

Práctica RAID 1 con Ubuntu

¿Qué es RAID?

RAID es un acrónimo del inglés que significa Redundant Array of Independent Disk,s literalmente «matriz de discos independientes redundantes», aunque o todos los sistemas RAID proporcionan redundancia.

Al crear un único volumen con varios discos duros funcionando en conjunto conseguimos redundancia(tolerancia a fallos en el caso de que uno falle, conocido como disk mirroring) o mayor velocidad(conocido como disk striping).

¿Qué es RAID 1?

-En este modo, los datos se escriben en los dos discos de manera

simultánea, siendo el uno una copia exacta del otro, motivo por el

que se conoce a este modo como «mirroring».

-En este caso, si se estropeara uno de los dos discos no pasaría nada

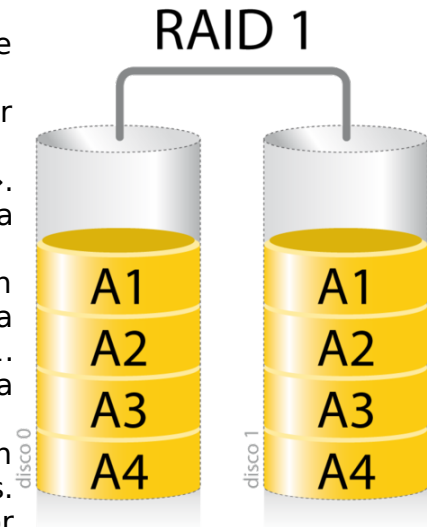
porque los datos estarían todavía en el otro, y bastaría con reemplazar el disco estropeado por uno nuevo para volver a restablecer el RAID 1.

-La parte mala de este modo de RAID es que no se gana velocidad

de acceso, más bien al contrario porque todos los datos deben escribirse dos veces.

-Además, el tamaño del volumen será el del disco de menor capacidad.

-Es decir, si usáramos un disco de 1 TB y otro de 500 GB, tendríamos un volumen de 500 GB en RAID 1.



PASOS PARA INSTALACIÓN DE RAID 1 EN UBUNTU

- Primero, en la máquina virtual, tendremos que añadir dos discos extras de 5 o 10GB, dependiendo de la capacidad que queramos.

En Ubuntu, en la terminal, comprobar que los discos están añadidas con el comando:

--> `lsblk -fm`

- Ahora, deberemos instalar una herramienta de administración de RAID, llamada mdadm.

El comando varía (pero no el software) depende la distribución de Linux que tengamos instalado en la Máquina Virtual o como SO principal:

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

- El siguiente paso consistirá en crear una partición para cada uno de los discos implicados. En cada disco, la partición ocupará todo el espacio disponible.
- El sdb varía dependiendo el tipo de unidad de almacenamiento que tenemos. Si tenemos, por ejemplo un nvsm, sería dev/nvme0, etc.

--> `sudo fdisk /dev/sdb`

- El comando `fdisk` actúa de forma interactiva, por lo que nos irá pidiendo información y nosotros deberemos ir completándola. Vayamos paso a paso:
 2. La primera vez que nos pide una orden, pulsamos la tecla `n`, que significa que queremos crear una nueva partición.
 3. A continuación, deberemos indicar que la partición es primaria. Por lo tanto, siguiendo las indicaciones del propio comando, pulsaremos la tecla `p`.
 4. Indicaremos que se trata de la primera partición del dispositivo, para lo que pulsaremos la tecla `1`.
 5. 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:

1. Como respuesta a la siguiente solicitud de una orden, pulsamos la tecla `t`, para indicar que queremos establecer el tipo de partición.
2. 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`.
3. Por último, escribimos las letras `fd`, que indican que se trata de una partición *RAID*.
4. Hacemos lo mismo para la segunda partición.

- Si todo ha salido bien, creamos el volumen de RAID 1
`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 **-l** 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.

- **Formatear el nuevo volumen creado**

Como pretendemos utilizar el sistema de archivos `ext4`, usaremos el comando `mkfs.ext4`:
`sudo mkfs.ext4 /dev/md0`

- Al finalizar todos los pasos haremos un `sudo reboot` desde la terminal para que todos los cambios se apliquen.