

16. NFS



16.1.- Introducción

NFS (Network File System) es un protocolo estándar de los sistemas Unix/Linux para compartir archivos en una red TCP/IP. Fue desarrollado en 1984 por Sun Microsystems que años más tarde se liberó bajo licencia GPL.

Permite a un usuario en un sistema acceder a los archivos de otro sistema a través de la red de igual forma a como lo hace con los archivos del sistema local, soportando de forma nativa los permisos y características de un sistema de archivos de Unix/Linux.

Las operaciones son síncronas, esto es que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado a esa operación; así se garantiza la integridad del archivo usado.

La última versión de NFS es NFSv4 (RFC 3530) que permite seguridad a través de Kerberos, trabajar con firewall, ACLs y utiliza operaciones con descripción de estado. Utiliza TCP para comunicar con el servidor, a diferencia de versiones anteriores que usaban TCP o UDP.

NFS exporta los datos en “claro”, por lo que no se recomienda exportar datos sin usar seguridad en autenticación y encriptación Kerberos.

Otros sistemas similares son: Samba, Andrew File System, SSHFS, Apple Talk, ...

→ Ventajas de usar NFS:

- Los sistemas locales utilizan menos espacio de disco ya que los datos están centralizados en un servidor, evitando tener que replicar la información.
- Los datos del servidor NFS pueden ser modificados y/o accedidos por varios clientes.
- Los directorios *home* de los usuarios pueden crearse en el servidor NFS y montarse en cualquier máquina desde donde el usuario se conecte, eliminando la necesidad de crear un *home* para el usuario en cada sistema de la organización.

El sistema NFS se divide en dos partes:

- Parte servidor: 1 servidor que exporta uno o más directorios (recursos) con los datos a compartir.
- Parte cliente: 1 o más clientes que acceden de forma remota a los datos almacenados en el servidor, montando en su sistema de archivos local el recurso exportado por el servidor.

16.2.- Configurar un servidor NFS

Los pasos a seguir son:

- 1.- Instalar el paquete *nfs-utils*. Normalmente en una instalación estándar está incluido no así en una instalación mínima.
- 2.- Crear el directorio a exportar. El usuario propietario debe ser *nfsnobody*.
- 3.- Configuración en el archivo */etc/exports*:

Añadiremos una línea por recurso a exportar con el formato:

<dir_exportar> <cliente>(opciones)

donde:

<dir_exportar>: directorio creado para almacenar los datos a exportar.

<cliente>: puede ser un nombre resoluble por DNS, una dirección IPv4 o IPv6, una dirección de red IPv4 o IPv6. Se pueden usar los comodines *, para representar cualquier número de caracteres alfanuméricos (incluido el nulo), o ?, para representar cualquier carácter alfanumérico, o un rango entre [].

Se pueden poner varios clientes en la misma línea del recurso a exportar separados por un espacio en blanco.

(opciones): pueden ser de acceso, de propiedad de los archivos o de seguridad.

- De acceso: **ro** sólo lectura, **rw** (por defecto) para lectura y escritura.
- De propiedad de los archivos: **no_root_squash** respeta el uid/gid 0 de *root*, **root_squash** (por defecto) cambia el uid/gid 0 de *root* por el del usuario *nfsnobody* y **all_squash** los archivos de todos los usuarios aparecen con el uid/gid *nfsnobody*.
- De seguridad: se pone **sec=<op_seguridad>**. Las opciones de seguridad posibles son:

none: acceso anónimo a los archivos, los archivos creados por un usuario remoto aparecen con usuario propietario *nfsnobody*. Para esto se necesita activar el booleano de SELinux *nfsd_anon_write*.

sys (por defecto): acceso a los archivos basado en los permisos estándar Linux.

krb5: el cliente proporciona una identidad basada en Kerberos y los permisos estándar de Linux.

krb5i: añade a krb5 integridad de los datos.

krb5p: añade a krb5i encriptación entre las peticiones cliente-servidor.

Estas tres últimas opciones necesitan el servicio *nfs-secure-server* arrancado y activado en el servidor y un archivo de keytab en */etc/krb5.keytab*.

4.- Abrir el firewall para permitir los accesos por NFS: puerto 2049/tcp para *nfsd*.

