

## Práctica 6:

### **Servidor de disco NFS**

#### ***Índice:***

- 1) Introducción*
- 2) Crear máquina virtual y configurarla como servidor NFS*
- 3) Configurar los clientes M1 y M2*
- 4) Seguridad en el servidor NFS*

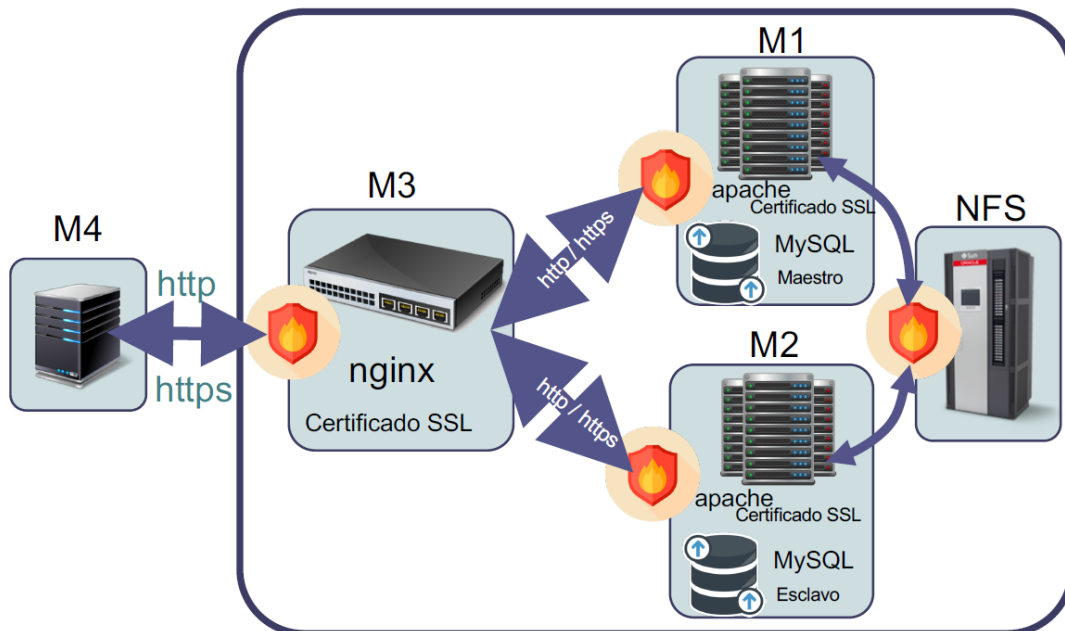
## 1) Introducción

El objetivo principal de esta práctica es configurar un servidor NFS para exportar un espacio en disco a los servidores finales de nuestra granja web, los cuales actuarán como clientes-NFS).

Los objetivos concretos de esta práctica son:

- Configurar una máquina como servidor de disco NFS y exportar una carpeta a los clientes.
- Montar en las máquinas cliente la carpeta exportada por el servidor.
- Comprobar que la información que se escribe en una máquina en dicha carpeta se ve actualizada en el resto de máquinas que comparten ese espacio.
- Añadir configuración de seguridad al servidor NFS.

Para así conseguir la siguiente estructura en nuestra granja web:



## 2) Crear máquina virtual y configurarla como servidor NFS

Instalaremos una nueva máquina virtual llamada NFS con Ubuntu Server como hicimos en la primera práctica (1024 MB de RAM y 10 GB de disco virtual). Una vez instalada deberemos configurar sus adaptadores de red NAT y solo-anfitrión (IP máquina NFS = 192.168.56.104/24).

```
joselepedraza@nfs:/etc/netplan$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:feaa:321e prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:aa:32:1e txqueuelen 1000 (Ethernet)
    RX packets 44 bytes 12882 (12.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 57 bytes 6430 (6.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.104 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe90:69d0 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:90:69:d0 txqueuelen 1000 (Ethernet)
    RX packets 57 bytes 5244 (5.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 656 (656.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 92 bytes 7036 (7.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 92 bytes 7036 (7.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Para comprobar que se comunica correctamente con las máquinas m1 y m2 tiramos sus cortafuegos (ejecutando en cada una el script: firewallresetconf.sh) y probamos haciendo ping desde la máquina NFS a la máquina m1 y m2:

