

Configuración de servicios CIFS y NFS

- Montaje de al menos un recurso compartido por cada protocolo (uno para CIFS y otro para NFS).

Protocolo CIFS

Creamos las carpetas de los recursos a compartir

```
sudo mkdir -p /srv/nuevo_cifs_publico
```

```
sudo chown nobody:nogroup /srv/nuevo_cifs_publico
```

```
sudo chmod 777 /srv/nuevo_cifs_publico
```

Montamos el recurso compartido en samba llamado CIFS_PublicoNuevo que apunta al path /srv/nuevo_cifs_publico.

Editamos en /etc/samba/smb.conf donde agregamos los datos del recurso a compartir

```
joel@tres: ~
GNU nano 7.2 /etc/samba/smb.conf
[global]
workgroup = WORKGROUP
server string = Servidor Samba
security = user
map to guest = Bad User
guest account = nobody
log file = /var/log/samba/log.%m
max log size = 1000
load printers = no
disable spoolss = yes

[CIFS_PublicoNuevo]
path = /srv/nuevo_cifs_publico
public = yes
writable = yes
guest ok = yes
browseable = yes
create mask = 0666
directory mask = 0777

[CompartidoCIFS]
path = /srv/compartido_cifs
valid users = cifsuser
read only = no
browseable = yes
create mask = 0660
```

En la sección global nos aseguramos de tener esta información, ya que este comando nos permitirá que los usuarios no autenticados se conecten como invitados

```
joel@tres: ~
GNU nano 7.2
[global]
workgroup = WORKGROUP
server string = Servidor Samba
security = user
map to guest = Bad User
guest account = nobody
log file = /var/log/samba/log.%m
max log size = 1000
load printers = no
disable spoolss = yes
```

Reiniciamos el servicio samba con `sudo systemctl restart smbd`

Verificamos que el recurso esta compartido correctamente

```
joel@tres:~$ testparm -s | grep -A5 CIFS_PublicoNuevo
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed by GnuTLS (e.g. NTLM as a compatibility fallback)

Server role: ROLE_STANDALONE

[CIFS_PublicoNuevo]
    create mask = 0666
    directory mask = 0777
    guest ok = Yes
    path = /srv/nuevo_cifs_publico
    read only = No
```

Protocolo NFS

Creamos el directorio con `sudo mkdir -p /srv/nuevo_nfs_publico`

Asignamos al grupo con `sudo chown nobody:nogroup /srv/nuevo_nfs_publico`

Damos permisos con `sudo chmod 777 /srv/nuevo_nfs_publico`

Configuramos la exportación en `/etc/exports`:

`/srv/nuevo_nfs_publico`

`192.168.40.0/24(rw,sync,no_subtree_check,all_squash,anonuid=65534,anongid=65534)`

Aplicamos la configuracion con `sudo exportfs -ra` y reiniciamos con `sudo systemctl restart nfs-server`

Verificamos que esta correctamente compartido con:

```
joel@tres:~$ sudo exportfs -v
[sudo] password for joel:
/srv/compartido_nfs
    192.168.40.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
/srv/nfs_analistas
    192.168.40.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
/srv/nfs_invitados
    192.168.40.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,ro,secure,root_squash,no_all_squash)
/srv/nfs/compartido
    192.168.40.0/24(sync,wdelay,hide,no_subtree_check,anonuid=3001,anongid=2002,sec=sys,rw,secure,root_squash,no_all_squash)
/srv/nuevo_nfs_publico
    192.168.40.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,all_squash)
/srv/nuevo_nfs_privado
    192.168.40.0/24(sync,wdelay,hide,no_subtree_check,anonuid=4001,anongid=4001,sec=sys,rw,secure,root_squash,no_all_squash)
```

- Creación de usuarios/grupos y permisos adecuados.

Vamos a crear dos usuarios, `analista1` con permisos de lectura y escritura. Y el usuario `invitado1` con permisos de solo lectura.