La versión NFSv4 dejó de depender del servicio de mapa de puertos (portmap) y de los servicios complementarios (mountd, lockd, ...) que están ahora incluidos en el propio NFS, con lo que sólo es necesario abrir el puerto 2049/tcp. En versiones anteriores NFSv2 y NFSv3 era necesario también abrir 111/udp 111/tcp, 662/tcp, 662/udp, 875/tcp, 875/udp, 892/tcp, 892/udp, 2049/tcp, 2049/udp, 32803/tcp y 32769/udp que son los puertos correspondientes a los mencionados servicios complementarios de *lockd*, *mountd*, *rquotad* y *statd*

5.- Configuración SELinux:

Se necesita configurar NFS para que exporte los directorios con los contextos de SELinux apropiados, para ello, en su archivo de configuración */etc/sysconfig/nfs*, hay que indicar que se use la versión 4.2; para esto, se configura esta línea así: `RPCNFSARGS="-V 4.2"`

y tras los cambios en el archivo de configuración, se debe reiniciar el servicio *nfs-server*. En el caso de estar utilizando seguridad Kerberos, también se debe reiniciar el servicio *nfs-secure-server*.

En los directorios exportados, hay que ponerles el contexto de SELinux adecuado para NFS, pudiendo ser *nfs_t* o *public_content_t*. Con ambos contextos se garantiza la lectura pero para permitir la escritura en el recurso exportado, se debe usar *nfs_t* o el contexto *public_content_rw_t* con el booleano *nfsd_anon_write* activo.

6.- Arranque y activación del servicio *nfs-server* y, si se usa autenticación Kerberos, también el servicio *nfs-secure-server*.

16.3.- Configurar un cliente NFS

Los pasos a seguir son:

1.- Identificar los recursos exportados por el servidor NFS.

En un punto de montaje provisional, p.e., */mnt* se monta con el comando *mount* el root del servidor NFS y con *ls* del punto de montaje provisional se ve qué recursos está exportando el servidor.

En el caso de usar versiones anteriores a NFSv4, se puede utilizar el comando *showmount -e <servidor_nfs>* para ver los recursos que exporta el servidor.

2.- Crear el punto de montaje donde se montará el recurso exportado.

3.- Montar el recurso remoto.

Se puede hacer de forma manual con el comando *mount*, pero no es persistente, si el sistema se reinicia perderíamos este montaje. Para hacerlo persistente, se puede hacer un montaje

automático con el archivo */etc/fstab* o con automonter.

→ Manual: usaremos el comando *mount* de la forma:

```
mount -t nfs -o <ops> <server_nfs>:<recurso_exportado> <pto_montaje>
```

donde:

<server_nfs>: sistema remoto que hace de servidor NFS.

<recurso_exportado>: directorio que exporta el servidor NFS.

<pto_montaje>: punto de montaje local creado en el punto anterior.

<ops>: opciones de montaje. Pueden ser:

- *sync* o *async* (por defecto) para indicar modo de funcionamiento síncrono o asíncrono respectivamente.
- *sec=<modo>*: opciones de seguridad, donde <modo> será *none*,*sys* (por defecto), *krb5*, *krb5p* o *krb5i* (visto en la parte servidor). Y si se quieren ver en el cliente los contextos SELinux que haya exportado el servidor, añadiremos *v4.2* en estas opciones de seguridad, separando con una coma, p.e., *sec=krb5,v4.2*.

→ Automático: usando el archivo */etc/fstab*: donde se añade una línea por recurso remoto con el formato:

```
<server_nfs>:<recurso_exportado> <pto_montaje> nfs <ops> 0 0
```

donde:

<server_nfs>: sistema remoto que hace de servidor NFS.

<recurso_exportado>: directorio que exporta el servidor NFS.

<pto_montaje>: punto de montaje local creado en el punto anterior.

<ops>: opciones de montaje. Las mismas usadas en el montaje manual.

→ Automonter: usando el servicio *autofs* que de forma automática monta bajo demanda recursos NFS compartidos y los desmonta cuando no están en uso.

Sus ventajas principales son:

- Los usuarios no necesitan tener privilegios de root para los comandos *mount* y *umount* y está accesible a todos los usuarios del sistema.
- No es necesario estar permanentemente conectado al recurso, dejando libres recursos de red y de sistema.
- Soporta mapeo de punto de montaje directo e indirecto proporcionando flexibilidad en las localizaciones de los puntos de montaje.
- Sólo necesita configuración en el lado cliente y usa las mismas opciones de

montaje que el comando *mount*.

- Los puntos de montaje indirectos se crean y borran por *autofs* eliminando la necesidad de gestionarlos manualmente.
- Soporta otros tipos de sistemas de archivos distintos a NFS como p.e. Samba.
- Es un servicio del sistema y se puede gestionar como tal.

16.3.1.- Crear un automontaje

Los pasos a seguir son:

1.- Instalar el paquete *autofs* en el cliente NFS

2.- Añadir un archivo de mapeo principal en */etc/auto.master.d/*

El nombre no es relevante, (mejor si es descriptivo) y debe tener la extensión *.autofs*. En este archivo añadir una entrada de mapeo de la forma:

- para mapeo indirecto: */<dir> /etc/auto.<detalles_montaje>*
- para mapeo directo: */- /etc/auto.<detalles_montaje>*

3.- Crear el archivo con los detalles del montaje con el nombre *auto.<detalles_montaje>* en */etc* con la línea:

- para mapeo indirecto: ** - <ops> <server_nfs>:<recurso_exportado>/&*
- para mapeo directo: *<pto_montaje> - <ops> <server_nfs>:<recurso_exportado>*

donde:

<server_nfs>: es el sistema remoto que hace de servidor NFS.

