

Configuración de un servidor NFS

NFS (*Network File System*) es un protocolo cliente-servidor de compartición de archivos y directorios, creado por Sun Microsystems en 1984, y que se convirtió en el estándar de facto de los entornos Unix. Fue definido por una serie de RFC, adaptado a Linux y a Windows.



La versión mayor, la más reciente del protocolo, es la versión 4 (RFC 7530), pero la versión 3 (RFC 1813) sigue siendo muy usada y constituye el objeto principal de este tema de la certificación LPIC-2.

NFS que es un protocolo de nivel aplicación. Se basa en un protocolo de nivel de sesión, los RPC (*Remote Procedure Call*, llamada de procedimiento remoto), creado también inicialmente por Sun Microsystems.

1. NFS versión 4

La versión 4 del protocolo NFS es el resultado de una reestructuración completa. Al contrario que para las versiones anteriores que funcionaban en modo sin estado (*stateless*) y utilizaban UDP o TCP, esta versión implementa un servidor con estado, y solamente usa TCP.

El protocolo ya no se basa en los RPC, lo que le permite evitar las restricciones relativas al daemon `rpcbind/portmapper` (uso posible en Internet y a través de firewalls).

Es mucho más seguro (identificación a través de Kerberos, comunicaciones cifradas, etc.).

Esta versión no se estudiará profundamente en el marco de la certificación LPIC-2, que se centra en la versión 3, la más usada hoy día.

2. El servicio rpcbind/portmapper

Este servicio tiene como función gestionar las solicitudes RPC, enviadas por los clientes hacia los diferentes servidores que se basan en este protocolo de nivel de sesión para comunicar con sus clientes.

a. Principio de funcionamiento

El daemon (`rpcbind` o `portmapper`, dependiendo de las versiones) se inicia y se pone en escucha en el puerto bien conocido de los RPC, el puerto 111 en UDP y en TCP.

Cada servidor que quiera utilizar los RPC se registra en el daemon `rpcbind/portmapper` (que tiene que haber sido iniciado anteriormente), indicándole su identificador de protocolo aplicativo RPC y su versión, así como el número de puerto dinámico en el que se pone en espera.

Cuando el daemon recibe una solicitud de un cliente RPC en el puerto 111 (UDP o TCP), especificando el protocolo de aplicación y la versión del servidor solicitada, el daemon le devuelve el número de puerto del servidor asociado, para que puedan comunicar directamente juntos.

b. Protección del servicio

Como el daemon `rpcbind/portmapper` es un punto de paso obligado para todos los clientes que utilizan los RPC, podemos asignarles reglas de control de acceso gracias a la biblioteca `tcp_wrappers` y a sus archivos `/etc/hosts.allow` y/o `/etc/hosts.deny`.

El servicio que se tiene que configurar en esos archivos tiene como identificador `portmapper` (como está declarado en el archivo `/etc/services`).



La sintaxis de declaración de las reglas están detalladas en la sección Depuración de la red en el capítulo Configuración de red.

3. Paro/inicio del servidor NFS

El inicio del servidor NFS es generalmente automático, gestionado por un script `init system V` o por `systemd`.

El servidor NFS está constituido por distintos daemons y depende de la activación anterior del daemon que administra la capa RPC, `rpcbind` (o `portmapper`).

Por lo tanto, se necesitan al menos tres daemons activos para que el servidor NFS esté operativo:

- `rpcbind` (o `portmapper`): este daemon gestiona las solicitudes RPC.
- `nfsd`: este daemon gestiona las solicitudes de los clientes NFS.
- `mountd`: este daemon gestiona las solicitudes de montaje de los clientes NFS.

El servidor NFS está generalmente instalado por defecto en la mayoría de las distribuciones, y lo proporciona el paquete de software `nfs-utils` (Red Hat) o `nfs-kernel-server` (Debian).

Puede haber otros daemons necesarios para el servidor NFS, en función de la versión instalada.

Ejemplo

Servidor NFS en una distribución CentOS 8.

