

# Volúmenes lógicos y RAID software en GNU/Linux

---

## Tabla de contenidos

---

- Instalación del entorno de prácticas
- Comprobación de recursos
- RAID5
  - Preparación de los dispositivos
  - Creación del RAID5
  - Formateo EXT3
  - Montaje del RAID5
  - Extensión de dispositivos
  - Extensión de almacenamiento
- Cuestiones
- Bibliografía

## Instalación del entorno de prácticas

---

Iniciamos el autoinstalador para Linux

```
curl -o- \
http://ccia.esei.uvigo.es/docencia/CDA/1819/practicas//ejercicio-lvm-raid.sh | \
bash -
```

Nos mandará poner un identificador único. Después de esto, se nos abrirá nuestro nuevo entorno de pruebas.

Para poder loguearnos en el sistema deberemos introducir el usuario `root` junto a la contraseña `purple`.

Si no se nos inicia el entorno gráfico, deberemos ejecutar el siguiente comando:

```
startx
```

Una vez tengamos nuestro entorno preparado, procederemos a ejecutar la aplicación `LXTerminal`.

# Comprobación de recursos

---

Comprobamos los discos disponibles con cualquiera de los siguientes comandos:

```
lsblk
fdisk -l
parted -l
```

Personalmente prefiero `lsblk`. Una vez ejecutado ésta es su salida:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdc	8:32	0	100M	0	disk	
sdd	8:48	0	100M	0	disk	
sde	8:64	0	200M	0	disk	
sdf	8:80	0	100M	0	disk	

Como podemos comprobar, el sistema está montado sobre el disco `sda` y la partición SWAP sobre el `sdb`, pero disponemos a mayores cuatro discos para poder trabajar a gusto.

## RAID5

---

Procederemos con la creación de un RAID5 utilizando los dispositivos `/dev/sdc1`, `/dev/sdd1` y `/dev/sde1`.

## Preparación de los dispositivos

---

Primeramente, crearemos una partición primaria en los dispositivos `/dev/sdc`, `/dev/sdd` y `/dev/sde` asignándole todo el espacio disponible usando el comando `parted`.

```
parted /dev/sdc
(parted) mklabel msdos
(parted) mkpart primary 1M 100%
(parted) set 1 raid on
(parted) quit
# Repetir el proceso usando /dev/sdd y /dev/sde
```

Comprobamos el resultado ejecutando `lsblk` de nuevo.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sd	8:32	0	100M	0	disk	
└─sd	8:33	0	99M	0	part	
sdd	8:48	0	100M	0	disk	
└─sdd1	8:49	0	99M	0	part	
sde	8:64	0	200M	0	disk	
└─sde1	8:65	0	199M	0	part	
sdf	8:80	0	100M	0	disk	

## Creación del RAID5

Una vez creados los dispositivos necesarios para nuestro RAID, usaremos `mdadm` para construir y gestionar nuestro RAID5. En concreto ejecutaremos este comando:

```
mdadm --create --verbose /dev/md/md_RAID5 --level=raid5 --raid-devices=3 \
/dev/sdc1 /dev/sdd1 /dev/sde1
```

El cual es bastante intuitivo y no necesita demasiada explicación:

```
mdadm [--create: Crear RAID] \
[--verbose: Muestra logs por pantalla] \
[/dev/md/md_RAID5: Crear el RAID en dicha ruta] \
[-level=raid5: Tipo de RAID] \
[--raid-devices=3: Utilizar 3 dispositivos para montarlo] \
[/dev/sdc1 /dev/sdd1 /dev/sde1: Dispositivos disponibles para ser usados]
```

Una vez ejecutado el comando, nos saldrá la siguiente confirmación:

```
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 100352K
mdadm: largest drive (/dev/sde1) exceeds size (100352K) by more than 1%
Continue creating array?
```

Se nos avisa de que el `sde1` excede en tamaño a los otros dos, pero nosotros confirmaremos dicha advertencia ya que somos conscientes.

