with the following way to achieve the goal (All computers are connected to one network switch).

A with one external IP/24 (eht0) and one with 10.0.0.1/16 (eth1). exports setup of A:

> **/home** 10.0.0.0/24(rw,no_subtree_check,fsid=10,no_root_squash)
> /data 10.0.0.0/16(rw,no_subtree_check,fsid=11,no_root_squash)

stab of A

> /dev/sda1 /data xfs defaults 0 0

A1 IP with 10.0.0.2/16, A2 IP with 10.0.0.3/16 fstab of A1's

> 10.0.0.1:/home /home nfs … 10.0.0.1:/data /data nfs

B with its one external IP/24 (eht0) and one with 10.0.1.1/24 (eth1).

exports setup of B

> /home 10.0.1.0/24(rw,no_subtree_check,fsid=13,no_root_squash)

fstab of B

> 10.0.0.1:/data /data nfs

B1 with 10.0.1.2/24, B2 with 10.0.1.3/24 … 10.0.1.1:/home /home nfs 10.0.0.1:/data /data nfs

I thought my goal can be achieved simply by IP/Mask, but it doesn't work. I can mount /data of A on B's compute nodes, but I cannot mount /home of its own B on B's compute nodes.

How can I set up the network and NFS to achieve my goal?

Reply

Pedro

April 30, 2021 at 7:12 pm

Hi,

```
"sudo: ufw command not found"
```

Can I help me?

I need free access to specific IP.

Reply

James Kiarie

May 3, 2021 at 3:02 pm

Hey Pedro, it seems the UFW firewall is not installed on your system, in which case you need to install it using the command:

```
$ sudo apt install ufw
```

Then enable it as follows:

```
$ sudo  ufw enable
```

Reply

Deepop
January 16, 2021 at 5:55 pm

Hi James,

Thank you for this useful tutorial. It worked for me like a charm on Ubuntu. But I would like to get a step further and, there, I fail. Would you be able to help? Let me explain:

I want to share over NFS the content of my secondary drive. On my NFS server, this drive, called **NewVolume**, is mounted on **/media/solaris/NewVolume** as soon as I try to access it under user 'Solaris'.

When I try to export **/media/solaris/NewVolume** (or any subfolder of it) in **/etc/exports**, the '**showmount -e**' shows me that it works. But when I try to mount **/media/solaris/NewVolume** on any folder on my NFS client side, it tells me that the NFS server does not allow…

So I tried a trick: I mounted **/media/solaris/NewVolume** on **/mnt/nfs_share/newvolume**. From my NFS client, I can indeed see the '**newvolume**' folder in **/mnt/nfs_clientshare** (and I can see the content of **/media/solaris/NewVolume** in **/mnt/nfs_share/newvolume** in my NFS server) but **/mnt/nfs_clientshare/newvolume** is empty!

Any hint towards a solution is welcome.
Thanks in advance,
Deepop

Reply

Iyot Bihagay

November 5, 2020 at 8:20 am

I noticed that when the client machine is restarted, I need to rerun the mount command.

Is there a way to retain the mount after restarts? like editing fstab or something?

Thanks :)

Reply

Igor

November 3, 2020 at 2:48 pm

You don’t really NEED to add/enable a firewall port if the firewall is disabled on your system. I wasn’t too careful and broke my ssh connection by enabling the firewall. Please specify in the tutorial to ONLY do that if the firewall is currently ACTIVE. Otherwise, great tutorial! Thanks!