Comprobamos que el paquete esté instalado:

```
yum list nfs*
```

```
Última comprobación de caducidad de metadatos hecha hace 2:33:47,  
el jue 04 nov 2021 12:14:35 EDT.
```

```
Paquetes disponibles
```

<code>nfs-utils.x86_64</code>	<code>1:2.3.3-46.el8</code>	<code>baseos</code>
<code>nfs4-acl-tools.x86_64</code>	<code>0.3.5-3.el8</code>	<code>baseos</code>

Comprobamos el estado del servicio RPC, `rpcbind`:

systemctl status rpcbind

```
rpcbind.service - RPC Bind
Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled;
vendor preset: enabled)
Active: active (running) since Mon 2020-05-25 15:20:34 CEST; 2 days ago
Docs: man:rpcbind(8)
Main PID: 979 (rpcbind)
Tasks: 1 (limit: 23516)
Memory: 1.7M
CGroup: /system.slice/rpcbind.service
        979 /usr/bin/rpcbind -w -f
```

mayo 25 15:20:33 centos8 systemd[1]: Starting RPC Bind...

mayo 25 15:20:34 centos8 systemd[1]: Started RPC Bind.

El servicio `rpcbind` está activo.

Comprobamos el servidor NFS y lo activamos si fuera necesario:

systemctl status nfs-server

```
? nfs-server.service - NFS server and services
Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled;
vendor preset: disabled)
Active: inactive (dead)
```

systemctl enable nfs-server

```
Created symlink /etc/systemd/system/multi-user.target.wants/
nfs-server.service ? /usr/lib/systemd/system/nfs-server.service.
```

systemctl start nfs-server

Observamos los diferentes procesos que corresponden a los dos servicios:

ps -ef | grep rpc

```
rpc    979   1 0 may25 ?    00:00:00 /usr/bin/rpcbind -w -f
root   983   2 0 may25 ?    00:00:00 [rpciod]
rpcuser 31896  1 0 16:24 ?    00:00:00 /usr/sbin/rpc.statd
root   31898  1 0 16:24 ?    00:00:00 /usr/sbin/rpc.idmapd
root   31910  1 0 16:24 ?    00:00:00 /usr/sbin/rpc.mountd
```

ps -ef | grep nfs

```
root  31917  2 0 16:24 ?    00:00:00 [nfsd]
root  31918  2 0 16:24 ?    00:00:00 [nfsd]
root  31919  2 0 16:24 ?    00:00:00 [nfsd]
root  31920  2 0 16:24 ?    00:00:00 [nfsd]
root  31921  2 0 16:24 ?    00:00:00 [nfsd]
root  31922  2 0 16:24 ?    00:00:00 [nfsd]
root  31923  2 0 16:24 ?    00:00:00 [nfsd]
root  31924  2 0 16:24 ?    00:00:00 [nfsd]
```

El servidor NFS está activo.

Servidor NFS en una distribución Debian 10.

Instalamos el paquete:

apt-get install nfs-kernel-server

Leyendo lista de paquetes... Hecho

Creando árbol de dependencias

Leyendo la información de estado... Hecho

Se instalarán los siguientes paquetes adicionales:

keyutils libnfsidmap2 nfs-common rpcbind

Paquetes sugeridos:

[open-iscsi](#) watchdog

Se instalarán los siguientes paquetes NUEVOS:

keyutils libnfsidmap2 nfs-common nfs-kernel-[server](#) rpcbind

0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 1 [no](#) actualizados.

Se necesita descargar 486 kB de archivos.

Se utilizarán 1.540 kB de espacio de disco adicional después de esta operación.

¿Desea continuar? [S/n] **S**

[...]

Creating config [file](#) /etc/[default](#)/nfs-kernel-[server with new version](#)

Procesando disparadores para systemd (241-7~deb10u8) ...

Procesando disparadores para man-db (2.8.5-2) ...

Procesando disparadores para libc-[bin](#) (2.28-10) ...

