

# NFS

**Autores:** Emiliano López ([emiliano.lopez@gmail.com](mailto:emiliano.lopez@gmail.com))

Maximiliano Boscovich ([maximiliano@boscovich.com.ar](mailto:maximiliano@boscovich.com.ar))

**Fecha:** 30/05/2018 02:03

NFS (Network File System) es un protocolo que permite a clientes acceder a recursos compartidos por un servidor como si fuesen locales. Fue desarrollado por Sun Microsystems en 1984.

Todas las versiones de NFS dependen de RPC (Remote Procedure Calls) entre clientes y servidor. RPC en Red Hat LE 7 son controlados por el servicio *rpcbind*.

## Instalación

```
yum install nfs-utils
```

## Configuración del servidor

Hay tres archivos principales para la configuración: */etc/exports*, */etc/hosts.allow*, y */etc/hosts.deny*, aunque con basta con modificar */etc/exports* para tener el servicio funcionando.

El formato de */etc/exports* consiste en líneas que contengan:

```
dir host1(opciones1) host2(opciones2)
```

Donde *dir* es el directorio que se desea compartir, *host1* y *host2* son las direcciones IPs o de red desde donde se permite el acceso con diferentes opciones.

La forma más fácil de configuración únicamente indica el directorio exportado y el host que tiene acceso al recurso. Esto sería:

```
/directorio/exportado yo.ejemplo.com
```

Aquí, *yo.ejemplo.com*, puede montar el directorio */directorio/exportado* del servidor NFS con sus opciones por defecto. Una vez modificado el archivo *exports* se debe ejecutar *exportfs -a* para que los cambios surtan efecto.

Luego basta con hacer `#systemctl start nfs` que inicia NFS y los procesos RPC apropiados.

### ACTIVIDAD 1

- Cree un directorio dentro de */home/vagrant* con su nombre y algunos archivos dentro
- Edite el archivo */etc/exports* de manera de compartir ese directorio con una única PC de su compañero

## Opciones por defecto

En caso que no se especifiquen opciones de configuración, se tomarán por defecto las siguientes:

- *ro* Se montan los sistemas de archivos como de sólo lectura (read-only). Los host remotos no pueden hacer cambios a los datos compartidos en el sistema de archivos. Para permitir que los hosts puedan hacer cambios, debe especificar la opción *rw* (lectura-escritura, read-write).

- *sync* El servidor NFS no responderá los nuevos pedidos hasta tanto no se hayan escrito en disco los pedidos previos. Para habilitar escritura asíncrona, especifique la opción *async*.
- *wdelay* Provoca que el servidor NFS retrase el escribir a disco si sospecha que otra petición de escritura es inminente. Esto puede mejorar el rendimiento reduciendo las veces que se debe acceder al disco por comandos de escritura separados. Use *no\_wdelay* para desactivar esta opción, la cual solo funciona si está usando la opción por defecto *sync*.
- *root\_squash* Previene a los usuarios root conectados remotamente de tener privilegios como root asignándoles el id del usuario de **nfsnobody**. Esto reconvierte el nivel de acceso del usuario root remoto al de usuario local más bajo, previniendo la alteración desautorizada de archivos en el servidor remoto. Alternativamente, la opción *no\_root\_squash* lo desactiva. Para reconvertir a todos los usuarios, incluyendo a root, use la opción *all\_squash*. Para especificar los ID de usuario y grupo para usar con usuarios remotos desde un host particular, utilice las opciones *anonuid* y *anongid*, respectivamente. De esta manera, puede crear una cuenta de usuario especial para que los usuarios NFS remotos compartan y especificar (*anonuid=<uid-value>,anongid=<gid-value>*), donde *<uid-value>* es el número de ID del usuario y *<gid-value>* es el número de ID del grupo.

Para conocer todas las opciones soportadas vea el manual haciendo `man exports`.

## Reglas de sintaxis

El archivo `/etc/exports` controla los directorios que son exportados hacia equipos remotos y especifica opciones. Tiene las siguientes **reglas de sintaxis**:

- Líneas en blanco son ignoradas .
- Para agregar un comentario, comience la línea con el caracter numeral (#).
- Cada directorio exportado debe tener su propia línea individual.
- Cualquier lista de equipos autorizados ubicados después de un directorio exportado debe ser separada por un espacio.
- Las opciones para cada equipo deben ir entre paréntesis. El paréntesis de apertura debe ser ubicado exactamente después del nombre del equipo o dirección IP, sin ningún espacio entre medio.

Por ejemplo, las siguientes dos líneas no significan lo mismo:

```
/home bob.example.com(rw)
/home bob.example.com (rw)
```

La primer línea permite únicamente usuarios del equipo bob.example.com con acceso de lectura/escritura al directorio /home. En cambio, la segunda línea permite a los usuarios de bob.example.com montarlo al directorio solo para lectura, mientras que el resto del mundo puede montarlo para lectura-escritura.

