

# Servidores Web de Altas Prestaciones

## Práctica 6

**Autor: Juan Ocaña Valenzuela**

En esta práctica se deben cumplir los siguientes objetivos obligatorios:

- 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 carpetas que comparten ese espacio
- Añadir configuración de seguridad al servidor NFS asegurando los accesos

### Configurar una máquina como servidor de disco NFS y exportar una carpeta a los clientes

Configuraremos una máquina nueva de Ubuntu 18.04 llamada NFS como servidor de disco NFS, comprobando que puede conectarse con el resto de máquinas. Le daremos la IP estática

192.168.56.105.

Una vez finalizada la instalación, instalamos los paquetes `nfs-kernel-server`, `nfs-common` y `rpcbind`.

```
Ubuntu 18.04.4 LTS nfs tty1
nfs login: patchispatch
Password:
Last login: Sun May 17 11:55:13 UTC 2020 on tty1
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-99-generic x86_64)

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

System information as of Sun May 17 17:42:24 UTC 2020

System load:  0.18           Processes:            104
Usage of /:   30.3% of 9.78GB Users logged in:      0
Memory usage: 33%           IP address for enp0s3: 10.0.2.15
Swap usage:   0%            IP address for enp0s8: 192.168.56.105

 * MicroK8s passes 9 million downloads. Thank you to all our contributors!

https://microk8s.io/

Pueden actualizarse 69 paquetes.
44 actualizaciones son de seguridad.

patchispatch@nfs:~$ sudo apt install nfs-kernel-server nfs-common rpcbind
```

Creamos la carpeta de datos `/datos/compartido`, y cambiamos el propietario con `sudo chown nobody:nogroup /datos/compartido`. Por último, le damos todos los permisos de forma recursiva con `sudo chmod -R 777 /datos/compartido`.

```
patchispatch@nfs:~$ ls -l /datos/
total 4
drwxrwxrwx 2 nobody nogroup 4096 may 17 12:02 compartido
patchispatch@nfs:~$ _
```

Para permitir que accedan las máquinas M1 ( 192.168.56.103 ) y M2 ( 192.168.56.101 ) a la carpeta compartida, debemos editar el archivo de configuración `/etc/exports` como sigue a continuación:

[illegible]

Finalmente, reiniciamos el servicio y comprobamos que queda activo. Pasamos a realizar la configuración en las máquinas cliente.

## Montar en las máquinas cliente la carpeta exportada por el servidor

En las máquinas cliente debemos instalar los paquetes `nfs-common` y `rpcbind`. Una vez hecho, crear una carpeta con todos los permisos para montar los datos:

```
patchispatch@m1:~$ ls -l
total 20
drwxrwxrwx 2 patchispatch patchispatch 4096 may 17 17:47 datos
-rwxr-xr-x 1 patchispatch patchispatch 922 may 3 12:56 iptables.sh
-rw-rw-r-- 1 patchispatch patchispatch 3226 may 2 18:30 less
drwxr-xr-x 3 patchispatch patchispatch 4096 mar 21 17:51 m2
-rwxrwxr-x 1 patchispatch patchispatch 164 may 3 12:01 reset.sh
patchispatch@m1:~$
```

```
patchispatch@m2:~$ ls -l
total 24
-rw-r--r-- 1 patchispatch patchispatch 1448 may 2 18:50 apache.crt
-rw----- 1 patchispatch patchispatch 1704 may 2 18:50 apache.key
-rw-r--r-- 1 root root 431 mar 21 17:42 config.yaml
drwxrwxrwx 2 patchispatch patchispatch 4096 may 17 17:47 datos
-rwxr-xr-x 1 patchispatch patchispatch 925 may 3 12:59 iptables.sh
-rwxrwxr-x 1 patchispatch patchispatch 164 may 15 11:04 reset.sh
patchispatch@m2:~$ _
```

