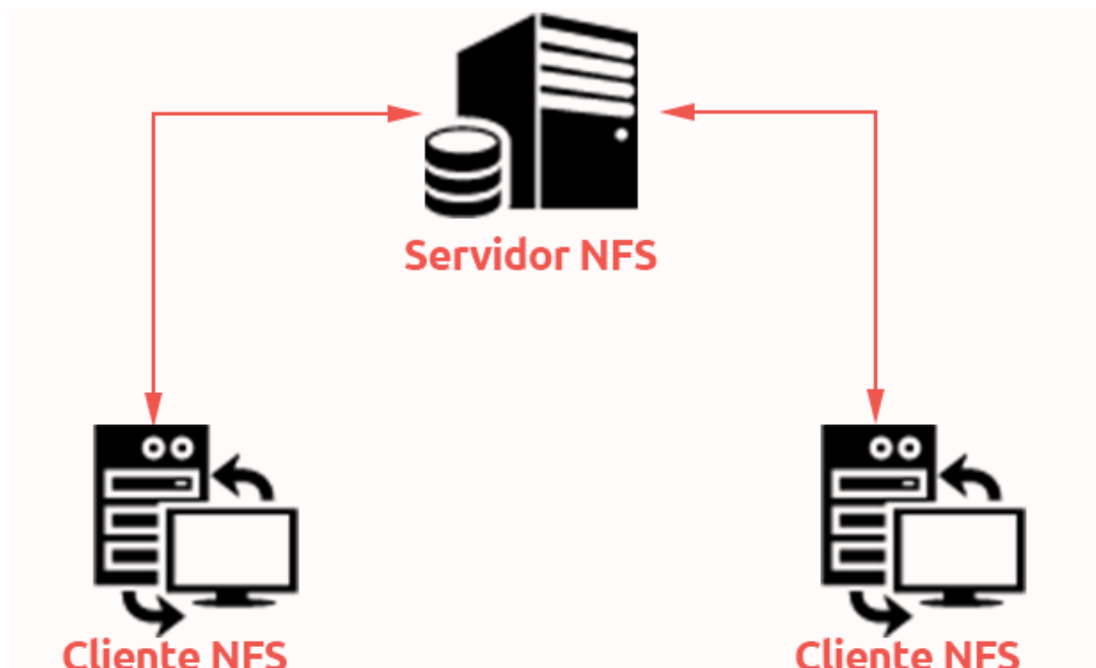


# PRÁCTICA 6: SERVIDOR DE DISCO NFS

Iñaki Melguizo Marcos



## 1. INTRODUCCIÓN

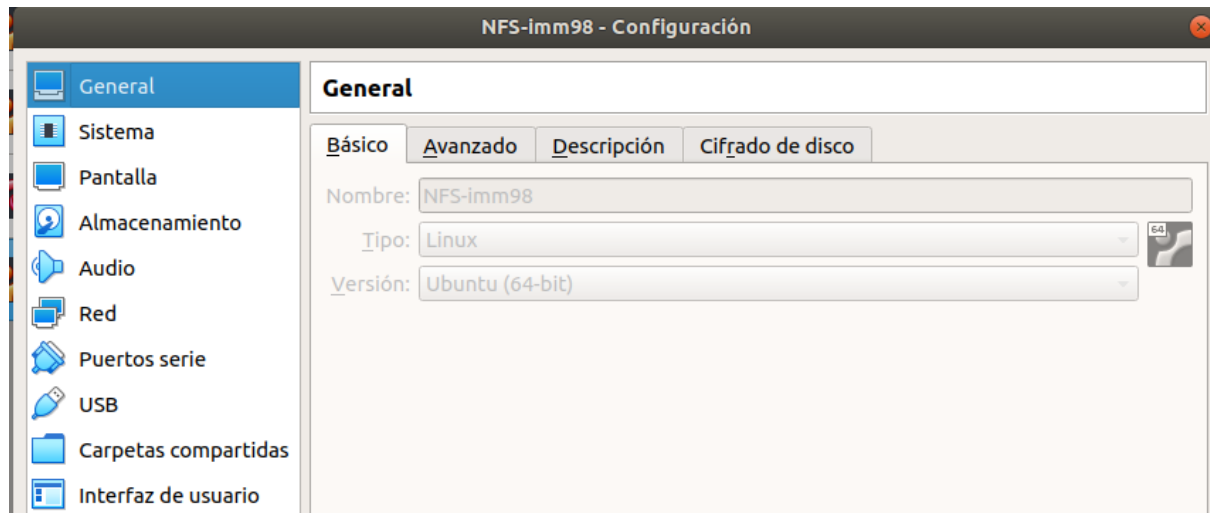
En esta última práctica vamos a terminar la configuración de nuestra granja web, en la que hemos estado trabajando durante el transcurso de la asignatura. En esta práctica configuraremos un servidor NFS para exportar un servicio en disco a los dos servidores finales que tenemos: m1-imm98 y m2-imm98. Esta máquina que hará de servidor NFS se llamará **nfs-imm98**. Las IPs de las 3 máquinas que utilizaremos en esta práctica serán:

m1-imm98	192.168.56.105
m2-imm98	192.168.56.104
nfs-imm98	192.168.56.109

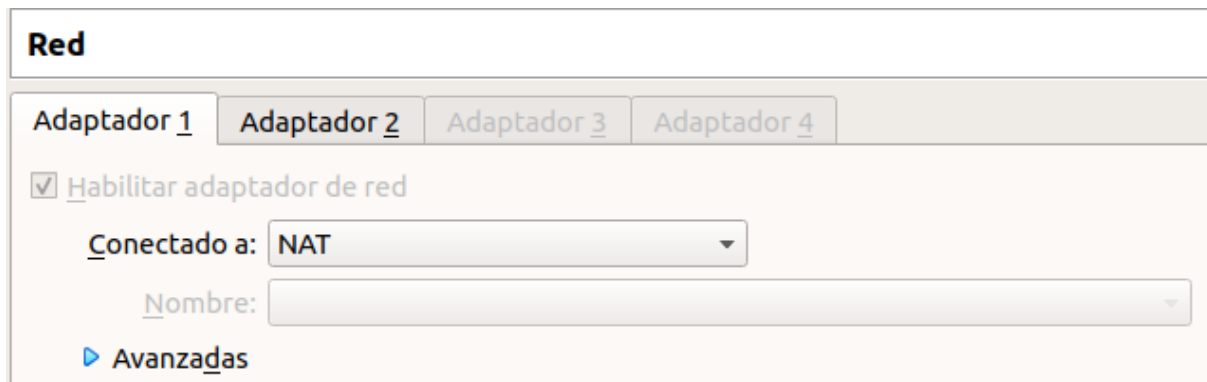
Posteriormente de esta configuración del servidor NFS exportaremos una carpeta a los clientes m1-imm98 y m2-imm98, comprobaremos la correcta compartición de esa carpeta y por último configuraremos el cortafuegos del servidor NFS mediante IPTABLES.