Se nos notificará del éxito de la operación con el siguiente mensaje:

```
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/md_RAID5 started.
```

Si ejecutamos de nuevo `lsblk` comprobaremos la creación del RAID

```
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda        8:0    0   16G  0 disk
└─sda1     8:1    0   16G  0 part  /
sdb        8:16   0    1G  0 disk
└─sdb1     8:17   0 1022M  0 part  [SWAP]
sdc        8:32   0   100M  0 disk
└─sdc1     8:33   0    99M  0 part
   └─md127  9:127   0   196M  0 raid5
sdd        8:48   0   100M  0 disk
└─sdd1     8:49   0    99M  0 part
   └─md127  9:127   0   196M  0 raid5
sde        8:64   0   200M  0 disk
└─sde1     8:65   0   199M  0 part
   └─md127  9:127   0   196M  0 raid5
sdf        8:80   0   100M  0 disk
```

Si necesitamos información extra, podremos obtenerla de la siguiente manera:

```
mdadm --detail /dev/md/md_RAID5
```

Salida:

```
/dev/md/md_RAID5:
  Version : 1.2
  Creation Time : Sat Oct 20 12:49:13 2018
  Raid Level : raid5
  Array Size : 200704 (196.00 MiB 205.52 MB)
  Used Dev Size : 100352 (98.00 MiB 102.76 MB)
  Raid Devices : 3
  Total Devices : 3
  Persistence : Superblock is persistent
  Update Time : Sat Oct 20 12:49:14 2018
  State : clean
  Active Devices : 3
  Working Devices : 3
  Failed Devices : 0
  Spare Devices : 0
  Layout : left-symmetric
  Chunk Size : 512K
  Name : datos.cda.net:md_RAID5 (local to host datos.cda.net)
  UUID : 5f6d514c:209e86ab:54764a4a:b2f9824a
  Events : 18

   Number   Major   Minor   RaidDevice State
     0         8       33         0     active sync  /dev/sdc1
     1         8       49         1     active sync  /dev/sdd1
     3         8       65         2     active sync  /dev/sde1
```

# Formateo EXT3

Para formatear nuestro RAID en formato `ext3` ejecutaremos

```
mkfs.ext3 /dev/md/md_RAID5
```

Si no tenemos ningún error nos mostrará la siguiente salida:

```
Se está creando un sistema de ficheros con 200704 bloques de 1k y 50200 nodos-i
UUID del sistema de ficheros: 4017da92-621e-46cc-8225-d786e3c4660b
Respaldo del superbloque guardado en los bloques:
    8193, 24577, 40961, 57345, 73729
Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (4096 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros:
0/2hecho
```

## Montaje del RAID5

Montaremos nuestro RAID5 de la siguiente manera

```
mkdir /mnt/raid
mount /dev/md/md_RAID5 /mnt/raid/
```

Comprobaremos que se ha montado correctamente usando `lsblk` :

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdс	8:32	0	100M	0	disk	
└─sdс1	8:33	0	99M	0	part	
└─md127	9:127	0	196M	0	raid5	/mnt/raid
sdd	8:48	0	100M	0	disk	
└─sdd1	8:49	0	99M	0	part	
└─md127	9:127	0	196M	0	raid5	/mnt/raid
sde	8:64	0	200M	0	disk	
└─sde1	8:65	0	199M	0	part	
└─md127	9:127	0	196M	0	raid5	/mnt/raid
sdf	8:80	0	100M	0	disk	

Para ver el espacio disponible usaremos el siguiente comando:

```
df -Th /mnt/raid
```

De esta forma nos mostrará que tenemos disponibles 175M en nuestro nuevo dispositivo

S.ficheros	Tipo	Tamaño	Usados	Disp	Uso%	Montado en
/dev/md127	ext3	186M	1,6M	175M	1%	/mnt/raid

Crearemos un archivo de prueba de 1MB en `/mnt/raid` para comprobar que funciona correctamente usando `dd` :

```
dd bs=1MB count=0 seek=1 of=/mnt/raid/allzeros  
ls -l /mnt/raid/allzeros
```

Como vemos, el archivo se ha creado correctamente:

```
0+0 registros leídos  
0+0 registros escritos  
0 bytes copied, 0,000191775 s, 0,0 kB/s  
-rw-r--r-- 1 root root 1000000 oct 20 13:52 /mnt/raid/allzeros
```

Curiosamente, si vemos el espacio disponible con el comando `df` , veremos que nada ha cambiado:

S.ficheros	Tipo	Tamaño	Usados	Disp	Uso%	Montado en
/dev/md127	ext3	186M	1,6M	175M	1%	/mnt/raid

Esto se debe a que el archivo `allzeros` no ocupa espacio real en el disco, dado que está formado por bytes nulos.