Buscamos los scripts de inicio `init system V`:

```
ls /etc/init.d/*nfs* /etc/init.d/*rpc*
/etc/init.d/nfs-common /etc/init.d/nfs-kernel-server
/etc/init.d/rpcbind
```

Comprobamos el estado del RPC, `rpcbind`:

```
systemctl status rpcbind
systemctl status rpcbind
rpcbind.service - RPC bind portmap service
Loaded: loaded (/lib/systemd/system/rpcbind.service; enabled;
vendor preset: enabled)
Active: active (running) since Wed 2020-05-27 16:33:51 CEST; 3min 19s ago
Docs: man:rpcbind(8)
Main PID: 12442 (rpcbind)
Tasks: 1 (limit: 4558)
Memory: 640.0K
CGroup: /system.slice/rpcbind.service
        12442 /sbin/rpcbind -f -w

mayo 27 16:33:51 debian10 systemd[1]: Starting RPC bind portmap service...
mayo 27 16:33:51 debian10 systemd[1]: Started RPC bind portmap service.
```

El servicio `rpcbind` está activo.

Comprobamos el servidor NFS y lo activamos si fuera necesario:

```
systemctl status nfs-server
nfs-server.service - NFS server and services
Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled;
vendor preset: enabled)
Active: active (exited) since Wed 2020-05-27 16:34:04 CEST; 4min 12s ago
Main PID: 13028 (code=exited, status=0/SUCCESS)
Tasks: 0 (limit: 4558)
Memory: 0B
```

CGroup: /system.slice/nfs-server.service

mayo 27 16:34:04 debian10 systemd[1]: Starting NFS server and services...

mayo 27 16:34:04 debian10 systemd[1]: Started NFS server and services.

Observamos los diferentes procesos que corresponden a los dos servicios:

ps -ef | grep rpc

```
_rpc  12442  1 0 16:33 ?    00:00:00 /sbin/rpcbind -f -w
root  12802  2 0 16:34 ?    00:00:00 [rpciod]
root  13025  1 0 16:34 ?    00:00:00 /usr/sbin/rpc.idmapd
root  13026  1 0 16:34 ?    00:00:00 /usr/sbin/rpc.mountd -manage-gids
```

ps -ef | grep nfs

```
root  13033  2 0 16:34 ?    00:00:00 [nfsd]
root  13034  2 0 16:34 ?    00:00:00 [nfsd]
root  13035  2 0 16:34 ?    00:00:00 [nfsd]
root  13036  2 0 16:34 ?    00:00:00 [nfsd]
root  13037  2 0 16:34 ?    00:00:00 [nfsd]
root  13038  2 0 16:34 ?    00:00:00 [nfsd]
root  13039  2 0 16:34 ?    00:00:00 [nfsd]
root  13040  2 0 16:34 ?    00:00:00 [nfsd]
```

El servidor NFS está activo.

4. Configuración de directorios compartidos

NFS permite a un servidor definir directorios compartidos, accesibles a través de la red a clientes NFS.

El servidor define un primer nivel de control de acceso, autorizando o no a los hosts a acceder a los recursos compartidos y compartiendo un directorio en lectura y escritura o en solo lectura.

Un segundo nivel de control de acceso se ejerce gracias a los permisos de acceso clásicos en los archivos y directorios, según la cuenta de usuario y el grupo asociado al

archivo o al directorio.

Por defecto, el servidor NFS considera que los identificadores de los usuarios (UID, *User Identifier*) y los grupos (GID, *Group Identifier*) del sistema cliente son iguales a los del sistema local (exceptuando el superusuario, UID 0).

a. El archivo de declaración de las comparticiones: `/etc/exports`

El archivo `/etc/exports` es un archivo de texto en el que están definidos los directorios que se van a compartir, con sus parámetros de compartición. Estas comparticiones se activan automáticamente al inicio del servidor NFS.

Una compartición está definida en una línea, según el formato siguiente:

```
CaminoDir [ IdentCliente1[(Opciones)] ... IdentClienteN[(Opciones)]
```

Donde:

CaminoDir	Camino de acceso al directorio que se va a compartir.
IdentCliente	Nombre de host, del dominio, dirección IP o red del o de los clientes NFS. Por defecto: todos los clientes.
Opciones	Opciones de compartición para el o los clientes especificados.



