



NFS

IES GONZALO NAZARENO

Que es nfs...

El sistema de archivos de red (*Network File System*, NFS) es una aplicación cliente/servidor que permite a un usuario de equipo ver y, opcionalmente, almacenar y actualizar archivos en un equipo remoto como si estuvieran en el propio equipo del usuario. El protocolo NFS es uno de varios estándares de sistema de archivos distribuidos para almacenamiento atado a la red (NAS).

Instalación en la parte de servidor, Debian

empezamos instalando el nfs-kernel-server y el nfs-common

apt install nfs-kernel-server

```
Configurando rpcbind (1.2.5-9) ...
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /lib/systemd/system/rpcbind.socket.
Configurando libevent-2.1-7:amd64 (2.1.12-stable-1) ...
Configurando keyutils (1.6.1-2) ...
Configurando libnfsidmap2:amd64 (0.25-6) ...
Configurando nfs-common (1:1.3.4-6) ...

Creating config file /etc/idmapd.conf with new version
Añadiendo el usuario del sistema 'statd' (UID 107) ...
Añadiendo un nuevo usuario 'statd' (UID 107) con grupo 'nogroup' ...
No se crea el directorio personal '/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
nfs-utils.service is a disabled or a static unit, not starting it.
Configurando nfs-kernel-server (1:1.3.4-6) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /lib/systemd/system/nfs-server.service.
Job for nfs-server.service canceled.

Creating config file /etc/exports with new version

Creating config file /etc/default/nfs-kernel-server with new version
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para libc-bin (2.31-13+deb11u3) ...
root@debian:~#
```

apt install nfs-common

```
root@debian:~# apt install nfs-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
nfs-common ya está en su versión más reciente (1:1.3.4-6).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@debian:~#
```

Iniciamos el servicio del servidor NFS y habilitarlo para el inicio del sistema con rpcbind

El servicio `rpcbind` asigna los servicios de llamada a procedimiento remoto (RPC) a los puertos en los que escuchan. Los procesos RPC notifican a `rpcbind` cuando se inician, registrando los puertos en los que escuchan y los números de programa RPC

`systemctl start rpcbind nfs-server`

`systemctl enable rpcbind nfs-server`

```
root@debian:~# systemctl start rpcbind nfs-server
root@debian:~# systemctl enable rpcbind nfs-server
Synchronizing state of rpcbind.service with SysV service script with /lib/systemd/systemd-sysv-inst
Executing: /lib/systemd/systemd-sysv-install enable rpcbind
root@debian:~#
```

comprobamos status del servicio del servidor NFS

```
root@debian:~# 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 Mon 2022-05-16 13:35:19 CEST; 12min ago
     Main PID: 1118 (code=exited, status=0/SUCCESS)
       Tasks: 0 (limit: 2341)
      Memory: 0B
         CPU: 0
      CGroup: /system.slice/nfs-server.service

may 16 13:35:18 debian systemd[1]: Starting NFS server and services...
may 16 13:35:19 debian systemd[1]: Finished NFS server and services.
root@debian:~#
```

Con `rpcinfo -p` enumeramos los servicios activos y decirle al cliente solicitante dónde enviar la solicitud RPC. Si un host escucha en el puerto 111, se puede usar `rpcinfo` para obtener números de programas y puertos y servicios en ejecución

Aquí podemos ver los puertos y el protocolo usado en el puerto 111 que es un puerto que suele ser muy atacado en las conexiones y como nfs usa el puerto 2049

```
root@debian:~# 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   48014 mountd
100005    1    tcp   57797 mountd
100005    2    udp   53167 mountd
100005    2    tcp   51157 mountd
100005    3    udp   53611 mountd
100005    3    tcp   46851 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   33068 nlockmgr
100021    3    udp   33068 nlockmgr
100021    4    udp   33068 nlockmgr
100021    1    tcp   43357 nlockmgr
100021    3    tcp   43357 nlockmgr
100021    4    tcp   43357 nlockmgr
root@debian:~#
```

Compartir/Exportar

por razones de seguridad usamos en una raiz la exportacion NFS sobre todo para que el usuario no se salga del punto de montaje y la acabemos liando

```
root@debian:~# mkdir /compartida_nfs
root@debian:~# mount -o bind /compartida_nfs/ /mnt/nfs/
root@debian:~# touch /compartida_nfs/jose
root@debian:~# ls -l /compartida_nfs/
total 0
-rw-r--r-- 1 root root 0 may 17 10:20 jose
root@debian:~# ls -l /mnt/nfs/
total 0
-rw-r--r-- 1 root root 0 may 17 10:20 jose
root@debian:~# mount -o bind /compartida_nfs /mnt/nfs_
```

