Network File System (NFS)

NFS allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems almost as if they were local files.

Some of the most notable benefits that NFS can provide are:

* Local workstations use less disk space because commonly used data can be stored on a single machine and still remain accessible to others over the network.
* There is no need for users to have separate home directories on every network machine. Home directories could be set up on the NFS server and made available throughout the network.
* Storage devices such as floppy disks, CDROM drives, and USB Thumb drives can be used by other machines on the network. This may reduce the number of removable media drives throughout the network.

Installation

At a terminal prompt enter the following command to install the NFS Server:

sudo apt install nfs-kernel-server

To start the NFS server, you can run the following command at a terminal prompt:

sudo systemctl start nfs-kernel-server.service

Configuration

You can configure the directories to be exported by adding them to the /etc/exports file. For example:

/srv \*(ro,sync,subtree\_check)

/home \*.hostname.com(rw,sync,no\_subtree\_check)

/scratch \*(rw,async,no\_subtree\_check,no\_root\_squash)

Make sure any custom mount points you’re adding have been created (/srv and /home will already exist):

sudo mkdir /scratch

Apply the new config via:

sudo exportfs -a

You can replace \* with one of the hostname formats. Make the hostname declaration as specific as possible so unwanted systems cannot access the NFS mount. Be aware that \*.hostname.com will match foo.hostname.com but not foo.bar.my-domain.com.

The sync/async options control whether changes are gauranteed to be committed to stable storage before replying to requests. async thus gives a performance benefit but risks data loss or corruption. Even though sync is the default, it’s worth setting since exportfs will issue a warning if it’s left unspecified.

subtree\_check and no\_subtree\_check enables or disables a security verification that subdirectories a client attempts to mount for an exported filesystem are ones they’re permitted to do so. This verification step has some performance implications for some use cases, such as home directories with frequent file renames. Read-only filesystems are more suitable to enable subtree\_check on. Like with sync, exportfs will warn if it’s left unspecified.

There are a number of optional settings for NFS mounts for tuning performance, tightening security, or providing conveniences. These settings each have their own trade-offs so it is important to use them with care, only as needed for the particular use case. no\_root\_squash, for example, adds a convenience to allow root-owned files to be modified by any client system’s root user; in a multi-user environment where executables are allowed on a shared mount point, this could lead to security problems.

## NFS Client Configuration

To enable NFS support on a client system, enter the following command at the terminal prompt:

sudo apt install nfs-common

Use the mount command to mount a shared NFS directory from another machine, by typing a command line similar to the following at a terminal prompt:

sudo mkdir /opt/example

sudo mount example.hostname.com:/srv /opt/example

**Warning**

The mount point directory /opt/example must exist. There should be no files or subdirectories in the /opt/example directory, else they will become inaccessible until the nfs filesystem is unmounted.

An alternate way to mount an NFS share from another machine is to add a line to the /etc/fstab file. The line must state the hostname of the NFS server, the directory on the server being exported, and the directory on the local machine where the NFS share is to be mounted.

The general syntax for the line in /etc/fstab file is as follows:

example.hostname.com:/srv /opt/example nfs rsize=8192,wsize=8192,timeo=14,intr