

Abstract geometric lines in the top left corner, consisting of several overlapping, irregular polygons and lines in a light beige color.

PRÁCTICA 8 ALMACENAMIENTO

Paloma Pérez de Madrid

ÍNDICE

- 2. Almacenamiento Local
- 3. Configuración de NFS
- 4. Configuración de SAMBA
- 5. Configuración de iSCSI

ÍNDICE

- 2. Almacenamiento Local**
- 3. Configuración de NFS
- 4. Configuración de SAMBA
- 5. Configuración de iSCSI


2. ALMACENAMIENTO LOCAL

Preparar una máquina virtual con la distribución de su elección y que contará con 10 discos duros adicionales de 20 GB de capacidad cada uno (sdb a sdk) y el mismo tipo de controladora (SATA, SCSI, NVMe, ...) con la que cuenta el disco principal. A continuación, arrancar el equipo y realizar las siguientes tareas de configuración:

- a) Comprobar la configuración de la primera unidad de disco (sda) de la máquina virtual. Tipos de particiones existentes (físicas y lógicas), identificación, capacidades y sistemas de archivos configurados.
- b) En la segunda unidad de disco (sdb) crear una partición primaria de 8 GB. Formatear la partición con el sistema de archivos xfs y montar la partición sobre el directorio /mnt/datos. Configurar el montaje automático de la misma (archivo /etc/fstab) y comprobar que se monta automáticamente al reiniciar el equipo. Nota: se recomienda usar el UUID de la partición para evitar problemas en caso de renombrado de particiones.
- c) Usando la utilidad mdadm configurar un RAID1 sobre las unidades sdc y sdd. ¿Qué capacidad tiene el array resultante? (justificar esta capacidad). Formatear el dispositivo RAID con el sistema de archivos ext4, y montarlo sobre el directorio /mnt/raid1. Añadir algún archivo a este directorio. A continuación, simular un fallo en la unidad sdc. Comprobar que los archivos siguen estando accesibles, aunque el array está degradado (utilizar la opción detail para ver el estado del array). Sustituir la unidad sdc por la unidad sde, y comprobar que el array se regenera automáticamente.
- d) Configurar un RAID 5 con cuatro unidades de disco (sdf, sdg, sdh y sdi) más un disco de repuesto o spare (sdj). ¿Qué capacidad tiene el array resultante? ¿Por qué?
- e) Crear un volumen físico para cada unidad RAID configurada en los apartados c) y d) (habrá que desmontar previamente el RAID1, y al crear el volumen físico se perderá el formato y contenido). A continuación, generar un grupo de volúmenes (vg-asi) formado por los dos volúmenes físicos. Visualizar las características y la capacidad del grupo de volúmenes creados. A continuación, crear dos volúmenes lógicos: files (de 50 GB) y apps (con el 50% de la capacidad restante del grupo de volúmenes). Formatear ambos volúmenes lógicos con xfs y montarlos en /mnt/files y /mnt/apps. Configurar el montaje automático de ambos volúmenes y reiniciar el equipo para comprobar que se montan automáticamente.
- f) Necesitamos extender el volumen lógico files hasta 75 GB, usando si es necesario un disco duro adicional (sdk). Realizar el proceso completo, incluyendo la extensión de la partición xfs una vez redimensionemos el volumen lógico (xfs_growfs). Una vez concluido el proceso, y con el volumen montado, adjuntar como evidencia pantallazos de los comandos lsblk y df.

2. ALMACENAMIENTO LOCAL

Preparar una máquina virtual con la distribución de su elección y que contará con 10 discos duros adicionales de 20 GB de capacidad cada uno (sdb a sdk) y el mismo tipo de controladora (SATA, SCSI, NVMe, ...) con la que cuenta el disco principal. A continuación, arrancar el equipo y realizar las siguientes tareas de configuración

Maximum disk size (GB): 

Recommended size for Ubuntu 64-bit: 20 GB


















☐ Allocate all disk space now.

Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.

☒ Store virtual disk as a single file

☐ Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Device	Summary
 Memory	1 GB
 Processors	2
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 New Hard Disk (SATA)	20 GB
 Hard Disk (SCSI)	40 GB
 CD/DVD (SATA)	Using unknown backend
 Network Adapter	NAT
 USB Controller	Present
 Display	Auto detect

2. ALMACENAMIENTO LOCAL

a) Comprobar la configuración de la primera unidad de disco (sda) de la máquina virtual. Tipos de particiones existentes (físicas y lógicas), identificación, capacidades y sistemas de archivos configurados.

Información del disco sda (los principales de la máquina):

`fdisk -l /dev/sda`

```
root@server:~# fdisk -l /dev/sda
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 86E3D186-E177-497E-B3A9-206DCDAA3697

Device        Start      End  Sectors  Size Type
/dev/sda1      2048      4095    2048    1M BIOS boot
/dev/sda2      4096  4198399  4194304    2G Linux filesystem
/dev/sda3  4198400 83884031 79685632   38G Linux filesystem
root@server:~# |
```

Disco Principal

- **/dev/sda1**: Una partición de arranque BIOS que comienza en el sector 2048 y termina en el sector 4095, con un tamaño de 1 MiB.
- **/dev/sda2**: Una partición de sistema de archivos Linux que comienza en el sector 4096 y termina en el sector 4,198,399, con un tamaño de 2 GiB.
- **/dev/sda3**: Otra partición de sistema de archivos Linux que comienza en el sector 4,198,400 y termina en el sector 83,884,031, con un tamaño de 38 GiB.

2. ALMACENAMIENTO LOCAL

a) Comprobar la configuración de la primera unidad de disco (sda) de la máquina virtual. Tipos de particiones existentes (físicas y lógicas), identificación, capacidades y sistemas de archivos configurados.

Listar discos del sistema:

`lsblk`

loopX: Dispositivos de bucle
(dispositivos virtuales que permiten
montar archivos como dispositivos).

Discos configurados
como Hard Disk SATA

```
root@server:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0                              7:0    0   63,3M  1 loop /snap/core20/1778
loop1                              7:1    0   40,4M  1 loop /snap/snapd/20671
loop2                              7:2    0    87M  1 loop /snap/lxd/28373
loop3                              7:3    0   49,8M  1 loop /snap/snapd/17950
loop4                              7:4    0  114,4M  1 loop /snap/lxd/26741
loop5                              7:5    0    62M  1 loop /snap/core20/1587
sda                                8:0    0   40G   0 disk
├─sda1                             8:1    0    1M   0 part
├─sda2                             8:2    0    2G   0 part /boot
├─sda3                             8:3    0   38G   0 part
└─ubuntu--vg-ubuntu--lv 253:0    0   19G   0 lvm  /
sdb                                8:16   0   20G   0 disk
sdc                                8:32   0   20G   0 disk
sdd                                8:48   0   20G   0 disk
sde                                8:64   0   20G   0 disk
sdf                                8:80   0   20G   0 disk
sdg                                8:96   0   20G   0 disk
sdh                                8:112  0   20G   0 disk
sdi                                8:128  0   20G   0 disk
sdj                                8:144  0   20G   0 disk
sdk                                8:160  0   20G   0 disk
sr0                                11:0    1 1024M   0 rom
```

2. ALMACENAMIENTO LOCAL

a) Comprobar la configuración de la primera unidad de disco (sda) de la máquina virtual. Tipos de particiones existentes (físicas y lógicas), identificación, capacidades y sistemas de archivos configurados.

Listar discos del sistema:

`lsblk`

- sda: Primer disco duro
- sda1, sda2, sda3: Particiones del primer disco duro.
- /boot: Punto de montaje de la partición sda2.
- /: Punto de montaje de la partición sda3.
- ubuntu--vg-ubuntu--lv: Nombre del volumen lógico (LVM) dentro de la partición sda3.
- lvm: Tipo de partición (en el caso del volumen lógico).

```
root@server:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0                              7:0      0   63,3M 1 loop /snap/core20/1778
loop1                              7:1      0   40,4M 1 loop /snap/snapd/20671
loop2                              7:2      0    87M 1 loop /snap/lxd/28373
loop3                              7:3      0   49,8M 1 loop /snap/snapd/17950
loop4                              7:4      0  114,4M 1 loop /snap/lxd/26741
loop5                              7:5      0    62M 1 loop /snap/core20/1587
sda                                 8:0      0   40G   0 disk
├─sda1                             8:1      0    1M   0 part
├─sda2                             8:2      0    2G   0 part /boot
└─sda3                             8:3      0   38G   0 part
   └─ubuntu--vg-ubuntu--lv 253:0    0   19G   0 lvm  /
sdb                                 8:16     0   20G   0 disk
sdc                                 8:32     0   20G   0 disk
sdd                                 8:48     0   20G   0 disk
sde                                 8:64     0   20G   0 disk
sdf                                 8:80     0   20G   0 disk
sdg                                 8:96     0   20G   0 disk
sdh                                 8:112    0   20G   0 disk
sdi                                 8:128    0   20G   0 disk
sdj                                 8:144    0   20G   0 disk
sdk                                 8:160    0   20G   0 disk
sr0                                11:0     1  1024M   0 rom
```

Unidad CD/DVD 1024M

2. ALMACENAMIENTO LOCAL

b) En la segunda unidad de disco (sdb) crear una partición primaria de 8 GB. Formatear la partición con el sistema de archivos xfs y montar la partición sobre el directorio /mnt/datos. Configurar el montaje automático de la misma (archivo /etc/fstab) y comprobar que se monta automáticamente al reiniciar el equipo. Nota: se recomienda usar el UUID de la partición para evitar problemas en caso de renombrado de particiones.

- 1) Crear partición primaria del disco sdb: `fdisk /dev/sdb`
- 2) Formatear la partición con XFS: `mkfs.xfs /dev/sdb1`
- 3) Crear archivo de montaje : `mkdir /mnt/datos`
- 4) Montar la partición en ese archivo: `mount /dev/sdb1 /mnt/datos`
- 5) Obtener el UUID de la partición (lo necesitaremos para configurar el montaje automático): `blkid /dev/sdb1`
- 6) Editar /etc/fstab y añadir `UUID=<UUID> /mnt/datos xfs defaults 0 0`
- 7) Comprobar montaje automático : `reboot`

Se debería observar la partición /dev/sdb1 montada en /mnt/datos

2. ALMACENAMIENTO LOCAL

b) En la segunda unidad de disco (sdb) crear una partición primaria de 8 GB. Formatear la partición con el sistema de archivos xfs y montar la partición sobre el directorio /mnt/datos. Configurar el montaje automático de la misma (archivo /etc/fstab) y comprobar que se monta automáticamente al reiniciar el equipo. Nota: se recomienda usar el UUID de la partición para evitar problemas en caso de renombrado de particiones.

```
Command (m for help): o
Created a new DOS disklabel with disk identifier 0xafdd7611.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1): 1
First sector (2048-41943039, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943039, default 41943039): +8200M

Created a new partition 1 of type 'Linux' and of size 8 GiB.
```

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

- 1) Crear partición primaria del disco sdb: `fdisk /dev/sdb`
 - a) Crear una nueva tabla de particiones: `Command : o`
 - b) Crear una nueva partición: `Command : n`
 - c) Tipo de partición : `p (primaria)`
 - d) Número de partición: `1`
 - e) First Sector: `default`
 - f) Last Sector: `+8200M` (8GB y 200M para ajustarse al tamaño de la partición)
 - g) Guardar: `w`

```
abanda -vg abanda -lv 255.0 0 198 0 tvm /
sdb      8:16  0    20G  0 disk
└─sdb1    8:17  0     8G  0 part
```

lsblk

2. ALMACENAMIENTO LOCAL

b) En la segunda unidad de disco (sdb) crear una partición primaria de 8 GB. Formatear la partición con el sistema de archivos xfs y montar la partición sobre el directorio /mnt/datos. Configurar el montaje automático de la misma (archivo /etc/fstab) y comprobar que se monta automáticamente al reiniciar el equipo. Nota: se recomienda usar el UUID de la partición para evitar problemas en caso de renombrado de particiones.