Cuidado, no hay que poner un espacio entre el o los identificadores de los clientes y las opciones de compartición entre paréntesis.

Los clientes pueden estar definidos por su nombre de host, su dirección IP o por su nombre de dominio. Se pueden usar los identificadores de red/subred o caracteres joker

en los nombres DNS para especificar un conjunto de clientes.

Si ningún identificador de cliente ha sido especificado, la compartición estará accesible a todos los clientes NFS.

Si uno o distintos identificadores son idénticos, los hosts que no correspondan a estos identificadores no pueden acceder a la compartición.

Las opciones de compartición son múltiples. Por defecto, sin opción, un directorio se comparte en solo lectura. Las opciones pueden estar combinadas, separándolas con una coma.

Las principales opciones son las siguientes:

<code>ro</code>	Acceso en solo lectura (por defecto).
<code>rw</code>	Acceso en lectura y escritura.
<code>sync</code>	Acceso en escritura síncrona. Los datos se escriben inmediatamente.
<code>async</code>	Acceso en escritura asíncrona (por defecto).
<code>root_squash</code>	La cuenta de superusuario (UID 0) en el cliente se reemplaza en el servidor por una cuenta de usuario no super usuario (por defecto).
<code>no_root_squash</code>	La cuenta de superusuario (UID 0) en el cliente será también la cuenta de superusuario en el servidor.

El archivo se lee cada vez que se inicia el servidor NFS.

Si modificamos el archivo, podemos reiniciar el servidor NFS o pedirle que vuelva a cargar el archivo de configuración (usando `systemctl reload`) para que tome en cuenta las modificaciones. También se puede ejecutar el comando `exportfs` con la opción `-a`.

Ejemplo

Definición de una compartición en lectura para el cliente `192.168.0.60` :

```

vi /etc/exports
# Compartición accesible en lectura para 192.168.0.60.
/var/nfs/public 192.168.0.60(ro,no_subtree_check)

```

Creamos el directorio local.

```
mkdir -p /var/nfs/public
```

Posicionamos los permisos de acceso.

```

chgrp users /var/nfs/public
chmod 775 /var/nfs/public

```

Creamos un archivo en el directorio compartido:

```

echo "Bienvenido al directorio compartido" >/var/nfs/public/info.txt
ls -l /var/nfs/public
total 4
-rw-r--r-- 1 root root 39 mayo 28 14:03 info.txt

```

b. Compartición dinámica de un directorio

El comando `exportfs` permite compartir dinámicamente un directorio. El directorio estará compartido hasta que solicitemos explícitamente terminar la compartición, o hasta que el sistema se pare.

Sintaxis

```
exportfs -o Opciones IdentCliente:/CaminoDir
```

Donde:

CaminoDir	Camino de acceso del directorio que se compartirá.
IdentClient	Nombre de host, de dominio, dirección IP o red del o de los clientes NFS. Por defecto: todos los clientes.
Opciones	Opciones de compartición para el o los clientes especificados.

Ejemplo

Definición de una compartición en lectura y escritura para el cliente `centos8`:

Creamos el directorio local:

```
mkdir -p /var/nfs/input
```

Posicionamos los permisos de acceso:

```
chgrp users /var/nfs/input  
chmod 775 /var/nfs/input
```

Compartimos el directorio:

```
exportfs -o rw centos8:/var/nfs/input
```

5. Gestión de las cuentas de usuarios clientes

Cuando un cliente NFS se conecta a un servidor NFS, no se efectúa ninguna solicitud de identificación del usuario. El acceso a la compartición es autorizado o denegado en

función de la configuración de la compartición con respecto al host que hace la solicitud (dirección IP o nombre de host).

a. Derechos de acceso del usuario cliente

NFS considera que los identificadores de las cuentas de usuarios son coherentes entre el servidor y los clientes, es decir, que los identificadores de las cuentas de usuarios (UID) y de los grupos (GID) son los mismos en el sistema del servidor y en el sistema del cliente.

