

## **PASOS DE LA CREACIÓN Y CONFIGURACIÓN DEL RAID 10**

1.	Configuración del almacenamiento VM VirtualBox: .....	2
1.1.	Añadir y crear 4 unidades de disco duro virtual .....	2
2.	Pasos para configurar RAID 1 en Ubuntu Server 24.04: .....	4
2.1.	Verifica los discos disponibles (Creados en VBOX). .....	4
2.2.	Crear particiones en los discos .....	5
2.3.	Crear el RAID 10 usando mdadm .....	8
2.4.	Verificar el estado del RAID .....	9
2.5.	Formatear el RAID 1 .....	10
2.6.	Montar el RAID.....	10
2.7.	Verificación del montaje: .....	11
2.8.	Configurar el montaje automático. ....	12
2.9.	Guardar la configuración del RAID .....	13
3.	Pasos para simular un fallo y recuperar el disco: .....	13
3.1.	Añadir un fichero de prueba en el RAID 10: .....	13
3.2.	Verificar el archivo:.....	13
3.3.	Simular el fallo de un disco: .....	14
3.4.	Verificar el estado del RAID después del fallo: .....	14
3.5.	Retirar el disco fallido: .....	14
3.6.	Añadir un disco nuevo:.....	15
3.7.	Verificar la reconstrucción del RAID:.....	15
3.8.	Comprobar que el archivo sigue en el RAID: .....	15

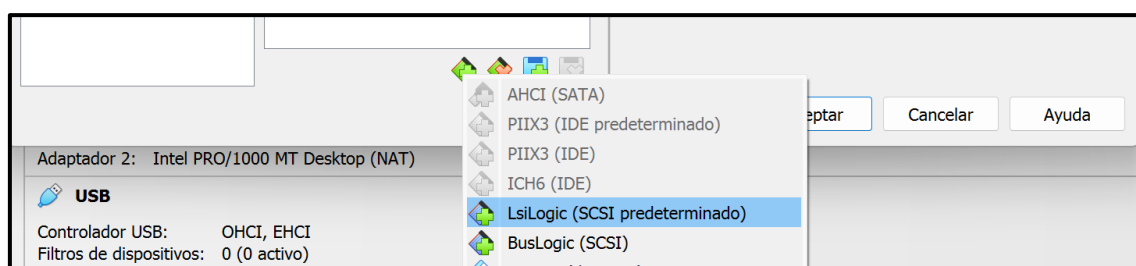
## 1. Configuración del almacenamiento VM VirtualBox:

### 1.1. Añadir y crear 4 unidades de disco duro virtual

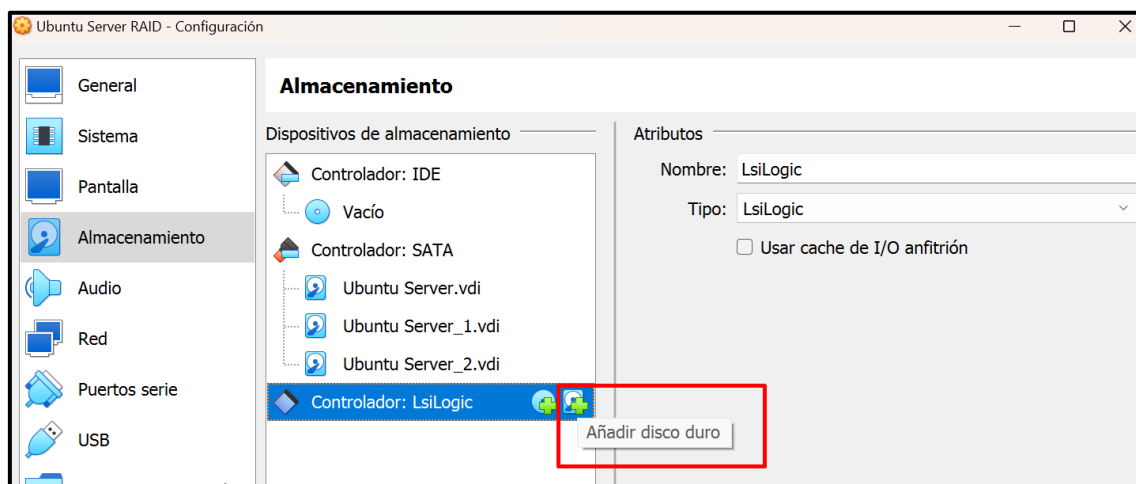
Una vez entendido el uso del RAID 10, procederemos a realizar la creación de los 4 discos virtuales para la máquina virtual.