Formatear la partición con XFS: `mkfs.xfs /dev/sdb1`

```
root@server:~# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1            isize=512    agcount=4, agsize=524800 blks
                     =          sectsz=512   attr=2, projid32bit=1
                     =          crc=1        finobt=1, sparse=1, rmapbt=0
                     =          reflink=1    bigtime=0 inobtcount=0
data        =                  bsize=4096   blocks=2099200, imaxpct=25
                     =          sunit=0      swidth=0 blks
naming      =version 2         bsize=4096   ascii-ci=0, ftype=1
log         =internal log     bsize=4096   blocks=2560, version=2
                     =          sectsz=512   sunit=0 blks, lazy-count=1
realtime    =none             extsz=4096   blocks=0, rtextents=0
```

Crear archivo de montaje : `mkdir /mnt/datos`

Montar la partición en ese archivo: `mount /dev/sdb1 /mnt/datos`

```
root@server:~# mkdir /mnt/datos
root@server:~# mount /dev/sdb1 /mnt/datos
```

2. ALMACENAMIENTO LOCAL

b) En la segunda unidad de disco (sdb) crear una partición primaria de 8 GB. Formatear la partición con el sistema de archivos xfs y montar la partición sobre el directorio /mnt/datos. Configurar el montaje automático de la misma (archivo /etc/fstab) y comprobar que se monta automáticamente al reiniciar el equipo. Nota: se recomienda usar el UUID de la partición para evitar problemas en caso de renombrado de particiones.

Obtener el UUID de la partición (lo necesitaremos para configurar el montaje automático): `blkid /dev/sdb1`

```
root@server: ~# blkid /dev/sdb1
/dev/sdb1: UUID="aae1789d-ea00-469a-971c-c10c74f2e5b9" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="a91bae48-01"
UUID="aae1789d-ea00-469a-971c-c10c74f2e5b9"
```

Editar /etc/fstab y añadir `UUID=<UUID> /mnt/datos xfs defaults 0 0`

`UUID= aae1789d-ea00-469a-971c-c10c74f2e5b9 /mnt/datos xfs defaults 0 0`

```
# / was on /dev/ubuntovg/ubuntolv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-sULEcfrthWWQC73L0FLBcmCyjsyq0ThNyqbhDGBmkV4ZUR5wKS8qpo8x774NMS00 / ext4 defaults 0
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/675d68d7-7cbc-4abb-8c90-e8bcb22ba155 /boot ext4 defaults 0 1
/swap.img          none        swap      sw        0          0
UUID=aae1789d-ea00-469a-971c-c10c74f2e5b9 /mnt/datos xfs defaults 0 0
```

2. ALMACENAMIENTO LOCAL

b) En la segunda unidad de disco (sdb) crear una partición primaria de 8 GB. Formatear la partición con el sistema de archivos xfs y montar la partición sobre el directorio /mnt/datos. Configurar el montaje automático de la misma (archivo /etc/fstab) y comprobar que se monta automáticamente al reiniciar el equipo. Nota: se recomienda usar el UUID de la partición para evitar problemas en caso de renombrado de particiones.

Comprobar montaje automático : **reboot**
(Se debería observar la partición
/dev/sdb1 montada en /mnt/datos)

```
root@server:~# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
loop0	7:0	0	49,8M	1	loop	/snap/snapd/17950
loop1	7:1	0	62M	1	loop	/snap/core20/1587
loop2	7:2	0	63,3M	1	loop	/snap/core20/1778
loop3	7:3	0	114,4M	1	loop	/snap/lxd/26741
loop4	7:4	0	87M	1	loop	/snap/lxd/28373
loop5	7:5	0	40,4M	1	loop	/snap/snapd/20671
sda	8:0	0	40G	0	disk	
├─sda1	8:1	0	1M	0	part	
├─sda2	8:2	0	2G	0	part	/boot
├─sda3	8:3	0	38G	0	part	
│ └─ubuntu--vg-ubuntu--lv	253:0	0	19G	0	lvm	/
sdb	8:16	0	20G	0	disk	
└─sdb1	8:17	0	8G	0	part	/mnt/datos

2. ALMACENAMIENTO LOCAL

c) Usando la utilidad mdadm configurar un RAID1 sobre las unidades sdc y sdd. ¿Qué capacidad tiene el array resultante? (justificar esta capacidad). Formatear el dispositivo RAID con el sistema de archivos ext4, y montarlo sobre el directorio /mnt/raid1. Añadir algún archivo a este directorio. A continuación, simular un fallo en la unidad sdc. Comprobar que los archivos siguen estando accesibles, aunque el array está degradado (utilizar la opción detail para ver el estado del array). Sustituir la unidad sdc por la unidad sde, y comprobar que el array se regenera automáticamente.

1. Descargar madam: `apt-get install mdadm`
2. Crear RAID: `mdadm --create --verbose /dev/md0 --level=mirror --raid-devices=2 /dev/sdc /dev/sdd`
 - `--verbose`: Muestra la salida detallada del proceso.
 - `/dev/md0`: nombre del dispositivo RAID que se creará.
 - `--level=mirror`: Nivel de RAID (en este caso es RAID1)
 - `--raid-devices=2`: Número de dispositivos que formarán parte del RAID, que en este caso son 2 (Raid 1 se duplica, espejo o “mirror”).
 - `/dev/sdc /dev/sdd`: Dispositivos que se usarán para el RAID1.

2. ALMACENAMIENTO LOCAL

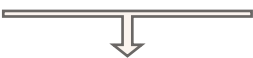
c) Usando la utilidad mdadm configurar un RAID1 sobre las unidades sdc y sdd. ¿Qué capacidad tiene el array resultante? (justificar esta capacidad).

```
root@server:~# mdadm --create --verbose /dev/md0 --level=mirror --raid-devices=2 /dev/sdc /dev/sdd
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
mdadm: size set to 20954112K
Continue creating array?
Continue creating array? (y/n) y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

La capacidad del array resultante se ha establecido en 20954112K

El RAID1 (espejo) duplica los datos en dos discos para proporcionar redundancia. Dado que estamos usando dos discos idénticos (sdc y sdd), la capacidad efectiva del RAID1 será igual a la capacidad de **uno** de los discos

Capacidad de un solo disco = $(20954112K / 1024) / 1024 \approx 20 \text{ GB}$


Conversión de KB a GB

2. ALMACENAMIENTO LOCAL

c) Usando la utilidad mdadm configurar un RAID1 sobre las unidades sdc y sdd. ¿Qué capacidad tiene el array resultante? (justificar esta capacidad). Formatear el dispositivo RAID con el sistema de archivos ext4, y montarlo sobre el directorio /mnt/raid1.

1. Formatear el dispositivo RAID con el sistema de archivos ext4: `mkfs.ext4 /dev/md0`
2. Montarlo en el Directorio /mnt/raid1:
 - `mkdir /mnt/raid1`
 - `mount /dev/md0 /mnt/raid1`

```
root@server:~# mkfs.ext4 /dev/md0
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5238528 4k blocks and 1310720 inodes
Filesystem UUID: 46da9a29-4d95-4e7f-bc5a-856e89083f12
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
root@server:~# mkdir /mnt/raid1
root@server:~# mount /dev/md0 /mnt/raid1
```

```
root@server:~# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	93M	1,3M	91M	2%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	19G	7,9G	9,8G	45%	/
tmpfs	461M	0	461M	0%	/dev/shm
tmpfs	5,0M	0	5,0M	0%	/run/lock
/dev/sda2	2,0G	251M	1,6G	14%	/boot
/dev/sdb1	8,0G	90M	8,0G	2%	/mnt/datos
tmpfs	93M	4,0K	93M	1%	/run/user/0
/dev/md0	20G	24K	19G	1%	/mnt/raid1

2. ALMACENAMIENTO LOCAL

c) Añadir algún archivo a este directorio. A continuación, simular un fallo en la unidad sdc. Comprobar que los archivos siguen estando accesibles, aunque el array está degradado (utilizar la opción detail para ver el estado del array). Sustituir la unidad sdc por la unidad sde, y comprobar que el array se regenera automáticamente.

1. Añadir archivo al directorio: `touch /mnt/raid1/archivo.txt`
2. Simular un fallo en la unidad:
 - a) Marcar la unidad sdc como fallida: `mdadm --manage /dev/md0 --fail /dev/sdc`
 - b) Eliminar la unidad fallida del array: `mdadm --manage /dev/md0 --remove /dev/sdc`
 - c) Comprobar que los archivos siguen estando accesibles aunque el array está degradado: `ls -l /mnt/raid1`

```
root@server:~# touch /mnt/raid1/archivo.txt
root@server:~# mdadm --manage /dev/md0 --fail /dev/sdc
mdadm: set /dev/sdc faulty in /dev/md0
root@server:~# mdadm --manage /dev/md0 --remove /dev/sdc
mdadm: hot removed /dev/sdc from /dev/md0
root@server:~# ls -l /mnt/raid1
total 16
-rw-r--r-- 1 root root    0 abr 30 16:35 archivo.txt
drwx----- 2 root root 16384 abr 30 16:29 lost+found
```

2. ALMACENAMIENTO LOCAL

c) Añadir algún archivo a este directorio. A continuación, simular un fallo en la unidad sdc. Comprobar que los archivos siguen estando accesibles, aunque el array está degradado (utilizar la opción detail para ver el estado del array). Sustituir la unidad sdc por la unidad sde, y comprobar que el array se regenera automáticamente.

1. Sustituir la unidad sdc por la unidad sde :
`mdadm --manage /dev/md0 --add /dev/sde`
2. Comprobar que el array se regenera automáticamente:
`mdadm --detail /dev/md0`

Active Devices : 1 → uno de los dispositivos del array está funcionando correctamente

Working Devices : 2 → Hay 2 dispositivos operativos (/dev/sdc y /dev/sdd)

Rebuild Status : 5% complete → el proceso de regeneración del array está en curso y ya ha progresado un 5%.

