

Memoria Práctica 5

Administración de Servidores

Administración avanzada de dispositivos de almacenamiento

Almudena García Jurado-Centurión

Índice

Introducción

Parte 1: RAID

1. [Creando un nuevo disco virtual](#)
2. [Creando particiones en el disco duro](#)
3. [Configurando las particiones en RAID 5](#)
4. [Montando los RAID](#)
5. [Creando ficheros](#)
6. [Simulando la rotura de uno de los discos del RAID](#)
7. [Accediendo a los ficheros del RAID](#)
8. [Creando nuevos ficheros en el RAID](#)
9. [Añadiendo un nuevo disco al RAID](#)
 - [Añadiendo el RAID de forma persistente](#)
 - [Creando el nuevo disco duro](#)
 - [Añadiendo la nueva partición al RAID](#)
 - [Eliminando otro elemento del RAID](#)

Parte 2: LVM

1. [Creando un nuevo disco duro virtual](#)
2. [Creando un volumen físico](#)
 - [Creando la partición con fdisk](#)
 - [Creando el PV dentro de la partición con pvcreate](#)
 - [Comprobando los datos del PV con pvs](#)
3. [Creando un grupo de volúmenes](#)
 - [Creando el grupo de volúmenes con vgcreate](#)
 - [Comprobando el grupo de volúmenes con pvs y vgs](#)
4. [Creando volúmenes lógicos](#)
 - [Creando volúmenes lógicos con lvcreate](#)
 - [Comprobando los volúmenes lógicos con lvs](#)
5. [Montando los volúmenes lógicos en el sistema](#)
 - [Desmontando /tmp y /home](#)
 - [Buscando la unidad asociada a los volúmenes lógicos](#)
 - [Formateando los volúmenes lógicos](#)
 - [Montando las particiones](#)
6. [Creando ficheros en /home](#)
7. [Incrementando el tamaño de un volumen lógico](#)
 - [Aumentando el volumen lógico](#)
 - [Redimensionando el sistema de ficheros](#)

Conclusión

Introducción

La gestión avanzada de dispositivos de almacenamiento, mediante técnicas como RAID o LVM, nos permite añadir características adicionales a los sistemas de ficheros tradicionales, tales como agrupar varias unidades de disco para que funcionen como una sola, o añadir mayor integridad a los ficheros.

En el caso de RAID, esta técnica nos puede permitir tener redundancia de datos, protección ante errores, y mayor velocidad en el acceso a estos. De esta forma, la rotura de uno de los discos no provocará pérdida de datos y, al estar la información repartida entre diferentes discos, el tiempo de acceso se reduce.

Por su parte, LVM nos permite dividir uno o varios discos duros en unidades virtuales fácilmente eliminables o redimensionables. Esto nos permite hacer una gestión mas limpia de las particiones, pudiendo redimensionar el sistema de ficheros en caliente, sin necesidad de desmontaje previo, y añadiendo mas discos de forma rápida.

En esta práctica aprenderemos a gestionar estas dos técnicas avanzadas de almacenamiento, creándolas desde cero y realizando diferentes tipos de modificaciones sobre las mismas para comprobar sus características.

Parte 1: RAID

1. Creando un nuevo disco virtual

Creamos un nuevo disco duro en la máquina virtual, de 512 MB, usando el asistente de VirtualBox.

Iniciamos la máquina virtual y, tras ejecutar lsblk, vemos que el disco duro es detectado correctamente por la máquina, con el nombre de sde

```
[root@localhost ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sr0	11:0	1	1024M	0	rom	
sda	8:0	0	20G	0	disk	
└─sda1	8:1	0	500M	0	part	/boot
└─sda2	8:2	0	19,5G	0	part	
└─VolGroup-lv_root (dm-0)	253:0	0	17,6G	0	lvm	/
└─VolGroup-lv_swap (dm-1)	253:1	0	2G	0	lvm	
sdb	8:16	0	384M	0	disk	
└─sdb1	8:17	0	133,3M	0	part	/tmp
└─sdb2	8:18	0	133,4M	0	part	/home
└─sdb3	8:19	0	109,8M	0	part	[SWAP]
sdc	8:32	0	8G	0	disk	
└─sdc1	8:33	0	8G	0	part	
└─privado (dm-2)	253:2	0	8G	0	crypt	
sdd	8:48	0	1G	0	disk	
sde	8:64	0	512M	0	disk	

2. Creando particiones en el disco duro

Tras crear el nuevo disco duro, pasamos a crear tres particiones del mismo tamaño, dentro del disco

Para ello usaremos la herramienta fdisk, ya vista en prácticas anteriores