```
joselepedraza@nfs:/etc/netplan$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data:
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.702 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.727 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.969 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.836 ms
^C
--- 192.168.56.102 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3056ms
rtt min/avg/max/mdev = 0.702/0.808/0.969/0.109 ms
joselepedraza@nfs:/etc/netplan$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data:
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.777 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.924 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.597 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.252 ms
^C
--- 192.168.56.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3035ms
rtt min/avg/max/mdev = 0.252/0.637/0.924/0.252 ms
joselepedraza@nfs:/etc/netplan$ _
```

La comunicación entre máquinas es correcta.

A continuación, en la máquina NFS debemos instalar las herramientas necesarias para utilizarla como servidor NFS con el siguiente comando:

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

Una vez instalado, creamos la carpeta que vamos a compartir con los clientes y cambiamos el propietario y permisos de esa carpeta como sigue:

```
sudo mkdir /datos/compartido
```

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

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

```
joselepedraza@nfs:~$ mkdir datos
joselepedraza@nfs:~$ cd datos/
joselepedraza@nfs:~/datos$ mkdir compartido
joselepedraza@nfs:~/datos$ cd ..
joselepedraza@nfs:~$ sudo chown nobody:nogroup /datos/compartido/
chown: cannot access '/datos/compartido/': No such file or directory
joselepedraza@nfs:~$ ls
datos
joselepedraza@nfs:~$ sudo chown nobody:nogroup /datos/compartido
chown: cannot access '/datos/compartido': No such file or directory
joselepedraza@nfs:~$ sudo chown nobody:nogroup datos/compartido
joselepedraza@nfs:~$ ls -la
total 44
drwxr-xr-x 6 joselepedraza joselepedraza 4096 may 19 20:51 .
drwxr-xr-x 3 root          root          4096 may 19 19:23 ..
-rw----- 1 joselepedraza joselepedraza  105 may 19 19:49 .bash_history
-rw-r--r-- 1 joselepedraza joselepedraza  220 abr  4 2018 .bash_logout
-rw-r--r-- 1 joselepedraza joselepedraza 3771 abr  4 2018 .bashrc
drwx----- 2 joselepedraza joselepedraza 4096 may 19 19:31 .cache
drwxrwxr-x 3 joselepedraza joselepedraza 4096 may 19 20:52 datos
drwx----- 3 joselepedraza joselepedraza 4096 may 19 19:31 .gnupg
-rw-r--r-- 1 joselepedraza joselepedraza  807 abr  4 2018 .profile
drwx----- 2 joselepedraza joselepedraza 4096 may 19 20:06 .ssh
-rw-r--r-- 1 joselepedraza joselepedraza    0 may 19 19:55 .sudo_as_admin_successful
-rw----- 1 root          root          1154 may 19 20:00 .viminfo
joselepedraza@nfs:~$ sudo chmod -R 777 datos/compartido/
joselepedraza@nfs:~$ ls -l
total 4
drwxrwxr-x 3 joselepedraza joselepedraza 4096 may 19 20:52 datos
joselepedraza@nfs:~$ _
```

Una vez hecho esto, debemos dar permiso a las máquinas clientes (m1 y m2) añadiendo las IP correspondientes al archivo `/etc/exports` como sigue:

```
/datos/compartido/ IP_M1(rw) IP_M2(rw)
```

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4       gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home/joselepedraza/datos/compartido/ 192.168.56.101(rw) 192.168.56.102(rw)
```