```
root@server:~# mdadm --manage /dev/md0 --add /dev/sde
mdadm: added /dev/sde
root@server:~# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
    Creation Time : Tue Apr 30 16:20:53 2024
    Raid Level : raid1
    Array Size : 20954112 (19.98 GiB 21.46 GB)
    Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Tue Apr 30 16:38:15 2024
    State : clean, degraded, recovering
    Active Devices : 1
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 1

    Consistency Policy : resync

    Rebuild Status : 5% complete

    Name : server:0 (local to host server)
    UUID : 051a8b47:36d3c947:1443b580:a25f18f7
    Events : 28

    Number Major Minor RaidDevice State
       2      8      64        0  spare rebuilding /dev/sde
       1      8      48        1  active sync /dev/sdd
root@server:~#
```

2. ALMACENAMIENTO LOCAL

Si se vuelve a dar al comando más tarde:

```
root@server:~# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
    Creation Time : Tue Apr 30 16:20:53 2024
    Raid Level : raid1
    Array Size : 20954112 (19.98 GiB 21.46 GB)
    Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Tue Apr 30 16:40:00 2024
    State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

Consistency Policy : resync

    Name : server:0 (local to host server)
    UUID : 051a8b47:36d3c947:1443b580:a25f18f7
    Events : 45

    Number Major Minor RaidDevice State
      2       8      64        0  active sync /dev/sde
      1       8      48        1  active sync /dev/sdd
root@server:~#
```

2. ALMACENAMIENTO LOCAL

d) Configurar un RAID 5 con cuatro unidades de disco (sdf, sdg, sdh y sdi) más un disco de repuesto o spare (sdj). ¿Qué capacidad tiene el array resultante? ¿Por qué?

Añadir archivo al directorio: `mdadm --create --verbose /dev/md1 --level=5 --raid-devices=4 --spare-devices=1 /dev/sdf /dev/sdg /dev/sdh /dev/sdi /dev/sdj`

```
root@server:~# mdadm --create --verbose /dev/md1 --level=5 --raid-devices=4 --spare-devices=1 /dev/sdf /dev/sdg /dev/sdh /dev/sdi /dev/sdj
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 20954112K
mdadm: Defaulting to version 1.2 metadata
```

Siguiendo el cálculo anterior:

- Capacidad de cada disco = $(20954112K / 1024) / 1024 \approx 20 \text{ GB}$
- Dado que son cuatro discos de datos y uno de repuesto, el número de discos de datos es $4 - 1 = 3$.
- Capacidad total del array $\approx 20 \text{ GB} \times 3 = 60 \text{ GB}$

2. ALMACENAMIENTO LOCAL

d) Configurar un RAID 5 con cuatro unidades de disco (sdf, sdg, sdh y sdi) más un disco de repuesto o spare (sdj). ¿Qué capacidad tiene el array resultante? ¿Por qué?

`cat /proc/mdstat` para ver el estado actual de todos los dispositivos RAID

```
root@server:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md1 : active raid5 sdi[5] sdj[4](S) sdh[2] sdg[1] sdf[0]
      62862336 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [UUU_]
      [=>.....] recovery = 5.6% (1176412/20954112) finish=13.1min speed
=25126K/sec

md0 : active raid1 sde[2] sdd[1]
      20954112 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

Md1: le quedan 13.1min para regenerar los datos

```
root@server:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md1 : active raid5 sdi[5] sdj[4](S) sdh[2] sdg[1] sdf[0]
      62862336 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/4] [UUUU]

md0 : active raid1 sde[2] sdd[1]
      20954112 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

2. ALMACENAMIENTO LOCAL

e) Crear un volumen físico para cada unidad RAID configurada en los apartados c) y d) (habrá que desmontar previamente el RAID1, y al crear el volumen físico se perderá el formato y contenido). A continuación, generar un grupo de volúmenes (vg-asi) formado por los dos volúmenes físicos. Visualizar las características y la capacidad del grupo de volúmenes creados. A continuación, crear dos volúmenes lógicos: files (de 50 GB) y apps (con el 50% de la capacidad restante del grupo de volúmenes). Formatear ambos volúmenes lógicos con xfs y montarlos en /mnt/files y /mnt/apps. Configurar el montaje automático de ambos volúmenes y reiniciar el equipo para comprobar que se montan automáticamente.

- 1) Desmontar RAID 1: `umount /dev/md0`
- 2) Crear volumen físico en RAID 1: `pvcreate /dev/md0` (usando LVM manager)
- 3) Crear volumen físico para RAID 5: `pvcreate /dev/md1`

2. ALMACENAMIENTO LOCAL

e) Crear un volumen físico para cada unidad RAID configurada en los apartados c) y d) (habrá que desmontar previamente el RAID1, y al crear el volumen físico se perderá el formato y contenido).

```
root@server:~# umount /dev/md0
root@server:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     93M        1,3M   91M    2% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 19G       7,9G    9,8G   45% /
tmpfs                     461M         0   461M    0% /dev/shm
tmpfs                     5,0M         0    5,0M    0% /run/lock
/dev/sda2                 2,0G      251M    1,6G   14% /boot
/dev/sdb1                 8,0G       90M    8,0G    2% /mnt/datos
tmpfs                     93M        4,0K    93M    1% /run/user/0
```

```
root@server:~# pvcreate /dev/md0
WARNING: ext4 signature detected on /dev/md0 at offset 1080. Wipe it? [y/n]: y
Wiping ext4 signature on /dev/md0.
Physical volume "/dev/md0" successfully created.
```

```
root@server:~# pvcreate /dev/md1
Physical volume "/dev/md1" successfully created.
```

2. ALMACENAMIENTO LOCAL

e) A continuación, generar un grupo de volúmenes (vg-asi) formado por los dos volúmenes físicos. Visualizar las características y la capacidad del grupo de volúmenes creados.

1) Crear grupo de volúmenes:

```
vgcreate vg-asi /dev/md0 /dev/md1
```

2) Visualizar características del grupo de volúmenes:

```
vgdisplay vg-asi
```

3) Visualizar capacidad grupo de volúmenes:

```
vgs vg-asi
```

```
root@server:~# vgcreate vg-asi /dev/md0 /dev/md1
Volume group "vg-asi" successfully created
root@server:~# vgdisplay vg-asi
--- Volume group ---
VG Name                vg-asi
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   1
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0
Open LV                 0
Max PV                 0
Cur PV                 2
Act PV                 2
VG Size                 <79,93 GiB
PE Size                 4,00 MiB
Total PE                20461
Alloc PE / Size         0 / 0
Free PE / Size          20461 / <79,93 GiB
VG UUID                 xcC0uh-CsfS-YP6l-cST5-U9kE-fBmq-FuiJd1

root@server:~# vgs vg-asi
VG      #PV #LV #SN Attr   VSize   VFree
vg-asi   2   0   0 wz--n- <79,93g <79,93g
```


2. ALMACENAMIENTO LOCAL

e) A continuación, crear dos volúmenes lógicos: files (de 50 GB) y apps (con el 50% de la capacidad restante del grupo de volúmenes). Formatear ambos volúmenes lógicos con xfs y montarlos en /mnt/files y /mnt/apps. Configurar el montaje automático de ambos volúmenes y reiniciar el equipo para comprobar que se montan automáticamente

- 1) Crear volúmenes lógicos:
 - File 50GB: `lvcreate -L 50G -n files vg-asi`
 - apps 50% del espacio restante:
 - Calcular 50% restante con `vgdisplay vg-asi | grep "Free PE / Size"`
 - Crear volume lógico: `lvcreate -l 50%FREE -n apps vg-asi`
 - 2) Formatear los volúmenes lógicos con XFS:
 - Files: `mkfs.xfs /dev/vg-asi/files`
 - Apps: `mkfs.xfs /dev/vg-asi/apps`
 - 3) Montar los volúmenes lógicos:
 - `mkdir -p /mnt/files + mount /dev/vg-asi/files /mnt/files`
 - `sudo mkdir -p /mnt/apps + mount /dev/vg-asi/apps /mnt/apps`
 - 4) Configurar el montaje automático :
 - `UUID=[UUID_files] /mnt/files xfs defaults 0 0`
 - `UUID=[UUID_apps] /mnt/apps xfs defaults 0 0`
- Nota: para encontrar los UUID: blkid

2. ALMACENAMIENTO LOCAL

1) Crear volúmenes lógicos:

- File 50GB: `lvcreate -L 50G -n files vg-asi`
- apps 50% del espacio restante:
 - Calcular 50% restante con `vgdisplay vg-asi | grep "Free PE / Size"`
 - Crear volume lógico: `lvcreate -l 50%FREE -n apps vg-asi`

```
root@server:~# lvcreate -L 50G -n files vg-asi
Logical volume "files" created.
```

```
root@server:~# vgdisplay vg-asi | grep "Free PE / Size"
Free PE / Size      7661 / <29,93 GiB
```

Convertir el tamaño del espacio libre a GiB: 29,93 GiB

Calcular el 50% del espacio libre: $50\% \text{ de } 29,93 \text{ GiB} = 0.5 * 29,93 = 14,965 \text{ GiB}$

Luego el comando será: `lvcreate -L 14.965G -n apps vg-asi`

```
Rounding up size to full physical extent <14,97 GiB
Logical volume "apps" created.
```

2. ALMACENAMIENTO LOCAL

2) Formatear los volúmenes lógicos con XFS:

- Files: `mkfs.xfs /dev/vg-asi/files`
- Apps: `mkfs.xfs /dev/vg-asi/apps`

```
root@server:~# mkfs.xfs /dev/vg-asi/files
log stripe unit (524288 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/vg-asi/files      isize=512    agcount=16, agsize=819200 blks
                                   =               sectsz=512    attr=2, projid32bit=1
                                   =               crc=1        finobt=1, sparse=1, rmapbt=0
                                   =               reflink=1     bigtime=0 inobtcount=0
data      =                       bsize=4096   blocks=13107200, imaxpct=25
                                   =               sunit=128     swidth=384 blks
naming    =version 2             bsize=4096   ascii-ci=0, ftype=1
log       =internal log         bsize=4096   blocks=6400, version=2
                                   =               sectsz=512    sunit=8 blks, lazy-count=1
realtime  =none                  extsz=4096    blocks=0, rtextents=0

root@server:~# mkfs.xfs /dev/vg-asi/apps
meta-data=/dev/vg-asi/apps      isize=512    agcount=4, agsize=980992 blks
                                   =               sectsz=512    attr=2, projid32bit=1
                                   =               crc=1        finobt=1, sparse=1, rmapbt=0
                                   =               reflink=1     bigtime=0 inobtcount=0
data      =                       bsize=4096   blocks=3923968, imaxpct=25
                                   =               sunit=0       swidth=0 blks
naming    =version 2             bsize=4096   ascii-ci=0, ftype=1
log       =internal log         bsize=4096   blocks=2560, version=2
                                   =               sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                  extsz=4096    blocks=0, rtextents=0
```

2. ALMACENAMIENTO LOCAL

3) Montar los volúmenes lógicos:

- `mkdir -p /mnt/files + mount /dev/vg-asi/files /mnt/files`
- `mkdir -p /mnt/apps + mount /dev/vg-asi/apps /mnt/apps`

4) Configurar el montaje automático :

- `UUID=[UUID_files] /mnt/files xfs defaults 0 0`
- `UUID=[UUID_apps] /mnt/apps xfs defaults 0 0`