```
[root@localhost ~]# fdisk /dev/sde

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Orden (m para obtener ayuda): o
Building a new DOS disklabel with disk identifier 0x08b03897.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Atención: el indicador 0x0000 inválido de la tabla de particiones 4 se corregirá mediante w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Orden (m para obtener ayuda): n
Acción de la orden
e   Partición extendida
p   Partición primaria (1-4)
p
Número de partición (1-4): 1
Primer cilindro (1-65, default 1):
Using default value 1
Last cilindro, +cilindros or +size{K,M,G} (1-65, default 65): +128M

Orden (m para obtener ayuda): n
Acción de la orden
e   Partición extendida
p   Partición primaria (1-4)
p
Número de partición (1-4): 2
Primer cilindro (18-65, default 18):
Using default value 18
Last cilindro, +cilindros or +size{K,M,G} (18-65, default 65): +128M
```

```
Orden (m para obtener ayuda): n
Acción de la orden
e   Partición extendida
p   Partición primaria (1-4)
p
Número de partición (1-4): 3
Primer cilindro (35-65, default 35): +128M
El valor está fuera del rango.
Primer cilindro (35-65, default 35):
Using default value 35
Last cilindro, +cilindros or +size{K,M,G} (35-65, default 65): +128M

Orden (m para obtener ayuda): w
¡Se ha modificado la tabla de particiones!
```

Ejecutamos lsblk para comprobar que las particiones se han creado correctamente

```
[root@localhost ~]# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sr0                                11:0    1  1024M  0  rom
sda                                 8:0     0    20G   0  disk
├─sda1                             8:1     0    500M   0  part  /boot
└─sda2                             8:2     0   19,5G   0  part
   ├─VolGroup-lv_root (dm-0) 253:0     0   17,6G   0  lvm    /
   └─VolGroup-lv_swap (dm-1) 253:1     0     2G   0  lvm
sdb                                 8:16     0    384M   0  disk
├─sdb1                             8:17     0   133,3M   0  part  /tmp
├─sdb2                             8:18     0   133,4M   0  part  /home
└─sdb3                             8:19     0   109,8M   0  part  [SWAP]
sdc                                 8:32     0     8G   0  disk
├─sdc1                             8:33     0     8G   0  part
│   └─privado (dm-2)         253:2     0     8G   0  crypt
sdd                                 8:48     0     1G   0  disk
sde                                 8:64     0    512M   0  disk
├─sde1                             8:65     0   133,3M   0  part
├─sde2                             8:66     0   133,4M   0  part
└─sde3                             8:67     0   133,4M   0  part
[root@localhost ~]#
```

3. Configurando las particiones en RAID 5

Para crear el RAID, usamos la herramienta mdadm, que nos permite crear un RAID a partir de diferentes discos.

La sintaxis de mdadm es la siguiente:

```
mdadm --create [dispositivo_destino] --level=[tipo] -raid-  
devices=[numero de dispositivos] [dispositivo1] [dispositivo2] ...  
[dispositivo n]
```

En nuestro caso, queremos crear un RAID 5 con 3 dispositivos: /dev/sde1, /dev/sde2 y /dev/sde3

Ejecutamos el comando con nuestros parámetros

```
[root@localhost ~]# mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sde1 /dev/sde2 /dev/sde3  
mdadm: Defaulting to version 1.2 metadata  
mdadm: array /dev/md0 started.
```

El programa nos avisa de que el RAID se ha creado correctamente.

Volvemos a comprobar el resultado con lsblk

```
[root@localhost ~]# lsblk  
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT  
sr0                                11:0    1 1024M  0 rom  
sda                                8:0     0   20G  0 disk  
├─sda1                             8:1     0   500M  0 part  /boot  
└─sda2                             8:2     0  19,5G  0 part  
   ├─VolGroup-lv_root (dm-0) 253:0     0  17,6G  0 lvm    /  
   └─VolGroup-lv_swap (dm-1) 253:1     0    2G  0 lvm  
sdb                                8:16    0   384M  0 disk  
├─sdb1                             8:17    0  133,3M  0 part  /tmp  
├─sdb2                             8:18    0  133,4M  0 part  /home  
└─sdb3                             8:19    0  109,8M  0 part  [SWAP]  
sdc                                8:32    0    8G  0 disk  
└─sdc1                             8:33    0    8G  0 part  
   └─privado (dm-2)          253:2     0    8G  0 crypt  
sdd                                8:48    0    1G  0 disk  
sde                                8:64    0   512M  0 disk  
├─sde1                             8:65    0  133,3M  0 part  
│   └─md0                         9:0     0   264M  0 raid5  
├─sde2                             8:66    0  133,4M  0 part  
│   └─md0                         9:0     0   264M  0 raid5  
└─sde3                             8:67    0  133,4M  0 part  
   └─md0                         9:0     0   264M  0 raid5
```

Vemos que de nuestras particiones cuelga un nuevo dispositivo llamado md0, correspondiente a nuestro RAID 5

Para comprobar que el RAID se ha creado y es funcional, usamos las opciones `--query` y `--detail` de `mdadm`, seguidos del nombre de dispositivo, en este caso `/dev/md0`

```
[root@localhost ~]# mdadm --query /dev/md0
/dev/md0: 264.00MiB raid5 3 devices, 0 spares. Use mdadm --detail for more details.
```

Tras ejecutar `mdadm --query`, vemos que hay un nuevo RAID con 3 dispositivos de 264MiB

El comando nos indica que podemos obtener mas información con el argumento `--detail`, así que lo ejecutamos.