Finalmente, debemos reiniciar el servicio y comprobar el estado:

```
sudo service nfs-kernel-server restart
```

```
sudo service nfs-kernel-server status
```

```
joselepedraza@nfs:~/datos/compartido$ sudo service nfs-kernel-server restart
joselepedraza@nfs:~/datos/compartido$ 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 Tue 2020-05-19 21:07:42 UTC; 13s ago
     Process: 2977 ExecStopPost=/usr/sbin/exportfs -f (code=exited, status=0/SUCCESS)
     Process: 2976 ExecStopPost=/usr/sbin/exportfs -au (code=exited, status=0/SUCCESS)
     Process: 2833 ExecStop=/usr/sbin/rpc.nfsd 0 (code=exited, status=0/SUCCESS)
     Process: 3018 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
     Process: 3017 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
    Main PID: 3018 (code=exited, status=0/SUCCESS)

may 19 21:07:42 nfs systemd[1]: Starting NFS server and services...
may 19 21:07:42 nfs exportfs[3017]: exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no_subtr
may 19 21:07:42 nfs exportfs[3017]: Assuming default behaviour ('no_subtree_check').
may 19 21:07:42 nfs exportfs[3017]: NOTE: this default has changed since nfs-utils version 1.0.x
may 19 21:07:42 nfs exportfs[3017]: exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no_subtr
may 19 21:07:42 nfs exportfs[3017]: Assuming default behaviour ('no_subtree_check').
may 19 21:07:42 nfs exportfs[3017]: NOTE: this default has changed since nfs-utils version 1.0.x
may 19 21:07:42 nfs systemd[1]: Started NFS server and services.
lines 1-18/18 (END)
```

### 3) Configurar los clientes M1 y M2

En los clientes (máquinas m1 y m2) debemos instalar los paquetes necesarios y crear el punto de montaje (el directorio “datos” en cada máquina cliente):

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

```
cd /home/joselepedraza
```

```
mkdir datos
```

```
chmod -R 777 datos
```

Mostraremos capturas del proceso en m1 y haremos lo mismo para la máquina m2:

```
nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /lib/systemd/system/n
fs-client.target.
nfs-utils.service is a disabled or a static unit, not starting it.
Procesando disparadores para systemd (237-3ubuntu10.39) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para ureadahead (0.100.0-21) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...
joselepedraza@m1:~$ cd /home/joselepedraza/
joselepedraza@m1:~$ mkdir datos
joselepedraza@m1:~$ chmod -R 777 datos
joselepedraza@m1:~$ ls -l
total 24
-rw-r--r-- 1 root      root          606 may  6 20:28 createxampleDB.sql
drwxrwxrwx 2 joselepedraza joselepedraza 4096 may 20 11:14 datos
-rwxrwxrwx 1 root      root           257 abr 25 22:57 firewallresetconf.sh
-rwxrwxr-x 1 joselepedraza joselepedraza 1125 may  8 12:54 iptablesfirewallconf.sh
-rw-rw-r-- 1 joselepedraza joselepedraza 3392 mar 26 20:32 tar.tgz
drwxrwxr-x 3 joselepedraza joselepedraza 4096 mar 26 20:35 var
joselepedraza@m1:~$ _
```

Ahora ya podemos montar la carpeta remota (la exportada en el servidor NFS) sobre el directorio recién creado:

*sudo mount IP\_NFS:/home/joselepedraza//datos/compartido datos*

```
joselepedraza@m1:~$ sudo mount 192.168.56.104:/home/joselepedraza/datos/compartido datos
joselepedraza@m1:~$ ls -l
total 24
-rw-r--r-- 1 root      root          606 may  6 20:28 createexampleDB.sql
drwxrwxrwx 2 nobody   nogroup       4096 may 19 20:52 datos
-rwxrwxrwx 1 root      root          257 abr 25 22:57 firewallresetconf.sh
-rwxrwxr-x 1 joselepedraza joselepedraza 1125 may  8 12:54 iptablesfirewallconf.sh
-rw-rw-r-- 1 joselepedraza joselepedraza 3392 mar 26 20:32 tar.tgz
drwxrwxr-x 3 joselepedraza joselepedraza 4096 mar 26 20:35 var
joselepedraza@m1:~$ ls -la datos
total 8
drwxrwxrwx 2 nobody   nogroup       4096 may 19 20:52 
drwxr-xr-x 8 joselepedraza joselepedraza 4096 may 20 11:14 ..
joselepedraza@m1:~$ touch datos/archivo1.txt
joselepedraza@m1:~$ _
```

Como vemos, podemos comprobar que se pueden leer y escribir los archivos almacenados en la carpeta compartida:

*ls -la datos*

*touch datos/archivo1.txt*

Entonces en ambas máquinas (m1 y m2) y también en el servidor NFS podemos acceder a todos los archivos que modifiquemos en la carpeta compartida, tanto para lectura como para escritura:

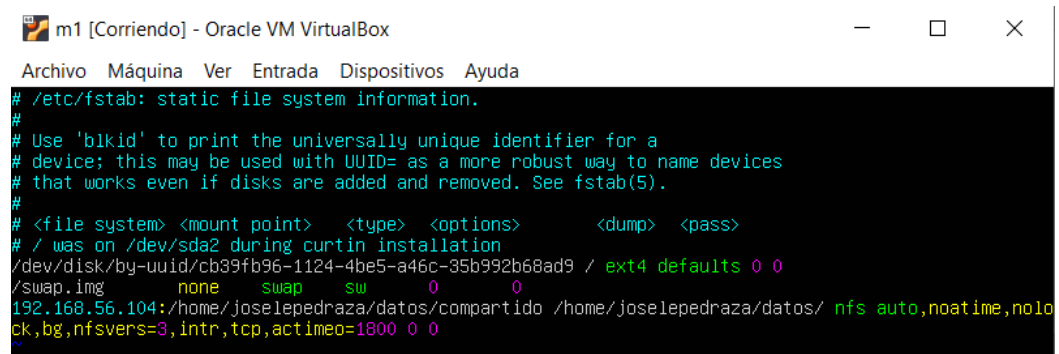
```
joselepedraza@m2:~$ sudo mount 192.168.56.104:/home/joselepedraza/datos/compartido datos
[sudo] password for joselepedraza:
joselepedraza@m2:~$ ls -la datos
total 8
drwxrwxrwx 2 nobody   nogroup       4096 may 20 11:33 
drwxr-xr-x 7 joselepedraza joselepedraza 4096 may 20 11:09 ..
-rw-rw-r-- 1 joselepedraza joselepedraza   0 may 20 11:33 archivo1.txt
joselepedraza@m2:~$ _
```

```
joselepedraza@nfs:~/datos/compartido$ ls -la
total 8
drwxrwxrwx 2 nobody   nogroup       4096 may 20 11:33 
drwxrwxr-x 3 joselepedraza joselepedraza 4096 may 19 20:52 ..
-rw-rw-r-- 1 joselepedraza joselepedraza   0 may 20 11:33 archivo1.txt
joselepedraza@nfs:~/datos/compartido$ _
```

Una vez comprobado que funciona correctamente, debemos hacer la configuración permanente (tanto en m1 como en m2), añadiendo una línea al archivo de configuración */etc/fstab* para que la carpeta compartida se monte al arrancar el sistema.

Entonces, añadiremos la siguiente línea editando el archivo con *vim*:

```
IP_NFS:/home/joselepedraza/datos/compartido /home/joselepedraza/datos nfs
auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```



```
m1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/cb39fb96-1124-4be5-a46c-35b992b68ad9 / ext4 defaults 0 0
/swap.img none swap sw 0 0
192.168.56.104:/home/joselepedraza/datos/compartido /home/joselepedraza/datos/ nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```

Tras reiniciar el sistema, la carpeta compartida debería montarse y quedar disponible de forma automática.

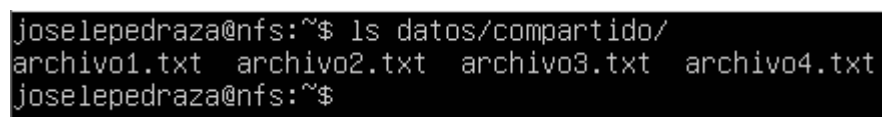
Entonces después de reiniciar volvemos a crear un archivo en el directorio compartido (a través de m1 o m2) y comprobamos que funciona correctamente:



```
joselepedraza@m1:~/datos$ ls
archivo1.txt  archivo2.txt
joselepedraza@m1:~/datos$ touch archivo3.txt
joselepedraza@m1:~/datos$ ls
archivo1.txt  archivo2.txt  archivo3.txt  archivo4.txt
joselepedraza@m1:~/datos$ _
```



```
joselepedraza@m2:~$ cd datos/
joselepedraza@m2:~/datos$ ls
archivo1.txt  archivo2.txt  archivo3.txt
joselepedraza@m2:~/datos$ touch archivo4.txt
joselepedraza@m2:~/datos$ ls
archivo1.txt  archivo2.txt  archivo3.txt  archivo4.txt
joselepedraza@m2:~/datos$ _
```



```
joselepedraza@nfs:~$ ls datos/compartido/
archivo1.txt  archivo2.txt  archivo3.txt  archivo4.txt
joselepedraza@nfs:~$
```