```
root@server:~# mkdir -p /mnt/files
root@server:~# mount /dev/vg-asi/files /mnt/files
```

```
root@server:~# mkdir -p /mnt/apps
root@server:~# ^C
root@server:~# mount /dev/vg-asi/apps /mnt/apps
```

Nota: para encontrar los UUID: `blkid`

```
root@server:~# blkid /dev/vg-asi/files
/dev/vg-asi/files: UUID="554687be-24f0-4710-b5cf-1b9e4bef64f9" BLOCK_SIZE="512" TYPE="xfs"
root@server:~# blkid /dev/vg-asi/apps
/dev/vg-asi/apps: UUID="69cdf7ae-13ee-4e30-9dc2-6ec9cb105682" BLOCK_SIZE="512" TYPE="xfs"
```

- `UUID= 554687be-24f0-4710-b5cf-1b9e4bef64f9 /mnt/files xfs defaults 0 0`
- `UUID= 69cdf7ae-13ee-4e30-9dc2-6ec9cb105682 /mnt/apps xfs defaults 0 0`

2. ALMACENAMIENTO LOCAL

- UUID= 554687be-24f0-4710-b5cf-1b9e4bef64f9 /mnt/files xfs defaults 0 0
- UUID= 69cdf7ae-13ee-4e30-9dc2-6ec9cb105682 /mnt/apps xfs defaults 0 0

vim /etc/fstab

```
/dev/disk/by-id/dm-uuid-LVM-sULEcfrthWWQC73L0FLBcmCyjsyq0ThNyqbhDGbmKV4ZUR5wKS8qpo8x774
NMS00 / ext4 defaults 0 1
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/675d68d7-7cbc-4abb-8c90-e8bcb22ba155 /boot ext4 defaults 0 1
/swap.img          none        swap        sw          0          0

UUID=aae1789d-ea00-469a-971c-c10c74f2e5b9 /mnt/datos  xfs  defaults  0  0
UUID=554687be-24f0-4710-b5cf-1b9e4bef64f9 /mnt/files  xfs  defaults  0  0
UUID=69cdf7ae-13ee-4e30-9dc2-6ec9cb105682 /mnt/apps  xfs  defaults  0  0
```

root@server:~# reboot



```
root@server:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                      93M        1,3M   91M   2% /run
/dev/mapper/ubuntu--vg-ubuntu--lv  19G       7,9G   9,8G  45% /
tmpfs                      461M         0   461M   0% /dev/shm
tmpfs                      5,0M         0    5,0M   0% /run/lock
/dev/sdf2                  2,0G      251M   1,6G  14% /boot
/dev/sda1                   8,0G       90M   8,0G   2% /mnt/datos
/dev/mapper/vg--asi-files  50G       390M   50G    1% /mnt/files
/dev/mapper/vg--asi-apps   15G      140M   15G    1% /mnt/apps
tmpfs                      93M        4,0K   93M   1% /run/user/0
```

2. ALMACENAMIENTO LOCAL

f) Necesitamos extender el volumen lógico files hasta 75 GB, usando si es necesario un disco duro adicional (sdk). Realizar el proceso completo, incluyendo la extensión de la partición xfs una vez redimensionemos el volumen lógico (xfs_growfs). Una vez concluido el proceso, y con el volumen montado, adjuntar como evidencia pantallazos de los comandos lsblk y df.

- 1) Extender el volumen lógico de files (ya tiene 50GB, vamos a necesitar extenderlo hasta 75GB)
 - Si suponemos que sí necesitamos extender a otro disco (25GB) restantes: `lvextend -L +55G /dev/vg-asi/files /dev/sdk`
 - Si suponemos que no: `lvextend -L 25G /dev/vg-asi/files` (vamos a seleccionar esta opción)
- 2) Redimensionar el sistema de archivos xfs: `xfs_growfs /mnt/files`

2. ALMACENAMIENTO LOCAL

fdisk -l

```
Disk /dev/mapper/vg--asi-files: 50 GiB, 53687091200 bytes, 104857600 sectors  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 524288 bytes / 1572864 bytes
```

lvextend -L 25G /dev/vg-asi/files

```
root@server:~# lvextend -L 25G /dev/vg-asi/files  
New size given (6400 extents) not larger than existing size (12800 extents)
```

El tamaño nuevo especificado para el volumen lógico (25G) no es mayor que el tamaño existente (12800 extents). Esto significa que el volumen lógico ya está en su tamaño máximo y no se puede extender más allá de eso.

2. ALMACENAMIENTO LOCAL

Verificamos el espacio libre en el grupo de volúmenes vg-asi: `vgdisplay vg-asi | grep "Free PE / Size"`

```
root@server:~# vgdisplay vg-asi | grep "Free PE / Size"
Free PE / Size      8948 / 34,95 GiB
```

Verificamos el espacio libre en el volumen lógico files: `lvdisplay /dev/vg-asi/files | grep "LV Size"`

```
root@server:~# lvdisplay /dev/vg-asi/files | grep "LV Size"
LV Size              50,00 GiB
```

Nota: hemos añadido al grupo vg-asi sdk antes

Extendemos +25GB (ya teníamos 50GB): `lvextend -L +25G /dev/vg-asi/files`

```
root@server:~# lvextend -L +25G /dev/vg-asi/files
Size of logical volume vg-asi/files changed from 50,00 GiB (12800 extents) to 75,00 GiB (19200 extents).
Logical volume vg-asi/files successfully resized.
```


2. ALMACENAMIENTO LOCAL

Ajustar sistema de archivos XFS: `xfs_growfs /mnt/files`

```
root@server:~# xfs_growfs /mnt/files
meta-data=/dev/mapper/vg--asi-files isize=512    agcount=16, agsize=819200 blks
        =                               sectsz=512   attr=2, projid32bit=1
        =                               crc=1        finobt=1, sparse=1, rmapbt=0
        =                               reflink=1    bigtime=0 inobtcount=0
data      =                               bsize=4096   blocks=13107200, imaxpct=25
        =                               sunit=128    swidth=384 blks
naming    =version 2                       bsize=4096   ascii-ci=0, ftype=1
log        =internal log                   bsize=4096   blocks=6400, version=2
        =                               sectsz=512   sunit=8 blks, lazy-count=1
realtime  =none                             extsz=4096   blocks=0, rtextents=0
data blocks changed from 13107200 to 19660800
```

`lsblk`

```
sdg                8:96    0    20G    0 disk
└─md126             9:126    0    60G    0 raid5
   └─vg--asi-files   253:1    0    75G    0 lvm    /mnt/files
sdh                8:112    0    20G    0 disk
└─md126             9:126    0    60G    0 raid5
   └─vg--asi-files   253:1    0    75G    0 lvm    /mnt/files
sdi                8:128    0    20G    0 disk
└─md126             9:126    0    60G    0 raid5
   └─vg--asi-files   253:1    0    75G    0 lvm    /mnt/files
sdj                8:144    0    20G    0 disk
└─md126             9:126    0    60G    0 raid5
   └─vg--asi-files   253:1    0    75G    0 lvm    /mnt/files
sdk                8:160    0    20G    0 disk
└─vg--asi-files      253:1    0    75G    0 lvm    /mnt/files
sr0               11:0     1 1024M    0 rom
```

ÍNDICE

- 2. Almacenamiento Local
- 3. **Configuración de NFS**
- 4. Configuración de SAMBA
- 5. Configuración de iSCSI

3. CONFIGURACIÓN DE NFS

- a) Instalar un servidor NFS en un equipo con SO Rocky Linux 9.3.
- b) Configurar el servidor NFS para compartir algún directorio. A lo largo de estos apartados vamos a evaluar las opciones de exportación. En la configuración inicial podemos compartir sin restricciones un directorio en modo lectura/escritura (/var/archivos, con derechos 777) y otro en modo de sólo lectura (/usr/share/doc).
- c) Utilizar otro equipo como cliente NFS, ahora con SO Ubuntu Server 22.04. Indicar el proceso seguido para la instalación. Nota: en el lado cliente NO hay que arrancar el servidor NFS. Desde el cliente listar los directorios NFS que comparte el servidor.
- d) En el equipo cliente montar (por ejemplo, en /mnt/nfs) el directorio NFS compartido por el servidor en modo lectura/escritura. Añadir un archivo desde el cliente en el recurso compartido y comprobar la propiedad y derechos del mismo, tanto si lo hacemos con un usuario no privilegiado como si usamos el perfil de root. A continuación, modificar las propiedades de compartición (ro, all_squash, no_root_squash, ...) y comprobar su efecto tras añadir nuevos archivos (si tenemos derecho de escritura, el UID de usuario con el que se crean los archivos, y permisos asociados a los mismos).
- e) Limitar las direcciones IP de acceso y comprobar que el recurso deja de estar disponible desde otras direcciones IP distintas.
- f) Indicar el proceso de instalación del servidor NFS en el SO Ubuntu Server 22.04, y del cliente en un SO Rocky Linux 9.3.

3. CONFIGURACIÓN DE NFS

- Instalar un servidor NFS en un equipo con SO Rocky Linux 9.3.
- Configurar el servidor NFS para compartir algún directorio. A lo largo de estos apartados vamos a evaluar las opciones de exportación. En la configuración inicial podemos compartir sin restricciones un directorio en modo lectura/escritura (/var/archivos, con derechos 777) y otro en modo de sólo lectura (/usr/share/doc).

1) Instalar el paquete NFS: `dnf install nfs-utils`

```
^C^C^C[root@server ~]# dnf install nfs-utils
Error al cargar el complemento "config_manager": '*prog'
Extra Packages for Enterprise Linux 9 - x86_64      37 kB/s | 33 kB    00:00
Extra Packages for Enterprise Linux 9 - x86_64      1.0 MB/s | 21 MB    00:20
Extra Packages for Enterprise Linux 9 openh264 (From Cisco) - x86_64 1.4 kB/s | 993 B    00:00
Rocky Linux 9 - BaseOS                             11 kB/s | 4.1 kB    00:00
Rocky Linux 9 - BaseOS                             5.5 MB/s | 2.2 MB    00:00
Rocky Linux 9 - AppStream                          14 kB/s | 4.5 kB    00:00
Rocky Linux 9 - AppStream                          8.8 MB/s | 7.4 MB    00:00
Rocky Linux 9 - Extras                             9.2 kB/s | 2.9 kB    00:00
Rocky Linux 9 - Extras                             29 kB/s | 14 kB    00:00
Dependencias resueltas.
=====
Paquete           Arquitectura  Versión           Repositorio      Tam.
=====
Instalando:
nfs-utils         x86_64       1:2.5.4-20.el9   baseos           425 k
Instalando dependencias:
rpcbind           x86_64       1.2.6-5.el9      baseos           56 k
Resumen de la transacción
=====
Instalar 2 Paquetes
```

2) Crear el directorio archivo para compartir:

```
[root@server ~]# mkdir /var/archivos
[root@server ~]# chmod 777 /var/archivos
[root@server ~]#
```

3. CONFIGURACIÓN DE NFS

- a) Instalar un servidor NFS en un equipo con SO Rocky Linux 9.3.
- b) Configurar el servidor NFS para compartir algún directorio. A lo largo de estos apartados vamos a evaluar las opciones de exportación. En la configuración inicial podemos compartir sin restricciones un directorio en modo lectura/escritura (/var/archivos, con derechos 777) y otro en modo de sólo lectura (/usr/share/doc).

El archivo /etc/exports define los directorios que serán compartidos con los clientes NFS

Escribimos en ese archivo:

```
/var/archivos *(rw,sync,no_root_squash)
/usr/share/doc *(ro,sync,no_root_squash)
```

- *: cualquier cliente puede acceder a estos directorios.
- rw: /var/archivos se comparte → lectura/escritura
- ro: /usr/share/doc → solo Lectura
- sync: Garantiza que los cambios se escriban de manera síncrona en el servidor NFS.
- no_root_squash: Permite al usuario root en el cliente tener los mismos privilegios que root en el servidor.

Nota: ¡es importante no poner espacios!

Por ejemplo, si añadimos espacios entre los campos, podría interpretarse erróneamente como si /var/archivos estuviera compartido con un cliente llamado *(rw,sync,no_root_squash) y /usr/share/doc con un cliente llamado *(ro,sync,no_root_squash)

3. CONFIGURACIÓN DE NFS

```
/var/archivos *(rw,sync,no_root_squash)
/usr/share/doc *(ro,sync,no_root_squash)
```

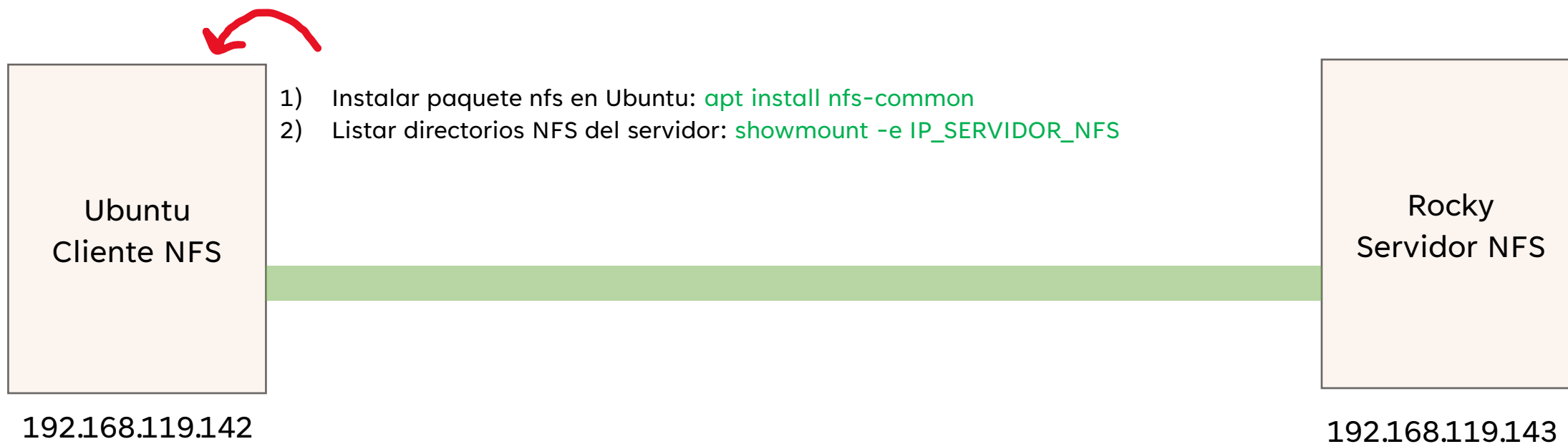
Después aplicar los cambios en la configuración de NFS: `exportfs -ra`

Iniciar servicio NFS: `systemctl start nfs` + `systemctl enable nfs`

```
[root@server ~]# vim /etc/exports
[root@server ~]# [root@server ~]# exportfs -ra
[root@server ~]# systemctl start nfs-server
[root@server ~]# systemctl enable nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.
[root@server ~]# systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
   Drop-In: /run/systemd/generator/nfs-server.service.d
            └─order-with-mounts.conf
   Active: active (exited) since Wed 2024-05-01 13:00:32 CEST; 11s ago
   Main PID: 4048 (code=exited, status=0/SUCCESS)
   CPU: 57ms
```

3. CONFIGURACIÓN DE NFS

c) Utilizar otro equipo como cliente NFS, ahora con SO Ubuntu Server 22.04. Indicar el proceso seguido para la instalación.
Nota: en el lado cliente NO hay que arrancar el servidor NFS. Desde el cliente listar los directorios NFS que comparte el servidor.



3. CONFIGURACIÓN DE NFS

c) Utilizar otro equipo como cliente NFS, ahora con SO Ubuntu Server 22.04. Indicar el proceso seguido para la instalación. Nota: en el lado cliente NO hay que arrancar el servidor NFS. Desde el cliente listar los directorios NFS que comparte el servidor.

```
root@server:~# apt install nfs-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  keyutils libnfsidmap1 rpcbind
Paquetes sugeridos:
  watchdog
Se instalarán los siguientes paquetes NUEVOS:
  keyutils libnfsidmap1 nfs-common rpcbind
```

```
root@server:~# showmount -e 192.168.119.143
Export list for 192.168.119.143:
/usr/share/doc *
/var/archivos *
root@server:~#
```


3. CONFIGURACIÓN DE NFS

d) En el equipo cliente montar (por ejemplo, en /mnt/nfs) el directorio NFS compartido por el servidor en modo lectura/escritura. Añadir un archivo desde el cliente en el recurso compartido y comprobar la propiedad y derechos del mismo, tanto si lo hacemos con un usuario no privilegiado como si usamos el perfil de root. A continuación, modificar las propiedades de compartición (ro, all_squash, no_root_squash, ...) y comprobar su efecto tras añadir nuevos archivos (si tenemos derecho de escritura, el UID de usuario con el que se crean los archivos, y permisos asociados a los mismos).

Montar el directorio nfs: `mkdir /mnt/nfs + mount -o rw IP_SERVIDOR_NFS:/var/archivos /mnt/nfs`

Agregar un archivo desde el cliente: `touch /mnt/nfs/nuevo_archivo.txt`

Comprobar la propiedad y los derechos del archivo (`ls -l /mnt/nfs/nuevo_archivo.txt`)

```
root@server:~# mkdir /mnt/nfs
root@server:~# mount -o rw 192.168.119.143:/var/archivos /mnt/nfs
root@server:~# touch /mnt/nfs/nuevo_archivo.txt
root@server:~# ls -l /mnt/nfs/nuevo_archivo.txt
-rw-r--r-- 1 root root 0 may  1 11:20 /mnt/nfs/nuevo_archivo.txt
```

Pertenece al usuario root y al grupo root, con permisos de lectura y escritura para el usuario root, y solo permisos de lectura para el grupo y otros usuarios.

3. CONFIGURACIÓN DE NFS

d) A continuación, modificar las propiedades de compartición (ro, all_squash, no_root_squash, ...) y comprobar su efecto tras añadir nuevos archivos (si tenemos derecho de escritura, el UID de usuario con el que se crean los archivos, y permisos asociados a los mismos).

Entrar en /etc/exports y modificar /var/archivos

- 1) Cambiar el a modo solo lectura: `/var/archivos *(ro,sync,no_root_squash)` (modificamos la línea)
- 2) Cambiar a all_squash (mapeo de todos los usuarios a un usuario anónimo): `/var/archivos *(ro,sync,all_squash)`
- 3) Cambiar a root_squash (no_root_squah → permite que el usuario root tenga los mismos privilegios en el cliente):
`/var/archivos *(rw,sync,root_squash)`

Después de cada cambio:

- Servidor NFS: `exportfs -ra + systemctl restart nfs-server`
- Cliente (Ubuntu): volver a montar el directorio `umount /mnt/nfs + mount -o rw/ro ip_servidor_nfs:/var/archivos /mnt/nfs`

3. CONFIGURACIÓN DE NFS

d) A continuación, modificar las propiedades de compartición (ro, all_squash, no_root_squash, ...) y comprobar su efecto tras añadir nuevos archivos (si tenemos derecho de escritura, el UID de usuario con el que se crean los archivos, y permisos asociados a los mismos).

Entrar en /etc/exports y modificar /var/archivos

1) Cambiar el a modo solo lectura: `/var/archivos *(ro,sync,no_root_squash)` (modificamos la línea)

En Rocky:

```
/var/archivos *(ro,sync,no_root_squash)
/usr/share/doc *(ro,sync,no_root_squash)
```

```
[root@server ~]# exportfs -ra
[root@server ~]# systemctl restart nfs
Failed to restart nfs.service: Unit nfs.service not found.
[root@server ~]# systemctl restart nfs-server
```

En Ubuntu:

```
root@server:~# umount /mnt/nfs
```

```
root@server:~# mount -o ro 192.168.119.143:/var/archivos /mnt/nfs
```

```
root@server:~# touch /mnt/nfs/rchivo2.txt
touch: cannot touch '/mnt/nfs/rchivo2.txt': Read-only file system
```

Comprobación:

No deja crear archivos porque es de lectura solo

3. CONFIGURACIÓN DE NFS

d) A continuación, modificar las propiedades de compartición (ro, all_squash, no_root_squash, ...) y comprobar su efecto tras añadir nuevos archivos (si tenemos derecho de escritura, el UID de usuario con el que se crean los archivos, y permisos asociados a los mismos).

Entrar en /etc/exports y modificar /var/archivos

2) Cambiar a all_squash (mapeo de todos los usuarios a un usuario anónimo): `/var/archivos *(rw, sync, all_squash)`

En Rocky:

```
/var/archivos *(rw, sync, all_squash)
/usr/share/doc *(ro, sync, no_root_squash)
```

```
[root@server ~]# vim /etc/exports
[root@server ~]# exportfs -ra
[root@server ~]# systemctl restart nfs-server
```

```
#/var/archivos *(rw, sync, root_squash)
/var/archivos *(rw, sync, all_squash)
/usr/share/doc *(ro, sync, no_root_squash)
```

En Ubuntu:

```
root@server:~# mount -o rw 192.168.119.143:/var/archivos /mnt/nfs
root@server:~# touch /mnt/nfs/archivo_allSquash.txt
root@server:~# ls -la /mnt/nfs/archivo_allSquash.txt
-rw-r--r-- 1 nobody nogroup 0 may  6 15:31 /mnt/nfs/archivo_allSquash.txt
```

all_squash mapea todos los usuarios a un usuario anónimo → nobody

3. CONFIGURACIÓN DE NFS

d) A continuación, modificar las propiedades de compartición (ro, all_squash, no_root_squash, ...) y comprobar su efecto tras añadir nuevos archivos (si tenemos derecho de escritura, el UID de usuario con el que se crean los archivos, y permisos asociados a los mismos).

Entrar en /etc/exports y modificar /var/archivos

- 3) Cambiar a root_squash (no_root_squash → permite que el usuario root tenga los mismos privilegios en el cliente):
/var/archivos *(rw,sync,root_squash)

En Rocky:

```
/var/archivos *(rw,sync,root_squash)
/usr/share/doc *(ro,sync,no_root_squash)
```

```
[root@server ~]# systemctl restart nfs-server
[root@server ~]# vim /etc/exports
[root@server ~]# exportfs -ra
[root@server ~]# systemctl restart nfs-server
```

En Ubuntu:

```
root@server:~# umount /mnt/nfs
```

```
root@server:~# mount -o rw 192.168.119.143:/var/archivos /mnt/nfs
```

```
root@server:~# touch /mnt/nfs/archivo4.txt
root@server:~# ls -l /mnt/nfs
total 0
-rw-r--r-- 1 nobody nogroup 0 may  1 12:04 archivo4.txt
-rw-r--r-- 1 root    root    0 may  1 11:20 nuevo_archivo.txt
```

3. CONFIGURACIÓN DE NFS

d) A continuación, modificar las propiedades de compartición (ro, all_squash, no_root_squash, ...) y comprobar su efecto tras añadir nuevos archivos (si tenemos derecho de escritura, el UID de usuario con el que se crean los archivos, y permisos asociados a los mismos).