```
[root@localhost ~]# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Mon Jan 15 15:49:55 2018
    Raid Level : raid5
    Array Size : 270336 (264.00 MiB 276.82 MB)
  Used Dev Size : 135168 (132.00 MiB 138.41 MB)
    Raid Devices : 3
  Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Mon Jan 15 16:04:26 2018
      State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 0
Spare Devices : 0

    Layout : left-symmetric
  Chunk Size : 512K

        Name : localhost.localdomain:0 (local to host localhost.localdomain)
        UUID : ce36d767:da9f7f04:2187c80c:397b7ea4
    Events : 19

   Number  Major   Minor  RaidDevice State
    -----
     0      8      65        0     active sync  /dev/sde1
     1      8      66        1     active sync  /dev/sde2
     3      8      67        2     active sync  /dev/sde3
```

Vemos que las 3 particiones están incluidas dentro del RAID y funcionan correctamente.

4. Montando las RAID

Antes de montar el RAID, debemos crearle un nuevo sistema de ficheros.

Para ello, usaremos la herramienta mkfs

```
[root@localhost ~]# mkfs -t ext4 /dev/md0
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de S0: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=512 blocks, Stripe width=1024 blocks
67584 nodos-i, 270336 bloques
13516 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67633152
33 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
2048 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729, 204801, 221185

Escribiendo las tablas de nodos-i: hecho
Creating journal (8192 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 36 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
[root@localhost ~]#
```

Hecho esto, creamos el directorio de nuestro punto de montaje, en /mnt/RAID, y montamos nuestro RAID sobre él.

```
[root@localhost ~]# mkdir /mnt/RAID
[root@localhost ~]# mount /dev/md0 /mnt/RAID
```

```
[root@localhost ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sr0                                  11:0    1  1024M  0 rom
sda                                  8:0     0   20G   0 disk
├─sda1                              8:1     0   500M   0 part  /boot
├─sda2                              8:2     0  19,5G   0 part
│   └─VolGroup-lv_root (dm-0) 253:0     0  17,6G   0 lvm    /
│       └─VolGroup-lv_swap (dm-1) 253:1     0    2G   0 lvm
sdb                                  8:16    0   384M   0 disk
├─sdb1                              8:17    0  133,3M   0 part  /tmp
├─sdb2                              8:18    0  133,4M   0 part  /home
└─sdb3                              8:19    0  109,8M   0 part  [SWAP]
sdc                                  8:32    0     8G   0 disk
├─sdc1                              8:33    0     8G   0 part
│   └─privado (dm-2)          253:2     0     8G   0 crypt
sdd                                  8:48    0     1G   0 disk
sde                                  8:64    0   512M   0 disk
├─sde1                              8:65    0  133,3M   0 part
│   └─md0                          9:0     0   264M   0 raid5  /mnt/RAID
├─sde2                              8:66    0  133,4M   0 part
│   └─md0                          9:0     0   264M   0 raid5  /mnt/RAID
├─sde3                              8:67    0  133,4M   0 part
│   └─md0                          9:0     0   264M   0 raid5  /mnt/RAID
```

Ejecutamos lsblk, y comprobamos que las 3 particiones pertenecientes al RAID están montadas en el mismo punto de montaje

5. Creando ficheros

Creamos varios ficheros de texto, comprobamos que existen y que su contenido está correcto

```
[root@localhost ~]# echo "12345" >> /mnt/RAID/nuevofichero.txt
[root@localhost ~]# echo "abc" >> /mnt/RAID/ficheroabc.txt
[root@localhost ~]# ls /mnt/RAID/
ficheroabc.txt  lost+found  nuevofichero.txt
[root@localhost ~]# more /mnt/RAID/ficheroabc.txt
abc
[root@localhost ~]# more /mnt/RAID/nuevofichero.txt
12345
```

6. Simulando la rotura de uno de los discos del RAID

Para simular la rotura de uno de los discos, vamos a usar la opción `--failure` de mdadm

La sintaxis es:

```
mdadm --failure [dispositivo_raid] [dispositivo_a_probar]
```

En nuestro caso, el dispositivo RAID es el md0, y como dispositivo a probar, vamos a usar el sde1