## 2. CREACIÓN DE MÁQUINA VIRTUAL Y CONFIGURACIÓN COMO SERVIDOR NFS

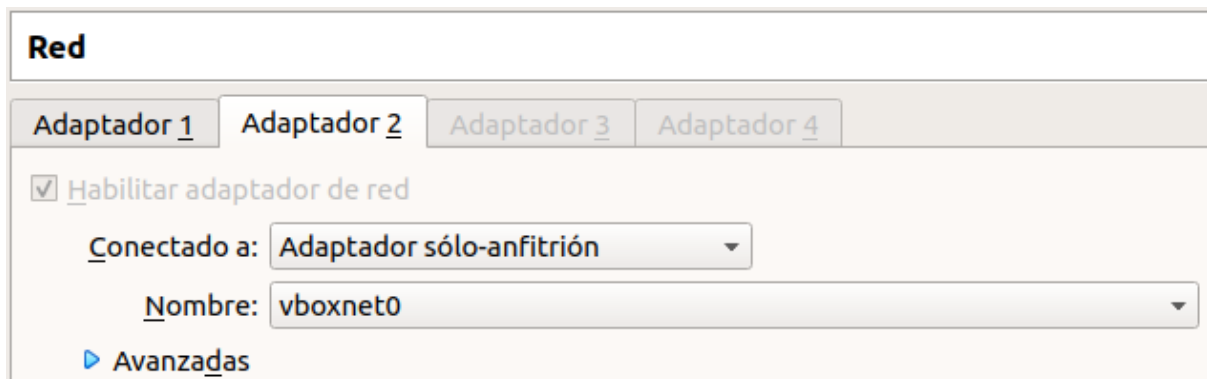
Primeramente, vamos a crear una nueva máquina virtual con Ubuntu Server, que hará de servidor NFS. La llamaremos nfs-imm98



y le añadiremos un adaptador de red NAT



y un adaptador de red Sólo-Anfitrión



Estos mismos adaptadores son los que habíamos añadido a la hora de crear las máquinas m1-imm98 y m2-imm98.

Una vez que hayamos iniciado la máquina nfs-imm98 con US vamos a instalar las herramientas requeridas para que funcione como un servidor NFS. Para ello, ejecutamos la orden

```
sudo apt-get install nfs-kernel-server nfs-common rpcbind
```

```
imm98@nfs-imm98:~$ sudo apt-get install nfs-kernel-server nfs-common rpcbind;  
[sudo] password for imm98:
```

Una vez que hemos instalado las herramientas anteriores vamos a crear una carpeta. Esta carpeta va a ser la que copartiremos con los clientes del servidor NFS, que como ya hemos dicho son, m1-imm98 y m2-imm98. Esta carpeta se llamará **compartido** y estará dentro del directorio **datos** que también tendremos que crear. Para crearlos ejecutaremos:

```
sudo mkdir datos  
cd datos  
sudo mkdir compartido
```

```
imm98@nfs-imm98:~$ sudo mkdir datos  
imm98@nfs-imm98:~$ cd datos  
imm98@nfs-imm98:~/datos$ sudo mkdir compartido  
imm98@nfs-imm98:~/datos$
```

Ahora necesitamos cambiar el propietario de la carpeta datos/compartido.

```
sudo chown nobody:nogroup datos/compartido/
```

```
imm98@nfs-imm98:~$ sudo chown nobody:nogroup datos/compartido/
```

Y por último le cambiamos los permisos a dicha carpeta.

```
sudo chmod -R 777 datos/compartido/
```

```
imm98@nfs-imm98:~$ sudo chmod -R 777 datos/compartido/
```

Ahora vamos a darle permiso de lectura y escritura sobre el directorio que acabamos de crear tanto a la máquina m1-imm98 como a la máquina m2-imm98. Para ello tenemos que modificar el archivo **/etc/exports** y añadir:

```
/home/imm98/datos/compartido 192.168.56.105(rw) 192.168.56.104(rw)
```

```
/home/imm98/datos/compartido 192.168.56.105(rw) 192.168.56.104(rw)
```

Por último, para ver que todo es correcto, reiniciamos el servicio **nfs-kernel-server**. Para ello ejecutamos el siguiente comando:

*sudo service nfs-kernel-server restart*

*sudo service nfs-kernel-server status*

```
imm98@nfs-imm98:~$ sudo service nfs-kernel-server restart
imm98@nfs-imm98:~$ sudo service nfs-kernel-server status
• nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Active: active (exited) since Mon 2021-05-24 09:00:58 UTC; 7s ago
     Process: 3362 ExecStopPost=/usr/sbin/exportfs -f (code=exited, status=0/SUCCESS)
     Process: 3361 ExecStopPost=/usr/sbin/exportfs -au (code=exited, status=0/SUCCESS)
     Process: 3351 ExecStop=/usr/sbin/rpc.nfsd 0 (code=exited, status=0/SUCCESS)
     Process: 3371 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
     Process: 3370 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
    Main PID: 3371 (code=exited, status=0/SUCCESS)

May 24 09:00:58 nfs-imm98 systemd[1]: Starting NFS server and services...
May 24 09:00:58 nfs-imm98 exportfs[3370]: exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no
May 24 09:00:58 nfs-imm98 exportfs[3370]: Assuming default behaviour ('no_subtree_check').
May 24 09:00:58 nfs-imm98 exportfs[3370]: NOTE: this default has changed since nfs-utils version 1
May 24 09:00:58 nfs-imm98 exportfs[3370]: exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no
May 24 09:00:58 nfs-imm98 exportfs[3370]: Assuming default behaviour ('no_subtree_check').
May 24 09:00:58 nfs-imm98 exportfs[3370]: NOTE: this default has changed since nfs-utils version 1
May 24 09:00:58 nfs-imm98 systemd[1]: Started NFS server and services.
```

### 3. CONFIGURACIÓN DE LOS CLIENTES M1 Y M2

Ahora vamos a configurar las máquinas m1-imm98 y m2-imm98 para crear el punto de montaje, que será un directorio llamado **datos**, en cada máquina cliente. Primeramente necesitamos instalar los siguientes paquetes:

```
sudo apt-get install nfs-common rpcbind
```

```
imm98@m1-imm98:~$ sudo apt-get install nfs-common rpcbind
```

Ahora crearemos el directorio datos en el directorio /home/imm98 y le daremos los permisos necesarios. Para ello ejecutaremos los siguientes comandos:

```
cd /home/imm98
mkdir datos
chmod -R 777 datos
```

```
imm98@m1-imm98:~$ cd /home/imm98
imm98@m1-imm98:~$ mkdir datos
imm98@m1-imm98:~$ chmod -R 777 datos
```

Una vez que ya tenemos el directorio datos creado vamos a montar la carpeta remota sobre este directorio datos. Tenemos que la IP de la máquina nfs-imm98 es la 192.168.56.109:

```
imm98@nfs-imm98:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c2:6a:39 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 83968sec preferred_lft 83968sec
    inet6 fe80::a00:27ff:fec2:6a39/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:25:06:e7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.109/24 brd 192.168.56.255 scope global dynamic enp0s8
        valid_lft 564sec preferred_lft 564sec
    inet6 fe80::a00:27ff:fe25:6e7/64 scope link
        valid_lft forever preferred_lft forever
```

Para montar dicha carpeta ejecutaremos el comando:

```
sudo mount 192.168.56.109:/home/imm98/datos/compartido /home/imm98/datos
```

```
imm98@m1-imm98:~$ sudo mount 192.168.56.109:/home/imm98/datos/compartido /home/imm98/datos_
```

Una vez que hemos montado la carpeta remota sobre el directorio datos, vamos a comprobar el correcto funcionamiento de dicha carpeta compartida. En el **servidor NFS** creamos un archivo en la carpeta remota **/datos/compartido** llamado **archivo\_compartido.txt**.

```
imm98@nfs-imm98:~/datos/compartido$ touch archivo_compartido.txt
imm98@nfs-imm98:~/datos/compartido$ ls
archivo_compartido.txt
```

Y en la máquina **m1-imm98** comprobamos mediante el comando ls que dicho archivo se encuentra en el directorio datos:

```
imm98@m1-imm98:~/datos$ ls
imm98@m1-imm98:~/datos$ ls
archivo_compartido.txt
```

Por tanto hemos comprobado el correcto funcionamiento del comando mount ejecutado anteriormente.

Una vez que hemos montado la carpeta remota sobre el directorio datos de m1-imm98 vamos a hacer lo mismo en la máquina **m2-imm98**. Para ello ejecutamos los comandos ejecutados en m1-imm98.