Cuando un cliente se conecta a una compartición NFS, muestra su identificador (UID) al servidor, el cual utiliza el identificador correspondiente en su sistema local (y los grupos a los que pertenece) para determinar sus derechos de acceso sobre los archivos y directorios del servidor.



Esta obligación de coherencia de los identificadores es una limitación importante de NFS. Sin embargo, es posible gestionar los identificadores de los usuarios y grupos de manera global en el seno de una organización con un servicio de anuario (NIS o LDAP), lo que permite resolver esta dificultad.

Si una compartición NFS está configurada con la opción de solo lectura, esta se impondrá a todos los usuarios de la compartición, independientemente de los derechos que tengan sobre el contenido de la compartición.

b. El caso particular del superusuario

Para evitar que el superusuario (UID 0) de un cliente NFS sea considerado como un superusuario del servidor, la opción de compartición `root_squash` está activada por defecto: cuando el servidor recibe una solicitud de conexión de una cuenta con el UID 0, le atribuye automáticamente el UID de una cuenta de usuario anónimo (en general el UID 65534). Esta cuenta de usuario (que puede ser, según las distribuciones y las versiones, `nfsnobody`, `nobody`...) estará sometida a un control de acceso normal (no superusuario) para los archivos y directorios de la compartición.

Si deseamos desactivar este mecanismo en una compartición, se puede usar la opción `no_root_squash`.

6. Monitorización del servidor NFS

Existen diferentes comandos que permiten seguir la actividad del servidor NFS y de la capa RPC subyacente.

a. Gestión de comparticiones: `exportfs`

El comando `exportfs` permite gestionar dinámicamente los directorios compartidos.

Sintaxis

```
exportfs [ Opciones ]
```

Opciones principales:

<code>-v</code>	Visualización detallada.
<code>-s</code>	Muestra las comparticiones definidas en <code>/etc/exports</code> .
<code>-a</code>	Activa todas las competiciones declaradas en <code>/etc/exports</code> .
<code>-u</code> <code>Cli:Dir</code>	Desactiva la o las comparticiones.
<code>...</code>	
<code>-ua</code>	Desactiva todas las competiciones declaradas en <code>/etc/exports</code> .

Sin argumento (o solo con la opción `-v`), muestra la lista de los directorios compartidos y

su uso por los clientes NFS.

[Ejemplo](#)

Activación de las comparticiones añadidas en el archivo `/etc/exports`:

exportfs -a

Lista de todas las comparticiones activas del servidor NFS local (con detalle). La opción `-v` muestra las opciones de compartición, explícitas o por defecto:

```
exportfs -v
/var/nfs/public
    192.168.0.60(ro,wdelay,root_squash,no_subtree_check,
sec=sys,ro,secure,root_squash,no_all_squash)
/var/nfs/input 192.168.0.60(rw,wdelay,root_squash,no_subtree_check,
sec=sys,rw,secure,root_squash,no_all_squash)
```

b. showmount

El comando `showmount` muestra la información sobre los clientes que han montado directorios compartidos de un servidor NFS.

[Sintaxis](#)

`showmount [Opciones] [IdServidor]`

Por defecto, el comando muestra las conexiones a las comparticiones del servidor local. La opción `-v` devuelve una visualización detallada.

[Ejemplo](#)

