# Oracle Linux 8 Managing the Network File System





Oracle Linux 8 Managing the Network File System,

G26605-03

Copyright © 2025, Oracle and/or its affiliates.

Documentation License
The content in this document is licensed under the <u>Creative Commons Attribution—Share Alike 4.0</u> (CC-BY-SA) license. In accordance with CC-BY-SA, if you distribute this content or an adaptation of it, you must provide attribution to Oracle and retain the original copyright notices.

# Contents

Preface	
About NFS	
Enabled Versions of NFS	
About NFS Services	
Configuring an NFS Server	
Editing the /etc/exports File	
The exportfs command	:
Using the exportfs Command	
Configuring NFS Clients	
Mounting an NES Share	



# **Preface**

This chapter includes information about managing the Network File System (NFS) in Oracle Linux 8, including tasks for configuring, administering, and using NFS.

# **About NFS**

NFS (Network File System) is a distributed file system that lets a client system access files over a network as though the files were on local storage.

An NFS server can share directory hierarchies in its local file systems with remote client systems over an IP-based network. After an NFS server exports a directory, NFS clients with the appropriate permissions can mount this directory. To the client systems, the directory appears as if it's a local directory.

The benefits of using NFS include centralized storage provisioning, improved data consistency, and reliability.

# **Enabled Versions of NFS**

The following versions of NFS are enabled in Oracle Linux:

- NFS version 3 (NFSv3), specified in RFC 1813.
- NFS version 4 (NFSv4), specified in <u>RFC 7530</u>.
- NFS version 4 minor version 1 (NFSv4.1), specified in RFC 8881.
- NFS version 4 minor version 2 (NFSv4.2), specified in RFC 7862.

NFSv3 provides safe, asynchronous writes, and efficient error handling. NFSv3 also uses 64-bit file sizes and offsets, which enable clients to access more than 2 GB of file data.

NFSv3 relies on Remote Procedure Call (RPC) services, which are controlled by the rpcbind service. The rpcbind service responds to requests for an RPC service and then sets up connections for the requested service. Separate services are used to handle locking and mounting protocols. Configuring a firewall to cope with the various ports that are used by all these services can be complex and error-prone.

### NFSv4:

- Works across firewalls and the Internet.
- Doesn't require the rpcbind service.
- Uses Access Control Lists (ACLs).
- Uses stateful operations.

NFSv4 requires the Transmission Control Protocol (TCP) running over an IP network. Instead of using rpcbind, the NFS server listens on TCP port 2049 for service requests. The mounting and locking protocols are also integrated into the NFSv4 protocol, which means that separate services are also not required for these protocols. These refinements make firewall configuration for NFSv4 no more difficult than for a service such as HTTP.

NFS clients mount by using NFSv4.2 (the default version), but fall back to NFSv4.1 when the server doesn't work with NFSv4.2. The mount later falls back to NFSv4.0 and then to NFSv3.



# **About NFS Services**

The NFS versions used by Oracle Linux 8 rely on Remote Procedure Calls (RPC) between clients and servers. To share or mount NFS file systems, the following required services work together, depending on which version of NFS is implemented. Note that all these services are started automatically:

### nfsd

Server kernel module that services requests for shared NFS file systems.

### rpcbind

Acts as a central "directory" service for NFS (and other RPC) services. Because some NFS services use dynamic ports, clients contact rpcbind on the server to find out which ports to use.

### rpc.mountd

Process that's used by an NFS server to process mount requests from NFSv3 clients. The service checks that the requested NFS share is exported by the NFS server.

### rpc.nfsd

Process that lets explicit NFS versions and protocols the server advertises to be defined.

### lockd

Kernel thread that runs on both clients and servers. The lockd process implements the Network Lock Manager (NLM) protocol, which lets NFSv3 clients lock files on the server. The daemon is started automatically whenever the NFS server is run and whenever an NFS file system is mounted.

### rpc-statd

Process that implements the Network Status Monitor (NSM) RPC protocol, which notifies NFS clients when an NFS server is restarted without first being brought down. The rpc-statd service is automatically started by the nfs-server service. This service doesn't require configuration by the user and isn't used with NFSv4.