*sudo apt-get install nfs-common rpcbind*

```
imm98@m2-imm98:~$ sudo apt-get install nfs-common rpcbind
[sudo] password for imm98: _
```

Creamos el directorio datos en el directorio /home/imm98 y le daremos los permisos necesarios. Para ello ejecutaremos los siguientes comandos:

```
cd /home/imm98
mkdir datos
chmod -R 777 datos
```

```
imm98@m2-imm98:~$ cd /home/imm98
imm98@m2-imm98:~$ mkdir datos
imm98@m2-imm98:~$ chmod -R 777 datos
```

Para montar dicha carpeta ejecutaremos el comando:

*sudo mount 192.168.56.109:/home/imm98/datos/compartido /home/imm98/datos*

```
imm98@m2-imm98:~$ sudo mount 192.168.56.109:/home/imm98/datos/compartido /home/imm98/datos
```

Comprobamos el correcto funcionamiento del comando anterior viendo que el archivo "archivo\_compartido.txt" creado en el directorio /datos/compartido de la máquina nfs-imm98 se encuentra también en el directorio datos de la máquina m2-imm98:

```
imm98@m2-imm98:~$ cd datos
imm98@m2-imm98:~/datos$ ls
archivo_compartido.txt
```

Una vez que hemos comprobado el correcto funcionamiento en las máquinas m1-imm98 y m2-imm98 vamos a hacer que dicha configuración sea permanente. Para ello debemos introducir en el directorio **/etc/fstab** de las máquinas m1-imm98 y m2-imm98 la siguiente línea:

*192.168.56.109:/home/imm98/datos/compartido/ /home/imm98/datos/ nfs auto, noatime, noblock, bg, nfsvers=3, intr, tcp, actimeo=1800 0 0*

```
192.168.56.109:/home/imm98/datos/compartido/ /home/imm98/datos/ nfs auto, noatime, noblock, bg,
$g,nfsvers=3, intr, tcp, actimeo=1800 0 0
```

Vamos a reiniciar la máquina m1-imm98 para ver si la carpeta se monta correctamente tras el reinicio. Para reiniciar la máquina m1-imm98 ejecutaremos el comando:

*sudo reboot*

```
imm98@m1-imm98:~/datos$ sudo reboot
```

```
New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

imm98@m1-imm98:~$ cd datos
imm98@m1-imm98:~/datos$ ls
archivo_compartido.txt
```

Y vemos que justo después de producirse el reinicio de la máquina m1-imm98 vemos que la carpeta /datos/compartido sigue montada sobre el directorio datos.

Hacemos lo mismo en la máquina m2-imm98. introducimos en el directorio /etc/fstab de la máquina m2-imm98 la siguiente línea:

*192.168.56.109:/home/imm98/datos/compartido/ /home/imm98/datos/ nfs auto, noatime, noblock, bg, nfsvers=3, intr, tcp, actimeo=1800 0 0*

```
imm98@m2-imm98:~/datos$ sudo cat /etc/fstab
```

```
192.168.56.109:/home/imm98/datos/compartido/ /home/imm98/datos/ nfs auto, noatime, noblock, bg, nfsve
rs=3, intr, tcp, actimeo=1800 0 0
```

Vamos a reiniciar la máquina m1-imm98 para ver si la carpeta se monta correctamente tras el reinicio. Para reiniciar la máquina m1-imm98 ejecutaremos el comando:

*sudo reboot*

```
imm98@m2-imm98:~/datos$ sudo reboot
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

64 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-re
or proxy settings

imm98@m2-imm98:~$ cd datos
imm98@m2-imm98:~/datos$ ls
adios.txt  archivo_compartido.txt
imm98@m2-imm98:~/datos$
```

Y vemos que justo después de reiniciar la máquina m2-imm98, vemos que la carpeta /datos/compartido sigue montada sobre el directorio datos, como queríamos.