Creamos los usuarios en el sistema y en **samba**:

```
joel@tres:~$ sudo useradd -M -s /sbin/nologin analista1
joel@tres:~$ sudo smbpasswd -a analista1
New SMB password:
Retype new SMB password:
Added user analista1.
joel@tres:~$ sudo useradd -M -s /sbin/nologin invitado1
joel@tres:~$ sudo smbpasswd -a invitado1
New SMB password:
Retype new SMB password:
Added user invitado1.
```

Creamos las carpetas por rol:

```
joel@tres:~$ sudo mkdir -p /srv/cifs_analistas
joel@tres:~$ sudo mkdir -p /srv/cifs_invitados
joel@tres:~$
```

Asignamos permisos:

```
joel@tres:~$ sudo chown analista1 /srv/cifs_analistas
joel@tres:~$ sudo chmod 770 /srv/cifs_analistas
joel@tres:~$ sudo chown invitado1 /srv/cifs_invitados
joel@tres:~$ sudo chmod 750 /srv/cifs_invitados
```

Vamos a la configuración de samba con el comando `sudo nano /etc/samba/smb.conf` y agregamos los recursos:

```
joel@tres: ~
GNU nano 7.2 /etc/samba/smb.conf *
max log size = 1000
logging = file
load printers = no
printing = bsd
disable spoolss = yes

[CompartidoCIFS]
path = /srv/compartido_cifs
valid users = cifsuser
read only = no
browseable = yes
create mask = 0660
directory mask = 0770

[CIFS_Analistas]
path = /srv/cifs_analistas
valid users = analista1
read only = no
browseable = yes

[CIFS_Invitados]
path = /srv/cifs_invitados
valid users = invitado1
read only = yes
browseable = yes
```

Guardamos los cambios y reiniciamos con `sudo systemctl restart smbd`

- En NFS, usa permisos basados en UID/GID.

Como ya tenemos a los usuarios creados vamos a verificar sus UID/GID con los comandos:

```
joel@tres:~$ id analista1
uid=2001(analista1) gid=2001(analista1) groups=2001(analista1)
joel@tres:~$ id invitado1
uid=1003(invitado1) gid=1003(invitado1) groups=1003(invitado1)
```

Creamos las carpetas a compartir con:

```
joel@tres:~$ sudo mkdir -p /srv/nfs_analistas
joel@tres:~$ sudo mkdir -p /srv/nfs_invitados
```

Asignamos propiedades a cada carpeta:

```
joel@tres:~$ sudo chown analista1:analista1 /srv/nfs_analistas
joel@tres:~$ sudo chmod 770 /srv/nfs_analistas
joel@tres:~$ sudo chown invitado1:invitado1 /srv/nfs_invitados
joel@tres:~$ sudo chmod 755 /srv/nfs_invitados
```

Exportamos los recursos a /etc/exports

```
joel@tres: ~
GNU nano 7.2 /etc/exports *
# /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)
#
/srv/compartido_nfs 192.168.40.0/24(rw,sync,no_subtree_check)
/srv/nfs_analistas 192.168.40.0/24(rw,sync,no_subtree_check)
/srv/nfs_invitados 192.168.40.0/24(ro,sync,no_subtree_check)
.
```

Guardamos y aplicamos los cambios con:

```
sudo exportfs -ra
```

```
sudo systemctl restart nfs-kernel-server
```

En el cliente agregamos los mismos usuarios y grupos incluyendo los mismos UID/GID

```
joel@cuatro:~$ sudo groupadd -g 2001 analistas
joel@cuatro:~$ sudo groupadd -g 1003 invitados
joel@cuatro:~$ sudo useradd -m -u 2001 -g analistas -s /bin/bash analista1
joel@cuatro:~$ sudo useradd -m -u 1003 -g invitados -s /bin/bash invitado1
```

Creamos puntos de montaje con:

```
joel@cuatro:~$ sudo mkdir -p /mnt/analistas
joel@cuatro:~$ sudo mkdir -p /mnt/invitados
joel@cuatro:~$
```

Montamos recursos como invitado1

```
joel@cuatro:~$ sudo mount 192.168.40.183:/srv/nfs_invitados /mnt/invitados
```