<recurso_exportado>: directorio que exporta el servidor NFS.

<pto_montaje>: punto de montaje local que debe existir previamente.

<ops>: opciones de montaje. Son las mismas que las que hemos visto en el montaje manual. Existen dos opciones especiales de automonter: *fstype* cuando el sistema de archivos exportado no es nfs y *strict* para tratar errores cuando se monte el sistema de archivos.

4.- Arrancar y activar el servicio *autofs* en el cliente.

16.4.- Caso práctico

Tenemos el sistema que estamos configurando **server1** que hará de servidor NFS exportando el recurso */publico* para que los sistemas de su propia red puedan tener acceso de lectura y escritura y el sistema con la IP 172.25.6.18 sólo tenga acceso de lectura. Se configurará NFS para que el acceso a los archivos se base en los permisos estándar Linux, exporte los contextos de SELinux

adecuados y se abrirá el *firewall* para poder permitir el acceso de los clientes.

Además como cliente NFS, el sistema **server1** montará en su sistema de archivos local */corporativo* el recurso exportado por el sistema **central** de */shares/corp* con la información accesible para toda la organización usando automonter y el recurso */shares/marketing* de forma automática en */folletos*. Los recursos que exporta **central**, se exportan en sólo lectura y securizados con kerberos.

RESOLUCIÓN

→ Configuración de **central** como servidor NFS:

- Instalamos el paquete *nfs-utils*:

```
[root@central ~]# yum install -y nfs-utils
```

- Creamos los directorios que contendrán los recursos a exportar **corp** y **marketing**:

```
[root@central ~]# mkdir -p /shares/{corp,marketing}
```

```
[root@central ~]# chown nfsnobody /shares/{corp,marketing}
```

Creamos algo de contenido en ellos para luego poder verlo desde el cliente **server1**:

```
[root@central ~]# mkdir /shares/corp/{viajes,normativa,vacaciones}
```

```
[root@central ~]# touch /shares/corp/viajes/plantilla-{dietas,despl,aloja}.xls
```

```
[root@central ~]# touch /shares/corp/normativa/iso-900{0,1,2}.pdf
```

```
[root@central ~]# touch /shares/corp/vacaciones/oferta{1,2}.txt
```

```
[root@central ~]# touch /shares/marketing/folleto{1..5}.pdf
```

- Configuración en el archivo */etc/exports*:

```
[root@central ~]# echo 'shares/corp 10.11.1.0/24(ro) ' >> /etc/exports
```

```
[root@central ~]# echo 'shares/marketing 10.11.1.0/24(rw) ' >> /etc/exports
```

```
[root@central ~]# exportfs -r (no es necesario ya que aun no se ha arrancado el servicio)
```

- Configuramos el firewall:

```
[root@central ~]# firewall-cmd --permanent --add-service=nfs --zone=internal
```

```
[root@central ~]# firewall-cmd --reload
```

```
[root@central ~]# firewall-cmd --list-services --zone=internal
```

```
mdns http dhcpv6-client kerberos nfs ssh dns ldap tftp samba-client
```

- Configuramos SELinux: se debe indicar la versión 4.2 en el archivo de configuración para que se puedan cargar los contextos SELinux en los recursos exportados:

```
[root@central ~]# sed 's/^RPCNFSDARGS="\"/RPCNFSDARGS="\-V 4.2"/g' /etc/sysconfig/nfs
```

Añado a las políticas de SELinux la política para que el directorio *publico* tenga el contexto *nfs_t* y la aplico:

```
[root@central ~]# semanage fcontext -a -t nfs_t '/shares/corp(/.*)?'  
[root@central ~]# semanage fcontext -a -t nfs_t '/shares/marketing(/.*)?'  
[root@central ~]# restorecon -vvFR /shares
```

- Arrancamos y activamos el servicio *nfs-server*:

```
[root@central ~]# systemctl start nfs-server  
systemctl start nfs-server  
[root@central ~]# systemctl enable nfs-server  
Created symlink from /etc/systemd/system/multi-user.target.wants/nfs-server.service  
to /usr/lib/systemd/system/nfs-server.service.
```

- Verificamos que el recurso se ha exportado:

```
[root@central ~]# exportfs -s  
/shares/corp  
10.11.1.0/24(ro,wdelay,root_squash,no_subtree_check,sec=sys,ro,secure,root_squash,no_all_squash)  
/shares/marketing  
10.11.1.0/24(rw,wdelay,root_squash,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
```

