

# Compartición de datos con NFS

NyFSes el protocolo tradicional de compartición de archivos en sistemas Unix. Aunque su antigüedad lo hace menos popular a ojos de los jóvenes usuarios de Linux, sigue siendo interesante conocerlo por la rapidez y la simplicidad en su despliegue para una compartición entre dos sistemas Linux o Unix. Además, NyFS está gozando recientemente de un renovado interés gracias a algunas aplicaciones que lo usan, como las infraestructuras Vmware para acceder a espacios de almacenamiento de bajo coste o las unidades multimedia domésticas que acceden a servidores de archivos.

## 1. Compartición de directorios

### a. Observación de particiones activas

Las particiones NyFS activas en un sistema se declaran mediante un directorio local y están accesibles para algunos clientes con ciertas opciones. Los clientes autorizados así como las opciones se declaran cuando se activa la partición. Si se encuentra un sistema ya configurado, puede ser útil hacer un diagnóstico de las particiones activas de este sistema. Este diagnóstico se realiza usando el comando **exportfs**.

Ejemplo de uso del comando exportfs para observar las particiones activas

En este ejemplo, el directorio /perso se comparte solamente para la dirección 192.168.0.20, mientras que /nas se comparte para todos los clientes.

```
alfa:~# exportfs
/data/perso    <192.168.0.20>
/nas           <world>
alfa:~#
```

Se pueden observar las estadísticas relacionadas con la actividad NyFS mediante el comando **nfsstat**.

Visualización de estadísticas de NFS

El comando **nfsstat** sirve sobre todo para verificar una actividad o una ausencia de actividad en un servidor NyFS.

```
usuario@servidor:~$ nfsstat
Server rpc stats:
calls      badcalls    badauth    badclnt    xdrcll
12         0             0          0          0

Server nfs v3:
null      getattr      setattr      lookup      access      readlink
2         18% 2        18% 0         0% 2        18% 1        9% 0         0%
read      write        create       mkdir        symlink      mknod
0         0% 0         0% 0         0% 0         0% 0         0% 0         0%
remove    rmdir        rename       link         readdir      readdirplus
0         0% 0         0% 0         0% 0         0% 0         0% 0         0%
fsstat    fsinfo       pathconf     commit
0         0% 3         27% 1         9% 0         0%

Client rpc stats:
calls      retrans      authrefrsh
0         0            0

usuario@servidor:~$
```

### b. Compartición puntual

El comando **exportfs** también permite declarar una partición de forma interactiva. Se utiliza para declarar particiones puntuales.

Sintaxis del comando exportfs para una partición puntual

Comando exportfs: opciones y parámetros	
<i>dirección_cliente</i>	Dirección IP del cliente o de la red que se puede conectar a la compartición. yEl comodín "*" autoriza a todos los clientes para que se puedan conectar.
<i>ruta_compartición</i>	Ruta absoluta del directorio que se comparte.

Por supuesto, hace yya mucho tiempo que el control de acceso basado en la dirección IP no es una garantía de seguridad.

### c. Servicio NyFS y compartición permanente

Naturalmente, se puede declarar una compartición permanente activa para el inicio del servicio NyFS y Establecer una declaración se realiza en el archivo **/etc/exports**. Hay que tener en cuenta que, según la distribución, este archivo posiblemente no se haya creado después de la instalación del servicio y hay que crearlo desde cero.

y Formato del archivo /etc/exports

```
compartición1 dirección_cliente1
compartición2 dirección_cliente2
```

y Este archivo se lee cada vez que arranca el servicio NyFS o en cada llamada al comando **exportfs** con la opción **-a**. Hay que tener en cuenta que las comparticiones se expresan siempre con su ruta absoluta, es decir, se expresan en relación a la raíz del sistema de archivos.

y El script de administración del servicio NyFS provoca la ejecución de tres daemons estándar:

- portmap: gestiona las peticiones RPC (*Remote Procedure Call*).
- nfsd: espacio de usuario del servicio NyFS Inicia los threads NyFS para las conexiones cliente.
- mountd: gestiona las peticiones de montaje de los clientes.

y El comando **rpcinfo** permite efectuar una petición RPC a un servidor y mostrar los daemons gestionados.

