

Capítulo 4: RAID

RAID es el acrónimo de **Redundant Array of Independent Disks**. Consiste en una solución de almacenamiento que combina dos o más discos en una misma unidad lógica a fin de proveer redundancia de datos y/o aumentar el rendimiento de las operaciones de lectura o escritura. Sin embargo, ambas características dependen de la manera en que los discos se hayan organizado en lo que se denomina un nivel (puede pensarse como un tipo) de RAID dado.

RAID permite acceder a todo el arreglo de discos como un solo conjunto, por más que en bajo nivel existan varios discos físicos o particiones. Utilizando el Calculador de RAID online podemos visualizar la capacidad de un arreglo de acuerdo al nivel de RAID utilizado y confirmar el tipo de tolerancia a fallos que provee.

Sin importar el nivel de RAID que hayamos elegido ni cuán confiable sea la tecnología de almacenamiento que usemos para implementarlo, un arreglo RAID no reemplaza la necesidad de una estrategia de backup apropiada – por más que provea redundancia de datos.

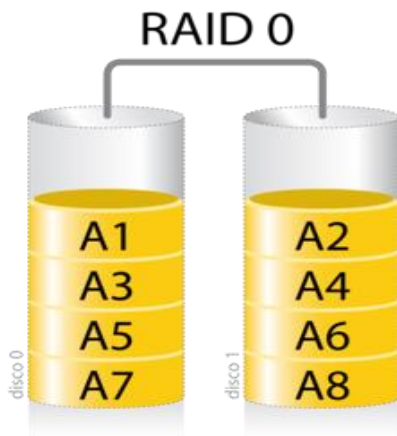
Tres niveles de RAID en Linux

Aunque hay más opciones de acuerdo a las necesidades que tengamos, para introducir el tema presentaremos solamente los tres niveles de RAID más utilizados RAID 0, RAID 1, y RAID 1+0 o RAID 10. En las imágenes ilustrativas, los bloques de datos escritos en los discos se indican como A1, A2, A3, y así sucesivamente.

RAID 0

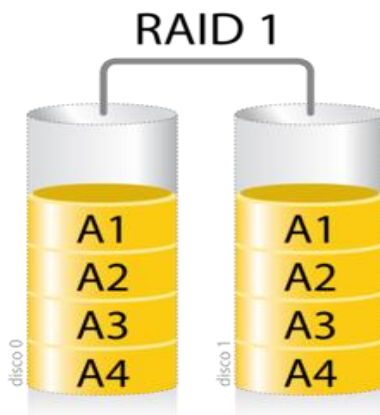
Conocido también por su nombre en inglés *stripe* (debido a que distribuye los datos de forma pareja entre dos o más discos, tal como se muestra en la imagen siguiente), este nivel de RAID provee un aumento en el rendimiento de las operaciones de lectura, pero no provee tolerancia a fallos. Requiere de al menos dos discos para implementarse, y el tamaño resultante es igual a la suma del tamaño de cada disco.





RAID 1

Más conocido como *mirror* (espejo), este arreglo de discos provee redundancia de datos al escribirlos simultáneamente en dos discos, de manera separada como indica la imagen. Esto permite que si uno de los discos experimenta un fallo, la misma información está disponible en el otro. Sin embargo, impacta negativamente en el rendimiento de escritura ya que un dato tiene que ser escrito dos veces. Al reemplazar el disco defectuoso por uno nuevo se puede reconstruir el arreglo y copiar la información en el mismo. Requiere de al menos dos discos para implementarse, y el tamaño resultante es igual al disco de menor tamaño.

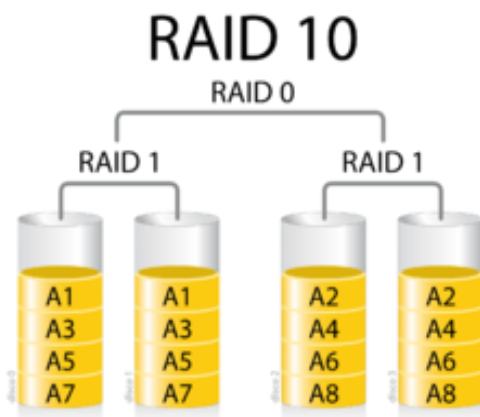


Como observamos en el esquema de arriba, los bloques de datos A1, A2, A3, y A4 son escritos en ambos dispositivos (de ahí el término *espejo* que se utiliza para describir este nivel de RAID).

RAID 1+0

También llamado RAID 10, este nivel de RAID es una especie de “combinación” de los dos anteriores: consiste en un RAID 0 (*stripe*) de dos arreglos RAID 1 (*mirror*), tal como se aprecia a continuación:





Como es de esperar, este arreglo provee tanto una mejora en el rendimiento general de lectura y escritura como redundancia, aunque admite como máximo el fallo de un disco en cada espejo. El tamaño resultante es igual a la suma de la capacidad de los dos RAIDs 1.

Crear RAID por software

La herramienta que se utiliza para crear RAIDs por software en Linux es `mdadm` (abreviación de **m**ultiple **d**isks **a**dmin), que podemos instalar utilizando

```
apt-get install mdadm
```

si es que no lo tenemos disponible todavía.

Para completar los ejemplos que mencionamos en esta sección, necesitaremos disponer de 4 discos particionados a los que en esta ocasión denominaremos **/dev/sdb1**, **/dev/sdc1**, **/dev/sdd1**, y **/dev/sde1**.

IMPORTANTE: Por el momento es conveniente que elijamos un solo nivel de RAID para comenzar.