→ Configuración de **server1** como servidor NFS:

- Instalamos el paquete *nfs-utils*:

```
[root@server1 ~]# yum install -y nfs-utils
```

- Creamos el directorio a exportar **publico**:

```
[root@server1 ~]# mkdir /publico  
[root@server1 ~]# chown nfsnobody /publico
```

- Configuración en el archivo */etc/exports*:

```
[root@server1 ~]# echo '/publico 172.25.1.0/24(rw) 172.25.6.18(ro) ' >> /etc/exports  
[root@server1 ~]# exportfs -r (no es necesario ya que aun no se ha arrancado el servicio)
```

- Configuramos el firewall:

```
[root@server1 ~]# firewall-cmd --permanent --add-service=nfs
```

```
[root@server1 ~]# firewall-cmd -- reload
```

- Configuramos SELinux: se debe indicar la versión 4.2 en el archivo de configuración para que se puedan cargar los contextos SELinux en los recursos exportados:

```
[root@server1 ~]# sed 's/^RPCNFSDARGS=\\""/RPCNFSDARGS=\\"-V 4.2\\"/g' /etc/sysconfig/nfs
```

Añadimos a las políticas de SELinux la política para que el directorio *publico* tenga el contexto *nfs_t* y la aplicamos:

```
[root@server1 ~]# semanage fcontext -a -t nfs_t '/publico(/.*)?'
```

```
[root@server1 ~]# restorecon -vvFR /publico
```

- Arrancamos y activamos el servicio *nfs-server*:

```
[root@server1 ~]# systemctl start nfs-server; systemctl enable nfs-server
```

- Verificamos que el recurso se ha exportado:

```
[root@server1 ~]# exportfs -s
```

→ Configuración de **server1** como cliente NFS de **central**:

- Instalamos el paquete *nfs-utils* (ya realizado en el punto anterior) y el paquete *autofs*:

```
[root@server1 ~]# yum install -y autofs
```

Comprobamos que accedo a los recursos de central montando el root de central en un punto de montaje provisional, p.e., */mnt*:

```
[root@server1 ~]# mount central:/ /mnt
```

```
[root@server1 ~]# tree /mnt
```

```
/mnt
├── shares
│   ├── corp
│   │   ├── normativa
│   │   │   ├── iso9000.pdf
│   │   │   ├── iso9001.pdf
│   │   │   └── iso9002.pdf
│   │   ├── vacaciones
│   │   │   ├── oferta1.txt
│   │   │   └── oferta2.txt
│   │   └── viajes
│   │       ├── platilla-aloja.xls
│   │       ├── platilla-despl.xls
│   │       └── platilla-dietas.xls
│   └── marketing
│       ├── folleto1.pdf
│       ├── folleto2.pdf
│       ├── folleto3.pdf
│       └── folleto4.pdf
```


└─ folleto5.pdf

```
[root@server1 ~]# umount /mnt
```

- Creamos el directorio local */folletos*: *corporativo no hace falta → automonter*

```
[root@server1 ~]# mkdir /folletos
```

- Para el montado automático del recurso */shares/marketing* del servidor NFS **central** con el archivo */etc/fstab*, añadimos la siguiente línea:

```
[root@server1 ~]# echo 'central:/shares/marketing /folletos nfs sync,sec=sys,v4.2 0 0'
>>/etc/fstab
```

```
[root@server1 ~]# ls /folletos
```

folleto1.pdf folleto2.pdf folleto3.pdf folleto4.pdf folleto5.pdf

- Para el montado indirecto del recurso */shares/corp* de **central** usando automonter, creamos el archivo de mapeo principal con la configuración que apunta al archivo con la descripción del montaje:

```
[root@server1 ~]# echo '/corporativo /etc/auto.corporativo' >
/etc/auto.master.d/corporativo.autofs
```

y creamos el archivo con la descripción del montaje:

```
[root@server1 ~]# echo '* -ro,sec=sys,v4.2 central:/shares/corp/&' >
/etc/auto.corporativo
```

- Arrancamos y activamos el servicio *autofs*:

```
[root@server1 ~]# systemctl start autofs; systemctl enable autofs
```

Created symlink from /etc/systemd/system/multi-user.target.wants/autofs.service to /usr/lib/systemd/system/autofs.service.

```
[root@server1 ~]# ls /corporativo/{viajes,vacaciones,normativa}
```

/corporativo/normativa:

iso9000.pdf iso9001.pdf iso9002.pdf

/corporativo/vacaciones:

oferta1.txt oferta2.txt

/corporativo/viajes:

platilla-aloja.xls platilla-despl.xls platilla-dietas.xls