

Archivos compartidos

1. NFS

a. Inicio

El sistema de archivos compartidos **NFS** (*Network File System*) o sistema de archivos de red permite compartir todo o parte de un sistema de archivos destinados a clientes NFS, a menudo otros Unix. En su versión básica, es un sistema simple y eficaz. Vamos a estudiar la versión 2, pero los ejemplos funcionan también en versión 3 (observe sin embargo, que el protocolo de transporte por defecto que pasa a TCP en lugar de UDP para algunas distribuciones).

NFS se basa en el **portmapper (portmap)**, el soporte **NFS** del núcleo y los servicios **rpc.nfsd** y **rpc.mountd**.

En las distribuciones no equipadas con systemd, para iniciar el servicio NFS, se debe ejecutar portmap y NFS (verificar el estado antes).

```
# service portmap status # /etc/init.d/portmap status o rpcinfo -p
# service nfs status
# service portmap start
# service nfs start
# service nfslock start
```

En algunas distribuciones, portmap se cambia por **rpcbind**. Hay que iniciar este servicio en lugar de portmap.

En las distribuciones con systemd, primero hay que instalar:

- ▾ En Ubuntu y openSUSE, el paquete nfs-kernel-server
- ▾ En Fedora, Red Hat y CentOS, el paquete nfs-utils

Los servicios se llamarán:

- ▾ En Ubuntu, Fedora, CentOS y Red Hat: rpcbind y nfs-server

- En openSUSE: rpcbind y nfsserver

Después podrá usar los comandos clásicos systemctl.

Para saber si el servicio está disponible en un anfitrión remoto:

```
# rpcinfo -p host
root@ubuntu:/etc# rpcinfo -p
  program vers proto  port  service
    100000   4   tcp    111  portmapper
...
    100005   1   tcp   33003  mountd
...
    100003   3   tcp    2049  nfs
...
    100021   1   tcp   40811  nlockmgr
...
```

b. Caso de NFS4

NFSv4 o NFS4, o por último 4.1, no utiliza ya las RPC y portmapper. La versión 4 está basada en TCP. Los ejemplos siguientes funcionan bajo NFS3 o 4, indistintamente. La versión 4 introduce la noción de NFS raíz vinculada a los fsids (identificadores de recurso compartido). Un export NFS de **fsid=0** indica de hecho una raíz de sistema de archivos NFS: solo los subdirectorios que contienen pueden montarse desde un cliente.

Si comparte /NFS con la opción fsid=0, y esta carpeta contiene los subdirectorios share1 y share2, un cliente solo verá los recursos share1 y share2.

NFS4 permite usar la autenticación Kerberos, sin embargo, este aspecto no se abordará en este capítulo.

c. Compartir por el lado servidor

La lista de directorios que hay que exportar se encuentra en **/etc/exports**. Contiene un sistema compartido por línea.

```
# Dir exportes Autorizaciones de acceso
/      puesto1(rw) puesto2(rw,no_root_squash)
/projects    *.midominio.org(rw)
/home/joe    puesto*.midominio.org(rw)
/pub        192.168.1.0/255.255.255.0(ro)
/srv/nfs4    *(rw,sync,fsid=0,no_subtree_check)
```

Cada línea se compone de dos partes. La primera es la ruta del directorio exportado. La segunda contiene las autorizaciones de acceso.

La autorización de acceso se compone de pares huésped/permisos según el formato siguiente:

host(permisos)

Si no se define el huésped, se aplicarán los permisos a toda la red (de alcance mundial). Si no se definen los permisos, la exportación será de sólo lectura. Sobre todo, no debe ponerse espacio entre el huésped y los permisos. El host puede ser:

- ~ un nombre de host único;
- ~ un dominio;
- ~ una red o subred;
- ~ una combinación del conjunto con caracteres de sustitución (*, ?).

Los permisos pueden ser:

- ~ **ro**: sólo lectura.
- ~ **rw**: lectura escritura.
- ~ **no_root_squash**: el root remoto equivale al root local: esto es un error grave de seguridad.
- ~ **root_squash**: si root se conecta al directorio compartido, su uid será reemplazado por el de un usuario anónimo. Así no hay riesgo de que el usuario root de un puesto local pueda ser root en un directorio remoto.
- ~ **all_squash**: extiende la regla precedente a todos los usuarios.

- **anonuid / anongid**: uid y gid para el usuario anónimo.