➤ Para añadir un nuevo disco, he detenido la máquina para luego:

1. Abrir Configuración del menú.
2. Crear un nuevo controlador para guardar los 4 nuevos discos para el raid 10.



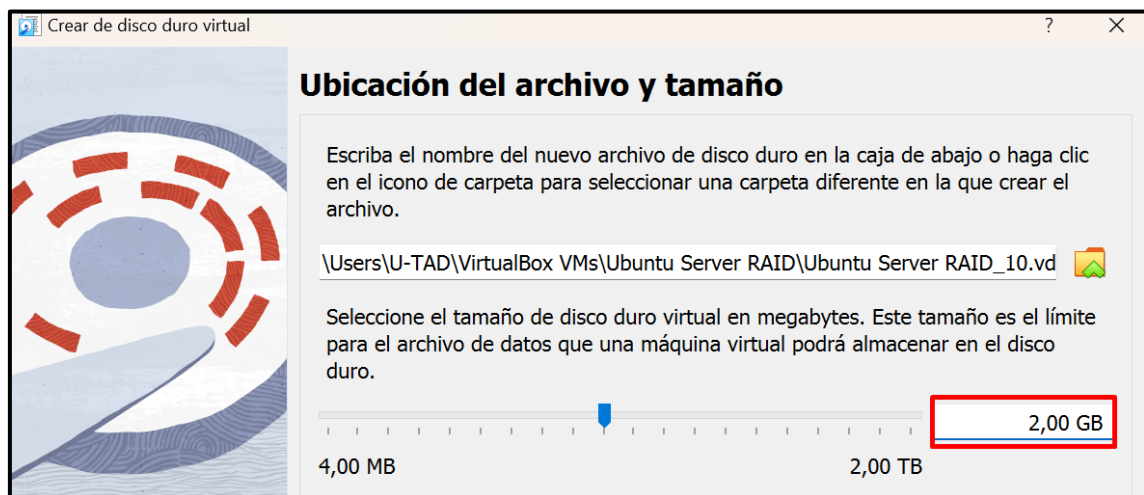
3. Seleccionar la pestaña Almacenamiento y añadir/crear un nuevo disco duro virtual.



- El tipo de archivo de disco virtual que se va a seleccionar será la primera opción: VDI (VirtualBox Disk Image)

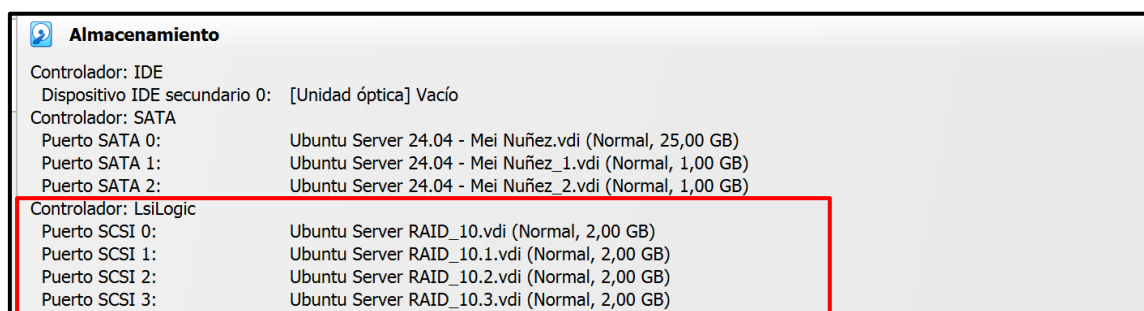


- Por último, se tendrá que elegir el tamaño de disco duro virtual que se le proporcionará para cada disco RAID



- Aquí se puede ver un resumen de los discos creados y configurados para proceder con los siguientes pasos.

Una vez realizados estos pasos aplico los cambios e inicio la máquina para que los cambios surtan efecto.



## 2. Pasos para configurar RAID 1 en Ubuntu Server 24.04:

### 2.1. Verifica los discos disponibles (Creados en VBOX).

Antes de configurar el RAID 10, es necesario verificar que los discos virtuales estén disponibles.