```
root@server:~# touch /mnt/nfs/archivo4.txt
root@server:~# ls -l /mnt/nfs
total 0
-rw-r--r-- 1 nobody nogroup 0 may  1 12:04 archivo4.txt
-rw-r--r-- 1 root    root    0 may  1 11:20 nuevo_archivo.txt
```

"root_squash" impide que el usuario root en el cliente tenga privilegios de root en el servidor remoto ("aplasta" los privilegios de root)

Como en Ubuntu estoy en un usuario root → el sistema NFS asigna al usuario root del cliente un UID y GID de "nobody" y "nogroup", respectivamente, para todas las operaciones realizadas en ese directorio.

3. CONFIGURACIÓN DE NFS

e) Limitar las direcciones IP de acceso y comprobar que el recurso deja de estar disponible desde otras direcciones IP distintas.

```
#/var/archivos *(rw,sync,root_squash)
/var/archivos 192.168.119.142(rw,sync,no_root_squash)
/usr/share/doc *(ro,sync,no_root_squash)
```

Permitirá el acceso de escritura al cliente ubicado en la dirección IP 192.168.119.142, mientras que otros clientes no tendrán acceso

```
[root@server ~]# vim /etc/exports
[root@server ~]# exportfs -ra
[root@server ~]# systemctl restart nfs-server
```

Ubuntu
Cliente NFS

```
root@server:~# umount /mnt/nfs
root@server:~# mount -o rw 192.168.119.143:/var/archivos /mnt/nfs
root@server:~# |
```

Desde el servidor Ubuntu me deja

3. CONFIGURACIÓN DE NFS

e) Limitar las direcciones IP de acceso y comprobar que el recurso deja de estar disponible desde otras direcciones IP distintas.

Voy a cambiar la IP para que no permita el acceso desde Ubuntu

```
#/var/archivos *(rw, sync, root_squash)
/var/archivos 192.168.119.200(rw, sync, no_root_squash)
/usr/share/doc *(ro, sync, no_root_squash)
```

```
[root@server ~]# vim /etc/exports
[root@server ~]# exportfs -ra
[root@server ~]# systemctl restart nfs-server
```

Ubuntu
Cliente NFS

```
root@server:~# mount -o rw 192.168.119.143:/var/archivos /mnt/nfs
Created symlink /run/systemd/system/remote-fs.target.wants/rpc-statd.service → /lib/systemd/system/rpc-statd.service.
mount.nfs: mounting 192.168.119.143:/var/archivos failed, reason given by server: No such file or directory
root@server:~#
```

Desde el servidor Ubuntu no permite el acceso

3. CONFIGURACIÓN DE NFS

f) Indicar el proceso de instalación del servidor NFS en el SO Ubuntu Server 22.04, y del cliente en un SO Rocky Linux 9.3. .

Servidor

Ubuntu:
`apt install nfs-kernel-server`

Rocky:
`yum install nfs-utils`

Ya
demostrado

Cliente

Ubuntu:
`apt install nfs-common`

Ya
demostrado

Rocky:
`yum install nfs-utils`

Ya
demostrado

3. CONFIGURACIÓN DE NFS

f) Indicar el proceso de instalación del **servidor NFS en el SO Ubuntu Server 22.04**, y del cliente en un SO Rocky Linux 9.3. .

Servidor

Ubuntu:
apt install nfs-kernel-server

```
root@server:~# apt install nfs-kernel-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  nfs-kernel-server
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 93 no actualizados.
```

3. CONFIGURACIÓN DE NFS

f) Indicar el proceso de instalación del servidor NFS en el SO Ubuntu Server 22.04, y del **cliente en un SO Rocky Linux 9.3.** .

Cliente

Rocky:
yum install nfs-utils

Ya
demostrado

```
[root@server archivos]# yum install nfs-utils
Error al cargar el complemento "config_manager": '*prog'
Extra Packages for Enterprise Linux 9 - x86_64          49 kB/s | 38 kB      00:00
Extra Packages for Enterprise Linux 9 - x86_64          4.5 MB/s | 21 MB     00:04
Rocky Linux 9 - BaseOS                                   4.5 kB/s | 4.1 kB    00:00
Rocky Linux 9 - AppStream                               9.8 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras                                  6.7 kB/s | 2.9 kB    00:00
El paquete nfs-utils-1:2.5.4-20.el9.x86_64 ya está instalado.
Dependencias resueltas.
Nada por hacer.
¡Listo!
```

ÍNDICE

- 2. Almacenamiento Local
- 3. Configuración de NFS
- 4. **Configuración de SAMBA**
- 5. Configuración de iSCSI

4. CONFIGURACIÓN DE SAMBA

- a) Instalar el servidor Samba en una máquina con SO Rocky Linux 9.3. y arrancar el servicio. Para comprobar el servicio usaremos nuestro propio equipo anfitrión (Windows, Mac o Linux) o un cliente Samba instalado en otra máquina virtual Linux.
- b) Crear un nuevo usuario de Samba a partir de un usuario del sistema Linux ya existente. A continuación, configurar un recurso compartido en el servidor Linux. Se denominará [shared] y permitirá el acceso a la carpeta /var/shared con derechos de escritura. El recurso estará visible y sólo será accesible para el usuario anteriormente creado
- c) Verificar que es posible trabajar sobre el recurso compartido accediendo desde una máquina cliente que cuente con un cliente SMB y validándose con el usuario samba previamente creado.
- d) Crear un nuevo recurso [public] que permitirá el acceso al directorio /var/public (situar en este directorio algún archivo) con derechos de lectura para todos los usuarios. Comprobar que es posible acceder a este directorio compartido y su contenido sin necesidad de autenticación.

4. CONFIGURACIÓN DE SAMBA

- a) Instalar el servidor Samba en una máquina con SO Rocky Linux 9.3. y arrancar el servicio. Para comprobar el servicio usaremos nuestro propio equipo anfitrión (Windows, Mac o Linux) o un cliente Samba instalado en otra máquina virtual Linux.

`yum install samba + systemctl start smb + systemctl enable smb`

```
Actualizado:
libsmbclient-4.18.6-102.el9_3.x86_64
samba-client-4.18.6-102.el9_3.x86_64
samba-common-4.18.6-102.el9_3.noarch
Instalado:
libnetapi-4.18.6-102.el9_3.x86_64
samba-common-tools-4.18.6-102.el9_3.x86_64
samba-ldb-ldap-modules-4.18.6-102.el9_3.x86_64
libwbclient-4.18.6-102.el9_3.x86_64
samba-client-libs-4.18.6-102.el9_3.x86_64
samba-common-libs-4.18.6-102.el9_3.x86_64
samba-4.18.6-102.el9_3.x86_64
samba-dcerpc-4.18.6-102.el9_3.x86_64
samba-libs-4.18.6-102.el9_3.x86_64
```

```
[root@server ~]# systemctl start smb
[root@server ~]# systemctl enable smb
Created symlink /etc/systemd/system/multi-user.target.wants/smb.service → /usr/lib/systemd/system/smb.service.
[root@server ~]# systemctl status smb
● smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled; preset: disabled)
   Active: active (running) since Thu 2024-05-02 12:05:58 CEST; 12s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 5442 (smbd)
```

4. CONFIGURACIÓN DE SAMBA

- a) Instalar el servidor Samba en una máquina con SO Rocky Linux 9.3. y arrancar el servicio. Para comprobar el servicio usaremos nuestro propio equipo anfitrión (Windows, Mac o Linux) o un cliente Samba instalado en otra máquina virtual Linux.

Comprobar desde un cliente (`yum install samba-client`)

```
root@192.168.119.138's password:
Last login: Mon Mar 25 17:18:53 2024 from 192.168.119.1
[root@server ~]# yum install samba-client
[root@server ~]# smbclient -L 192.168.119.143
Password for [SAMBA\root]:
Anonymous login successful

  Sharename      Type      Comment
  -----
  print$         Disk      Printer Drivers
  IPC$           IPC       IPC Service (Samba 4.18.6)
SMB1 disabled -- no workgroup available
[root@server ~]# |
```

Otra máquina Rocky en mi caso

Acceso a los recursos compartidos del servidor Samba desde el cliente :

`smbclient -L IP_SERVIDOR_SAMBA`

4. CONFIGURACIÓN DE SAMBA

b) Crear un nuevo usuario de Samba a partir de un usuario del sistema Linux ya existente. A continuación, configurar un recurso compartido en el servidor Linux. Se denominará [shared] y permitirá el acceso a la carpeta /var/shared con derechos de escritura. El recurso estará visible y sólo será accesible para el usuario anteriormente creado

palomaSamba

Usaremos este usuario

```
[root@server samba]# smbpasswd -a palomaSamba
New SMB password:
Retype new SMB password:
Added user palomaSamba.
```

Creamos el usuario `smbpasswd -a <<usuario>>`

Crear recurso compartido:

- Crear carpeta: `mkdir /var/shared`
- Establecer permisos : `chown palomaSamba:palomaSamba /var/shared`
- Configurar el recurso compartido en `/etc/samba/smb.conf`

4. CONFIGURACIÓN DE SAMBA

b) Crear un nuevo usuario de Samba a partir de un usuario del sistema Linux ya existente. A continuación, configurar un recurso compartido en el servidor Linux. Se denominará [shared] y permitirá el acceso a la carpeta /var/shared con derechos de escritura. El recurso estará visible y sólo será accesible para el usuario anteriormente creado

```
[root@server samba]# mkdir /var/shared
[root@server samba]# chown palomaSamba:palomaSamba /var/shared
[root@server samba]#
```

/etc/samba/smb.conf

```
[shared]
    comment = Carpeta compartida PR8
    path = /var/shared
    valid users = palomaSamba
    writable = yes
    browseable = yes
```

Writable: con derechos de escritura

Browsable: recurso compartido será visible cuando los usuarios naveguen por la red

```
[root@server samba]# systemctl restart smb
```

4. CONFIGURACIÓN DE SAMBA

c) Verificar que es posible trabajar sobre el recurso compartido accediendo desde una máquina cliente que cuente con un cliente SMB y validándose con el usuario samba previamente creado.

Con la máquina Rocky cliente samba del apartado a:

`smbclient -L IP_SERVIDOR_SAMBA -U palomaSamba`