Reply

Anadi Mishra

July 23, 2020 at 3:11 pm

Didn’t work on KVM based VM.

```
$ sudo mount 192.168.43.234:/mnt/nfs_share  /mnt/nfs_clientshare
```

This worked for me.

```
$ sudo mount -t nfs -o nfsvers=3 192.168.43.234:/mnt/nfs_share  /mnt/nfs
```

Reply

myri
July 10, 2020 at 2:12 pm

Is the parameter “**re**” a typing mistake ??

```
/mnt/nfs_share  client_IP_1 (re,sync,no_subtree_check)
```

Reply

Ravi Saive
July 10, 2020 at 3:13 pm

@Myri,

The command is correct in the article.

```
/mnt/nfs_share  192.168.43.0/24(rw,sync,no_subtree_check)
```

Reply

Dennis Gentry

June 27, 2020 at 5:18 pm

This is the most helpful tutorial for setting up NFS on ubuntu. But it didn't work for me until, besides opening port **2049** for NFS, I opened port 111 for RPC. Until I figured that out, showmount just failed with some unhelpful RPC error.

Reply

James Kiarie

June 29, 2020 at 2:28 pm

Hey Dennis, that's a bit strange that you had to open port 111. But either way, I'm glad you finally hacked it.

Reply

Rich

August 7, 2020 at 2:15 pm

+1 for mentioning you needed 111. I also needed it but didn't figure that out until I read your comment and later was tailing **/var/log/syslog**. It also wanted a 16bit port number up at 40699, I think. When I then ran `rpcinfo -p`, it showed me that 111 was for portmapper and the 40699 port was being used by mountd. So I had to do something like:

```
$ sudo ufw allow proto tcp from 172.16.21.0/24 to any port 111,40699
$ sudo ufw allow proto tcp from 172.16.81.0/24 to any port 111,46999
```

Thanks to James for the original article. Very useful for setting up an NFS path between my OSX laptop and an Ubuntu VM running on my laptop. I was then able to do programming and testing that would have been much harder if I had to rely on standalone components.

Reply

James Kiarie

August 7, 2020 at 5:26 pm

Thank you Rich for your valueable input.

Reply

Got something to say? Join the discussion.

Have a question or suggestion? Please leave a comment to start the discussion. Please keep in mind that all comments are moderated and your email address will NOT be published.

Name *

Email *

Website

□ Save my name, email, and website in this browser for the next time I comment.

□ Notify me of followup comments via e-mail. You can also subscribe without commenting.

Post Comment

This site uses Akismet to reduce spam. Learn how your comment data is processed.

Learn Linux Commands and Tools

Terminalizer – Record Your Linux Terminal and Generate Animated GIF

5 Ways to Find a ‘Binary Command’ Description and Location on File System

How to Run MySQL/MariaDB Queries Directly from the Linux Command Line

5 ‘stat’ Command Examples for Linux Newbies

6 Interesting Funny Commands of Linux (Fun in Terminal)

How to Remove Packages with Dependencies Using Yum

If You Appreciate What We Do Here On TecMint, You Should Consider:

Linux Server Monitoring Tools

How to Monitor System Usage, Outages and Troubleshoot Linux Servers – Part 9

Dstat – A Resourceful Tool to Monitor Linux Server Performance in Real-Time

How To Install and Connect an Agent to Pandora FMS Server

How to Monitor Linux Commands Executed by System Users in Real-time

How to Monitor Nginx Performance Using Netdata on CentOS 7

iftop – A Real Time Linux Network Bandwidth Monitoring Tool

Learn Linux Tricks & Tips

How to Record and Replay Linux Terminal Sessions using ‘script’ and ‘scriptreplay’ Commands

How to Generate/Encrypt/Decrypt Random Passwords in Linux

Useful Commands to Create Commandline Chat Server and Remove Unwanted Packages in Linux

How to Count Number of Files and Subdirectories inside a Given Directory

How to View Configuration Files Without Comments in Linux

5 Useful Tools to Remember Linux Commands Forever

Best Linux Tools

6 Online Tools for Generating and Testing Cron Jobs for Linux

The 8 Best Open Source Web Servers

10 Best File and Disk Encryption Tools for Linux

7 Best Calendar Apps for Linux Desktop in 2020

The Top 5 Open-Source Microsoft 365 Alternatives for Linux

4 Best Linux Boot Loaders

Donate to TecMint Contact Us Advertise on TecMint Linux Services Copyright Policy
Privacy Policy Career Sponsored Post