```
[root@localhost ~]# mdadm --fail /dev/md0 /dev/sde1
mdadm: set /dev/sde1 faulty in /dev/md0
```

La herramienta nos indica que la partición /dev/sde1 ha sido marcada como fallida

Ejecutamos mdadm --detail para comprobar el estado del RAID

```
[root@localhost ~]# mdadm --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Mon Jan 15 15:49:55 2018
  Raid Level : raid5
  Array Size : 270336 (264.00 MiB 276.82 MB)
  Used Dev Size : 135168 (132.00 MiB 138.41 MB)
  Raid Devices : 3
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Mon Jan 15 16:36:58 2018
  State : clean, degraded
Active Devices : 2
Working Devices : 2
Failed Devices : 1
Spare Devices : 0

  Layout : left-symmetric
  Chunk Size : 512K

  Name : localhost.localdomain:0 (local to host localhost.localdomain)
  UUID : ce36d767:da9f7f04:2187c80c:397b7ea4
  Events : 48

  Number Major Minor RaidDevice State
    0      0      0        0     removed
    1      8      66        1     active sync  /dev/sde2
    3      8      67        2     active sync  /dev/sde3
    0      8      65        -     faulty    /dev/sde1
```

Vemos que hay 2 dispositivos activos y 1 fallido, correspondiente al /dev/sde1, y que el estado del cluster es “degradado”

7. Accediendo a los ficheros del RAID

Con la partición anterior marcada como errónea, intentamos acceder a los ficheros almacenados en nuestro RAID

```
[root@localhost ~]# more /mnt/RAID/nuevofichero.txt
12345
[root@localhost ~]# more /mnt/RAID/ficheroabc.txt
abc
[root@localhost ~]# ls /mnt/RAID/
ficheroabc.txt  lost+found  nuevofichero.txt
```

Vemos que los ficheros se conservan correctamente

8. Creando nuevos ficheros en el RAID

Creamos nuevos ficheros y comprobamos que son accesibles

```
[root@localhost ~]# echo "qwerty" >> /mnt/RAID/qwerty.txt
[root@localhost ~]# echo "nueva prueba" >> /mnt/RAID/prueba.txt
[root@localhost ~]# echo "otrofichero" >> /mnt/RAID/fichero.txt
[root@localhost ~]# ls /mnt/RAID/
ficheroabc.txt  fichero.txt  lost+found  nuevofichero.txt  prueba.txt  qwerty.txt
[root@localhost ~]# more /mnt/RAID/fichero
/mnt/RAID/fichero: No existe el fichero o el directorio
[root@localhost ~]# more /mnt/RAID/fichero.txt
otrofichero
[root@localhost ~]# more /mnt/RAID/ficheroabc.txt
abc
[root@localhost ~]# more /mnt/RAID/nuevofichero.txt
12345
[root@localhost ~]# more /mnt/RAID/prueba.txt
nueva prueba
[root@localhost ~]# more /mnt/RAID/qwerty.txt
qwerty
```

9. Añadiendo un nuevo disco al RAID

Comprobado que el RAID funciona correctamente y es resistente ante fallos, pasamos a crear un nuevo disco y añadirlo al RAID

- **Añadiendo el RAID de forma persistente**

Por defecto, Linux no recuerda automáticamente los datos del RAID, de forma que, si se apaga la máquina, se pierden todos los datos sobre el mismo.

Para evitarlo, debemos almacenar dicha información en el fichero `/etc/mdadm.conf`

Así pues, antes de apagar la máquina para crear el nuevo disco, creamos este fichero con los datos de nuestro RAID

Empezamos añadiendo los nombres de nuestras particiones, precedidas por la palabra `DEVICE`, al fichero

Para ello, nos valemos del comando `mdadm --detail --scan`, que nos da toda la información sobre los dispositivos de nuestro RAID

Ejecutamos dicho comando redireccionando la salida a `/etc/mdadm.conf`, y comprobamos que se ha ejecutado correctamente, mostrando el contenido del fichero con el comando `more`

```
[root@localhost ~]# echo "DEVICE /dev/sde1 /dev/sde2 /dev/sde3" > /etc/mdadm.conf
[root@localhost ~]# mdadm --detail --scan >> /etc/mdadm.conf
[root@localhost ~]# more /etc/mdadm.conf
DEVICE /dev/sde1 /dev/sde2 /dev/sde3
ARRAY /dev/md0 metadata=1.2 name=localhost.localdomain:0 UUID=ce36d767:da9f7f04:2187c80c:397b7ea4
```

Tras reiniciar, comprobamos que el RAID se mantiene, pero solo con las particiones sde2 y sd3