RAID 0

Si quisiéramos crear un RAID 0 utilizando **/dev/sdb1** y **/dev/sdc1**, utilizaremos el siguiente comando:

```
mdadm --create --verbose /dev/md0 --level=stripe --raid-devices=2 /dev/sdb1 /dev/sdc1
```

donde la opción `--verbose` nos permitirá obtener cierto grado de detalle sobre el progreso de la creación del arreglo, mientras que `--level` y `--raid-devices` se emplean para indicar el nivel y la cantidad de dispositivos a utilizar para este ejercicio, respectivamente.



RAID 1

Si nos inclináramos por este caso, el comando es esencialmente idéntico al del ejemplo anterior. La única diferencia radica en el valor de la opción `--level`:

```
mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1
```

Este tipo de arreglo será el que ilustraremos mediante imágenes en este capítulo. Más adelante veremos cómo verificar el estado de `/dev/md0`.

RAID 1+0

En este caso deberíamos emplear los cuatro discos:

```
mdadm --create --verbose /dev/md0 --level=10 --raid-devices=4 /dev/sd[b-e]1
```

En el ejemplo anterior podemos ver que `mdadm` admite el uso de una expresión regular para especificar los discos que utilizaremos sin necesidad de listarlos uno a uno.

Crear un sistema de archivos y montar el dispositivo

Luego de haber creado nuestro RAID (sea del nivel que fuere), es momento de crear un sistema de archivos sobre el mismo y de montarlo para poder utilizarlo. Sin embargo, antes de esto deberemos agregar el detalle de nuestro arreglo en el archivo de configuración `/etc/mdadm/mdadm.conf` como primera medida para lograr persistencia. De otra forma, es posible que cuando el sistema reinicie no lo reconozca con el nombre que lo creamos.

```
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Ahora sí utilizaremos **ext4** como sistema de archivos y **/mnt/miraid** como punto de montaje de ejemplo:

```
mkfs.ext4 /dev/md0  
mount -t ext4 /dev/md0 /mnt/miraid
```

De más está decir que el punto de montaje debe ser un directorio existente en nuestro sistema. Por lo tanto, podemos crear **/mnt/miraid** o bien utilizar otro directorio de nuestra elección.

El último paso para lograr persistencia consiste agregar la línea necesaria en `/etc/fstab` y actualizar el disco RAM inicial:

```
echo "/dev/md0 /mnt/miraid ext4 defaults 0 0" >> /etc/fstab  
update-initramfs -u
```

De la misma manera en que lo hemos hecho en capítulos anteriores, podemos asignar un **LABEL** al sistema de archivos o averiguar el UUID del arreglo mediante el uso de

blkid. Cualquiera de estas dos alternativas nos permitirá montar el arreglo más fácilmente mediante una entrada en el archivo **/etc/fstab**.

Administración de RAID

En esta sección aprenderemos cómo monitorear el estado de creación del arreglo, como agregar un disco extra al mismo (para tener listo en el caso de una falla), y cómo simular una falla. Finalmente, explicaremos cómo “reciclar” uno o más discos que hayan formado parte de un arreglo previamente a fin de usarlos en otro.

Una vez que ejecutamos el comando para crear un arreglo, podemos monitorear el estado de la creación mediante cualquiera de los siguientes comandos:

```
cat /proc/mdstat  
mdadm --detail /dev/md0
```

En particular, el segundo comando nos advertirá de problemas (en el caso de que los haya) al listar con detalle el estado del arreglo.

```
root@debiancla:~# cat /proc/mdstat  
Personalities : [raid1]  
md0 : active raid1 sdc1[1] sdb1[0]  
      153408 blocks super 1.2 [2/2] [UU]  
  
unused devices: <none>  
root@debiancla:~# mdadm --detail /dev/md0  
/dev/md0:  
      Version : 1.2  
      Creation Time : Thu Oct  4 19:55:51 2018  
      Raid Level : raid1  
      Array Size : 153408 (149.81 MiB 157.09 MB)  
      Used Dev Size : 153408 (149.81 MiB 157.09 MB)  
      Raid Devices : 2  
      Total Devices : 2  
      Persistence : Superblock is persistent  
  
      Update Time : Thu Oct  4 19:55:53 2018  
      State : clean  
      Active Devices : 2  
      Working Devices : 2  
      Failed Devices : 0
```

Para simular una falla del disco **/dev/sdc1**, utilizaremos el siguiente comando, prestando atención a la opción **--fail**:

```
mdadm /dev/md0 --fail /dev/sdc1
```

A continuación, detendremos el arreglo para evitar problemas y quitaremos el dispositivo con problemas utilizando **--remove**:

```
mdadm --stop /dev/md0  
mdadm /dev/md0 --remove /dev/sdc1
```



Luego de haber conectado el disco físicamente a nuestro sistema y habiendo creado la partición correspondiente identificada como **/dev/sdg1**, podemos utilizar para reemplazar la dañada y luego reiniciamos el arreglo:

```
mdadm /dev/md0 --add /dev/sdg1  
mdadm --assemble /dev/md0
```

Si quisiéramos reciclar uno o varios discos a fin de utilizarlos en otro arreglo podemos hacerlo de la siguiente manera. Luego de haber 1) tomado las precauciones necesarias, 2) resguardado los datos importantes, y 3) removido un disco en particular del arreglo, procederemos a eliminar la información del superbloque utilizando a **/dev/sdc1** como ejemplo ilustrativo:

```
mdadm --zero-superblock /dev/sdc1  
update-initramfs -u
```

En este punto ya podemos utilizar **/dev/sdc1** en otro arreglo si así lo deseamos.