Para identificar los 4 discos nuevos que se han añadido, el comando **fdisk -l** muestra una lista de todos los discos disponibles. Otro comando que se utiliza para listar los discos es: **lsblk**

Por ello, voy a ejecutar los dos comandos **sudo fdisk -l** y **lsblk** para listar los discos y particiones conectados a mi sistema, asegurándome de que los discos que voy a usar para el RAID estén presentes y correctos.

```
sudo fdisk -l
```

```
ubuntu_mei@ubuntu_mei:~$ sudo fdisk -l
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Esta salida nos proporciona una visión general del esquema de particiones del sistema con el nombre del directorio, donde están montados los discos (/dev)

```
Disk /dev/sda: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdc: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdd: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

## lsblk

```
ubuntu_mei@ubuntumei:~$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0     4K  1 loop /snap/bare/5
loop1       7:1      0    74,2M  1 loop /snap/core22/1621
loop2       7:2      0   271,4M  1 loop /snap/firefox/4955
loop3       7:3      0   505,1M  1 loop /snap/gnome-42-2204/176
loop4       7:4      0    91,7M  1 loop /snap/gtk-common-themes/1535
loop5       7:5      0    38,8M  1 loop /snap/snapd/21759
sda         8:0      0     2G  0 disk
sdb         8:16     0     2G  0 disk
sdc         8:32     0     2G  0 disk
sdd         8:48     0     2G  0 disk
sde         8:64     0    25G  0 disk
```

Realizar este paso es fundamental antes de realizar cualquier operación que pueda afectar la estructura del disco.

## 2.2. Crear particiones en los discos

Como los discos aún no tienen particiones, utilizo **sudo fdisk /dev/sdX** donde X es la letra que se le asigna a cada disco para crear las 4 particiones primarias en cada disco.

## sudo fdisk /dev/sdX

Antes de realizar las particiones, podemos ver la ayuda de este comando con los diferentes usos de este.

```
Help:

DOS (MBR)
a  toggle a bootable flag
b  edit nested BSD disklabel
c  toggle the dos compatibility flag

Generic
d  delete a partition
F  list free unpartitioned space
l  list known partition types
n  add a new partition
p  print the partition table
t  change a partition type
v  verify the partition table
i  print information about a partition

Misc
m  print this menu
u  change display/entry units
x  extra functionality (experts only)

Script
I  load disk layout from sfdisk script file
O  dump disk layout to sfdisk script file

Save & Exit
w  write table to disk and exit
q  quit without saving changes

Create a new label
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty MBR (DOS) partition table
s  create a new empty Sun partition table
```

Ahora que ya se tiene una idea de cómo usar este comando, comenzamos a proceder con la partición del primer disco.

### Primer disco:

Tipo de partición → primary (p) | Guardar cambios → tecla w

```
sudo fdisk /dev/sda
```

```
ubuntu_mei@ubuntu:~$ sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0x296c6993.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

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

### Segundo disco:

Tipo de partición → primary (p) | Guardar cambios → tecla w

```
sudo fdisk /dev/sdb
```

```
ubuntu_mei@ubuntu:~$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0xf2ecc911.

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):
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

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

## Tercer disco:

Tipo de partición → primary (p) | Guardar cambios → tecla w

```
sudo fdisk /dev/sdc
```

```
ubuntu_mei@ubuntu:~$ sudo fdisk /dev/sdc
Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0x3e36f59a.

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):
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

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

## Cuarto disco:

Tipo de partición → primary (p) | Guardar cambios → tecla w

```
sudo fdisk /dev/sdd
```

```
ubuntu_mei@ubuntu:~$ sudo fdisk /dev/sdd
Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0x50799787.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

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

Una vez creadas las dos particiones podemos ejecutar el comando lsblk para comprobar que se han creado las 4 particiones con partición primaria.

```
ubuntu_mei@ubuntu:~$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0     4K  1 loop /snap/bare/5
loop1       7:1      0    74,2M  1 loop /snap/core22/1621
loop2       7:2      0   271,4M  1 loop /snap/firefox/4955
loop3       7:3      0   505,1M  1 loop /snap/gnome-42-2204/176
loop4       7:4      0    91,7M  1 loop /snap/gtk-common-themes/1535
loop5       7:5      0    38,8M  1 loop /snap/snapd/21759
sda         8:0      0     2G  0 disk
└─sda1      8:1      0     2G  0 part
sdb         8:16     0     2G  0 disk
└─sdb1      8:17     0     2G  0 part
sdc         8:32     0     2G  0 disk
└─sdc1      8:33     0     2G  0 part
sdd         8:48     0     2G  0 disk
└─sdd1      8:49     0     2G  0 part
```

### 2.3. Crear el RAID 10 usando mdadm

Una vez que los discos están listos, creo el RAID con el uso del paquete mdadm.

