

21 de mayo de 2020

## PRÁCTICA 6. SERVIDOR DE DISCO NFS

### I. CONFIGURAR MÁQUINA COMO SERVIDOR NFS

Creamos una máquina nfs como hicimos en la práctica 1. Instalamos los paquetes necesarios para NFS, como se indica a continuación.

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

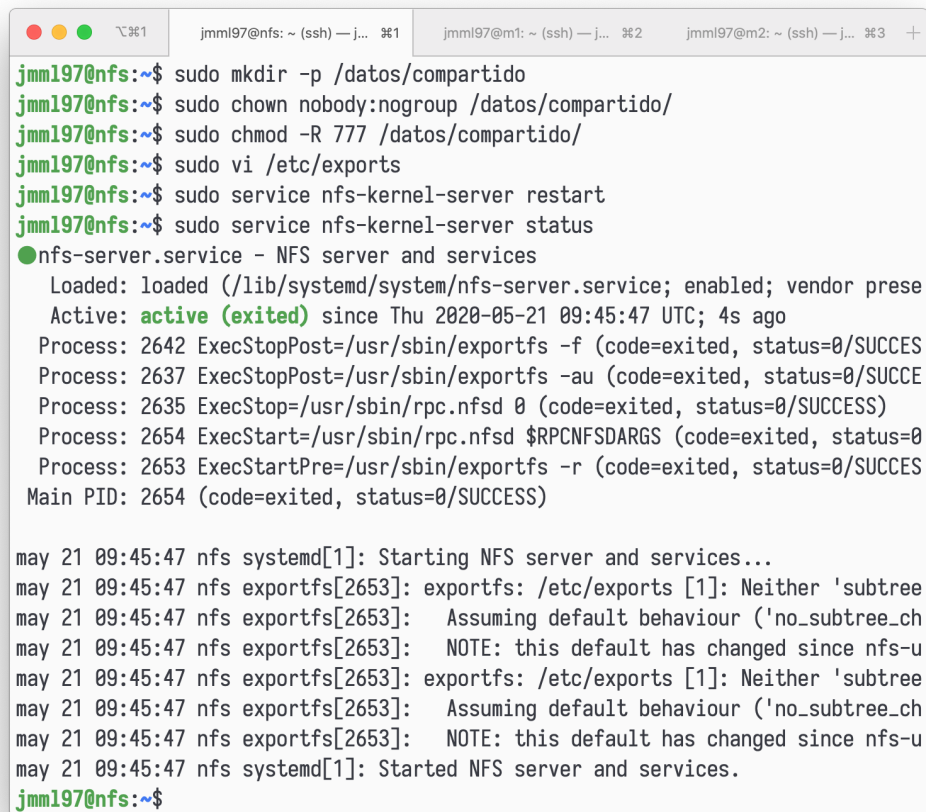
Creamos el directorio y asignamos permisos:

```
sudo mkdir -p /datos/compartido  
sudo chown nobody:nogroup /datos/compartido/  
sudo chmod -R 777 /datos/compartido/
```

Modificamos el archivo `/etc/exports/` para que las máquinas m1 y m2 puedan escribir:

```
/datos/compartido/ 192.168.56.103(rw) 192.168.56.102(rw)
```

Reiniciamos el servicio y comprobamos que todo funciona correctamente.



```
jmm197@nfs:~$ sudo mkdir -p /datos/compartido
jmm197@nfs:~$ sudo chown nobody:nogroup /datos/compartido/
jmm197@nfs:~$ sudo chmod -R 777 /datos/compartido/
jmm197@nfs:~$ sudo vi /etc/exports
jmm197@nfs:~$ sudo service nfs-kernel-server restart
jmm197@nfs:~$ sudo service nfs-kernel-server status
●nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor prese
   Active: active (exited) since Thu 2020-05-21 09:45:47 UTC; 4s ago
   Process: 2642 ExecStopPost=/usr/sbin/exportfs -f (code=exited, status=0/SUCCE
   Process: 2637 ExecStopPost=/usr/sbin/exportfs -au (code=exited, status=0/SUCCE
   Process: 2635 ExecStop=/usr/sbin/rpc.nfsd 0 (code=exited, status=0/SUCCESS)
   Process: 2654 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0
   Process: 2653 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCE
   Main PID: 2654 (code=exited, status=0/SUCCESS)

may 21 09:45:47 nfs systemd[1]: Starting NFS server and services...
may 21 09:45:47 nfs exportfs[2653]: exportfs: /etc/exports [1]: Neither 'subtree
may 21 09:45:47 nfs exportfs[2653]: Assuming default behaviour ('no_subtree_ch
may 21 09:45:47 nfs exportfs[2653]: NOTE: this default has changed since nfs-u
may 21 09:45:47 nfs exportfs[2653]: exportfs: /etc/exports [1]: Neither 'subtree
may 21 09:45:47 nfs exportfs[2653]: Assuming default behaviour ('no_subtree_ch
may 21 09:45:47 nfs exportfs[2653]: NOTE: this default has changed since nfs-u
may 21 09:45:47 nfs systemd[1]: Started NFS server and services.
jmm197@nfs:~$
```