Para una gestión correcta de los derechos y permisos, **los usuarios de mismo nombre (login) deben tener los mismos UID y GID en el servidor y el cliente**. NFS se basa, en efecto, en estos valores para garantizar la seguridad de los datos del directorio compartido. El nombre de login solo no es suficiente. En el ejemplo siguiente, el usuario **joe** está autorizado a acceder al directorio compartido **/home/joe** (se supone que es su directorio personal) en todos los puestos del dominio. Se debe declarar al usuario joe de la misma manera (mismo UID) en el servidor y en todos los puestos. Por eso se utiliza a menudo NIS con NFS.

El comando **exportfs** permite controlar los directorios compartidos.

- **exportfs -r** : refresca la lista de directorios compartidos después de la modificación de /etc/exports;
- **exportfs -v** : lista de los directorios compartidos;
- **exportfs -a** : exporta (o vuelve a cargar) todos los directorios compartidos de /etc/exports o un directorio compartido dado;
- **exportfs -u** : detiene la compartición dada. **-a** para todos.

El comando **showmount** muestra los directorios compartidos de un anfitrión dado.

```
# exportfs -av
...
exporting */srv/nfs4
# showmount -e
Export list for ubuntu:
/srv/nfs4 *
```

d. Montaje por el lado cliente

El soporte NFS está incluido en forma de módulo del núcleo. Se carga automáticamente cuando se utiliza un acceso NFS.

En **/etc/fstab** observe las modificaciones:

```
server1:/pub /mnt/pub nfs defaults 0 0
```

La ruta del directorio compartido con la forma **servidor:ruta** ha reemplazado al periférico. El sistema de archivos es **NFS**. Es lo mismo con el comando **mount**:

```
mount -t nfs servidor1:/pub /mnt/pub
```



Si se definen los montajes NFS en /etc/fstab, mount **-a** no los va a montar. Un servicio suele estar presente en cada distribución para montarlos y desmontarlos en las paradas/reinicios. En Red Hat, el servicio /etc/rc.d/init.d/netfs los montará de manera automática durante el inicio.

NFS dispone de opciones de montaje específicas:

- ✓ **nolock**: opción de compatibilidad con antiguos servidores NFS;
- ✓ **intr**: interrumpe una petición NFS si el servidor no contesta;
- ✓ **hard**: bloquea los procesos que intentan acceder a un directorio compartido inaccesible;
- ✓ **soft**: un proceso devolverá un error en caso de acceso infructuoso;
- ✓ **rsize=8192, wsize=8192**: tamaño de los bloques de lectura/escritura en el servidor. Una escritura de 8 kB es más rápida que 8 escrituras de 1 kB.

¡Atención al tipo de montaje NFS! Durante mucho tiempo, los montajes NFS utilizaban el protocolo UDP. Sin embargo, esta opción de montaje se ha modificado y ahora se utiliza TCP. En UDP, cuando una transmisión fallaba, saltaba un timeout. Más allá de un valor determinado, significa que se tendrá que retransmitir un paquete RPC completo con cada error de transmisión, que es bastante pesado.

En TCP, el control de transmisión se deja en manos del propio protocolo para los paquetes más pequeños. Ahora bien, esto puede provocar efectos secundarios importantes cuando

el servidor está trabajando con demasiada carga o cuando la entrada/salida es demasiado grande en un montaje, ya que el valor de timeout por defecto no ha cambiado. Entonces, puede ser que suceda lo siguiente: un archivo se escribe en el montaje NFS; por una razón cualquiera, el paquete TCP no llega en el tiempo previsto (por unas décimas de segundo) y NFS solicita la retransmisión de los datos sin esperar el timeout TCP, lo que provoca un conflicto, una corrupción de datos y la aparición de mensajes de error.

```
nfs: server not responding, timed out
```

o incluso:

```
touch: cannot touch `/home/seb/test': Input/output error
```

Hay que modificar la opción de montaje dejando TCP pero modificando el timeout NFS, expresado en décimas de segundo, para hacerlo corresponder al de TCP. Utilice un valor de 600, es decir, 60 segundos. En todas las últimas versiones de NFS para Linux, se ha convertido en el valor predeterminado, pero para estar seguro de que no se encontrará con este problema, utilice el comando como en el ejemplo siguiente:

```
mount -o timeo=600,retrans=2, tcp -t nfs servidor1:/pub /mnt/pub
```