### d. Opciones de compartición

Algunas opciones modifican el comportamiento del servidor NyFS para cada una de las comparticiones que alberga. Las opciones se especifican mediante el comando **exportfs** si se utiliza dinámicamente o en el archivo **/etc/exports** si se utiliza NyFS como servicio.

Opciones NyFS comunes	
ro	Acceso en modo sólo lectura.
rw	Acceso en modo lectura / escritura.
sync	Acceso en modo escritura síncrona. Los datos se escriben inmediatamente.
async	Acceso en modo escritura asíncrona. Utilización de una caché de escritura.
root_squash	Comportamiento por defecto. La cuenta root pierde sus privilegios en la compartición.
no_root_squash	La cuenta root conserva sus privilegios en la compartición.
nolock	No se bloquean los archivos a los que se accede.

yEjemplode uso del comando exportfs con la opción de sólo lectura

Si hayvarias opciones, se deben separar por comas.

```
root@servidor# exportfs -o ro */data
root@servidor#
```

yEjemplode archivo /etc/exports con la opción de sólo lectura

yEparámetro \* o una dirección IP de un cliente autorizado son indispensables para un buen funcionamiento.

```
/data      *(ro)
```

yEjemplode visualización de las comparticiones activas con sus opciones

Se muestran las opciones explícitas así como las opciones por defecto.

```
alfa:~# exportfs -v
/perso      192.168.0.20 (rw,wdelay,root_squash,no_subtree_check)
/data       <world> (ro,wdelay,root_squash,no_subtree_check)
alfa:~#
```

## 2. Configuración de clientes

### a. Visualización de las comparticiones remotas

yEtomando **showmount** permite mostrar la información de un servidor NyFSremoto.

Visualización de las comparticiones remotas con showmount

```
showmount --exports servidor
```

Donde *servidor* representa la dirección IP del servidor del que queremos obtener las comparticiones.

### b. Montaje de un directorio remoto

Los ordenadores cliente acceden a una compartición NyFSmediante una operación de montaje. Usan la compartición montada como si de una estructura de directorios local se tratase.

Montaje de una compartición NFS

```
mount -t nfs dirección_servidor:/ruta_compartición punto_de_montaje
```

Montaje NyFSopciones yparámetros	
-t nfs	Indica que el dispositivo que se montará es una compartición NyFSremota e invoca al subprograma cliente NyFS.
dirección_servidor	La dirección IP del servidor NyFS.
ruta_compartición	La ruta absoluta del directorio compartido en el servidor.
punto_de_montaje	yEDirectorio local del cliente en el que se montará la compartición NyFS.

## 3. Administración de las identidades

### a. Los permisos del cliente

Puede ser bastante sorprendente comprobar que no se solicita ningún tipo de identificación en la conexión a una compartición NyFS. Se encontrará conectado sin tener que haber proporcionado credenciales. De hecho, NyFS considera que los identificadores de los usuarios son coherentes entre el servidor y sus clientes, es decir, que todas las cuentas de usuario son idénticas en todas las máquinas y que sus identificadores de usuario (uid) son todos los mismos.

Cuando un cliente se conecta a una compartición NyFS, presenta su uid y tiene exactamente los permisos que el usuario con el mismo uid en el servidor. No se realiza ningún control.

### b. El caso particular del superusuario

Como la cuenta root tiene el uid 0 sea cual sea el sistema Linux, un cliente que se conecta al servidor con su cuenta de superusuario en teoría debería tener todos los privilegios en la compartición. Esta embarazosa situación se resuelve mediante la aplicación implícita de una opción de compartición: **root\_squash**. Efecto, si un servidor recibe una solicitud de conexión de una cuenta con el uid 0, modifica su identificador y le aplica en la compartición el uid de una cuenta de servicio NyFS. Esta cuenta (según la distribución `nfsnobody`, `nfsnobody`, `nfsnobody`...) en general sólo tendrá permisos del conjunto de usuarios "otros" en el sistema del servidor.