### rpc-idmapd

Process that provides NFSv4 client and server upcalls, which map between on-the-wire NFSv4 names (strings in the form of user@domain) and local UIDs and GIDs. The ID mapping service isn't enabled by default and must be explicitly started when using NFSv4. For the idmapd process to operate correctly, you must configure the /etc/idmapd.conf file with the appropriate domain and settings for the network. Note that only NFSv4 uses the rpc-idmapd service, earlier NFS versions don't require it, because they use numeric IDs directly.

# (i) Note

The mounting and locking protocols are incorporated into the NFSv4 protocol. Also, the server listens on TCP port 2049. For this reason, NFSv4 doesn't need to interact with the rpcbind, lockd, and rpc-statd services. However, the nfs-mountd service is still required to set up exports on the NFS server; but, the service isn't involved in any over the wire operations.

The rpc-idmapd service only handles upcalls from the kernel and isn't itself directly involved in any over the wire operations. The service, however, might make naming service calls, which do result in over the wire lookups.



# Configuring an NFS Server

You configure an NFS server in Oracle Linux 8 by first editing the <code>/etc/exports</code> file to grant directory access to NFS clients, and then making those shared directories available using the <code>exportfs</code> command.

# Editing the /etc/exports File

The following steps describe how to configure shared directories using the /etc/exports file.

Configure the directories that an NFS server exports, including which clients can access those directories and what permissions they have, by editing the /etc/exports file.

# Note

You can also configure exports in files that you create under the /etc/exports.d directory. For example, /etc/exports.d/myexports.

1. Install nfs-utils.

If it's not already installed, install the nfs-utils package.

```
sudo dnf install nfs-utils
```

2. Configure the /etc/exports file.

Edit the /etc/exports file to define the directories that the server makes available for clients to mount, for example:

```
/var/folder 192.0.2.102(rw,async)
/usr/local/apps *(all_squash,anonuid=501,anongid=501,ro)
/var/projects/proj1 192.168.1.0/24(ro) mgmtpc(rw)
```

Each entry includes the local path to the exported directory, followed by a list of clients that can mount the directory and then client-specific export options (in parentheses). There can't be any spaces between the client specifier and the parenthesized list of options that apply to that client.

The following information explains the example export file entries in greater detail:

- Only the client system with the IP address 192.0.2.102 can mount the /var/folder directory with read and write permissions. All writes to the disk are asynchronous. This means that the server doesn't wait for write requests to be written to disk before responding to further requests from the client.
- As indicated by the wildcard (\*), all clients can mount the /usr/local/apps directory as read-only. All connecting users, including root users, are mapped to the local, unprivileged user with UID 501 and GID 501.



All clients on the 192.168.1.0/24 subnet can mount the /var/projects/proj1 directory as read-only. However, the client system named mgmtpc can mount the directory with read/write permissions.

For more information on the format of the etc/exports file, see the exports(5) manual page.

3. Configure the /etc/idmapd.conf file for NFSv4 clients.

If the server serves NFSv4 clients, edit the /etc/idmapd.conf file's definition for the Domain parameter by specifying the server's domain name.

```
Domain = mydom.com
```

This setting prevents the owner and group from being incorrectly listed as the anonymous user or group (nobody or nogroup) on NFS clients when the all\_squash mount option isn't specified.

4. Configure the firewall to enable access only for NFSv4 clients.

To enable access through the firewall for NFSv4 clients only, use the following commands:

```
sudo firewall-cmd --permanent --zone=zone --add-service=nfs
```

This configuration assumes that rpc.nfsd listens for client requests on the default TCP port 2049.

Configure the firewall to enable access for NFSv3 and NFSv4 clients.

To enable access through the firewall for NFSv3 and NFSv4 clients, do the following:

a. Edit the /etc/nfs.conf file to specify the port settings for handling network mount requests (mountd section) and status monitoring (statd section). Also, set the TCP port on which the network lock manager listens in the lockd section. For example:

```
# Ports that various services should listen on.
```

```
[mountd]
port = 892
[statd]
port = 662
[lockd]
port = 32803
```

If any of these ports are already in use, NFS fails to start. Use the lsof-i command to find an unused port and then change the setting in the /etc/nfs.conf file as appropriate.

