**Setup NFS Server on CentOS 7 / RHEL 7**

[NFS](https://www.itzgeek.com/tag/nfs) stands for Network File System, helps you to share files and folders between [Linux](https://www.itzgeek.com/category/how-tos/linux) / Unix systems

## Benefits of NFS

* File / Folder sharing between \*nix systems
* Allows to mount remote filesystems locally
* Can be acted as Centralized Storage system
* It can be used as a Storage Domain ( Datastore) for VMware and other Virtualization Platform.
* Allows applications to share configuration and data files with multiple nodes.
* Allows having updated files across the share.

## Important Services

The following are the important NFS services, included in nfs-utils packages.

**rpcbind**: The rpcbind server converts RPC program numbers into universal addresses.

**nfs-server**: It enables clients to access NFS shares.

**nfs-lock / rpc-statd**: NFS file locking. Implement file lock recovery when an NFS server crashes and reboots.

**nfs-idmap**: It translates user and group ids into names, and to translate user and group names into ids

## Important Configuration Files

You would be working mainly on below configuration files to setup NFS server and Clients.

**/etc/exports**: It is the main configuration file, controls which file systems are exported to remote hosts and specifies options.

**/etc/fstab**: This file is used to control what file systems including NFS directories are mounted when the system boots.

**/etc/sysconfig/nfs**: This file is used to control which ports the required RPC services run on.

**/etc/hosts.allow** and **/etc/hosts.deny**: These files are called TCP wrappers, controls the access to the NFS server. It is used by NFS to decide whether or not to accept a connection coming in from another IP address.

### NFS Server side configuration

1. Install NFS package

#yum install -y nfs-utils

1. Start NFS and rpcbind services

#systemctl start nfs-server rpcbind

#systemctl enable nfs-server rpcbind

1. Create LVM mount point and mount it.

Ex: /u01

1. Change mount point permission to 777

#chmod 777 /u01

1. Add NFS share mount point to configuration file /etc/exports.

#vi /etc/exports

/u01 \*(rw,sync,no\_root\_squash)

1. Add entry to fstab

#vi /etc/fstab

/dev/oraclevg/oraclelv /u01 nfs defaults 0 0

**/u01**: shared directory

**rw**: Writable permission to shared folder

**sync**: All changes to the according filesystem are immediately flushed to disk; the respective write operations are being waited for.

**no\_root\_squash:**By default, any file request made by user root on the client machine is treated as by user nobody on the server. (Exactly which UID the request is mapped to depends on the UID of user “nobody” on the server, not the client.) If no\_root\_squash is selected, then root on the client machine will have the same level of access to the files on the system as root on the server.

1. Allow nfs-server and rpmbind ports in firewalld

firewall-cmd --permanent --add-service=rpc-bind

firewall-cmd --permanent --add-service=mountd

firewall-cmd --permanent --add-port=2049/tcp

firewall-cmd --permanent --add-port=2049/udp

firewall-cmd --reload

1. Export mount point

#exportfs –r

Exportfs options:

exportfs -v: Displays a list of shares files and export options on a server.  
exportfs -a: Exports all directories listed in /etc/exports.  
exportfs -u: UnExport one or more directories.  
exportfs -r: ReExport all directories after modifying /etc/exports.

### Client Server side configuration

1. Install NFS package

# yum install -y nfs-utils

1. Start the services

#systemctl start nfs-server

#systemctl start rpcbind

1. Stop firewalld

#systemctl stop firewalld

1. Check for the exported mount NFS shares

#showmount –e 192.168.0.3

1. Create mount point and mount it.

#mkdir /u01

#mount /t nfs 192.168.0.3:/u01 /u01

1. Add entry to fstab

#vi /etc/fstab

192.168.0.3:/u01 /u01 nfs defaults 0 0

NFS Server ports:

111 – for rpcbind enables server and connectivity

2049 – NFS service