La sintaxis general para crear un raid con madam es:

```
sudo mdadm --create /dev/md0 --level=TipoDeRaid --raid-devices=Nºdiscos  
/dev/sdb /dev/sdc....
```

Donde:

- Create /dev/md0 → crea un dispositivo RAID llamado /dev/md0 utilizando los 4 discos especificados.
- Level=TipoDeRaid es el nivel RAID, en este caso raid (0,1,5,6...).
- Raid-devices=Nºdiscos → donde esta sección indica el número de discos que se van a utilizar en el RAID y el directorio donde se han creado cada uno de ellos separado por espacios (/dev/sdb /dev/sdc...)

Entonces la sintaxis final para realizar la configuración del RAID 10 sería:

```
mdadm --create /dev/md0 --level=10 --raid-devices=4 /dev/sda1 /dev/sdb1 /dev/sdc1  
/dev/sdd1
```



Este comando crea el RAID 10, donde los datos se duplicarán en dos discos para asegurar redundancia.

```
ubuntu_mei@ubuntu:~$ sudo mdadm --create /dev/md0 --level=10 --raid-devices=4 /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

## 2.4. Verificar el estado del RAID

Después de crear el RAID, verifico que todo esté correcto ejecutando `cat /proc/mdstat`.

Este comando muestra el estado de los arreglos RAID en tiempo real.

**cat /proc/mdstat**

```
ubuntu_mei@ubuntu:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdd1[3] sdc1[2] sdb1[1] sda1[0]
      4188160 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>
```

También puedo usar `mdadm --detail /dev/md0` para obtener detalles más completos del RAID.

**mdadm --detail /dev/md0**

```
ubuntu_mei@ubuntu:~$ sudo mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Thu Oct  3 11:43:26 2024
    Raid Level : raid10
    Array Size : 4188160 (3.99 GiB 4.29 GB)
  Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
    Raid Devices : 4
    Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Thu Oct  3 11:43:47 2024
      State : clean
    Active Devices : 4
    Working Devices : 4
    Failed Devices : 0
    Spare Devices : 0

    Layout : near=2
    Chunk Size : 512K

Consistency Policy : resync

    Name : ubuntu:0 (local to host ubuntu)
    UUID : 731ce915:57ceea7e:1d2ca18b:5e3745a8
    Events : 17

   Number Major Minor RaidDevice State
    0         8       1        0     active sync set-A   /dev/sda1
    1         8      17        1     active sync set-B   /dev/sdb1
    2         8      33        2     active sync set-A   /dev/sdc1
    3         8      49        3     active sync set-B   /dev/sdd1
```

## 2.5. Formatear el RAID 1

Una vez que el RAID está creado, lo formateo con `mkfs.ext4 /dev/md0`. Este comando crea un sistema de archivos ext4 en el dispositivo RAID, permitiéndome almacenar archivos en él.

**mkfs.ext4 /dev/md0**

```
ubuntu_mei@ubuntumei:~$ sudo mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 261632 4k blocks and 65408 inodes
Filesystem UUID: 96c24100-808c-4f0e-bf21-08bc0cc73d22
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

## 2.6. Montar el RAID

Después de crear el sistema de archivos, montaremos el RAID 10 en un directorio.

Cuando se monta un dispositivo de almacenamiento en un directorio específico, el sistema operativo crea una conexión entre el sistema de archivos del dispositivo de almacenamiento y el árbol de directorios del sistema.

Primero creo un punto de montaje en el directorio `/mnt` con **`sudo mkdir /mnt/raid10`** y monto el RAID con **`sudo mount /dev/md0 /mnt/raid10`**

Directorio → **`sudo mkdir /mnt/raid10`**

Montar el RAID en el directorio → **`sudo mount /dev/md0 /mnt/raid10`**

**`sudo mkdir /mnt/raid10 && sudo mount /dev/md0 /mnt/raid10`**

```
ubuntu_mei@ubuntumei:~$ sudo mkdir /mnt/raid10 && sudo mount /dev/md0 /mnt/raid10
```

Montar el RAID en un directorio es necesario porque permite que el sistema operativo acceda y gestione los archivos y directorios dentro del RAID.

```
ubuntu_mei@ubuntumei:~$ cd /mnt/raid10
ubuntu_mei@ubuntumei:/mnt/raid10$ ls
lost+found
ubuntu_mei@ubuntumei:/mnt/raid10$ |
```

Si volvemos a usar el comando lsblk podemos ver que los 4 discos se han montado correctamente en el directorio creado.