```
[root@server ~]# smbclient -L 192.168.119.143 -U palomaSamba
Password for [SAMBA\palomaSamba]:

      Sharename      Type      Comment
      -
      print$         Disk      Printer Drivers
      shared          Disk      Carpeta compartida PR8
      IPC$            IPC       IPC Service (Samba 4.18.6)
      palomaSamba     Disk      Home Directories
SMB1 disabled -- no workgroup available
[root@server ~]# |
```

4. CONFIGURACIÓN DE SAMBA

d) Crear un nuevo recurso [public] que permitirá el acceso al directorio /var/public (situar en este directorio algún archivo) con derechos de lectura para todos los usuarios. Comprobar que es posible acceder a este directorio compartido y su contenido sin necesidad de autenticación.

1) `mkdir /var/public`

2) Config public en /etc/samba/smb.conf → `public = yes` + `guest ok = yes` (Permite el acceso de invitado al recurso compartido sin necesidad de autenticación)

3) `systemctl restart smb`

```
[root@server samba]# mkdir /var/public
[root@server samba]# vim /etc/samba/smb.conf
[root@server samba]# systemctl restart smb
```

Comprobar que es posible acceder a este directorio compartido y su contenido sin necesidad de autenticación:

`smbclient IP_SERVIDOR_SAMBA/public`

```
[public]
        comment = Carpeta pública PR8
        path = /var/public
        public = yes
        guest ok = yes
```

Por defecto:

- **Writable:** con derechos de escritura
- **Browsable:** yes

4. CONFIGURACIÓN DE SAMBA

d) Crear un nuevo recurso [public] que permitirá el acceso al directorio /var/public (situar en este directorio algún archivo) con derechos de lectura para todos los usuarios. Comprobar que es posible acceder a este directorio compartido y su contenido sin necesidad de autenticación.

Comprobar que es posible acceder a este directorio compartido y su contenido sin necesidad de autenticación:

`smbclient IP_SERVIDOR_SAMBA/public`

```
[root@server ~]# smbclient //192.168.119.143/public
Password for [SAMBA\root]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> |
```

Sin poner la contraseña (pulsando Enter)

ÍNDICE

- 2. Almacenamiento Local
- 3. Configuración de NFS
- 4. Configuración de SAMBA
- 5. **Configuración de iSCSI**

5. CONFIGURACIÓN DE ISCSI

Vamos a trabajar con dos máquinas virtuales Rocky Linux 9.3. La primera actuará como un iniciador iSCSI y la segunda como nodo target, que proporcionará una unidad de almacenamiento en modo bloque (un volumen lógico LVM en este ejemplo)

- Configurar la máquina virtual que actuará como servidor iSCSI (target). Contará con un segundo disco duro de 20 GB (sdb) gestionado con LVM, en el que configuraremos un volumen lógico de 10 GB y nombre LV01. Arrancar la máquina y configurar el volumen lógico.
- Usar el shell targetcli para configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01. La nueva unidad no tendrá restricciones de acceso (no requerirá autenticación, ni tampoco limitaremos la IP desde la que se podrá conectar al recurso). Identificar el iqn del recurso creado.
- Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete iscsi-initiator-utils y configurar el nombre iqn del recurso en el archivo initiatorname.iscsi. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos ext4 en esta nueva unidad y montarlo en el directorio /mnt/iscsi.

Iniciador iSCSI
192.168.119.135



Nodo target
192.168.119.138

5. CONFIGURACIÓN DE ISCSI

- a) Configurar la máquina virtual que actuará como servidor iSCSI (target). Contará con un segundo disco duro de 20 GB (sdb) gestionado con LVM, en el que configuraremos un volumen lógico de 10 GB y nombre LV01. Arrancar la máquina y configurar el volumen lógico.

Nodo target
192.168.119.138

Añadimos un segundo disco de 20GB

 Hard Disk (SATA)	20 GB
 New Hard Disk (SCSI)	20 GB

reboot

```
^[[A[root@server ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   20G  0 disk
├─sda1       8:1    0    1G  0 part /boot
└─sda2       8:2    0   19G  0 part
   └─rl-root 253:0    0   17G  0 lvm  /
      └─rl-swap 253:1    0    2G  0 lvm  [SWAP]
sdb          8:16   0   20G  0 disk
sr0         11:0    1 1024M  0 rom
```

Configurar volumen lógico de 10GB y nombre LV01:

- 1) Crear una partición en el disco sdb: `fdisk /dev/sdb`
- 2) Crear volumen físico: `pvcreate /dev/sdb1`
- 3) Crear un grupo de volúmenes LVM: `vgcreate vg-iscsi /dev/sdb1`
- 4) Crear el volumen lógico: `lvcreate -L 10G -n LV01 vg-iscsi`
- 5) Formatea el volumen lógico (suponemos que con ext4 como en el apartado 3 de almacenamiento local): `mkfs.ext4 /dev/vg-iscsi/LV01`
- 6) Montar el volumen lógico:
`mkdir /mnt/LV01 + mount /dev/vg-iscsi/LV01 /mnt/LV01`
- 7) Añadirlo a `/etc/fstab` (para asegurar la persistencia)
`UUID=<UUID_del_sistema_de_archivos> /mnt/LV01 ext4 defaults 0 0`
- 8) Reiniciar Máquina

5. CONFIGURACIÓN DE ISCSI

- a) Configurar la máquina virtual que actuará como servidor iSCSI (target). Contará con un segundo disco duro de 20 GB (sdb) gestionado con LVM, en el que configuraremos un volumen lógico de 10 GB y nombre LV01. Arrancar la máquina y configurar el volumen lógico.

Nodo target
192.168.119.138

Creamos el volumen lógico:

```
[root@server ~]# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created.
```

Creamos un grupo de volúmenes:

```
[root@server ~]# vgcreate vg-iscsi /dev/sdb1
Volume group "vg-iscsi" successfully created
```

Creamos una partición del disco (de los 20GB) para poder crear un volumen lógico sobre ella:

`fdisk /dev/sdb`

```
Orden (m para obtener ayuda): o
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0x8e096a9e.

Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1):
Primer sector (2048-41943039, valor predeterminado 2048):
Último sector, +/-sectores o +/-tamaño{K,M,G,T,P} (2048-41943039, valor predeterminado 41943039):

Crea una nueva partición 1 de tipo 'Linux' y de tamaño 20 GiB.

Orden (m para obtener ayuda): w
Se ha modificado la tabla de particiones.
Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.
```


5. CONFIGURACIÓN DE ISCSI

- a) Configurar la máquina virtual que actuará como servidor iSCSI (target). Contará con un segundo disco duro de 20 GB (sdb) gestionado con LVM, en el que configuraremos un volumen lógico de 10 GB y nombre LV01. Arrancar la máquina y configurar el volumen lógico.

Nodo target
192.168.119.138

Crear el volumen lógico

```
Volume group 'vg-iscsi' successfully created  
[root@server ~]# lvcreate -L 10G -n LV01 vg-iscsi  
Logical volume 'LV01' created.
```

Montar el volumen lógico:

```
[root@server ~]# mkdir /mnt/LV01  
[root@server ~]# mount /dev/vg-iscsi/LV01 /mnt/LV01
```

Formatear el volumen lógico (ext4)

```
[root@server ~]# mkfs.ext4 /dev/vg-iscsi/LV01  
mke2fs 1.46.5 (30-Dec-2021)  
Se está creando un sistema de ficheros con 2621440 bloques de 4k y 655360 nodos-i  
UUID del sistema de ficheros: c0f12936-9fa9-4cb1-9e2d-9c21cf54221d  
Respaldo del superbloque guardados en los bloques:  
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632  
  
Reservando las tablas de grupo: hecho  
Escribiendo las tablas de nodos-i: hecho  
Creando el fichero de transacciones (16384 bloques): hecho  
Escribiendo superbloques y la información contable del sistema de ficheros: hecho
```

UUID del sistema de ficheros: c0f12936-9fa9-4cb1-9e2d-9c21cf54221d

5. CONFIGURACIÓN DE ISCSI

- a) Configurar la máquina virtual que actuará como servidor iSCSI (target). Contará con un segundo disco duro de 20 GB (sdb) gestionado con LVM, en el que configuraremos un volumen lógico de 10 GB y nombre LV01. Arrancar la máquina y configurar el volumen lógico.

Nodo target
192.168.119.138

Finalmente añadimos una entrada del volumen lógico a **/etc/fstabs** (para asegurar la persistencia)

```
UUID=c0f12936-9fa9-4cb1-9e2d-9c21cf54221d /mnt/LV01 ext4 defaults 0 0
```

Reiniciamos y verificamos:

```
[root@server ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                 8:0    0   20G  0 disk
├─sda1                             8:1    0    1G  0 part /boot
└─sda2                             8:2    0   19G  0 part
   ├─rl-root                       253:0    0   17G  0 lvm  /
   └─rl-swap                       253:1    0    2G  0 lvm  [SWAP]
sdb                                 8:16    0   20G  0 disk
├─sdb1                             8:17    0   20G  0 part
└─vg--iscsi-LV01                 253:2    0    10G  0 lvm
sr0                                11:0    1 1024M  0 rom
```

5. CONFIGURACIÓN DE ISCSI

b) Usar el shell targetcli para configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01. La nueva unidad no tendrá restricciones de acceso (no requerirá autenticación, ni tampoco limitaremos la IP desde la que se podrá conectar al recurso). Identificar el iqn del recurso creado.

Abrir shel de targetcli (**targetcli**):

```
[root@server ~]# targetcli
Warning: Could not load preferences file /root/.targetcli/prefs.bin.
targetcli shell version 2.1.53
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> |
```

Importante:

systemctl enable target

```
[root@server ~]# systemctl enable target
Created symlink /etc/systemd/system/multi-user.target
.wants/target.service → /usr/lib/systemd/system/targe
t.service.
```

Configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01:

- 1) Crear un nuevo backstore para LV01
- 2) Crear un nuevo iSCSI target
- 3) Dentro del iSCSI target crear un LUN asociado al backstore creado
- 4) Configura el iSCSI target para no requerir autenticación ni limitar las IP de acceso

5. CONFIGURACIÓN DE ISCSI

b) Usar el shell targetcli para configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01. La nueva unidad no tendrá restricciones de acceso (no requerirá autenticación, ni tampoco limitaremos la IP desde la que se podrá conectar al recurso). Identificar el iqn del recurso creado.

Configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01:

1) Crear un nuevo backstore para LV01

```
cd /backstores/block
```

```
create lv01_block /dev/vg-iscsi/LV01
```

2) Crear un nuevo iSCSI target

```
cd /iscsi
```

```
create
```

```
/> cd /backstores/block  
/backstores/block> create lv01_block /dev/vg-iscsi/LV01  
Created block storage object lv01_block using /dev/vg-iscsi/LV01.
```

5. CONFIGURACIÓN DE ISCSI

b) Usar el shell targetcli para configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01. La nueva unidad no tendrá restricciones de acceso (no requerirá autenticación, ni tampoco limitaremos la IP desde la que se podrá conectar al recurso). Identificar el iqn del recurso creado.

2) Crear un nuevo iSCSI target

`cd /iscsi`

`create`