```
[root@localhost ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sr0                                  11:0    1  1024M  0 rom
sdc                                  8:32    0    8G  0 disk
├─sdc1                              8:33    0    8G  0 part
│   └─privado (dm-2)                253:2    0    8G  0 crypt
sdb                                  8:16    0   384M  0 disk
├─sdb1                              8:17    0  133,3M  0 part  /tmp
├─sdb2                              8:18    0  133,4M  0 part  /home
└─sdb3                              8:19    0  109,8M  0 part  [SWAP]
sda                                  8:0     0    20G  0 disk
├─sda1                              8:1     0   500M  0 part  /boot
├─sda2                              8:2     0   19,5G  0 part
│   └─VolGroup-lv_root (dm-0)      253:0    0   17,6G  0 lvm  /
│       └─VolGroup-lv_swap (dm-1)  253:1    0    2G  0 lvm
sde                                  8:64    0   512M  0 disk
├─sde1                              8:65    0  133,3M  0 part
├─sde2                              8:66    0  133,4M  0 part
│   └─md0                          9:0     0   264M  0 raid5
└─sde3                              8:67    0  133,4M  0 part
    └─md0                          9:0     0   264M  0 raid5
```

Ejecutamos `mdadm --detail /dev/md0` para comprobar el estado del cluster

```
[root@localhost ~]# mdadm --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Mon Jan 15 15:49:55 2018
    Raid Level : raid5
    Array Size : 270336 (264.00 MiB 276.82 MB)
  Used Dev Size : 135168 (132.00 MiB 138.41 MB)
    Raid Devices : 3
    Total Devices : 2
 Persistence : Superblock is persistent

    Update Time : Mon Jan 15 19:56:02 2018
      State : clean, degraded
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

    Layout : left-symmetric
    Chunk Size : 512K

         Name : localhost.localdomain:0 (local to host localhost.localdomain)
        UUID : ce36d767:da9f7f04:2187c80c:397b7ea4
        Events : 64

   Number   Major   Minor   RaidDevice State
     0         0         0         0   removed
     1         8         66         1   active sync  /dev/sde2
     3         8         67         2   active sync  /dev/sde3
```

Vemos que en el campo state pone "clean, degraded", indicando que el RAID esta dañado, manteniendo lo obtenido en los apartados anteriores.

- **Creando el nuevo disco duro**

Apagamos la máquina para crear un nuevo disco duro, al que le pondremos de tamaño 256 MB

Reiniciamos, y comprobamos que el nuevo disco duro es detectado correctamente, con el nombre de sdf

```
Last login: Mon Jan 15 19:56:18 from 10.0.2.2
[root@localhost ~]# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sr0                                  11:0    1 1024M  0 rom
sda                                  8:0     0   20G  0 disk
├─sda1                              8:1     0   500M  0 part  /boot
├─sda2                              8:2     0  19,5G  0 part
│   └─VolGroup-lv_root (dm-0) 253:0     0  17,6G  0 lvm  /
│       └─VolGroup-lv_swap (dm-1) 253:1     0    2G  0 lvm
sdb                                  8:16    0  384M  0 disk
├─sdb1                              8:17    0 133,3M  0 part  /tmp
├─sdb2                              8:18    0 133,4M  0 part  /home
├─sdb3                              8:19    0 109,8M  0 part  [SWAP]
sdc                                  8:32    0    8G  0 disk
├─sdc1                              8:33    0    8G  0 part
│   └─privado (dm-2)          253:2     0    8G  0 crypt
sdd                                  8:48    0    1G  0 disk
sde                                  8:64    0   512M  0 disk
├─sde1                              8:65    0 133,3M  0 part
├─sde2                              8:66    0 133,4M  0 part
│   └─md0                          9:0     0   264M  0 raid5
├─sde3                              8:67    0 133,4M  0 part
│   └─md0                          9:0     0   264M  0 raid5
sdf                                  8:80    0   256M  0 disk
[root@localhost ~]# _
```

Creamos una nueva partición, de 128 MB, en el nuevo disco duro. Esta partición será la que reemplazará a la partición dañada del RAID.

```
[root@localhost ~]# fdisk /dev/sdf
El dispositivo no contiene una tabla de particiones DOS válida ni una etiqueta de disco Sun o SGI o OSF
Building a new DOS disklabel with disk identifier 0x6e862169.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Atención: el indicador 0x0000 inválido de la tabla de particiones 4 se corregirá mediante w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Orden (m para obtener ayuda): n
Acción de la orden
e   Partición extendida
p   Partición primaria (1-4)
p
Número de partición (1-4): 1
Primer cilindro (1-32, default 1): 1
Last cilindro, +cilindros or +size{K,M,G} (1-32, default 32): +128M

Orden (m para obtener ayuda): w
¡Se ha modificado la tabla de particiones!

Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.
```

Comprobamos que la partición ha sido creada correctamente

```
[root@localhost ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sr0	11:0	1	1024M	0	rom	
sda	8:0	0	20G	0	disk	
└─sda1	8:1	0	500M	0	part	/boot
└─sda2	8:2	0	19,5G	0	part	
└─VolGroup-lv_root (dm-0)	253:0	0	17,6G	0	lvm	/
└─VolGroup-lv_swap (dm-1)	253:1	0	2G	0	lvm	
sdb	8:16	0	384M	0	disk	
└─sdb1	8:17	0	133,3M	0	part	/tmp
└─sdb2	8:18	0	133,4M	0	part	/home
└─sdb3	8:19	0	109,8M	0	part	[SWAP]
sdc	8:32	0	8G	0	disk	
└─sdc1	8:33	0	8G	0	part	
└─privado (dm-2)	253:2	0	8G	0	crypt	
sdd	8:48	0	1G	0	disk	
sde	8:64	0	512M	0	disk	
└─sde1	8:65	0	133,3M	0	part	
└─sde2	8:66	0	133,4M	0	part	
└─md0	9:0	0	264M	0	raid5	
└─sde3	8:67	0	133,4M	0	part	
└─md0	9:0	0	264M	0	raid5	
sdf	8:80	0	256M	0	disk	
└─sdf1	8:81	0	133,3M	0	part	

- **Añadiendo la nueva partición al RAID**

Para incorporar un nuevo disco al RAID, se usa la opción `--add` de `mdadm`, con la siguiente sintaxis:

```
mdadm --add [dispositivo_raid] [dispositivo_a_añadir]
```

En nuestro caso, el comando será:

```
mdadm --add /dev/md0 /dev/sdf1
```

Lo ejecutamos en nuestra máquina para añadir la nueva partición, y comprobamos el nuevo estado del RAID

```
[root@localhost ~]# mdadm --add /dev/md0 /dev/sdf1
mdadm: added /dev/sdf1
[root@localhost ~]# mdadm --details /dev/md0
mdadm: unrecognized option '--details'
Usage: mdadm --help
       for help
[root@localhost ~]# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Mon Jan 15 15:49:55 2018
    Raid Level : raid5
    Array Size : 270336 (264.00 MiB 276.82 MB)
  Used Dev Size : 135168 (132.00 MiB 138.41 MB)
    Raid Devices : 3
    Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Mon Jan 15 21:09:44 2018
      State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 0
Spare Devices : 0

    Layout : left-symmetric
    Chunk Size : 512K

           Name : localhost.localdomain:0 (local to host localhost.localdomain)
          UUID : ce36d767:da9f7f04:2187c80c:397b7ea4
         Events : 114

   Number   Major   Minor   RaidDevice State
     4         8       81         0   active sync  /dev/sdf1
     1         8       66         1   active sync  /dev/sde2
     3         8       67         2   active sync  /dev/sde3
[root@localhost ~]#
```

Tras añadir el disco, comprobamos que el estado del RAID es correcto, y ya no se muestra degradado

Añadimos la nueva partición al fichero de configuración del RAID

```
DEVICE /dev/sde1 /dev/sde2 /dev/sde3 /dev/sdf1
ARRAY /dev/md0 metadata=1.2 name=localhost.localdomain:0 UUID=ce36d767:da9f7f04:2187c80c:397b7ea4
~
~
~
```

Aprovechamos para volver a añadir la partición sde1, ya sin errores.

```
[root@localhost ~]# mdadm --add /dev/md0 /dev/sde1
mdadm: re-added /dev/sde1
[root@localhost ~]# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Mon Jan 15 15:49:55 2018
    Raid Level : raid5
    Array Size : 270336 (264.00 MiB 276.82 MB)
  Used Dev Size : 135168 (132.00 MiB 138.41 MB)
    Raid Devices : 3
  Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Mon Jan 15 21:28:31 2018
      State : clean
 Active Devices : 3
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 1


    Layout : left-symmetric
  Chunk Size : 512K

       Name : localhost.localdomain:0 (local to host localhost.localdomain)
       UUID : ce36d767:da9f7f04:2187c80c:397b7ea4
       Events : 115

   Number   Major   Minor   RaidDevice State
     4         8       81         0   active sync   /dev/sdf1
     1         8       66         1   active sync   /dev/sde2
     3         8       67         2   active sync   /dev/sde3

     0         8       65         -   spare        /dev/sde1
```

Tras añadirlo, comprobamos que la partición se configura como hot-spare

Actualizamos el mdadm.conf con la nueva configuración

```
DEVICE /dev/sde1 /dev/sde2 /dev/sde3 /dev/sdf1
ARRAY /dev/md0 metadata=1.2 spares=1 name=localhost.localdomain:0 UUID=ce36d767:da9f7f04:2187c80c:397b7ea4
~
~
```

