

LINUX : herramientas de administración de RAID

MDADM (***M**ultiple **D**evice **A**dministrator*) es un conjunto de herramientas que son utilizadas en GNU/Linux para la gestión de RAID (***R**edundant **A**rray of **I**ndependent **D**isks*, que se traduce como conjunto redundante de discos independientes) administrado a través de software, distribuido bajo los términos de la Licencia Pública General de GNU versión 2 (GNU/GPLv2).

El RAID por software permite: Una solución de RAID de muy bajo costo pues prescinde del uso de costosas tarjetas RAID y unidades de almacenamiento especiales y que carece de restricciones que regularmente imponen los fabricantes de dispositivos de hardware privativos. Proceso de reconstrucción de arrays en subprocesos.

- Configuración basada sobre el núcleo del sistema.
- Permite portar de manera transparente los arrays entre sistemas GNU/Linux sin necesidad de reconstruir éstos.
- Mejor aprovechamiento de los recursos del sistema pues la reconstrucción de los arrays se hace utilizando recursos libres.
- Soporte para unidades de almacenamiento con capacidad para realizar cambios en caliente (hot-swap).
- Detección automática del número de núcleos del microprocesador a fin de aprovechar mejor los recursos del sistema.

Más información:

URL: kernel.org/pub/linux/utils/raid/mdadm/.

Tipos de arrays soportados.

MDADM permite configurar los siguientes arrays RAID:

- **RAID0:** Incrementa el índice de transmisión del sistema (*throughput*) mediante el uso de varias unidades de disco en paralelo. El rendimiento del sistema con la configuración RAID0 es superior a la que proporcionan las configuraciones RAID1 o RAID5, pero se incrementa la posibilidad de pérdida de datos porque no hay forma de recuperar o reconstruir los datos almacenados en cualquier disco que falle. Requiere 2 discos.
- **RAID1:** Espejo. El array más lento de todos. Requiere 2 unidades de almacenamiento.
- **RAID4:** Igual que RAID0 pero con una unidad de almacenamiento extra para mejorar el desempeño a través de la paridad y al mismo tiempo lograr redundancia. Requiere 4 unidades de almacenamiento.
- **RAID5:** Igual a RAID4 pero con una paridad distribuida en todas las unidades de almacenamiento. Requiere un mínimo de 3 unidades de almacenamiento.
- **RAID6:** Igual que RAID5 pero con dos paridades. Requiere un mínimo de 4 unidades de almacenamiento.
- **RAID10:** Se le conoce mejor como RAID1+0. Combina RAID1 y RAID0. Requiere un mínimo de 4 unidades de almacenamiento.

EJEMPLO CONFIGURAR RAID-1

Ejecutar el siguiente comando, que nos aporta información sobre los dispositivos de almacenamiento de bloques disponibles:

```
lsblk -fm
```

El argumento `f` nos ofrece información sobre los sistemas de archivos instalados y el argumento `m` nos informa de quién es su propietario, su tamaño, etc.

Si la detección ha sido correcta, estaremos listos para comenzar, aunque deberemos instalar `mdadm` (multiple device administrator), que es la herramienta que nos permitirá configurar y administrar dispositivos RAID en Ubuntu.

El paquete `mdadm` tiene algunas dependencias, pero las resolveremos todas escribiendo la siguiente orden en la terminal:

```
sudo apt install mdadm rsync initramfs-tools -y
```

Con el argumento `y` le decimos al comando `apt` que, para todas las preguntas que nos realice, la respuesta será afirmativa. Así, evitamos que nos pregunte más adelante.

Crear particiones para los discos

Usaremos el comando `fdisk` con privilegios de administración:

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

La primera vez que nos pide una orden, pulsamos `la tecla n`, que significa que queremos crear una nueva partición.

A continuación, deberemos indicar que la partición es primaria. Por lo tanto, siguiendo las indicaciones del propio comando, pulsaremos `la tecla p`.

Indicaremos que se trata de la primera partición del dispositivo, para lo que pulsaremos `la tecla 1`.

