

REDES:

## CONFIGURACIÓN DE NFS

[UNIVERSIDAD TECNOLÓGICA NACIONAL]



[FACULTAD REGIONAL DELTA]

# CONFIGURACIÓN DE NFS



## ÍNDICE

|                                                              |   |
|--------------------------------------------------------------|---|
| Servicio NFS (Net File System)                               | 3 |
| Cliente NFS                                                  | 6 |
| Utilizar autofs para montar un directorio compartido vía NFS | 8 |

# CONFIGURACIÓN DE NFS

## Servicio NFS (Net File System)



NFS es un acrónimo de **Network File System** (**Sistemas de Archivos en Red**). Se trata de un protocolo a nivel de aplicación para el compartimiento recursos en red, permitiendo que el sistema operativo cliente los vea como propios.

Viene incluido en todos los sistemas UNIX y la mayoría de distribuciones GNU/Linux.



NFS utiliza las listas de control de acceso (ACL) y DNS para su seguridad. Tenemos que tener especial cuidado en la configuración, ya que **no posee un sistema de seguridad por usuarios**.

Algo común en GNU/Linux es que los servicios, en este caso NFS, se compongan de una serie de Daemons. O sea, en lugar de tener un solo proceso Daemon que controle la totalidad del servicio, se tiene una serie de daemons donde cada uno realiza una tarea específica. En el caso de NFS son:

- **autofs**: monta automáticamente directorios bajo demanda. O sea, si ingresamos a un directorio configurado con automount, lo va a montar automáticamente. Y lo desmonta luego de unos minutos si ese directorio no fue vuelto a acceder.
- **automountd**: este daemont maneja los requests de mount y unmount del servicio autofs.
- **lockd**: este daemon se encarga de las operaciones de lock por escritura. O sea, para que un archivo no sea escrito por dos usuarios al mismo tiempo.
- **mountd**: este es un RPC que maneja los requests de montado de filesystems remotos. A diferencia de automountd, no es automático. O sea, es para los requests de montados manuales.
- **nfsd**: maneja otros requests de parte del cliente nfs.
- **statd**: junto con lockd permite recuperarse de caídas.

Para poder usar nfs es necesario que instalemos los siguientes paquetes:

- **nfs-kernel-server**, habilita el soporte NFS para nuestro kernel
- **nfs-common**, conjunto de herramientas cliente y servidor para el funcionamiento de NFS.
- **autofs**, necesario si queremos configurar que algunos de nuestros directorios se monten bajo demanda.

```
# apt-get -y update
# apt-get install nfs-kernel-server nfs-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libgssglue1 libnfsidmap2 libtirpc1 nfs-common rpcbind
Paquetes sugeridos:
  open-iscsi watchdog
Se instalarán los siguientes paquetes NUEVOS:
  libgssglue1 libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server
  rpcbind
0 actualizados, 6 se instalarán, 0 para eliminar y 33 no actualizados.
Necesito descargar 417 kB de archivos.
Se utilizarán 1.816 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Una vez instalado, para poder compartir tenemos que editar el archivo `/etc/exports`, que es donde indicaremos los recursos compartidos, la configuración de la red y los permisos.



Cada recurso ocupa una línea, en la primera parte indicamos la ubicación del recurso; en la segunda, los datos de red (IP y máscara) de los clientes (hosts); por último, los permisos otorgados al recurso.

En la carpeta /mnt/datos, tenemos la subcarpeta aula1, que vamos a compartir en sólo lectura, para que los usuarios puedan verlos pero no modificarlos ni crear nuevos.

```
# /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) hostna-
# me2(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/datos/aula1      10.0.0.0/8(ro,no_subtree_check,sync)
```

En el ejemplo podemos ver las opciones de ejemplo que el servicio trae configuradas por default y en nuestra configuración podemos ver que la primera columna contiene el directorio que vamos a exportar (compartir) y, en la segunda columna, desde que número IP o red y con qué permisos se exporta el directorio, veamos abajo que significa cada parámetro en la configuración:

- **ro/rw**: indica que el recurso compartido sera de solo lectura, en otro caso podemos configurarlo con rw para dejarlo en modo escritura.
- **async/sync**: Con la opcion sync no se responden a las peticiones antes de que los cambios realizados sean escritos al disco. Con la opción async se permite mejorar el rendimiento y agilizar el funcionamiento global, pero supone un riesgo de corrupción de archivos o del sistema de archivos en casos de caídas del servidor y/o errores de éste.
- **root\_squash/no\_root\_squash/all\_squash**: root\_squash indica que si el cliente se identifica como root tendrá acceso al directorio con privilegios de un usuario anónimo. Si seleccionamos la opción no\_root\_squash evitaremos esto y tendremos que tener mucho cuidado ya que si se identifica como root, será root dentro del filesystem montado.



Si indicamos `all_squash`, entonces aplicaremos esto último a todos los usuarios, no sólo `root`.

· `no_subtree_check/subtree_check`: `subtree_check` es la encargada de comprobar que los directorios exportados están en un mismo volumen, `no_subtree_check` no hace esa comprobación, por lo que la transferencia de la lista de archivos es más rápida.

Solo nos falta levantar el servicio de NFS:

```
# /etc/init.d/nfs-kernel-server start
```

o bien:

```
# systemctl start nfs-kernel-server
```

## Cliente NFS

En las máquinas que van a trabajar como clientes tenemos que instalar el paquete `nfs-common`:

```
# apt-get install nfs-common
```

Luego vamos a comprobar si el directorio en el server esta exportado:

```
# showmount --exports 10.10.3.1
```

```
Export list for 10.10.3.1:
```

```
/mnt/datos/aula1 *
```



Lo que vemos aquí es que dentro del servidor **10.10.3.1**, que es el que configuramos anteriormente, el mismo tiene exportado el recurso `/mnt/datos/aula1` para cualquier IP de origen.

Ahora estamos en condiciones de montar el directorio exportado:

**01** Primero creamos el punto de montaje, en nuestro caso lo vamos a hacer en el directorio /mnt el punto de montaje se llama aula1.

**02** Luego usamos el comando mount para montar el dispositivo exportado.

```
# cd /mnt  
# mkdir aula1  
# mount -t nfs 10.10.3.1:/mnt/datos/aula1 /mnt/aula1
```

Los parámetros de mount son:

-t: tipo de filesystem en nuestro caso nfs-common

Luego tenemos la dirección de red del servidor que exporta y luego el punto de montaje local, el punto de montaje tiene que haber sido creado previamente.

**03** Si dentro del servidor creamos un archivo en el export:

```
# touch /mnt/datos/aula1
```

Lo veremos dentro de la carpeta que utilizamos como punto de montaje en el cliente:

```
# cd /mnt/aula1  
# ls
```

## Utilizar autofs para montar un directorio compartido vía NFS



**Autofs** nos permite montar sistemas de archivos locales y remotos bajo demanda y desmontarlos automáticamente cuando no se usen.

¿Qué quiero decir con bajo demanda? Simplemente que los sistemas de archivos se van a montar cuando el usuario acceda a ellos y se desmontarán después de un tiempo de inactividad.

Supongamos que tenemos un servidor en la máquina con IP 10.10.3.30 que exporta un directorio /mnt/aula1 mediante NFS, y queremos que este directorio se encuentre disponible para las máquinas de nuestra red local tal cual lo hicimos en el punto anterior.

Ahora vamos a dar un paso más, vamos a ver como automontar este directorio en un cliente sin necesidad de hacerlo usando el comando mount.

Primero, instalamos autofs en el cliente:

```
apt-get install autofs
```

Este servicio crea el archivo de configuración por defecto llamado auto.master.

Una vez instalado, editamos el archivo /etc/auto.master agregamos la siguiente línea:

```
/datos /etc/auto.aula1 --timeout=60
```



Esta línea le informa al servicio que si alguien se posiciona en el directorio **/datos**, entonces el programa **autofs** tiene que montar vía nfs el directorio contenido en el archivo **auto.aula1**.



Ahora creamos el archivo `/etc/auto.aula1` con el siguiente contenido:

```
aula1 -rw,soft,intr,rsize=8192,wsiz=8192,tcp 10.10.3.30:/mnt/aula1
```

Los parámetros son:

```
aula1: punto de montaje
-rw,soft,intr,rsize=8192,wsiz=8192,tcp: son las opciones de montaje.
10.10.3.30:/mnt/aula1: directorio exportado por el servidor.
```



De este modo, cuando el usuario acceda a `/datos/aula1`, se realizará automáticamente el montaje, es importante entender que en el cliente no tienen que existir los puntos de montaje, estos van a crearse en el momento de acceso.

Reiniciamos en el cliente, el autofs:

```
# /etc/init.d/autofs restart
```

E ingresamos directamente, por ejemplo, listando los archivos:

```
# ls /datos/aula1
```

Y como podemos observar, el recurso se montó automáticamente, sin necesidad de ejecutar ningún `mount`:

```
#mount | grep datos
```



### Conclusiones finales:

NFS es un servicio muy usado es muy rápido y seguro a la hora de trabajar. Lo usamos mucho en la industria porque es uno de los filesystems usados en storage cuando trabajamos con clusters de servidores. Es muy sencillo de configurar, por lo tanto les recomiendo usarlo y aprenderlo bien.