#### lsblk

```
ubuntu_mei@ubuntumei:/mnt/raid10$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
loop0       7:0      0   4K  1 loop  /snap/bare/5
loop1       7:1      0  74,2M  1 loop  /snap/core22/1621
loop2       7:2      0 271,4M  1 loop  /snap/firefox/4955
loop3       7:3      0 505,1M  1 loop  /snap/gnome-42-2204/176
loop4       7:4      0  91,7M  1 loop  /snap/gtk-common-themes/1535
loop5       7:5      0  38,8M  1 loop  /snap/snapd/21759
sda         8:0      0    2G  0 disk
├─sda1      8:1      0    2G  0 part
└─┬md0      9:0      0    4G  0 raid10 /mnt/raid10
sdb         8:16     0    2G  0 disk
├─sdb1      8:17     0    2G  0 part
└─┬md0      9:0      0    4G  0 raid10 /mnt/raid10
sdc         8:32     0    2G  0 disk
├─sdc1      8:33     0    2G  0 part
└─┬md0      9:0      0    4G  0 raid10 /mnt/raid10
sdd         8:48     0    2G  0 disk
├─sdd1      8:49     0    2G  0 part
└─┬md0      9:0      0    4G  0 raid10 /mnt/raid10
sde         8:64     0   25G  0 disk
```

### 2.7. Verificación del montaje:

Podemos verificar que el RAID 10 se ha montado correctamente para ver la lista de sistemas de archivos montados.

#### df -h

```
ubuntu_mei@ubuntumei:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           392M  1,2M  391M   1% /run
/dev/sde2       20G   7,0G  12G  38% /
tmpfs           2,0G    0  2,0G   0% /dev/shm
tmpfs           5,0M   8,0K  5,0M   1% /run/lock
tmpfs           392M   12K  392M   1% /run/user/1000
/dev/md0        3,9G   24K  3,7G   1% /mnt/raid10
```

## 2.8. Configurar el montaje automático.

Para que el RAID se monte automáticamente en cada reinicio, edito el archivo `/etc/fstab` con `sudo nano /etc/fstab` y añado una línea que defina el montaje automático del RAID. Esto asegura que el RAID esté disponible tras reiniciar el servidor.

Fichero → `/etc/fstab`

Modificar Fichero → `nano /etc/fstab`

```
ubuntu_mei@ubuntumei:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/55980fca-be00-4bb4-87f2-771e8d028e55 / ext4 defaults 0 1
/dev/disk/by-uuid/e07b7271-657f-48e2-bbbd-7b3c7c57febd none swap sw 0 0
ubuntu_mei@ubuntumei:~$ sudo nano /etc/fstab
```

Dentro del archivo habría que agregar una línea para definir el montaje automático del RAID 10 con el nombre del dispositivo RAID llamado `/dev/md0` la ruta del punto de montaje en el directorio `/mnt/raid10` el tipo de sistema de archivos `ext4` y utilizando las configuraciones por defecto y sin realizar copias de seguridad (0) ni revisiones automáticas del sistema de archivos (0).

`/dev/md0 /mnt/raid10 ext4 defaults 0 0`

```
ubuntu_mei@ubuntumei: ~  ×  +  ▾
GNU nano 7.2 /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/55980fca-be00-4bb4-87f2-771e8d028e55 / ext4 defaults 0 1
/dev/disk/by-uuid/e07b7271-657f-48e2-bbbd-7b3c7c57febd none swap sw 0 0
/dev/md0 /mnt/raid10 ext4 defaults 0 0
```

## 2.9. Guardar la configuración del RAID

Por último, guardo la configuración del RAID con `mdadm --detail --scan | tee -a /etc/mdadm/mdadm.conf`. Este comando agrega la configuración del RAID al archivo de configuración de mdadm, asegurando que el sistema mantenga la configuración después de reiniciar.

```
sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
```

```
ubuntu_mei@ubuntu_mei:~$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
ARRAY /dev/md0 metadata=1.2 UUID=731ce915:57ceea7e:1d2ca18b:5e3745a8
ubuntu_mei@ubuntu_mei:~$
```

## 3. Pasos para simular un fallo y recuperar el disco:

### 3.1. Añadir un fichero de prueba en el RAID 10:

Primero, vamos a crear un fichero de prueba dentro del directorio donde está montado el RAID 10, para la simulación de fallo.

```
sudo touch /mnt/raid10/prueba_fallo.txt
```

```
ubuntu_mei@ubuntu_mei:~$ sudo touch /mnt/raid10/prueba_fallo.txt
[sudo] password for ubuntu_mei:
```

### 3.2. Verificar el archivo:

Comprobamos que el archivo está creado.

```
ls /mnt/raid10/
```

```
ubuntu_mei@ubuntu_mei:~$ ls /mnt/raid10/
lost+found  prueba_fallo.txt
```

Al ejecutar `ls`, el comando lista los archivos dentro del directorio `/mnt/raid10/`, donde está el fichero ver `prueba_fallo.txt`.