Visualización de las comparticiones locales montadas por los clientes NFS. La opción `-a` muestra el cliente y la compartición para cada conexión en tiempo real:


```

0      0%  0      0%  0      0%  0      0%  216  21%
getfh      link      lock      lockt      locku
43      4%  0      0%  0      0%  0      0%  0      0%
lookup      lookup_root  nverify      open      openattr
35      3%  0      0%  0      0%  18      1%  0      0%
open_conf      open_dgrd      putfh      putpubfh      putrootfh
0      0%  0      0%  231  23%  0      0%  19      1%
read      readdir      readlink      remove      rename
0      0%  6      0%  0      0%  0      0%  0      0%
renew      restorefh      savefh      secinfo      setattr
0      0%  0      0%  0      0%  0      0%  0      0%
setctid      setctidconf  verify      write      rellockowner
0      0%  0      0%  0      0%  2      0%  0      0%
bc_ctl      bind_conn      exchange_id      create_ses      destroy_ses
0      0%  0      0%  17      1%  10      1%  8      0%
free_stateid  getdirdeleg      getdevinfo      getdevlist      layoutcommit
0      0%  0      0%  0      0%  0      0%  0      0%
layoutget      layoutreturn      secinfoonam      sequence      set_ssv
0      0%  0      0%  9      0%  318  32%  0      0%
test_stateid  want_deleg      destroy_clid      reclaim_comp      allocate
0      0%  0      0%  7      0%  9      0%  0      0%
copy      copy_notify      deallocate      ioadvise      layouterror
0      0%  0      0%  0      0%  0      0%  0      0%
layoutstats  offloadcancel  offloadstatus  readplus      seek
0      0%  0      0%  0      0%  0      0%  0      0%
write_same
0      0%

```

El servidor funciona en modo NFS versión 3 y versión 4.

d. rpcinfo

El comando `rpcinfo` permite efectuar una solicitud RPC en un servidor y mostrar los servicios de red registrados en el daemon `rpcbind` (o `portmapper`).

Sintaxis

`rpcinfo [Opciones] [IdServidor]`

Por defecto, el comando muestra la información RPC para el servidor local.

La opción `-s` nos da un resultado más condensado.

Ejemplo

```
rpcinfo -s
program version(s) netid(s)      service  owner
100000  2,3,4  local,udp,tcp,udp6,tcp6      portmapper superuser
100024  1      tcp6,udp6,tcp,udp          status    29
100005  3,2,1  tcp6,udp6,tcp,udp          mountd    superuser
100003  4,3    tcp6,tcp                   nfs       superuser
100227  3      tcp6,tcp                   nfs_acl   superuser
100021  4,3,1  tcp6,udp6,tcp,udp          nlockmgr  superuser
```

Los servicios NFS que usan los RPC están activos.

7. Implementación del cliente NFS

Un cliente NFS Linux puede acceder a un directorio compartido por un servidor NFS, montándolo en la arborescencia global de su sistema de archivos, como lo haría con un sistema de archivos local.

a. Montaje de un directorio compartido NFS

Para montar el directorio compartido remoto, se usa el comando estándar de montaje, `mount`.

Sintaxis

```
mount -t nfs [ -o Opciones ] IdServidor:/CaminoDir PuntoMontaje
```

Donde:

<code>-t nfs</code>	Tipo de sistema de archivos.
<code>-o Opciones</code>	Opciones de montaje (<code>intr</code> , <code>ro</code> , <code>noexec</code> , <code>nosuid</code> ...).
<code>IdServidor:/CaminoDir</code>	Dirección o nombre del servidor y de la compartición que se va a montar.
<code>PuntoMontaje</code>	Directorio de montaje.

Sin la opción `-t nfs`, el comando determina automáticamente el tipo de sistema de archivos que se tendrá que montar.

Las opciones de montaje son comunes a los otros tipos de sistemas de archivos, así como ciertas específicas del tipo NFS.

La opción `intr` (*interrupt*) interrumpe las solicitudes de acceso en el cliente si el servidor no responde transcurrido de un cierto periodo de tiempo, evitando que las aplicaciones queden en una espera indefinida.



La opción `vers=3` fuerza el uso de la versión 3 de NFS, en lugar de la versión por defecto (versión cuatro en las distribuciones recientes). Esto puede ser útil para las pruebas de preparación a la certificación LPIC-2.

Una vez montado, el directorio compartido remoto se usa como un directorio local.

Podemos declarar el montaje en el archivo `/etc/fstab`.

Podemos comprobar el montaje, con el comando `mount` sin argumento, o consultando el contenido del archivo `/proc/mounts` del sistema de archivos virtual `procfs`.

Ejemplo

Uso de las particiones creadas en los ejemplos precedentes en la máquina `debian10`.

Cuidado, la cuenta de usuario de prueba debe tener el mismo UID en el cliente y en el servidor, y formar parte del grupo `users` en el servidor.