Como opciones avanzadas tenemos varias opciones para ver los dispositivos montados en nuestro sistema. Con estas opciones que ejecutaremos en la máquina m1-imm98 (sería idéntico en la máquina m2-imm98) comprobaremos que efectivamente la carpeta remota /datos/compartido está montada sobre la carpeta datos. La primera forma que tenemos para comprobarlo es consultar el archivo /proc/mounts. Para ello ejecutaremos el comando:

*cat /proc/mounts | grep datos*

```
imm98@m1-imm98:~/datos$ cat /proc/mounts | grep datos
192.168.56.109:/home/imm98/datos/compartido /home/imm98/datos nfs4 rw,relatime,vers=4.2,rsz
,wsz=131072,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.56.105,local
lock=none,addr=192.168.56.109 0 0
```

y comprobamos que efectivamente está esa línea en el archivo con información acerca del mount.



La segunda forma que tenemos de verlo es mediante el comando `mount -l` cuya salida es muy parecida al contenido del archivo `/proc/mounts`. Vamos a probarlo:

*`mount -l | grep datos`*

```
imm98@m2-imm98:~$ mount -l | grep datos
192.168.56.109:/home/imm98/datos/compartido on /home/imm98/datos type nfs4 (rw,relatime,vers=4.2,rsi
ze=131072,wsize=131072,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.56.1
04,local_lock=none,addr=192.168.56.109)
```

y vemos que efectivamente nos muestra dichos directorios que hemos montado el uno sobre el otro.

La última forma que tenemos de comprobar que el montaje realizado anteriormente se ha realizado con éxito tenemos es el siguiente comando que nos muestra el directorio que se monta y el directorio de montaje además de información asociada a la memoria de dichos montajes :

*`df -aTh`*

```
imm98@m1-imm98:~/datos$ df -aTh
```

cgroup	cgroup	0	0	0	-	/sys/fs/cgroup/cpuset
systemd-1	-	-	-	-	-	/proc/sys/fs/binfmt_m
isc						
hugetlbfs	hugetlbfs	0	0	0	-	/dev/hugepages
debugfs	debugfs	0	0	0	-	/sys/kernel/debug
mqueue	mqueue	0	0	0	-	/dev/mqueue
sunrpc	rpc_pipefs	0	0	0	-	/run/rpc_pipefs
fusectl	fusectl	0	0	0	-	/sys/fs/fuse/connecti
ons						
configfs	configfs	0	0	0	-	/sys/kernel/config
/dev/sda2	ext4	976M	284M	626M	32%	/boot
lxcfs	fuse.lxcfs	0	0	0	-	/var/lib/lxcfs
192.168.56.109:/home/imm98/datos/compartido	nfs4	8.8G	3.6G	4.8G	44%	/home/imm98/datos

y vemos que efectivamente nos muestra que el directorio que se monta es `192.168.56.109:/home/imm98/datos/compartido` y el directorio de montaje es `/home/imm98/datos`

## 4. SEGURIDAD EN EL SERVIDOR NFS

Vamos a configurar el cortafuegos del servidor NFS que acabamos de crear. Recordemos que ya habíamos configurado en anteriores prácticas el cortafuegos de las máquinas m1-imm98 y m2-imm98. Para dicha configuración del cortafuegos utilizaremos IPTABLES.

Al utilizar los servicios mountd y nlockmgr puertos dinámicos, vamos a fijar dichos puertos para poder añadir las reglas del cortafuegos. Primeramente vamos a fijar el puerto del servicio **mountd**. Para ello, vamos a modificar el archivo **/etc/defaults/nfs-kernel-server** y vamos a utilizar el puerto 2000 tanto tcp como udp para este servicio. Para ello vamos a insertar esta línea en dicho archivo:

```
RPCMOUNTDOPTS="--manage-gids -p 2000"
```

```
# Number of servers to start up
RPCNFSDCOUNT=8

# Runtime priority of server (see nice(1))
RPCNFSDPRIORITY=0

# Options for rpc.mountd.
# If you have a port-based firewall, you might want to set up
# a fixed port here using the --port option. For more information,
# see rpc.mountd(8) or http://wiki.debian.org/SecuringNFS
# To disable NFSv4 on the server, specify '--no-nfs-version 4' here
RPCMOUNTDOPTS="--manage-gids -p 2000"

# Do you want to start the svcgssd daemon? It is only required for Kerberos
# exports. Valid alternatives are "yes" and "no"; the default is "no".
NEED_SVCSSD=""

# Options for rpc.svcgssd.
RPCSVCGSSDOPTS=""
```