### 3.3. Simular el fallo de un disco:

Para ello escogeremos uno de los discos para simular que uno de ellos falla (en este caso el disco /dev/sdb1).

```
sudo mdadm --manage /dev/md0 --fail /dev/sdb1
```

```
ubuntu_mei@ubuntumei:~$ sudo mdadm --manage /dev/md0 --fail /dev/sdb1
mdadm: set /dev/sdb1 faulty in /dev/md0
```

Este comando marca el disco /dev/sdb1 como fallido dentro del dispositivo RAID /dev/md1.

### 3.4. Verificar el estado del RAID después del fallo:

Comprueba el estado del RAID para asegurarte de que se ha marcado correctamente el fallo y el RAID sigue funcionando:

```
cat /proc/mdstat
```

```
ubuntu_mei@ubuntumei:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdd1[3] sdc1[2] sdb1[1](F) sda1[0]
      4188160 blocks super 1.2 512K chunks 2 near-copies [4/3] [U_UU]

unused devices: <none>
```

Aquí podemos ver que este comando nos muestra el estado actual del RAID, con el disco fallido. (4/3) y sdb1(F)

### 3.5. Retirar el disco fallido:

```
sudo mdadm --manage /dev/md0 --remove /dev/sdb1
```

Una vez simulado el fallo, ahora retiraremos el disco que ha fallado.

```
ubuntu_mei@ubuntumei:~$ sudo mdadm --manage /dev/md0 --remove /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md0
```

### 3.6. Añadir un disco nuevo:

```
sudo mdadm --manage /dev/md0 --add /dev/sdb1
```

Este comando simula la recuperación del sistema, y vuelve a añadir de nuevo el disco al RAID.

```
ubuntu_mei@ubuntumei:~$ sudo mdadm --manage /dev/md0 --add /dev/sdb1
mdadm: added /dev/sdb1
```

### 3.7. Verificar la reconstrucción del RAID:

```
cat /proc/mdstat
```

Volvemos a verificar el estado del RAID para asegurarnos de que el disco se está reconstruyendo.

```
ubuntu_mei@ubuntumei:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdb1[4] sdd1[3] sdc1[2] sda1[0]
      4188160 blocks super 1.2 512K chunks 2 near-copies [4/3] [U_UU]
      [=====>.....] recovery = 58.2% (1221120/2094080) finish=0.0min speed=203520K/sec

unused devices: <none>
```

```
ubuntu_mei@ubuntumei:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdb1[4] sdd1[3] sdc1[2] sda1[0]
      4188160 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>
```

### 3.8. Comprobar que el archivo sigue en el RAID:

```
ls /mnt/raid10/
```

Aquí podemos ver que aun fallar el disco y eliminarlo, el archivo creado sigue en el directorio.

```
ubuntu_mei@ubuntu:~$ sudo mdadm --manage /dev/md0 --remove /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md0
ubuntu_mei@ubuntu:~$ sudo mdadm --manage /dev/md0 --add /dev/sdb1
mdadm: added /dev/sdb1
ubuntu_mei@ubuntu:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdb1[4] sdd1[3] sdc1[2] sda1[0]
      4188160 blocks super 1.2 512K chunks 2 near-copies [4/3] [U_UU]
      [=====>.....] recovery = 58.2% (1221120/2094080) finish=0.0min speed=203520K/sec

unused devices: <none>
ubuntu_mei@ubuntu:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdb1[4] sdd1[3] sdc1[2] sda1[0]
      4188160 blocks super 1.2 512K chunks 2 near-copies [4/3] [U_UU]
      [=====>.....] recovery = 81.4% (1706880/2094080) finish=0.0min speed=213360K/sec

unused devices: <none>
ubuntu_mei@ubuntu:~$ ls /mnt/raid10/
lost+found  prueba_fallo.txt
ubuntu_mei@ubuntu:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdb1[4] sdd1[3] sdc1[2] sda1[0]
      4188160 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>
```