Probamos acceso como invitado1 donde nos indica que solo podemos leer y no escribir.

```
joel@cuatro:~$ sudo umount /mnt/invitados
joel@cuatro:~$ sudo mount 192.168.40.183:/srv/nfs_invitados /mnt/invitados
joel@cuatro:~$ sudo -u invitado1 ls /mnt/invitados
joel@cuatro:~$ sudo -u invitado1 touch /mnt/invitados/prueba.txt
touch: cannot touch '/mnt/invitados/prueba.txt': Read-only file system
```

Montamos recursos como analista1

```
joel@cuatro:~$ sudo mount 192.168.40.183:/srv/nfs_analistas /mnt/analistas
joel@cuatro:~$
```

Probamos acceso escribiendo el archivo ok.txt:

```
joel@cuatro:~$ sudo -u analista1 touch /mnt/analistas/ok.txt
joel@cuatro:~$ ls -l /mnt/analistas
```

Verificamos el contenido del directorio y vemos el archivo:

```
joel@cuatro:~$ sudo -u analista1 ls /mnt/analistas
archivo.txt ok.txt
```

- Verificación de accesos desde distintos clientes (Linux y/o Windows).

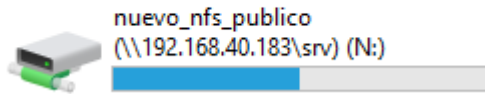
Accedemos a NFS desde Windows:

```
CA. Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.3650]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Administrador>mount \\192.168.40.183\srv\nuevo_nfs_publico N:
N: está conectado ahora correctamente a \\192.168.40.183\srv\nuevo_nfs_publico

El comando se completó correctamente.
```

Ubicaciones de red (1)



Accedemos a CIFS desde Windows:

```
C:\Users\Administrador>net use Z: \\192.168.40.183\CIFS_PublicoNuevo /user:guest "" /persistent:no
Se ha completado el comando correctamente.
```

Ubicaciones de red (2)



Accedemos a NFS desde Linux:

```
joel@cuatro:~$ sudo mkdir -p /mnt/nuevo_nfs_publico
joel@cuatro:~$ sudo mount -t nfs 192.168.40.183:/srv/nuevo_nfs_publico /mnt/nuevo_nfs_publico
joel@cuatro:~$ ls -l /mnt/nuevo_nfs_publico
total 4
-rwxr-xr-x 1 nobody nogroup 0 jul 29 02:06 'archivo 1.txt'
-rwxr-xr-x 1 nobody nogroup 0 jul 29 02:06 'archivo 2.txt'
-rwxr-xr-x 1 nobody nogroup 0 jul 29 02:06 'archivo 3.txt'
drwxr-xr-x 2 nobody nogroup 4096 jul 29 02:06 Pruebas
```

Accedemos CIFS desde Linux:

```
joel@cuatro:~$ sudo mkdir -p /mnt/nuevo_cifs_publico
joel@cuatro:~$ sudo mount -t cifs //192.168.40.183/CIFS_PublicoNuevo /mnt/nuevo_cifs_publico -o guest
joel@cuatro:~$ ls -l /mnt/nuevo_cifs_publico
total 0
drwxr-xr-x 2 root root 0 jul 29 02:45 actas
-rwxr-xr-x 1 root root 0 jul 29 02:46 inicio.txt
drwxr-xr-x 2 root root 0 jul 29 02:45 otros
-rwxr-xr-x 1 root root 0 jul 29 02:45 otros.txt
drwxr-xr-x 2 root root 0 jul 29 02:45 publico
```

Seguridad y automatización de respaldos

- Realice un respaldo automático de las carpetas compartidas (CIFS y NFS) cada 12 horas.
- Guarde los respaldos en una carpeta específica.
- Incluya control de acceso a los respaldos (propietario/permisos).
- Explique su funcionamiento.