Editamos ahora el fichero de exportación agregandole la ip que tenemos
nano /etc/exports

```
# 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)
#
/mnt/nfs 172.22.2.231/16(ro,all_squash)
```

el siguiente paso sera usar el comando `exportfs` con este comando podemos s permite a root exportar o no directorios concretos sin reiniciar los servicios NFS. Cuando se le pasan las opciones apropiadas a `exportfs`

para ello usamos

`exportfs -o ro,all_squash*:/mnt/nfs`

```
root@debian:~# exportfs -o ro,all_squash*:/mnt/nfs/  
/mnt/nfs      172.22.2.231  
/mnt/nfs      172.22.2.231/16  
root@debian:~#
```

Ahora llega el momento de aplicar los cambios realizados para ellos usamos

`exportfs -a`

`exportfs -v`

```
root@debian:~# exportfs -v  
/mnt/nfs      172.22.2.231(ro,wdelay,root_squash,all_squash,no_subtree_check,sec=sys,ro,secure,root_squash,all_squash)  
/mnt/nfs      172.22.2.231/16(ro,wdelay,root_squash,all_squash,no_subtree_check,sec=sys,ro,secure,root_squash,all_squash)  
root@debian:~# exportfs -a  
exportfs: /etc/exports [2]: Neither 'subtree_check' or 'no_subtree_check' specified for export "172.22.2.231/16:/mnt/nfs".  
    Assuming default behaviour ('no_subtree_check').  
    NOTE: this default has changed since nfs-utils version 1.0.x
```

Cliente NFS con Windows

En mi caso hice un ipconfig para comprobar la ip

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.1288]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erjos>ipconfig

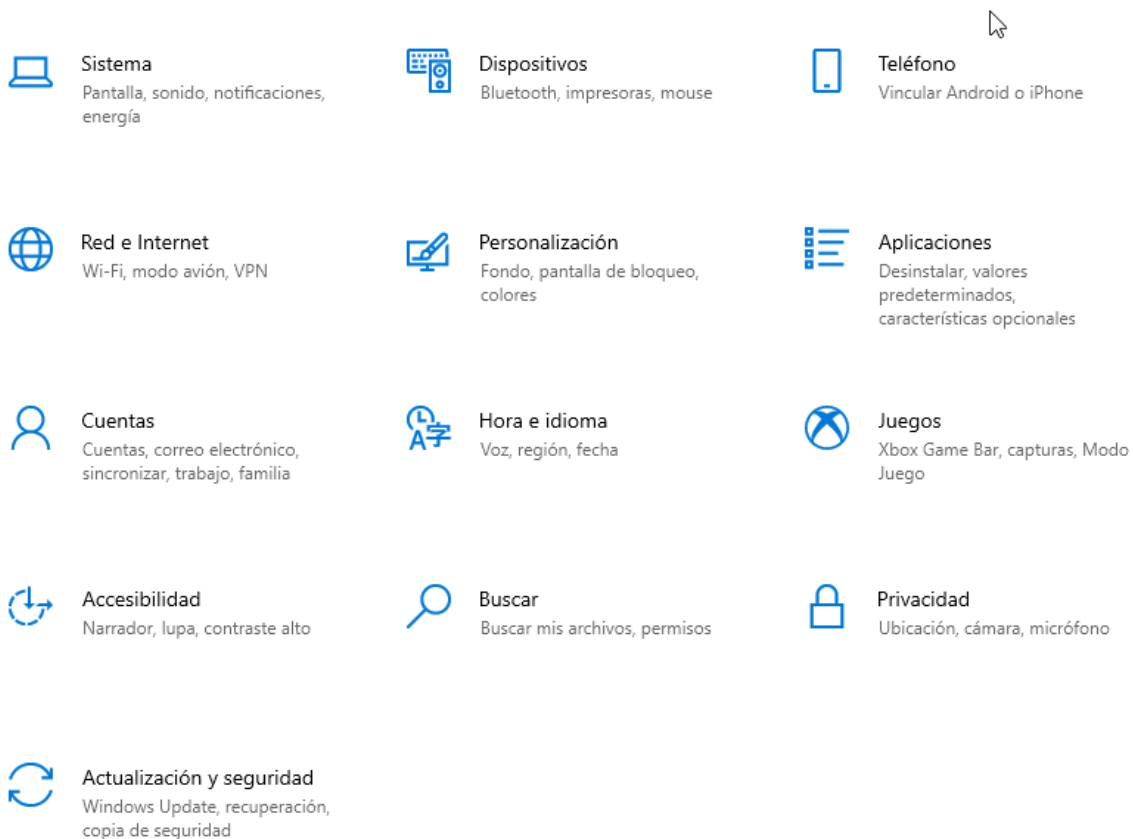
Configuración IP de Windows

Adaptador de Ethernet Instancia de Ethernet 0:

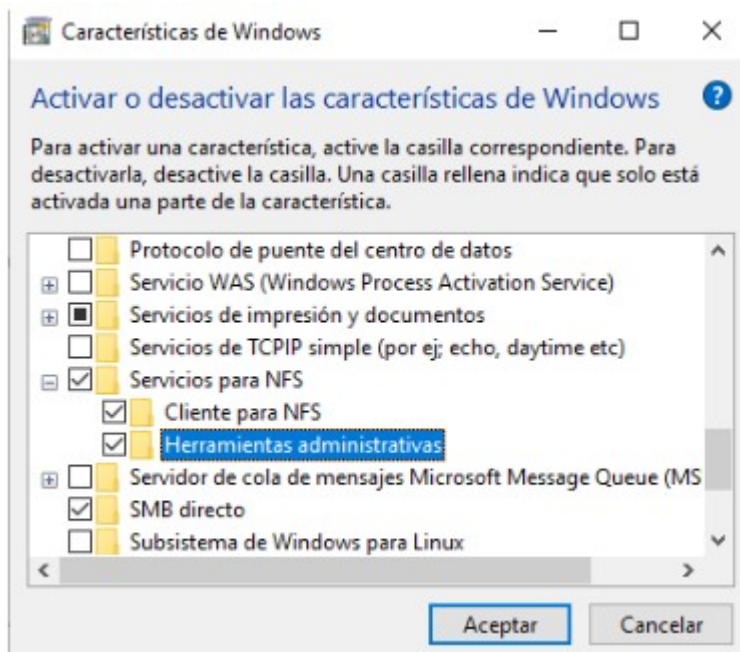
    Sufixo DNS específico para la conexión. . . : gonzalonazareno.org
    Vínculo: dirección IPv6 local. . . . . : fe80::f0f4:8fd7:e450:6b9e%6
    Dirección IPv4. . . . . : 172.22.8.130
    Máscara de subred . . . . . : 255.255.0.0
    Puerta de enlace predeterminada . . . . . : 172.22.0.1

C:\Users\erjos>
```

luego entre en sistema en el panel de control



y active el cliente y las herramientas de nfs



Despues con el comando compruebe que el servicio nfs estaba perfectamente en mi windows

```
mount -o anon
```

```
C:\Users\erjos>mount -o anon \\172.22.2.231\mnt\nfs p:
p: está conectado ahora correctamente a \\172.22.2.231\mnt\nfs

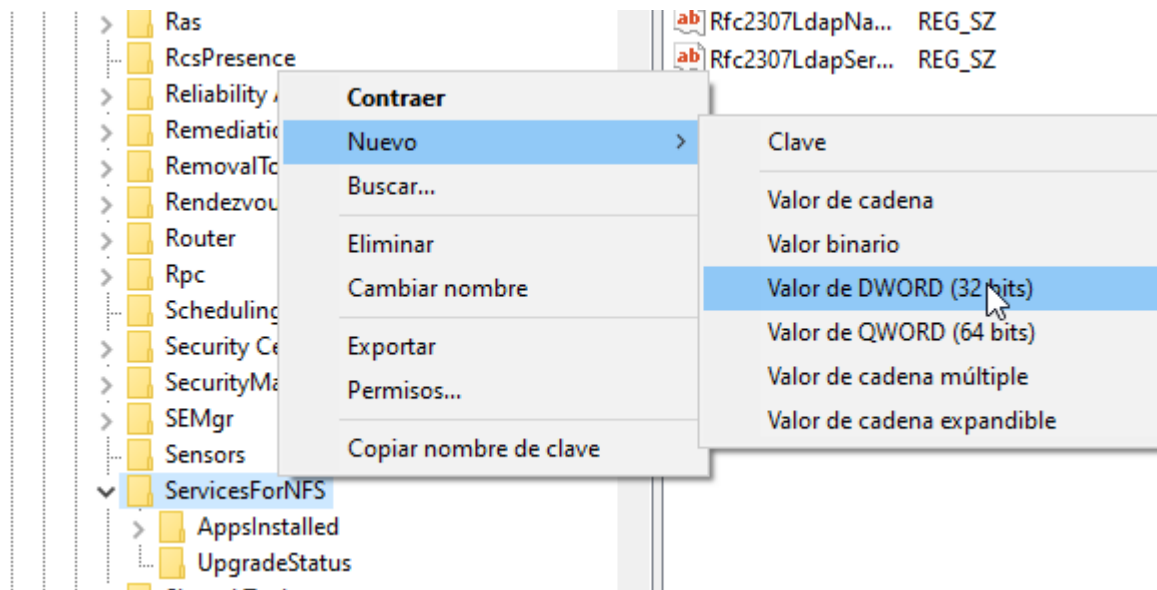
El comando se completó correctamente.