Si copiásemos cualquier otro fichero, el espacio sí que variaría:

```
cp /boot/initrd.img-4.9.0-7-amd64 /mnt/raid/  
df -Th /mnt/raid/
```

Dando como resultado lo siguiente:

S.ficheros	Tipo	Tamaño	Usados	Disp	Uso%	Montado en
/dev/md127	ext3	186M	22M	155M	13%	/mnt/raid

## Extensión de dispositivos

Lo primero será desmontar nuestro RAID

```
umount /mnt/raid
```

Particionaremos el dispositivo `sdf` tal y como hicimos con el resto:

```
parted /dev/sdf
(parted) mklabel msdos
(parted) mkpart primary 1M 100%
(parted) set 1 raid on
(parted) quit
```

Comprobamos el resultado con `lsblk`:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdс	8:32	0	100M	0	disk	
└─sdс1	8:33	0	99M	0	part	
└─md127	9:127	0	196M	0	raid5	
sdd	8:48	0	100M	0	disk	
└─sdd1	8:49	0	99M	0	part	
└─md127	9:127	0	196M	0	raid5	
sde	8:64	0	200M	0	disk	
└─sde1	8:65	0	199M	0	part	
└─md127	9:127	0	196M	0	raid5	
sdf	8:80	0	100M	0	disk	
└─sdf1	8:81	0	99M	0	part	

Para agregar nuestro nuevo dispositivo al RAID5, usaremos:

```
mdadm --add /dev/md/md_RAID5 /dev/sdf1
```

Si comprobamos el resultado con `mdadm --detail /dev/md/md_RAID5` veremos que podemos usarlo, pero aún no está dentro de nuestro array.

```
/dev/md/md_RAID5:
  Version : 1.2
  Creation Time : Sat Oct 20 12:49:13 2018
  Raid Level : raid5
  Array Size : 200704 (196.00 MiB 205.52 MB)
  Used Dev Size : 100352 (98.00 MiB 102.76 MB)
  Raid Devices : 3
  Total Devices : 4
  Persistence : Superblock is persistent
  Update Time : Sat Oct 20 14:14:45 2018
  State : clean
  Active Devices : 3
  Working Devices : 4
```

```

Failed Devices : 0
Spare Devices : 1
  Layout : left-symmetric
  Chunk Size : 512K
    Name : datos.cda.net:md_RAID5 (local to host datos.cda.net)
    UUID : 5f6d514c:209e86ab:54764a4a:b2f9824a
    Events : 19
  Number   Major   Minor   RaidDevice State
    0       8       33       0      active sync  /dev/sdc1
    1       8       49       1      active sync  /dev/sdd1
    3       8       65       2      active sync  /dev/sde1
    4       8       81       -      spare   /dev/sdf1

```

Para ello necesitaremos aumentar el número de dispositivos de nuestro RAID de la siguiente forma:

```
mdadm --grow --raid-devices=4 /dev/md/md_RAID5
```

Si comprobamos nuestro RAID5 con el comando `mdadm --detail /dev/md/md_RAID5`, veremos que ya está disponible dentro del array.

```

/dev/md/md_RAID5:
  Version : 1.2
  Creation Time : Sat Oct 20 12:49:13 2018
  Raid Level : raid5
  Array Size : 301056 (294.00 MiB 308.28 MB)
  Used Dev Size : 100352 (98.00 MiB 102.76 MB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent
  Update Time : Sat Oct 20 15:50:00 2018
  State : clean
  Active Devices : 4
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 0
    Layout : left-symmetric
    Chunk Size : 512K
      Name : datos.cda.net:md_RAID5 (local to host datos.cda.net)
      UUID : 5f6d514c:209e86ab:54764a4a:b2f9824a
      Events : 39
    Number   Major   Minor   RaidDevice State
      0       8       33       0      active sync  /dev/sdc1
      1       8       49       1      active sync  /dev/sdd1
      3       8       65       2      active sync  /dev/sde1
      4       8       81       3      active sync  /dev/sdf1

```

## Extensión de almacenamiento

---



Aunque nuestro RAID cuenta con un nuevo dispositivo, debemos redimensionar el espacio disponible.