Después, deberemos indicar dónde comienza la partición (primer sector) y dónde termina (último sector). Como queremos ocupar todo el dispositivo, dejamos los valores predeterminados. Es decir, nos limitamos a pulsar `la tecla Intro dos veces`.

Con esto habremos definido la partición, pero, antes de dar por concluido el trabajo de `fdisk`, debemos indicar el tipo de partición del que se trata: Como respuesta a la siguiente

solicitud de una orden, pulsamos la tecla **t**, para indicar que queremos establecer el tipo de partición.

A continuación, deberíamos indicar el número de la partición sobre la que actuaremos, pero como solo existe una, el comando asume automáticamente el valor 1.

Por último, escribimos las letras **fd**, que indican que se trata de una partición RAID.

Tipos de partición	
Valor	Tipo
82	Swap
83	Linux
86 / 87	NTFS
51 / 52	FAT32
8E	LVM
FD	RAID

A continuación, usamos la orden **p**, para conseguir que fdisk nos muestre un resumen de lo que acabamos de hacer y, si todo es correcto, escribimos la orden **w**, para guardar los cambios.

Para continuar, aplicaremos los mismos cambios sobre la segunda unidad de disco, por lo que escribiremos una nueva orden como esta:

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

... Y seguiremos los mismos pasos del apartado anterior

Una característica de *RAID-1* es que utiliza particiones del mismo tamaño. Si tus discos no son iguales, asegúrate de que las particiones sí lo sean.

De esta forma, podrás utilizar el espacio sobrante en la unidad de mayor tamaño, creando una partición independiente del volumen *RAID*.

Después de esto, puedes volver a utilizar *lsblk* para comprobar la nueva estructura:

```
lsblk -fm
```

para crear el nuevo volumen RAID-1. Lo conseguiremos usando el siguiente comando:

```
sudo mdadm -C /dev/md0 -l raid1 -n 2 /dev/sd[b-c]1
```

Donde el argumento `-c` significa crear y va seguido del archivo de dispositivo que utilizará. El argumento `-1` indica el nivel, que en este caso debe ser `raid1` (o `mirroring`), pero para otros niveles RAID usaremos `raid0`, `raid4`, `raid5`, etc. El argumento `-n` indica el número de dispositivos implicados. Y al final de la línea, debemos identificar dichos dispositivos.

Para asegurarnos de que todo es correcto, escribimos el siguiente comando.

```
sudo mdadm --detail /dev/md0
```

Formatear el nuevo volumen

Aunque ya casi hemos terminado, aún falta un detalle importante: darle formato al volumen.

Como pretendemos utilizar el sistema de archivos `ext4`, usaremos el comando `mkfs.ext4`:

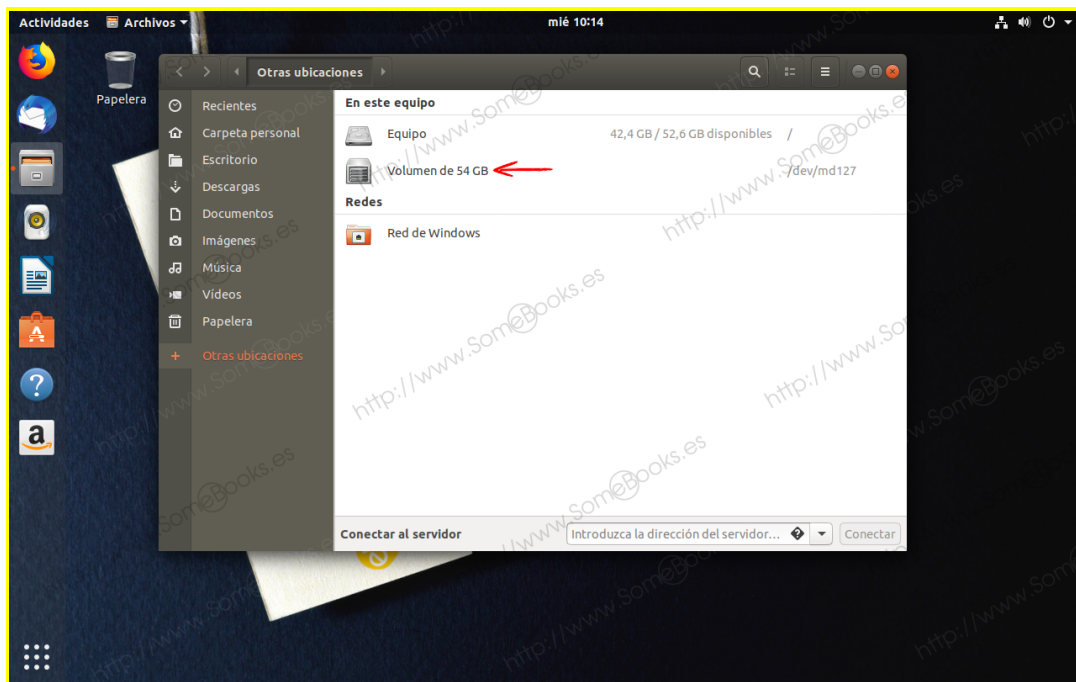
```
sudo mkfs.ext4 /dev/md0
```

Reiniciar el equipo

Para comprobar que todo es correcto, solo tenemos que reiniciar el equipo. Algo tan sencillo como recurrir a la siguiente orden:

```
sudo reboot
```

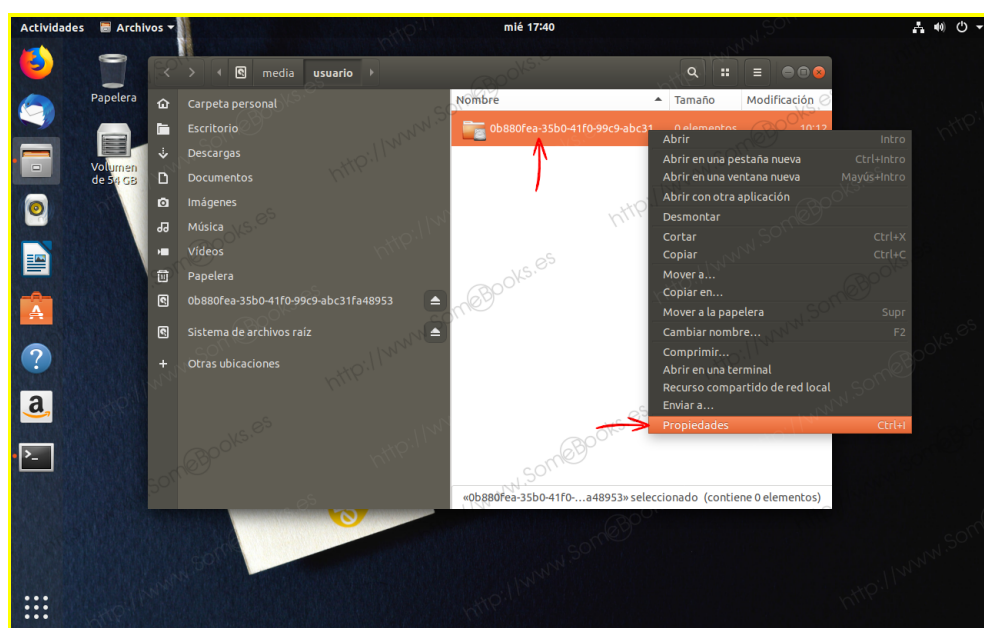
Cuando haya terminado el reinicio, solo tendrás que abrir el explorador de archivos y, en el panel izquierdo, elegir *Otras ubicaciones*. Verás que, en el panel derecho, aparece ya el nuevo volumen, con una capacidad equivalente a uno de los discos implicados. No olvides que funcionarán en espejo.



Cambiar los permisos del nuevo volumen

El propietario actual del dispositivo es root y tenemos que cambiar los permisos sobre su carpeta de montaje. Algo que podríamos hacer desde la terminal, pero que resultará más sencillo si utilizamos la interfaz gráfica.

```
sudo nautilus /media/usuario
```



Una vez que se abra la ventana Propiedades de la carpeta, hacemos clic sobre la solapa Permisos.