Posteriormente fijamos el puerto para el servicio **nlockmgr**. Vamos a hacer que este servicio escuche en los puertos 2001 (tcp) y 2002 (udp). Para fijar estos puertos crearemos el fichero **swap-nfs-ports.conf** en el directorio **/tc/sysctl.d/** en el cual añadiremos:

```
imm98@nfs-imm98:~$ sudo nano /etc/sysctl.d/swap-nfs-ports.conf
```

```
fs.nfs.nlm_tcpport=2001
fs.nfs.nlm_udpport=2002
```

```
fs.nfs.nlm_tcpport=2001
fs.nfs.nlm_udpport=2002
```

Este fichero donde hemos indicado los puertos de nlockmgr lo lanzaremos como nuevo archivo de configuración con la orden:

***sudo sysctl --system***

```
imm98@nfs-imm98:~$ sudo sysctl --system
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1
* Applying /etc/sysctl.d/10-link-restrictions.conf ...
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
* Applying /etc/sysctl.d/10-lxd-inotify.conf ...
fs.inotify.max_user_instances = 1024
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...
kernel.sysrq = 176
* Applying /etc/sysctl.d/10-network-security.conf ...
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.tcp_syncookies = 1
* Applying /etc/sysctl.d/10-ptrace.conf ...
kernel.yama.ptrace_scope = 1
* Applying /etc/sysctl.d/10-zero-page.conf ...
vm.mmap_min_addr = 65536
* Applying /usr/lib/sysctl.d/50-default.conf ...
net.ipv4.conf.all.promote_secondaries = 1
net.core.default_qdisc = fq_codel
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.d/swap-nfs-ports.conf ...
fs.nfs.nlm_tcpport = 2001
fs.nfs.nlm_udpport = 2002
* Applying /etc/sysctl.conf ...
```

Y reiniciamos el servidor NFS con el comando:

***sudo /etc/init.d/nfs-kernel-server restart***

```
imm98@nfs-imm98:~$ sudo /etc/init.d/nfs-kernel-server restart
[ ok ] Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
```

Para ver que esta configuración de los puertos se ha realizado correctamente ejecutamos el comando con el que vemos los puertos asociados a cada servicio:

***sudo rpcinfo -p localhost***

```
imm98@nfs-imm98:~$ sudo rpcinfo -p localhost
program vers proto port service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100005 1 udp 2000 mountd
100005 1 tcp 2000 mountd
100005 2 udp 2000 mountd
100005 2 tcp 2000 mountd
100005 3 udp 2000 mountd
100005 3 tcp 2000 mountd
100003 3 tcp 2049 nfs
100003 4 tcp 2049 nfs
100227 3 tcp 2049
100003 3 udp 2049 nfs
100227 3 udp 2049
100021 1 udp 2002 nlockmgr
100021 3 udp 2002 nlockmgr
100021 4 udp 2002 nlockmgr
100021 1 tcp 2001 nlockmgr
100021 3 tcp 2001 nlockmgr
100021 4 tcp 2001 nlockmgr
```

Por tanto a modo de resumen, tenemos los siguientes servicios y puertos asociados que tenemos que añadir como reglas en el cortafuegos de la máquina nfs-imm98:

SERVICIO	PUERTO
nfs	2049
portmapper	111
mountd	2000
nlockmgr (tcp)	2001
nlockmgr (udp)	2002