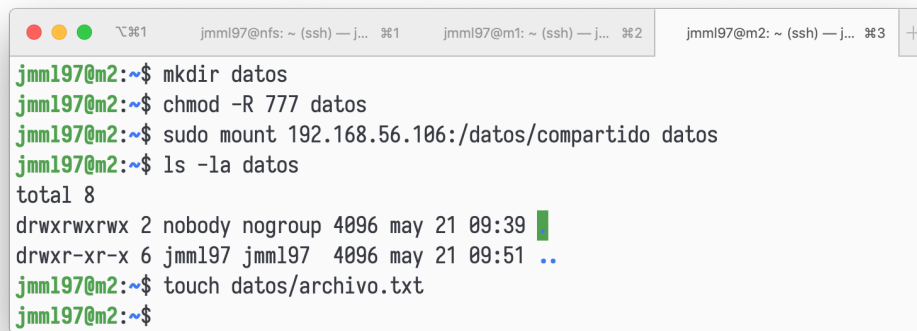
FIGURA 1: Estado correcto de NFS

## 2. CONFIGURACIÓN DE LOS CLIENTES M<sub>1</sub> Y M<sub>2</sub>

Tenemos que instalar el cliente NFS, crear una carpeta y enlazarla, en ambas máquinas.

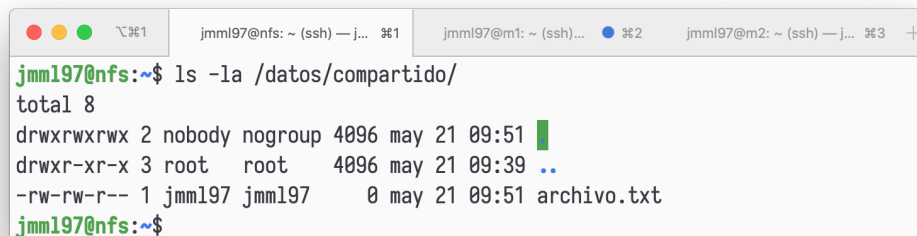
```
sudo apt-get install nfs-common rpcbind
mkdir datos
chmod -R 777 datos
sudo mount 192.168.56.106:/datos/compartido datos
```

Podemos comprobar ahora que si añadimos un archivo desde una máquina, digamos m<sub>2</sub>, aparecerá en nfs. Lo vemos en las siguientes imágenes.

A terminal window with three tabs. The active tab is 'jmm197@m2: ~ (ssh) — j... %3'. The terminal shows the following commands and output:

```
jmm197@m2:~$ mkdir datos
jmm197@m2:~$ chmod -R 777 datos
jmm197@m2:~$ sudo mount 192.168.56.106:/datos/compartido datos
jmm197@m2:~$ ls -la datos
total 8
drwxrwxrwx 2 nobody nogroup 4096 may 21 09:39 .
drwxr-xr-x 6 jmm197 jmm197 4096 may 21 09:51 ..
jmm197@m2:~$ touch datos/archivo.txt
jmm197@m2:~$
```

FIGURA 2: Configuración de NFS en m2 y creación de archivo

A terminal window with three tabs. The active tab is 'jmm197@nfs: ~ (ssh) — j... %1'. The terminal shows the following command and output:

```
jmm197@nfs:~$ ls -la /datos/compartido/
total 8
drwxrwxrwx 2 nobody nogroup 4096 may 21 09:51 .
drwxr-xr-x 3 root root 4096 may 21 09:39 ..
-rw-rw-r-- 1 jmm197 jmm197 0 may 21 09:51 archivo.txt
jmm197@nfs:~$
```

FIGURA 3: Comprobación de que el archivo se ha creado

Finalmente, para que la carpeta se monte con el arranque, editamos el archivo `fstab` en ambas máquinas, añadiendo:

```
192.168.56.106:/datos/compartido /home/jmm197/datos/ nfs auto,noatime,
nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```

```
jmm197@m2:~$ sudo reboot
Connection to 192.168.56.102 closed by remote host.
Connection to 192.168.56.102 closed.

~ took 14m 36s
→ ssh jmm197@192.168.56.102
jmm197@192.168.56.102's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-101-generic x86_64)

* 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

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

https://microk8s.io/

Pueden actualizarse 36 paquetes.
0 actualizaciones son de seguridad.