Creamos los dos directorios de montaje en el cliente:

```
mkdir -p /var/debian10/info
mkdir /var/debian10/envio
```

Montamos las particiones del servidor `debian10` (en NFS versión 3):

```
mount -t nfs -o vers=3 debian10:/var/nfs/public /var/debian10/info
mount -t nfs -o vers=3 debian10:/var/nfs/input /var/debian10/envio
```

Comprobamos el montaje usando el comando `mount`, y después a través de `/proc/mounts`:

```
mount | grep '/var/nfs'
debian10:/var/nfs/public on /var/debian10/info type nfs
(rw,relatime,vers=3,rsize=524288,wsiz=524288,namlen=255,hard,proto=
tcp,timeo=600,retrans=2,sec=sys,mountaddr=192.168.0.70,mountvers=3,
mountport=39062,mountproto=udp,local_lock=none,addr=192.168.0.70)
debian10:/var/nfs/input on /var/debian10/envio type nfs
(rw,relatime,vers=3,rsize=524288,wsiz=524288,namlen=255,hard,proto=tcp,
timeo=600,retrans=2,sec=sys,mountaddr=192.168.0.70,mountvers=3,
mountport=39062,mountproto=udp,local_lock=none,addr=192.168.0.70)
grep '/var/nfs' /proc/mounts
debian10:/var/nfs/public /var/debian10/info nfs rw,relatime,vers=3,
rsize=524288,wsiz=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,
```

```
sec=sys,mountaddr=192.168.0.70,mountvers=3,mountport=39062,mountproto=udp,
local_lock=none,addr=192.168.0.70 0 0
debian10:/var/nfs/input /var/debian10/envio nfs rw,relatime,
vers=3,rsize=524288,wsiz=524288,namlen=255,hard,proto=tcp,timeo=600,
retrans=2,sec=sys,mountaddr=192.168.0.70,mountvers=3,mountport=39062,
mountproto=udp,local_lock=none,addr=192.168.0.70 0 0
```

Abrimos una sesión con la cuenta de usuario `pba` (ID `1000` en el cliente y en el servidor):

```
su - pba
id
uid=1000(pba) gid=1000(pba) grupos=1000(pba),100(users)
```

Usamos la compartición en solo lectura, montada en `/var/debian10/infos`:

```
ls -l /var/debian10/info
total 4
-rw-r--r--. 1 root root 39 28 mayo 14:03 infos.txt
cat /var/debian10/infos/infos.txt
Bienvenido al directorio compartido
echo "Prueba desde centos8" > /var/debian10/infos/pba-centos8.txt
-bash: /var/debian10/infos/pba.txt: Sistema de archivos accesible en solo lectura
```

No podemos escribir en el directorio compartido.

Usamos la compartición en lectura y escritura, montada en `/var/debian10/envio`:

```
echo " Prueba desde centos8" > /var/debian10/envio/pba-centos8.txt
```

El usuario `pba` puede escribir en el directorio compartido.

Comprobamos en el servidor, en el directorio compartido `/var/nfs/input`:

```
ls -l /var/nfs/input/
total 4
-rw-rw-r-- 1 pba pba 21 mayo  28 16:02 pba-centos8.txt
cat /var/nfs/input/pba-centos8.txt
Prueba desde centos8
```

Intentamos usar una compartición del servidor `debian10`, desde una máquina no autorizada:

```
mount -t nfs -o vers=3 debian10:/var/nfs/public /var/debian10/infos
mount.nfs: access denied by server while mounting debian10:/var/nfs/public
```

Acceso denegado.

b. Visualización de las comparticiones de los servidores

La opción `-e` del comando `showmount` permite mostrar la información de un servidor NFS remoto.

Sintaxis

```
showmount -e IdServidor
```

Donde `IdServidor` representa la dirección IP o el nombre del servidor del que queremos obtener la lista de comparticiones.

Ejemplo

Visualización de las comparticiones del servidor `debian10` desde una máquina remota:

```
showmount -e debian10
Export list for debian10:
/var/nfs/input 192.168.0.60
/var/nfs/public 192.168.0.60
```