Montamos el recurso NFS

```
sudo mkdir -p /mnt/nuevo_nfs_publico
```

```
sudo mount -t nfs 192.168.40.183:/srv/nuevo_nfs_publico /mnt/nuevo_nfs_publico
```

Montamos el recurso CIFS

```
sudo mkdir -p /mnt/nuevo_cifs_publico
```

```
sudo mount -t cifs //192.168.40.183/CIFS_PublicoNuevo /mnt/nuevo_cifs_publico -o guest
```

Verificamos con `mount | grep -Ei 'nfs|cifs'`

```
joel@tres:~$ mount | grep -Ei 'nfs|cifs'
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)
192.168.40.183:/srv/nuevo_nfs_publico on /mnt/nuevo_nfs_publico type nfs4 (rw,relatime,vers=4.2,rsz=262144,wsz=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.40.183,local_lock=none,addr=192.168.40.183)
//192.168.40.183/CIFS_PublicoNuevo on /mnt/nuevo_cifs_publico type cifs (rw,relatime,vers=3.1.1,sec=none,cache=strict,upcall_target=app,uid=0,noforceuid,gid=0,noforcegid,addr=192.168.40.183,file_mode=0755,dir_mode=0755,soft,nounix,serverino,mapposix,rsz=4194304,wsz=4194304,bsz=1048576,retrans=1,echo_interval=60,actimeo=1,closetimeo=1)
```

Creamos la carpeta donde se guardaran los archivos del respaldo

```
sudo mkdir -p /backups/compartidos
```

```
sudo chown root:respaldo /backups/compartidos
```

```
sudo chmod 750 /backups/compartidos
```

Se crea el grupo de respaldo para el acceso con `sudo groupadd respaldo`

```
joel@tres:~$ sudo mkdir -p /backups/compartidos
joel@tres:~$ sudo chown root:respaldo /backups/compartidos
chown: invalid group: 'root:respaldo'
joel@tres:~$ sudo groupadd respaldo
joel@tres:~$ sudo chown root:respaldo /backups/compartidos
joel@tres:~$ sudo chmod 750 /backups/compartidos
```

Creamos el script de respaldo con `sudo nano /usr/local/bin/respaldo_cifs_nfs.sh`

El script:

```
#!/bin/bash
```

```
FECHA=$(date '+%Y-%m-%d_%H-%M-%S')
```

```
DESTINO="/backups/compartidos"
```

```
LOG="/var/log/respaldo_cifs_nfs.log"
```

```
# Crear subcarpetas con marca de tiempo
```

```
NFS_BACKUP="$DESTINO/nfs_publico_$FECHA"
```

```
CIFS_BACKUP="$DESTINO/cifs_publico_$FECHA"
```

```
mkdir -p "$NFS_BACKUP" "$CIFS_BACKUP"
```

```
echo "[$FECHA] Iniciando respaldo" >> "$LOG"
```

```
# Respalda NFS
```

```
rsync -av --delete /mnt/nuevo_nfs_publico/ "$NFS_BACKUP" >> "$LOG" 2>&1
```

```
if [ $? -eq 0 ]; then
```

```
    echo "[$FECHA] NFS respaldado correctamente" >> "$LOG"
```

```
else
```

```
    echo "[$FECHA] ERROR al respaldar NFS" >> "$LOG"
```

```
fi
```

```
# Respalda CIFS
```

```
rsync -av --delete /mnt/nuevo_cifs_publico/ "$CIFS_BACKUP" >> "$LOG" 2>&1
```

```
if [ $? -eq 0 ]; then
```

```
    echo "[$FECHA] CIFS respaldado correctamente" >> "$LOG"
```

```
else
```

```
    echo "[$FECHA] ERROR al respaldar CIFS" >> "$LOG"
```

```
fi
```

```
# Permisos finales
```

```
chown -R root:respaldo "$DESTINO"
```

```
chmod -R 750 "$DESTINO"
```

```
echo "[$FECHA] Respaldo finalizado" >> "$LOG"
```

Hacemos ejecutable el script


```
joel@tres:~$ sudo nano /usr/local/bin/respaldo_cifs_nfs.sh
joel@tres:~$ sudo chmod +x /usr/local/bin/respaldo_cifs_nfs.sh
joel@tres:~$
```