Para ello utilizaremos el siguiente comando:

```
resize2fs /dev/md/md_RAID5
```

En mi caso me saltó un error dado que el RAID debía ser comprobado antes de poder redimensionar su espacio:

```
resize2fs 1.43.4 (31-Jan-2017)
Por favor ejecute antes 'e2fsck -f /dev/md/md_RAID5'.
```

Como bien indica, comprobamos el RAID con `e2fsck` antes de redimensionarlo:

```
e2fsck -f /dev/md/md_RAID5
resize2fs /dev/md/md_RAID5
```

Y si todo sale bien, veremos la siguiente salida:

```
Paso 1: Verificando nodos-i, bloques y tamaños
Paso 2: Verificando la estructura de directorios
Paso 3: Revisando la conectividad de directorios
Paso 4: Revisando las cuentas de referencia
Paso 5: Revisando el resumen de información de grupos
/dev/md/md_RAID5: 12/50200 ficheros (0.0% no contiguos),
12001/200704 bloques
resize2fs 1.43.4 (31-Jan-2017)
Cambiando el tamaño del sistema de ficheros en /dev/md/md_RAID5
a 301056 (1k) bloques.
El sistema de ficheros en /dev/md/md_RAID5 tiene ahora 301056
bloques (de 1k).
```

Comprobamos que no haya ocurrido nada raro durante el proceso usando `mdadm --detail /dev/md/md_RAID5` :

```
/dev/md/md_RAID5:
  Version : 1.2
  Creation Time : Sat Oct 20 12:49:13 2018
  Raid Level : raid5
  Array Size : 301056 (294.00 MiB 308.28 MB)
  Used Dev Size : 100352 (98.00 MiB 102.76 MB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent
  Update Time : Sat Oct 20 16:35:09 2018
  State : clean
  Active Devices : 4
```

```

Working Devices : 4
Failed Devices : 0
Spare Devices : 0
  Layout : left-symmetric
  Chunk Size : 512K
    Name : datos.cda.net:md_RAID5 (local to host datos.cda.net)
    UUID : 5f6d514c:209e86ab:54764a4a:b2f9824a
    Events : 41
  Number   Major   Minor   RaidDevice State
    0       8       33       0      active sync  /dev/sdc1
    1       8       49       1      active sync  /dev/sdd1
    3       8       65       2      active sync  /dev/sde1
    4       8       81       3      active sync  /dev/sdf1

```

Como no ha ocurrido nada raro, procederemos a montarlo y a comprobar el nuevo espacio disponible.

```

mount -t ext3 /dev/md/md_RAID5 /mnt/raid
df -Th /mnt/raid

```

Podemos comprobar que ahora hay más almacenamiento disponible, casi 100M, que es el espacio que nos concede el nuevo dispositivo:

```

S.ficheros      Tipo Tamaño Usados  Disp Uso% Montado en
/dev/md127      ext3   281M   2,1M  265M   1% /mnt/raid

```

También podemos comprobar que sigue estando nuestro archivo de prueba llamado `allzeros` a pesar de todas la modificaciones realizadas:

```

-rw-r--r-- 1 root root 1000000 oct 20 13:56 /mnt/raid/allzeros

```

¡Todo perfecto!

## Cuestiones

#	Pregunta	Respuesta
1	¿Por qué en este caso es conveniente que las dos particiones de /dev/sde esten	Porque si falla el dispositivo <code>sde</code> , quedará el 'mirror' en los dispositivos <code>sdc</code> y <code>sdd</code> . Si no fuera así, uno de los RAID 1

	asignadas a "subarrays" RAID1 distintos?	dejaría de funcionar y fallaría el sistema de almacenamiento
2	¿Por qué en este caso sí es conveniente que las dos particiones de /dev/sde estén asignadas al mismo "subarray" RAID0?	En este caso, si falla el dispositivo <code>sde</code> , quedará el mirror en los dispositivos <code>sd</code> y <code>sd</code> , por lo que el sistema de almacenamiento seguiría funcionando de forma correcta

## Bibliografía

- <https://github.com/Student-Puma/HomeLab>
- <http://ccia.esei.uvigo.es/docencia/CDA/1819/practicas/ejercicio-lvm-raid/ejercicio-lvm-raid.html>
- <https://zackreed.me/adding-an-extra-disk-to-an-mdadm-array/>
- <https://unix.stackexchange.com/questions/102613/create-a-test-file-with-lots-of-zero-bytes>