Por lo tanto vamos a crear un archivo llamado **cort\_m6.sh** en el que vamos a establecer las reglas para que se deniegue todo el tráfico entrante salvo los servicios anteriores. Para ello incluimos en dicho archivo cort\_m6.sh lo siguiente:

```
# (1) Eliminar todas las reglas
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# (2) Denegar todo el trafico entrante
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# (3) Abrir el puerto 2049 para NFS(TCP)
iptables -A INPUT -p tcp --dport 2049 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 2049 -j ACCEPT

# (4) Abrir el puerto 2049 para NFS (UDP)
iptables -A INPUT -p udp --dport 2049 -j ACCEPT
iptables -A OUTPUT -p udp --dport 2049 -j ACCEPT

#(5) Abrir el puerto 111 para PORTMAPPER(TCP)
iptables -A INPUT -p tcp --dport 111 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 111 -j ACCEPT
# (6) Abrir el puerto 111 para PORTMAPPER(UDP)
iptables -A INPUT -p udp --dport 111 -j ACCEPT
iptables -A OUTPUT -p udp --dport 111 -j ACCEPT

# (7) Abrir el puerto 2000 para MOUNT(TCP)
iptables -A INPUT -p tcp --dport 2000 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 2000 -j ACCEPT

# (8) Abrir el puerto 2000 para MOUNT(UDP)
iptables -A INPUT -p udp --dport 2000 -j ACCEPT
iptables -A OUTPUT -p udp --dport 2000 -j ACCEPT

# (9) Abrir el puerto 2001 para NLOCKMGR TCP
iptables -A INPUT -p tcp --dport 2001 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 2001 -j ACCEPT

# (10) Abrir el puerto 2002 para NLOCKMGR UDP
iptables -A INPUT -p udp --dport 2002 -j ACCEPT
iptables -A OUTPUT -p udp --dport 2002 -j ACCEPT

iptables -L -n -v
```

Ejecutamos dicho script y este es el resultado de esa ejecución (no se ve completamente el resultado porque la salida es bastante grande):

```
sudo ./cort_p6.sh
```

```
imm98@nfs-imm98:~$ sudo ./cort_p6.sh
```

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	ACCEPT	all	--	*	*	0.0.0.0/0	0.0.0.0/0	state RELATED,ESTABLISHED
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:204
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:204
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:111
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:111
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:200
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:200
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:200
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:200

Chain FORWARD (policy DROP 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:204
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:204
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:111
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:111
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:200
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:200
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:200
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:200

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)

pkts	bytes	target	prot	opt	in	out	source	destination	
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:204
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:204
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:111
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:111
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:200
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:200
0	0	ACCEPT	tcp	--	*	*	0.0.0.0/0	0.0.0.0/0	tcp dpt:200
0	0	ACCEPT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:200

Y vemos que las reglas de iptables que se han creado son correctas. Vamos a comprobar ahora el correcto funcionamiento de la carpeta remota que habíamos montado en las máquinas clientes m1-imm98 y m2-imm98. Para dicha comprobación crearemos un archivo en la carpeta /datos/compartido de la máquina nfs-imm98 llamado

**archivo\_post\_iptables.txt :**

```
imm98@nfs-imm98:~$ cd datos/
imm98@nfs-imm98:~/datos$ ls
compartido
imm98@nfs-imm98:~/datos$ cd compartido/
imm98@nfs-imm98:~/datos/compartido$ touch archivo_post_iptables.txt
```

Y comprobamos que dicho archivo se encuentra en el directorio datos de m1-imm98:

```
imm98@m1-imm98:~/datos$ ls
adios.txt archivo_compartido.txt archivo_post_iptables.txt
```

y en el directorio datos de la máquina m2-imm98:

```
imm98@m2-imm98:~/datos$ ls  
adios.txt  archivo_compartido.txt  archivo_post_iptables.txt
```

Con esta prueba hemos comprobado el funcionamiento correcto de las reglas iptables que hemos establecido anteriormente.