Montamos la carpeta remota sobre el directorio con la orden `sudo mount`

`192.168.56.105:/datos/compartido datos:`

```
patchispatch@m1:~$ sudo mount 192.168.56.105:/datos/compartido datos
[sudo] password for patchispatch:
patchispatch@m1:~$
```

```
patchispatch@m2:~$ sudo mount 192.168.56.105:/datos/compartido datos
[sudo] password for patchispatch:
patchispatch@m2:~$
```

Para comprobar que funciona, vamos a crear un archivo en M1 y veremos si está disponible en M2 y NFS:

```
patchispatch@m1:~/datos$ touch prueba.md
patchispatch@m1:~/datos$ ls -l
total 0
-rw-rw-r-- 1 patchispatch patchispatch 0 may 17 18:35 prueba.md
patchispatch@m1:~/datos$ ls -la
total 8
drwxrwxrwx 2 nobody nogroup 4096 may 17 18:35 .
drwxr-xr-x 8 patchispatch patchispatch 4096 may 17 17:47 ..
-rw-rw-r-- 1 patchispatch patchispatch 0 may 17 18:35 prueba.md
patchispatch@m1:~/datos$
```

```
patchispatch@m2:~$ ls -la datos
total 8
drwxrwxrwx 2 nobody nogroup 4096 may 17 18:35 .
drwxr-xr-x 6 patchispatch patchispatch 4096 may 17 17:47 ..
-rw-rw-r-- 1 patchispatch patchispatch 0 may 17 18:35 prueba.md
patchispatch@m2:~$
```

```
patchispatch@nfs:~$ ls -la /datos/compartido
total 8
drwxrwxrwx 2 nobody nogroup 4096 may 17 18:35 .
drwxr-xr-x 3 root root 4096 may 17 12:02 ..
-rw-rw-r-- 1 patchispatch patchispatch 0 may 17 18:35 prueba.md
patchispatch@nfs:~$ _
```

## Añadir configuración de seguridad al servidor NFS asegurando los accesos

Partimos de una política por defecto de denegación implícita:

```
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Antes de configurar los servicios **mountd** y **nlockmgr**, debemos fijar sus puertos, ya que utilizan puertos dinámicos.

**mountd**

Añadimos `-p 2000` para que utilice el puerto 2000 tanto para TCP como UDP:

[illegible]

## nlockmgr

`/etc/sysctl.d/swap-nfs-ports.conf`, e introducimos lo siguiente:

[illegible]

Lanzamos el archivo de configuración con `sudo sysctl --system/etc/init.d/nfs-kernel-server restart`

```

patchispatch@nfs:~$ sudo sysctl --system /etc/init.d/nfs-kernel-server restart
* 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
fs.nfs.nlm_udpport = 2002
* Applying /etc/sysctl.conf ...
patchispatch@nfs:~$

```

Nuestro script de IPTABLES quedaría así:

```

11 #!/bin/bash
10
9 # Denegación implícita
8 iptables -P INPUT DROP
7 iptables -P OUTPUT ACCEPT
6 iptables -P FORWARD DROP
5 iptables -A INPUT -m state --state ESTABLISHED, RELATED -j ACCEPT
4
3 # Puertos NFS
2 iptables -A INPUT -s 192.168.56.103, 192.168.56.101 -p tcp -m multiport --ports 111,
2000, 2001, 2049 -j ACCEPT
1
12 iptables -A INPUT -s 192.168.56.103, 192.168.56.101 -p udp -m multiport --ports 111,
2000, 2002, 2049 -j ACCEPT

```

Ejecutamos el script:

```

patchispatch@nfs:~$ sudo ./iptables.sh
patchispatch@nfs:~$

```

Y comprobamos que todo funciona añadiendo un nuevo archivo desde M2 esta vez:

```

patchispatch@m2:~$ touch datos/prueba2.md
patchispatch@m2:~$ _

```

```
patchispatch@m1:~/datos$ ls -l
total 0
-rw-rw-r-- 1 patchispatch patchispatch 0 may 17 19:22 prueba2.md
-rw-rw-r-- 1 patchispatch patchispatch 0 may 17 18:35 prueba.md
patchispatch@m1:~/datos$ _
```