Easy way to Setup NFS Server on Ubuntu 20.04

By **koromicha** - June 10, 2021

In this tutorial, we will discuss how to setup NFS server on Ubuntu 20.04 server. Network File system (NFS) *is a commonly used file-based storage system that allows remote systems to access files over a computer network and interact with them as if they were locally mounted. This enables system Administrators to group resources onto centralized servers on a network for easy sharing.*

Setup NFS Server on Ubuntu 20.04

Install NFS Server on Ubuntu 20.04

In order to setup NFS server on Ubuntu 20.04, you need to install the NFS kernel server package, which is the currently recommended NFS server for use with Linux, featuring features such as NFSv3 and NFSv4, Kerberos support via GSS, and much more.

NFS operates in server-client architecture. Hence, install the NFS kernel server on the NFS server system.

```
apt update
```

```
apt install nfs-kernel-server
```

Setup NFS Server on Ubuntu 20.04

Next, optionally update the NFS domain in the **/etc/idmapd.conf** configuration file. This will by default is set to your system's DNS domain if not specified. This may result in other systems in the different DNS domain not being able to mount the share.

For example, the domain of my system is;

```
hostname -d
```

```
kifarunix-demo.com
```

Hence, edit the `/etc/idmapd.conf` file and uncomment line 6 and set the value if **Domain** to the correct domain name.

```
sed -i "s/# Domain = localdomain/Domain = $(hostname -d)/" /etc/idmapd.conf
```

To verify;

```
grep Domain /etc/idmapd.conf
```

Sample output;

```
Domain = kifarunix-demo.com
```

Setup NFS Exports on Ubuntu 20.04

NFS Exports are file systems or directories on an NFS server that are shared or accessible to NFS clients.

In this tutorial, we will create two directories/NFS shares. A public and a private directory.

```
mkdir /media/{public,private}
```

Next, edit the **/etc/exports** configuration file and configure the above directories as NFS shares.

To set up an NFS share:

- Specify the directory to be shared
- IP addresses or domain names (if you have DNS server) of the systems to share
- the options associated with shared directory.

The format of the NFS share in the **/etc/exports** should look like:

```
nfshare nfsclient_IP/network/domain [sharingoptions]
```

```
vim /etc/exports
```

In our setup, the general directory is shared with anyone and the private directory with specific clients.

In that case, below are our NFS exports configurations look like below;

```
/media/public  *(ro,sync,root_squash,subtree_check)
```

```
/media/private 192.168.59.21(rw,sync,no_root_squash,no_subtree_check)
```

So, let’s place the lines above in the **/etc/exports**.

```
echo -e "/media/public  *(ro,sync,root_squash,subtree_check)\n/media/private
192.168.59.21(rw,sync,no_root_squash,no_subtree_check)" >> /etc/exports
```

The NFS share mount options used above are;

- **ro** mounts the directory on the client with read only permissions.
- **rw** mounts the shared directory on the client with read write permissions.
- **sync** ensures that any changes made to the shared directory is synchronized between the server and the client.
- **root_squash** maps the remote root user privileges into a non-privileged user on the NFS server.
- **no_root_squash** allows remote user to access the share with full privileges of the root user on the NFS server.
- **subtree_check** ensures that in case a directory instead of a block device is exported, the NFS server must check the existence of files in the shared directory for every request made.
- **no_subtree_check** specifies that the NFS server should not verify the availability of the files in the export for every request.

Consult **man 5 exports** for more NFS export mount options.

Restart NFS Server on Ubuntu 20.04;

```
systemctl restart nfs-server
```

Export NFS shares

Next, run the following command to export the shared directories.

```
exportfs -arvf
```

Sample output;

```
exporting 192.168.59.21:/media/private
exporting *:/media/public
```

For more information on **exportfs** options used above, **man exportfs** .

Allow NFS Share Access on Firewall

If firewall is running on Ubuntu 20.04 NFS server, allow access to the nfs share from the clients.

For example, to allow specific IPs/Networks to access the shares above;

```
ufw allow from 192.168.59.0/24 to any port nfs
```

nfs actually opens port 2049.

Check with:

```
rpcinfo -p | grep nfs
```

Next, since we are using a port based firewall, you need to configure a fixed port for NFS rpc.mountd (more on SecuringNFS). It uses random ports by default, making access control a bit difficult.

For example, let us say we want it to use port **50001** , then edit the file,/ **etc/default/nfs-kernel-server** , and replace add the port to the line;

```
RPCMOUNTDOPTS="--manage-gids"
```

such that it looks like;

```
RPCMOUNTDOPTS="--manage-gids --port 50001"
```

Save and exit the file once done editing.

Open the RPC mount port on firewall;

```
ufw allow from 192.168.59.0/24 to any port 50001
```

Open NFS portmapper port as well;

```
ufw allow from 192.168.59.0/24 to any port 111
```

Restart NFS server;

```
systemctl restart nfs-server
```

Configure NFS Client

Install NFS Client Packages

On a system that acts as an NFS client, install NFS client packages:

```
sudo apt install nfs-common -y
```

Similarly, update the NFS share domain:

```
sed -i "s/# Domain = localdomain/Domain = $(hostname -d)/" /etc/idmapd.conf
```

Create a directory to mount the remote NFS share

To access the remote shared directories on the NFS client, you need to mount those directories on the NFS client.

```
mkdir -p /nfs-shares/{public,private}
```

Listing NFS Shares

Run the following command to show mount information for an NFS server, whose IP address is 192.168.59.14.

```
showmount -e 192.168.59.14
```

```
Export list for 192.168.59.14:
/media/public  *
/media/private 192.168.59.21
```

Mount NFS Shares on NFS client

Mount the exports on NFS client as shown below;

```
sudo mount -t nfs 192.168.59.14:/media/public /nfs-shares/public
```

```
sudo mount -t nfs 192.168.59.14:media/private /nfs-shares/private/
```

Verify the mounting with the following command;

```
df -hT -P /nfs-shares/private/ /nfs-shares/public/
```

```
Filesystem                   Type  Size  Used Avail Use% Mounted on
192.168.59.14:/media/private nfs4   14G  4.4G  8.7G  34% /nfs-shares/private
192.168.59.14:/media/public  nfs4   14G  4.4G  8.7G  34% /nfs-shares/public
```

As you can see, both of the shares have been mounted.

To finalize on this, let us create some files on the NFS server and verify that the same becomes available to the client.

```
touch /media/public/public-files.txt touch /media/private/private-files.txt
```

On the client:

```
ls -1 /nfs-shares/public/ /nfs-shares/private/
```

```
/nfs-shares/private/:
private-files.txt

/nfs-shares/public/:
public-files.txt
```