Last login: Thu May 21 09:46:38 2020 from 192.168.56.1
jmm197@m2:~$ ls -la datos
total 8
drwxrwxrwx 2 nobody nogroup 4096 may 21 09:51 
drwxr-xr-x 6 jmm197 jmm197 4096 may 21 10:01 ..
-rw-rw-r-- 1 jmm197 jmm197    0 may 21 09:51 archivo.txt
jmm197@m2:~$
```

FIGURA 4: Carpeta montada tras reinicio

### 3. SEGURIDAD EN EL SERVIDOR NFS

Aplicamos la configuración de denegación por defecto del tráfico entrante:

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

Modificamos los puertos de mountd y nlockmgr. Editamos el archivo `/etc/default/nfs-kernel-server`, cambiando la siguiente línea:

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

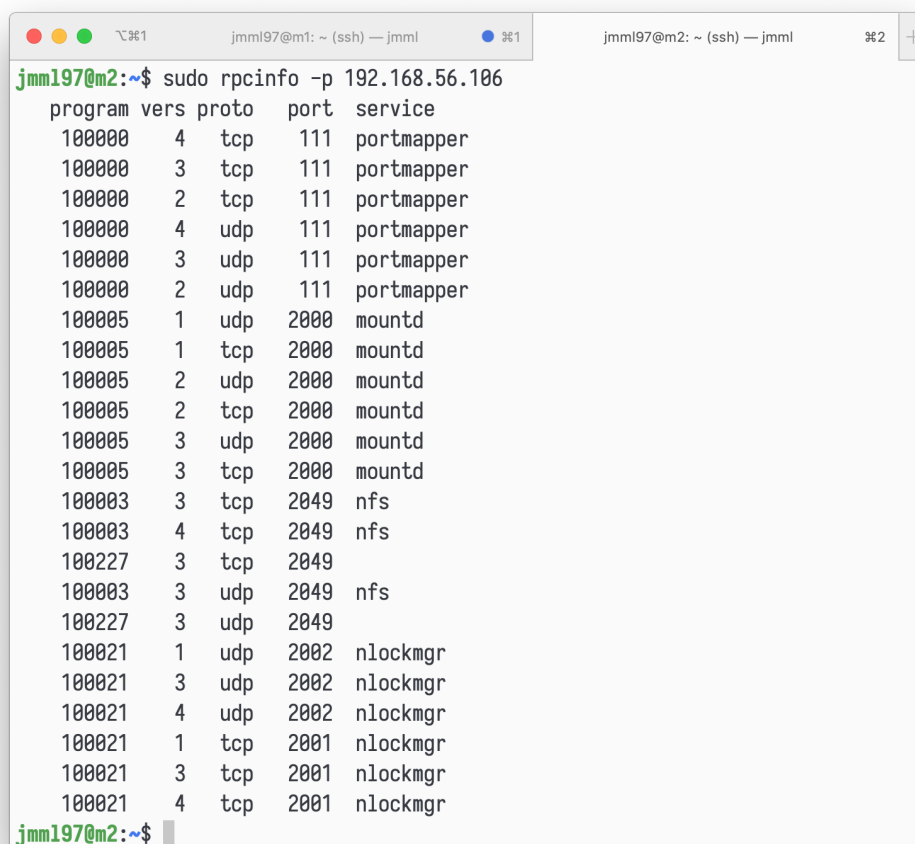
Creamos el archivo `swap-nfs-ports.conf` en `/etc/sysctl.d/`, añadiendo:

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

Aplicamos la configuración y reiniciamos.

```
sudo sysctl --system
/etc/init.d/nfs-kernel-server restart
```

Podemos comprobar los puertos con `rpcinfo`, como vemos en la siguiente imagen. (Por algún motivo me daba `connection timed out` desde la propia máquina, así que lo comprobé desde m2.)



```
jmm197@m2:~$ sudo rpcinfo -p 192.168.56.106
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
```

FIGURA 5: Puertos en la máquina nfs

Añadimos permisos a iptables en los puertos que necesitamos para NFS:

```
iptables -A INPUT -s 192.168.56.103,192.168.56.102 -p tcp -m
multiport --ports 111,2000,2001,2049 -j ACCEPT
iptables -A INPUT -s 192.168.56.103,192.168.56.102 -p udp -m
multiport --ports 111,2000,2002,2049 -j ACCEPT
```

En las siguientes imágenes comprobamos que todo funciona correctamente.

```
jmm197@nfs:~$ sudo iptables -A INPUT -s 192.168.56.103,192.168.56.102 -p tcp -m multiport --ports 11
1,2000,2001,2049 -j ACCEPT
jmm197@nfs:~$ sudo iptables -A INPUT -s 192.168.56.103,192.168.56.102 -p udp -m multiport --ports 11
1,2000,2001,2049 -j ACCEPT
jmm197@nfs:~$ touch /datos/compartido/archivo2.txt
jmm197@nfs:~$
```

FIGURA 6: Reglas iptables y creación de archivo

```
jmm197@m2:~$ ls -la datos
total 8
drwxrwxrwx 2 nobody nogroup 4096 may 21 10:19 
drwxr-xr-x 6 jmm197 jmm197 4096 may 21 10:01 ..
-rw-rw-r-- 1 jmm197 jmm197 0 may 21 10:19 archivo2.txt
-rw-rw-r-- 1 jmm197 jmm197 0 may 21 09:51 archivo.txt
jmm197@m2:~$
```

FIGURA 7: Comprobación en m2 del archivo creado