#### 4) Seguridad en el servidor NFS

Antes de continuar, levantamos los cortafuegos de las máquinas m1 y m2 (ejecutando el script `./iptablesfirewallconf.sh` desarrollado en la práctica anterior) y comprobamos que sigue funcionando correctamente (lo cual es lógico debido a que los cortafuegos de las máquinas m1 y m2 no restringen nada respecto a la máquina NFS, pero los desactivamos por si la configuración hubiese sido otra).

```
joselepedraza@m1:~$ cd datos/
joselepedraza@m1:~/datos$ ls
archivo1.txt archivo2.txt archivo3.txt archivo4.txt
joselepedraza@m1:~/datos$ touch pruebaCONcortafuegosM1.txt
joselepedraza@m1:~/datos$ ls
archivo1.txt archivo3.txt pruebaCONcortafuegosM1.txt
archivo2.txt archivo4.txt pruebaCONcortafuegosM2.txt
joselepedraza@m1:~/datos$ _
```

```
joselepedraza@m2:~$ cd datos/
joselepedraza@m2:~/datos$ ls
archivo1.txt archivo2.txt archivo3.txt archivo4.txt pruebaCONcortafuegosM1.txt
joselepedraza@m2:~/datos$ touch pruebaCONcortafuegosM2.txt
joselepedraza@m2:~/datos$ _
```

```
joselepedraza@nfs:~$ ls datos/compartido/
archivo1.txt archivo3.txt pruebaCONcortafuegosM1.txt
archivo2.txt archivo4.txt pruebaCONcortafuegosM2.txt
joselepedraza@nfs:~$ _
```

Una vez hecho esto, partimos de configurar IPTABLES con política por defecto de denegación implícita de tráfico entrante en la máquina NFS:

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD DROP
```

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
#!/bin/bash

#Eliminar todas las reglas (configuración limpia):
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#Política por defecto de denegación implícita de tráfico entrante:
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -L -n -y
```



Ejecutamos el script para la configuración por defecto:

```
joselepedraza@nfs:~$ ls
datos firewallresetconf.sh
joselepedraza@nfs:~$ sudo chmod 777 firewallresetconf.sh
joselepedraza@nfs:~$ sudo ./firewallresetconf.sh
Chain INPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source destination state
  0      0 ACCEPT     all  --  *      *       0.0.0.0/0  0.0.0.0/0  state RELATED,ESTABLISHED

Chain FORWARD (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source destination
joselepedraza@nfs:~$ _
```

Una vez hecho esto, debemos abrir los distintos puertos asociados a servicios NFS. Para ello, mientras que el servicio *nfs*, por defecto utiliza el puerto 2049 (tcp y udp) y el servicio *portmapper*, por defecto utiliza el puerto 111 (tcp y udp), los servicios *mountd* y *nlockmgr* utilizan puertos dinámicos por lo que no podemos abrir dinámicamente puertos con iptables. Para ello vamos a fijar puertos a los servicios mountd y nlockmgr para poder añadir reglas a iptables.

Entonces para el servicio *mountd* modificamos el archivo */etc/default/nfs-kernel-server* añadiendo *-p 2000* (esto es, por ejemplo, que vamos a utilizar el puerto 2000 tanto tcp como udp):

*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=""
```

Para el servicio *nlockmgr* es un poco más complejo porque es parte de un módulo del kernel, pero a través de *systemd* se puede reconfigurar ese módulo sin reiniciar el sistema. Para ello, vamos a crear el archivo *swap-nfs-ports.conf* en */etc/sysctl.d/* donde añadiremos las opciones correspondientes:

*fs.nfs.nlm\_tcpport = 2001*

*fs.nfs.nlm\_udpport = 2002*

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

Con estas opciones estamos configurando el servicio **nlockmgr** para que escuche en los puertos 2001 (tcp) y 2002 (udp). Ahora lanzamos el nuevo archivo de configuración que hemos creado y reiniciamos el servidor NFS:

```
sudo sysctl --system
```

```
/etc/init.d/nfs-kernel-server restart
```

```
joselepedraza@nfs:/$ 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 ...
* 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
* Applying /etc/sysctl.conf ...
joselepedraza@nfs:/$ /etc/init.d/nfs-kernel-server restart
[....] Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service==== AUTHENTICATING FOR
R org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'nfs-server.service'.
Authenticating as: Jose Luis Pedraza Román (joselepedraza)
Password:
==== AUTHENTICATION COMPLETE ====
. ok
joselepedraza@nfs:/$ _
```

Ahora podemos comprobar tras las configuraciones de los puertos anteriormente realizadas, los puertos asociados a cada servicio con el comando siguiente:

```
sudo rpcinfo -p
```

```
joselepedraza@nfs:~/datos/compartido$ sudo rpcinfo -p
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 37663 nlockmgr
100021 3 udp 37663 nlockmgr
100021 4 udp 37663 nlockmgr
100021 1 tcp 2001 nlockmgr
100021 3 tcp 2001 nlockmgr
100021 4 tcp 2001 nlockmgr
```

Para terminar con la configuración, abrimos los puertos correspondientes en la máquina NFS para que las máquinas m1 y m2 puedan acceder a NFS:

```
iptables -A INPUT -s IP_M1, IP_M2 -p tcp -m multiport --ports 111,2000,2001,2049 -j ACCEPT
```

```
iptables -A INPUT -s IP_M1, IP_M2 -p udp -m multiport --ports 111,2000,2002,2049 -j ACCEPT
```

```
#!/bin/bash

#Eliminar todas las reglas (configuración limpia):
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#Política por defecto de denegación implícita de tráfico entrante:
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

#Abrir puertos para que las máquinas m1 y m2 puedan acceder a NFS:
iptables -A INPUT -s 192.168.56.101,192.168.56.102 -p tcp -m multiport --ports 111,2000,2001,2049 -j ACCEPT
iptables -A INPUT -s 192.168.56.101,192.168.56.102 -p udp -m multiport --ports 111,2000,2002,2049 -j ACCEPT

iptables -L -n -v
```

```
"firewallresetconf.sh" 20L, 649C escritos
joselepedraza@nfs:~$ sudo ./firewallresetconf.sh
Chain INPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination        state
  0      0 ACCEPT     all  --  *      *       0.0.0.0/0         0.0.0.0/0          state RELATED,ESTABLISHED
  0      0 ACCEPT     tcp  --  *      *       192.168.56.101    0.0.0.0/0          multiport ports 111,2000,2001,2049
  0      0 ACCEPT     tcp  --  *      *       192.168.56.102    0.0.0.0/0          multiport ports 111,2000,2001,2049
  0      0 ACCEPT     udp   --  *      *       192.168.56.101    0.0.0.0/0          multiport ports 111,2000,2002,2049
  0      0 ACCEPT     udp   --  *      *       192.168.56.102    0.0.0.0/0          multiport ports 111,2000,2002,2049

Chain FORWARD (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
joselepedraza@nfs:~$ _
```

Para comprobar que funciona correctamente la configuración realizada, simplemente creamos un archivo en la carpeta compartida y debería aparecer en las distintas máquinas:

```
joselepedraza@nfs:~$ cd datos/compartido/
joselepedraza@nfs:~/datos/compartido$ touch pruebaCONcortafuegosNFS.txt
joselepedraza@nfs:~/datos/compartido$ ls
archivo1.txt  archivo3.txt  pruebaCONcortafuegosM1.txt  pruebaCONcortafuegosNFS.txt
archivo2.txt  archivo4.txt  pruebaCONcortafuegosM2.txt
```

```
joselepedraza@m1:~$ cd datos/
joselepedraza@m1:~/datos$ ls
archivo1.txt  archivo3.txt  pruebaCONcortafuegosM1.txt  pruebaCONcortafuegosNFS.txt
archivo2.txt  archivo4.txt  pruebaCONcortafuegosM2.txt
```

```
joselepedraza@m2:~$ cd datos/
joselepedraza@m2:~/datos$ ls
archivo1.txt  archivo3.txt  pruebaCONcortafuegosM1.txt  pruebaCONcortafuegosNFS.txt
archivo2.txt  archivo4.txt  pruebaCONcortafuegosM2.txt
```

Como hemos comprobado, funciona de maravilla.

Servidores Web de Altas Prestaciones