Programamos la ejecución automática cada 12 horas con crontab

sudo crontab -e

```
joel@tres: ~
GNU nano 7.2 /tmp/crontab.ZMDdLm/crontab *
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/2 * * * * /home/joel/backup_automatico.sh
*/2 * * * * /home/joel/backup_automatico.sh >> /var/log/backup_cron.log 2>&1
0 */12 * * * /usr/local/bin/respaldo_cifs_nfs.sh
```

Probamos que funciona y verificamos que solo podemos acceder a los archivos como root o si el usuario pertenece al grupo respaldos

```
joel@tres:~$ sudo /usr/local/bin/respaldo_cifs_nfs.sh
joel@tres:~$ ls /backups/compartidos/
ls: cannot open directory '/backups/compartidos/': Permission denied
joel@tres:~$ sudo ls /backups/compartidos/
cifs_publico_2025-08-03_21-40-16  nfs_publico_2025-08-03_21-40-16
cifs_publico_2025-08-03_21-40-22  nfs_publico_2025-08-03_21-40-22
```

- **Monitoreo y optimización básica**

Configuración de herramientas básicas de monitoreo (por ejemplo: iostat, df, du, nfsstat, smbstatus).

Con la herramienta du podemos ver el espacio que está consumiendo el recurso, por ejemplo:

```
joel@tres:~$ sudo du -sh /mnt/nuevo_nfs_publico
8,0K    /mnt/nuevo_nfs_publico
joel@tres:~$ sudo du -sh /mnt/nuevo_cifs_publico
0       /mnt/nuevo_cifs_publico
```

Con smbstatus podemos ver los usuarios que están conectados y los recursos que están usando por ejemplo

```
joel@tres:~$ sudo smbstatus

Samba version 4.19.5-Ubuntu
PID      Username   Group      Machine
-----
3378     nobody    nogroup    192.168.40.182 (ipv4:192.168.40.182:60082) SMB3_11
4141     nobody    nogroup    192.168.40.183 (ipv4:192.168.40.183:33688) SMB3_11

Service  pid      Machine    Connected at      Encryption  Signing
-----
CIFS_PublicoNuevo 4141     192.168.40.183 dom ago 3 21:24:36 2025 UTC - -
CIFS_PublicoNuevo 3378     192.168.40.182 dom ago 3 20:55:38 2025 UTC - -
IPC$     4141     192.168.40.183 dom ago 3 21:24:36 2025 UTC - -

Locked files:
Pid      User(ID)   DenyMode   Access    R/W      Oplock      SharePath  Name  Time
-----
3378     65534     DENY_ALL   0x100080  RDONLY   NONE        /srv/nuevo_cifs_publico .    Sun Aug 3 20:55:38 2025
```

- Identificación de al menos una métrica clave para CIFS y una para NFS.

Para CIFS

```
joel@tres:~$ sudo smbstatus | grep -A2 "Service"
Service  pid      Machine    Connected at      Encryption  Signing
-----
CIFS_PublicoNuevo 4141     192.168.40.183 dom ago 3 21:24:36 2025 UTC - -
```

Para NFS

```
joel@tres:~$ nfsstat | grep -A5 'Server rpc'
Server rpc stats:
calls    badcalls   badfmt     badauth    badclnt
403       2          2          0          0

Server nfs v3:
null     getattr    setattr    lookup     access
```

Propuesta de al menos una acción de mejora para optimizar el rendimiento del sistema.

Separar los respaldos en un disco dedicado

Objetivo:

Reducir la carga de E/S (entrada/salida) sobre el disco principal del sistema y evitar posibles cuellos de botella durante las tareas automáticas de respaldo.

Descripción:

Actualmente, los respaldos se están guardando en /backups/compartidos, que se encuentra en el mismo volumen que el sistema operativo y los recursos montados.

Una mejora sería mover los respaldos a un disco secundario exclusivo, ya sea:

Un segundo disco físico (/dev/sdb)

Un volumen lógico (LVM)

Una unidad externa (/mnt/disco_backup)

Esto permitiría:

Evitar competencia de E/S entre el sistema, Samba y NFS.

Reducir el impacto de los respaldos sobre los servicios compartidos.

Aumentar la vida útil del disco principal al reducir escrituras.