```
/iscsi> create
Created target iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
```

```
/iscsi> ls
o- iscsi ..... [Targets: 1]
  o- iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2 ..... [TPGs: 1]
    o- tpg1 ..... [no-gen-acls, no-auth]
      o- acls ..... [ACLs: 0]
      o- luns ..... [LUNs: 0]
      o- portals ..... [Portals: 1]
        o- 0.0.0.0:3260 ..... [OK]
/iscsi>
```

5. CONFIGURACIÓN DE ISCSI

b) Usar el shell targetcli para configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01. La nueva unidad no tendrá restricciones de acceso (no requerirá autenticación, ni tampoco limitaremos la IP desde la que se podrá conectar al recurso). Identificar el iqn del recurso creado.

Una vez tenemos el Backstore y el Target:

3) Dentro del iSCSI target crear un LUN asociado al backstore creado (lv01_block)

Entramos en luns

```
/iscsi/iqn.20...1f692efbb93e> cd tpg1
/iscsi/iqn.20...2efbb93e/tpg1> cd luns
/iscsi/iqn.20...93e/tpg1/luns> ls
o- luns .....
```

Creamos una unidad lógica iSCSI (LUN): `create /backstores/block/lv01_block`

```
/iscsi/iqn.20...948/tpg1/luns> create /backstores/block/lv01_block
Created LUN 0.
/iscsi/iqn.20...948/tpg1/luns> ls
o- luns ..... [LUNs: 1]
  o- lun0 ..... [block/lv01_block (/dev/vg-iscsi/LV01) (default_tg_pt_gp)]
```

5. CONFIGURACIÓN DE ISCSI

b) Usar el shell targetcli para configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01. La nueva unidad no tendrá restricciones de acceso (no requerirá autenticación, ni tampoco limitaremos la IP desde la que se podrá conectar al recurso). Identificar el iqname del recurso creado.

Una vez tenemos el Backstore, el Target y hemos creado un LUN asociado al backstore:

4) Configura el iSCSI target para no requerir autenticación ni limitar las IP de acceso

```
/iscsi/iqn.20...4aef9948/tpg1> ls
o- tpg1 ..... [no-gen-acls, no-auth]
  o- acls ..... [ACLs: 0]
  o- luns ..... [LUNs: 1]
    | o- lun0 ..... [block/lv01_block (/dev/vg-iscsi/LV01) (default_tg_pt_gp)]
  o- portals ..... [Portals: 1]
    o- 0.0.0.0:3260 ..... [OK]
/iscsi/iqn.20...4aef9948/tpg1> |
```

La configuración indica que no se acepta la conexión desde cualquier IP (0.0.0.0), configuraremos el acceso si autenticación

```
/iscsi/iqn.20...2efbb93e/tpg1> set auth userid=None
Parameter userid is now 'None'.
/iscsi/iqn.20...2efbb93e/tpg1> set auth password=None
Parameter password is now 'None'.
```

```
/iscsi> set discovery_auth enable = 0
Parameter enable is now 'False'.
```

5. CONFIGURACIÓN DE ISCSI

b) Usar el shell targetcli para configurar una unidad lógica iSCSI (LUN) sobre el volumen LV01. La nueva unidad no tendrá restricciones de acceso (no requerirá autenticación, ni tampoco limitaremos la IP desde la que se podrá conectar al recurso). Identificar el iqn del recurso creado.

Queda configurado la unidad lógica iSCSI (LUN) sobre el volumen LV01
Verificamos:

```
cat /etc/target/saveconfig.json
```

```
{
  "portals": [
    {
      "ip_address": "0.0.0.0",
      "iser": false,
      "offload": false,
      "port": 3260
    }
  ],
  "tag": 1
}

qn": "iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2"
```

```
"tpgs": [
  {
    "attributes": {
      "authentication": 0,
      "cache_dynamic_acls": 0,
      "default_cmdsn_depth": 64,
      "default_ernl": 0,
      "demo_mode_discovery": 1,
      "demo_mode_write_protect": 1,
      "fabric_prot_type": 0,
      "generate_node_acls": 0,
      "login_keys_workaround": 1,
      "login_timeout": 15,
      "netif_timeout": 2,
      "prod_mode_write_protect": 0,
      "t10_pi": 0,
      "tpg_enabled_sendtargets": 1
    }
  }
]
```

```
"luns": [
  {
    "alias": "b1cb23bb10",
    "alua_tg_pt_gp_name": "default_tg_pt_gp",
    "index": 0,
    "storage_object": "/backstores/block/lv01_block"
  }
]
```


5. CONFIGURACIÓN DE ISCSI

```
/iscsi> set discovery_auth enable = 0  
Parameter enable is now 'False'.
```

→ autenticación de descubrimiento en el servidor iSCSI

```
/iscsi/iqn.20...23d/tpg1/luns> cd ../portals/  
/iscsi/iqn.20.../tpg1/portals> create  
Using default IP port 3260  
Binding to INADDR_ANY (0.0.0.0)  
Created network portal 0.0.0.0:3260.
```

→ Crear un portal (no hecho en esta práctica)

```
/iscsi/iqn.20...f5a6fe5d/tpg1> set attribute authentication = 0  
Parameter authentication is now '0'.
```

→

Dentro de TPG (no cambia nada realmente, está a 0 por defecto)

```
/iscsi/iqn.20...f5a6fe5d/tpg1> set attribute demo_mode_write_protect = 0  
Parameter demo_mode_write_protect is now '0'.
```

→

Activar o desactivar el modo de protección contra escritura en modo de demostración

Las ACLs se generan automáticamente para los nodos que se conectan al TPG.

←

```
/iscsi/iqn.20...f5a6fe5d/tpg1> set attribute generate_node_acls = 1  
Parameter generate_node_acls is now '1'.
```

Cachear dinámicamente las ACLs (Listas de Control de Acceso) para los nodos (iniciadores) que se conectan al grupo de destino.

←

```
/iscsi/iqn.20...f5a6fe5d/tpg1> set attribute cache_dynamic_acls = 1  
Parameter cache_dynamic_acls is now '1'.
```

5. CONFIGURACIÓN DE ISCSI

c) Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete `iscsi-initiator-utils` y configurar el nombre `iqn` del recurso en el archivo `initiatorname.iscsi`. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Iniciador iSCSI
192.168.119.135

`yum install iscsi-initiator-utils`

Obtener iqns del servidor: `iscsiadm -m discovery -t sendtargets -p IP_SERVIDOR_ISCSI`

```
[root@server ~]# iscsiadm -m discovery -t sendtargets -p 192.168.119.138
192.168.119.138:3260,1 iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2
```

5. CONFIGURACIÓN DE ISCSI

c) Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete `iscsi-initiator-utils` y configurar el nombre `iqn` del recurso en el archivo `initiatorname.iscsi`. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Iniciador iSCSI
192.168.119.135

Añadir en `initiatorname.iscsi` el `iqn` del recurso: `vim /etc/iscsi/initiatorname.iscsi`

```
InitiatorName=iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2
```

5. CONFIGURACIÓN DE ISCSI

c) Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete `iscsi-initiator-utils` y configurar el nombre `iqn` del recurso en el archivo `initiatorname.iscsi`. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Iniciador iSCSI
192.168.119.135

Conectarse al recurso: `iscsiadm -m node --targetname NOMBRE_IQN--portal IP_SERVIDOR_ISCSI --login`
`iscsiadm -m node --targetname iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2 --portal 192.168.119.138 --login`


```
[root@server ~]# iscsiadm -m node --targetname iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2 --portal 192.168.119.138 --login
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2, portal: 192.168.119.138,3260]
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.server.x8664:sn.2f5fa5f07fa2, portal: 192.168.119.138,3260] successful.
```

5. CONFIGURACIÓN DE ISCSI

c) Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete `iscsi-initiator-utils` y configurar el nombre `iqn` del recurso en el archivo `initiatorname.iscsi`. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Iniciador iSCSI
192.168.119.135

Comprobamos que se ha añadido una nueva unidad de 10Gb al equipo



```
[root@server ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   20G  0 disk
├─sda1       8:1    0    1G  0 part /boot
└─sda2       8:2    0   19G  0 part
   └─rl-root 253:0    0   17G  0 lvm  /
      └─rl-swap 253:1    0    2G  0 lvm  [SWAP]
sdb          8:16    0   10G  0 disk
sr0         11:0    1 1024M  0 rom
```

5. CONFIGURACIÓN DE ISCSI

c) Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete `iscsi-initiator-utils` y configurar el nombre `iqn` del recurso en el archivo `initiatorname.iscsi`. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Iniciador iSCSI
192.168.119.135

Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Repetimos los mismos pasos de siempre:

- 1) Crear un sistema de archivos, esta vez sobre `/dev/sdb`
- 2) Crear el directorio de montaje + montarlo
- 3) Verificación

5. CONFIGURACIÓN DE ISCSI

c) Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete `iscsi-initiator-utils` y configurar el nombre `iqn` del recurso en el archivo `initiatorname.iscsi`. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Iniciador iSCSI
192.168.119.135

1) Crear un sistema de archivos, esta vez sobre `/dev/sdb`

```
Se está creando un sistema de ficheros con 2621440 bloques de 4k y 655360 nodos-i
UUID del sistema de ficheros: e47bb352-3987-4d10-9c51-214aa3543a3b
Respalos del superbloque guardados en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

5. CONFIGURACIÓN DE ISCSI

c) Levantar una segunda máquina virtual que actuará como iniciador iSCSI del recurso exportado por el target. Instalar el paquete `iscsi-initiator-utils` y configurar el nombre `iqn` del recurso en el archivo `initiatorname.iscsi`. Conectar al recurso iSCSI remoto y comprobar que se ha añadido una nueva unidad de 10 GB al equipo. Crear un sistema de archivos `ext4` en esta nueva unidad y montarlo en el directorio `/mnt/iscsi`.

Iniciador iSCSI
192.168.119.135

- 2) Crear el directorio de montaje + montarlo
- 3) Verificación

```
[root@server ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   20G  0 disk
├─sda1       8:1    0    1G  0 part /boot
└─sda2       8:2    0   19G  0 part
   └─rl-root 253:0    0   17G  0 lvm  /
      └─rl-swap 253:1    0    2G  0 lvm  [SWAP]
sdb          8:16   0   10G  0 disk /mnt/iscsi
sr0         11:0    1 1024M  0 rom
```

```
[root@server ~]# mkdir /mnt/iscsi
[root@server ~]# mount /dev/sdb /mnt/iscsi
[root@server ~]# df -h
S.ficheros          Tamaño Usados  Disp Uso% Montado en
devtmpfs             4,0M      0   4,0M  0% /dev
tmpfs                363M      0   363M  0% /dev/shm
tmpfs                146M   5,4M   140M  4% /run
/dev/mapper/rl-root   17G   2,9G   15G  17% /
/dev/sda1            1014M   310M   705M  31% /boot
tmpfs                 73M    4,0K    73M  1% /run/user/0
/dev/sdb              9,8G    24K    9,3G  1% /mnt/iscsi
[root@server ~]#
```