To confirm on which ports RPC services are listening, use the rpcinfo -p command.

b. Restart the firewall service and configure the firewall to let NFSv3 connections through:

```
sudo firewall-cmd --permanent --zone=zone --add-port=2049/tcp --add-
port=111/tcp --add-port=32803/tcp --add-port=892/tcp --add-port=662/tcp
```



c. Reboot the server.

sudo systemctl reboot

6. Start the nfs-server service.

Start the nfs-server service and configure the service to start automatically when the system boots:

```
sudo systemctl enable --now nfs-server
```

7. Verify which versions of NFS the server works with.

Run the following command to check that the server provides the NFS versions that you have configured:

```
sudo cat /proc/fs/nfsd/versions
```

For example, the following output shows that the server provides NFSv3, NFSv4, NFSv4.1, and NFSv4.2:

```
+3 +4 +4.1 +4.2
```

8. List the exported directories.

Display a list of the exported directories.

```
sudo showmount -e
```

```
Export list for host01.mydom.com
/var/folder 192.0.2.102
/usr/local/apps *
/var/projects/proj1 192.168.1.0/24 mgmtpc
```

The exportfs command on the server displays the same information as the showmount - e command.

```
sudo /usr/sbin/exportfs -v
```

The showmount —a command displays all the current clients and all the exported directories that the clients have mounted.



To enable use of the showmount command from NFSv4 clients, specify a port number to the MOUNTD\_PORT parameter in /etc/nfs.conf. Then, create a firewall rule to enable access to this TCP port.

# The exportfs command

Describes the exportfs command.



The exportfs command lets an administrator export or unexport directories selectively, without needing to restart the NFS service. When provided with the appropriate options, the exportfs command writes the exported directories to the /var/lib/nfs/etab file.

Changes to the list of exported directories are effective immediately because the nfs-mountd service refers to the etab file for a specific directory's access privileges.

# Using the exportfs Command

If used without any options, the exportfs command displays a list of exported directories. Providing options to the exportfs command let you be selective about what gets exported.

The exportfs command options include the following:

Refreshes the list of exported directories in the /var/lib/nfs/etab file by incorporating any changes that were made to the list in the /etc/exports file.

-a

Exports all the directories that are specified in the /etc/exports file. This option can be combined with other options, to specify what action is performed on the directories.

-u

Unexports one or more shared directories.



### (i) Note

The exportfs -ua command suspends NFS file sharing, but keeps all NFS services running. To reenable NFS sharing, use the exportfs -r command.

Specifies verbose logging, which displays detailed information about the file systems that are being exported or unexported.

### Example 2-1 Export all directories in the /etc/exports file

To export every directory share defined in the /etc/exports file:

exportfs -a

### Example 2-2 Export a single directory from the /etc/exports file

To export only the /var/projects/proj1 directory from the /etc/exports file:

exportfs /var/projects/proj1

# Example 2-3 Unexport a directory defined in the /etc/exports file

To unexport the /var/projects/proj1 directory from the /etc/exports file:

exportfs -u /var/projects/proj1



# Example 2-4 Show detailed information about all exported directories

To show verbose information about all the directories being exported from the /etc/exports file:

exportfs -v

For more information on the exportfs command, see the exportfs(8), exports(5), and showmount(8) manual pages.

# **Configuring NFS Clients**

This section describes how to configure clients to access NFS shared directories over the network as if they were local directories.

# Mounting an NFS Share

Describes how to mount an NFS share on a client.

For more information on mounting NFS shares, see the mount(8), nfs(5), and showmount(8) manual pages.

1. Install nfs-utils.

If it's not already installed, install the nfs-utils package.

```
sudo dnf install nfs-utils
```

List the NFS server's exported directories.

Display a list of the directories that the NFS server exports. For example:

```
sudo showmount -e host01.mydom.com
```

The output of the previous command would be similar to the following:

```
Export list for host01.mydom.com
/var/folder 192.0.2.102
/usr/local/apps *
/var/projects/proj1 192.168.1.0/24 mgmtpc
```



Some servers don't accept querying the list of exports.

3. Mount an exported directory.

Mount an exported NFS directory on an available mount point. For example:

```
sudo mount -t nfs -r -o nosuid host01.mydoc.com:/usr/local/apps /apps
```

This example mounts the /usr/local/apps directory that's exported by host01.mydoc.com with read-only permissions on /apps. The nosuid option prevents remote users from gaining greater privileges by running a setuid program.





Typically, the -t (or --type) nfs option can be omitted and the mount command guesses the file type.

4. (Optional) Mount the NFS share when the system boots.

To configure the system to mount an NFS share at boot time, add an entry for the share to the /etc/fstab file, as shown in the following example:

host01.mydoc.com:/usr/local/apps /apps nfs ro,nosuid 0 0