## Puertos

NFS requiere *rpcbind*, que asigna dinámicamente los puertos para los servicios RPC (Remote Process Call) y puede causar problemas para configurar reglas del firewall. Para permitir el acceso de los clientes a los recursos compartidos del servidor, edite el archivo `/etc/sysconfig/nfs` para especificar en cuales puertos deben correr los servicios RPC. En caso que el archivo no exista se lo debe crear y especificar lo siguiente:

```
RPCMOUNTDOPTS="-p port"
```

Esto agrega "-p port" al comando `rpc.mount`: `rpc.mount -p port`. Para especificar los puertos a ser usados por el servicio `nlockmgr`, especifique el número de puerto para la opción `nlm_tcpport` y `nlm_udpport` en el archivo `/etc/modprobe.d/lockd.conf`.

Si falla el inicio de NFS, se debe observar los logs en `/var/log/messages`. Comunmente NFS falla el inicio si se indica un puerto que ya se encuentra en uso. Luego de editar `/etc/sysconfig/nfs`, se debe reiniciar el servicio `nfs-config` para que los nuevos valores tengan efecto, haciendo:

```
systemctl restart nfs-config
```

## Configuración del Cliente

Una vez instalado `nfs-utils` se debe montar localmente el directorio remoto. Esto se puede hacer mediante el comando o utilizando el archivo `/etc/fstab`. Para el siguiente ejemplo, el servidor posee la dirección ip 10.10.10.13, el directorio compartido es `/home/usuario/compartido` y el directorio local donde se lo monta es `traidoxnfs`. De modo que el comando `#mount -t nfs IP_SERVER:DIR_REMOTO DIR_LOCAL` quedaría:

```
mount -t nfs 10.10.10.13:/home/vagrant/juan /home/vagrant/recursos
```

### ACTIVIDAD 2

- Arme una tabla con las direcciones IP y directorios compartidos
- Cree un directorio `/home/vagrant/recursos`
- Monte en el directorio previo el recurso servido por la PC de su compañero
- Intente escribir datos en el directorio y en caso de no ser posible solicite a su compañero que cambie los parámetros del recurso compartido para tener permisos de escritura.
- Desmonte el recurso remoto y cree bajo `/home/vagrant/recursos/` un directorio por cada recurso compartido en la red.
- Permita que toda la red pueda montar el recurso compartido
- Monte cada uno de los recursos compartidos en el directorio destinado para tal fin

El comando previo monta el directorio remoto mientras el sistema no se reinicie, para hacerlo permanente se debe utilizar el montado automático agregando la línea correspondiente en el archivo `/etc/fstab`:

```
10.10.10.13:/home/vagrant/juan /home/vagrant/recursos/juan nfs defaults 0 0
```

Para saber más sobre las opciones de montado vea `man fstab`.

### ACTIVIDAD 3

- Configure el archivo `/etc/fstab` para montar automáticamente alguno de los recursos.

## automount

El problema de usar `fstab` es que, independientemente de la frecuencia del uso, el sistema destina recursos para mantenerlo montado. Esto no suele ser un problema para algunos pocos directorios o equipos, pero si se deben mantener montados muchos sistemas remotos a la vez el desempeño se verá afectado.

Una alternativa a `/etc/fstab` es la herramienta basada en el kernel `automount`. Consiste en dos componentes:

- un módulo del kernel que implementa el sistema de archivos, y
- un demonio en el espacio de usuario que realiza todas las otras funciones

La utilidad **automount** puede montar y desmontar el sistema de archivos NFS automáticamente (bajo demanda), por lo que ahorra recursos de sistema. Se encuentra en el paquete **autofs**, para instalarlo:

```
yum install autofs
```

Primeramente se debe configurar el archivo `/etc/auto.master`. El formato consiste en un punto de montaje, un mapa y opciones, es decir, el directorio local, el archivo de configuración que indicará el recurso externo y opciones generales de autofs.

- El *punto de montaje* es el directorio local **padre** donde se montarán los recursos remotos, por ejemplo `/home/vagrant/recursos`.
- El *mapa* la ruta a otro archivo de configuración donde se especificarán las entradas de cada uno de los recursos remotos, por ejemplo `/etc/auto.misnfs`.
- Las *opciones* -en caso de existir- serán aplicadas a todos los montajes explicitados en el mapa previo.

```
/home/vagrant/recursos /etc/auto.misnfs --timeout=60
```

Luego se debe configurar el archivo que realiza el mapeo, con información similar a la que previamente hemos suministrado a `fstab`. Usando el nombre de archivo mencionado en el ítem previo, creamos `/etc/auto.misnfs` y cargamos una entrada por cada recurso:

```
juan -fstype=nfs 10.10.10.13:/home/vagrant/juan
```

La primer columna en el archivo de mapeo indica el directorio punto de montaje (`nfslocaldir` debe existir). La segunda columna indica las opciones de montado para autofs, mientras que la tercera indica la fuente de montado. Siguiendo la configuración realizada, el punto de montaje será `/home/nfslocaldir`.

Por último, reiniciar el servicio autofs.

```
# systemctl restart autofs
```

## ACTIVIDAD 4

- Desde el rol de cliente desmonte los recursos, esto es, comente el contenido del `fstab`.
- Monte los recursos externos usando `automount` para cada uno de los recursos de las PCs del laboratorio

## Referencias

- [NFS Server Config Storage Administration Guide Red Hat](#)
- [Enterprise Linux-7-Storage\\_Administration\\_Guide](#)
- [NFSProject](#)
- [AutoFS](#)