C:\Users\erjos>mount
```

Local	Remoto	Propiedades
p:	\\172.22.2.231\mnt\nfs	UID=-2, GID=-2 rsize=262144, wsize=262144 mount=soft, timeout=0.8 retry=1, locking=yes fileaccess=755, lang=ANSI casesensitive=no seg.=sys

```
C:\Users\erjos>
```


Volvi atrás para buscar en en el regedit o editor de registro y vemos quetenemos el servicio nfs



Cliente debian

como siempre instalamos nuestro paquete nfs-common

```
root@debian:~# apt install nfs-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
nfs-common ya está en su versión más reciente (1:1.3.4-6).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@debian:~#
```

con showmount -e nos Imprime una lista de los archivos compartidos o exportados.

```
root@debian:~# showmount -e 172.22.2.231
Export list for 172.22.2.231:
/mnt/nfs 172.22.2.231/16
root@debian:~#
```

Creamos ahora una carpeta y en ellas montamos con mount nosotros elegimos la forma manual

mount -t nfs -o options host:/remote/export /local/directory

mount -t nfs 172.22.2.231:/mnt servidorjose/

options

Una lista delimitada por comas de opciones de montaje.

host

El nombre de host, la dirección IP o el nombre de dominio completo del servidor que exporta el sistema de archivos que desea montar.

/remote/export

El sistema de archivos o el directorio que se exporta desde el servidor, es decir, el directorio que se desea montar.

/local/directory

La ubicación del cliente donde */remote/export* está montado.

```
root@debian:~# mkdir servidorjose
root@debian:~# mount -t nfs 172.22.2.231:/mnt/nfs servidorjose/
root@debian:~#
```

Ahora usamos un `df -h` que no imprime en pantalla información sobre los sistemas de archivos accesibles en forma de tabla. También es importante que nos muestre la tabla mirar la información que nos proporciona

```
root@debian:~# df -h
S.ficheros          Tamaño Usados  Disp Uso% Montado en
udev                976M      0  976M  0% /dev
tmpfs               199M    440K  198M  1% /run
/dev/sda1           31G    1,5G   28G  6% /
tmpfs               992M      0  992M  0% /dev/shm
tmpfs               5,0M      0   5,0M  0% /run/lock
tmpfs               199M      0  199M  0% /run/user/1000
172.22.2.231:/mnt/nfs 31G    1,5G   28G  6% /root/servidorjose
root@debian:~#
```

- **S.ficheros:** El nombre de lo archivo del sistema.
- **Bloques de 1k:** Indica el tamaño total de la unidad de almacenamiento.
- **Usado:** Muestra cuando espacio está siendo ocupado en cada archivo del sistema.
- **Disponible:** Muestra cuando espacio disponible queda para cada archivo del sistema.
- **Uso%:** Muestra el porcentaje del espacio que está siendo usado.
- **Montado en:** Nos muestra el directorio donde esta cada archivo del sistema.

Y para acabar hacemos un mount | egrep nfs para encontrar la información

```
root@debian:~# mount | egrep nfs
172.22.2.231:/mnt/nfs on /root/servidorjose type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=172.22.3.88,local_lock=none,addr=172.22.2.231)
root@debian:~#
```

Conclusión

Pues he aprendido muchas cosas en esta practica sobre todo a usar siempre el su – (perdona rafa por el error) la verdad es que me ha costado un poco hacerla porque no la acababa de entender pero bueno mas o menos creo que ya la entiendo

fuentes consultadas

<https://denovatoanovato.net/comando-df/>

<https://www.computerweekly.com/es/definicion/Sistema-de-archivos-de-red-NFS>

<https://wikimho.com/es/q/security/80799>

https://dit.gonzalonazareno.org/moodle/pluginfile.php/40374/mod_resource/content/1/nfs.pdf