- **Eliminando otro elemento del RAID**

Para probar la configuración, provocamos un fallo en la partición sde2 del RAID, y comprobamos el estado del RAID tras ese fallo.

```
[root@localhost ~]# mdadm --fail /dev/md0 /dev/sde2
mdadm: set /dev/sde2 faulty in /dev/md0
[root@localhost ~]# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Mon Jan 15 15:49:55 2018
    Raid Level : raid5
    Array Size : 270336 (264.00 MiB 276.82 MB)
  Used Dev Size : 135168 (132.00 MiB 138.41 MB)
    Raid Devices : 3
  Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Mon Jan 15 23:00:56 2018
      State : clean
 Active Devices : 3
Working Devices : 3
 Failed Devices : 1
  Spare Devices : 0

    Layout : left-symmetric
  Chunk Size : 512K

       Name : localhost.localdomain:0 (local to host localhost.localdomain)
      UUID : ce36d767:da9f7f04:2187c80c:397b7ea4
     Events : 141

   Number   Major   Minor   RaidDevice State
     4         8       81         0   active sync   /dev/sdf1
     0         8       65         1   active sync   /dev/sde1
     3         8       67         2   active sync   /dev/sde3
     1         8       66         -   faulty        /dev/sde2
```

Tras provocar el fallo en la partición sde2, vemos que la partición sde1 toma su lugar en el RAID, reemplazando la partición dañada.

Montamos el RAID, y comprobamos que los datos se conservan correctamente.

```
[root@localhost ~]# mount /dev/md0 /mnt/RAID/
[root@localhost ~]# ls /mnt/RAID/
ficheroabc.txt fichero.txt lost+found nuevofichero.txt prueba.txt qwerty.txt
[root@localhost ~]# more /mnt/RAID/ficheroabc.txt
abc
```

Vemos que, en efecto, se conservan todos los ficheros, y los datos almacenados en ellos se leen correctamente

Parte 2: LVM

1. Creando un nuevo disco duro virtual

Usando el asistente de VirtualBox, creamos un nuevo disco duro de 512 MB

Ejecutamos `lsblk` y comprobamos que la máquina lo detecta correctamente, con el nombre de `sdg`

```
[root@localhost ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sr0	11:0	1	1024M	0	rom	
sda	8:0	0	20G	0	disk	
├─sda1	8:1	0	500M	0	part	/boot
├─sda2	8:2	0	19,5G	0	part	
│ └─VolGroup-lv_root (dm-0)	253:0	0	17,6G	0	lvm	/
│ └─VolGroup-lv_swap (dm-1)	253:1	0	2G	0	lvm	
sdb	8:16	0	384M	0	disk	
├─sdb1	8:17	0	133,3M	0	part	/tmp
├─sdb2	8:18	0	133,4M	0	part	/home
└─sdb3	8:19	0	109,8M	0	part	[SWAP]
sdc	8:32	0	8G	0	disk	
├─sdc1	8:33	0	8G	0	part	
│ └─privado (dm-2)	253:2	0	8G	0	crypt	
sdd	8:48	0	1G	0	disk	
sde	8:64	0	512M	0	disk	
├─sde1	8:65	0	133,3M	0	part	
│ └─md0	9:0	0	264M	0	raid5	
├─sde2	8:66	0	133,4M	0	part	
├─sde3	8:67	0	133,4M	0	part	
│ └─md0	9:0	0	264M	0	raid5	
sdf	8:80	0	256M	0	disk	
├─sdf1	8:81	0	133,3M	0	part	
│ └─md0	9:0	0	264M	0	raid5	
sdg	8:96	0	512M	0	disk	

2. Creando un volumen físico

El primer paso para crear una sistema LVM es crear un volumen físico. El volumen físico (o PV) es la estructura que nos permitirá unir varios dispositivos en una única unidad virtual.

- **Creando la partición con fdisk**

El primer paso para crear un volumen físico es crear una partición de tipo LVM dentro del disco duro real.

Creamos una partición que ocupe todo el disco, usando la opción 'n' ya vista en pasos anteriores, y asignamos el tipo a la nueva partición

Para asignar el tipo usaremos la opción 't', que nos preguntará el código del tipo que queremos asignar a la partición. En este caso, el tipo será Linux LVM, y su código será 8e.

```
[root@localhost ~]# fdisk /dev/sdg
El dispositivo no contiene una tabla de particiones DOS válida ni una etiqueta de disco Sun o SGI o OSF
Building a new DOS disklabel with disk identifier 0x4b18fb5a.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Atención: el indicador 0x0000 inválido de la tabla de particiones 4 se corregirá mediante w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Orden (m para obtener ayuda): n
Acción de la orden
e Partición extendida
p Partición primaria (1-4)
p
Número de partición (1-4): 1
Primer cilindro (1-65, default 1):
Using default value 1
Last cilindro, +cilindros or +size{K,M,G} (1-65, default 65):
Using default value 65

Orden (m para obtener ayuda): t
Se ha seleccionado la partición 1
Código hexadecimal (escriba L para ver los códigos): 8e
Se ha cambiado el tipo de sistema de la partición 1 por 8e (Linux LVM)

Orden (m para obtener ayuda): w
¡Se ha modificado la tabla de particiones!

Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.
```

- **Creando el PV dentro de la partición con pvcreate**

Una vez creada la partición LVM, usamos el comando `pvcreate`, que inicializa una unidad o disco para usar LVM, creando un volumen físico sobre ella.

La sintaxis de `pvcreate` es:

`pvcreate [unidad]`

En nuestro caso, la unidad es `/dev/sdg1`, así que nuestro comando quedaría como

`pvcreate /dev/sdg1`

```
[root@localhost ~]# pvcreate /dev/sdg1
Physical volume "/dev/sdg1" successfully created
```

El comando nos indica que el volumen físico se ha creado correctamente.

- **Comprobando los datos del PV con pvs**

Lo confirmamos usando el comando `pvs`, que nos devuelve información de ocupación de cada volumen físico

```
[root@localhost ~]# pvs
PV          VG          Fmt  Attr PSize  PFree
/dev/sda2   VolGroup    lvm2  a--u  19,51g      0
/dev/sdg1                   lvm2  ---  509,84m  509,84m
```

Vemos que el comando nos muestra dos volúmenes físicos: `/dev/sda2`, correspondiente a nuestro sistema; y `/dev/sdg1`, que es el que acabamos de crear.

Vemos que la información se muestra en forma de tabla, con varios campos

Los campos incluidos en la tabla son los siguientes:

- PV y VG: dispositivo y grupo de volúmenes al que pertenece
- Fmt: formato del volumen físico
- Attr: Atributos del volumen
- PSize y PFree: Tamaño total y libre (no asignado a lv).

En nuestro caso, vemos que nuestro PV no está asociado a ningún grupo de volúmenes, que su formato es `lvm2`, no tiene atributos, y todo su espacio está libre.

3. Creando un grupo de volúmenes

Una vez creado el volumen físico, pasamos a crear un grupo de volúmenes, que albergará nuestras unidades lógicas en el LVM, y en el que vamos a incluir nuestra unidad física.

- **Creando el grupo de volúmenes con vgcreate**

Para crear el grupo de volúmenes usaremos el comando `vgcreate`, que nos permite unir varios volúmenes físicos dentro de un mismo grupo de volúmenes.

La sintaxis de `vgcreate` es:

```
vgcreate [opciones] [nombre_grupo] [unidad_1] [unidad_2] ... [unidad_n]
```

En nuestro caso, vamos a crear un grupo de nombre `lvmtest`, y vamos a añadir la unidad `/dev/sdg1`. No vamos a usar ninguna opción.

Así pues, el comando quedaría de la siguiente forma:

```
vgcreate lvmtest /dev/sdg1
```

Ejecutamos este comando en la terminal:

```
[root@localhost ~]# vgcreate lvmtest /dev/sdg1
Volume group "lvmtest" successfully created
```

- **Comprobando el grupo de volúmenes con pvs y vgs**

El comando nos avisa de que el grupo de volúmenes ha sido creado correctamente. Lo comprobamos con el comando `pvs`

```
[root@localhost ~]# pvs
PV          VG          Fmt  Attr PSize  PFree
/dev/sda2   VolGroup    lvm2 a--u 19,51g    0
/dev/sdg1   lvmtest     lvm2 a--u 508,00m 508,00m
```

Comprobamos que, efectivamente, la unidad física `sdg1` aparece asociada al grupo de volúmenes `lvmtest`

También comprobamos el estado del propio grupo de volúmenes, usando el comando `vgs`.

El comando `vgs` es un comando similar a `pvs`, que nos devuelve información sobre los grupos de volúmenes que tenemos en nuestro sistema.

```
[root@localhost ~]# vgs
VG          #PV #LV #SN Attr   VSize  VFree
VolGroup    1   2   0 wz--n- 19,51g    0
lvmtest     1   0   0 wz--n- 508,00m 508,00m
```

De nuevo, la información se nos muestra en forma de tabla, con los siguientes campos:

- VG: nombre del grupo de volúmenes
- PV: número de volúmenes físicos que lo forman
- LV: número de volúmenes lógicos que lo usan
- Attr: Atributos
- VSize y VFree: Tamaño total y libre

En nuestro caso, vemos que el grupo *lvmttest* que acabamos de crear solo tiene asociado un volumen físico, de 508 MB, y que está completamente libre.

4. Creando volúmenes lógicos

Una vez creado el grupo de volúmenes, vamos a crear dos volúmenes lógicos dentro de él.

- **Creando volúmenes lógicos con lvcreate**

Para ello, usaremos el comando *lvcreate*, con la sintaxis:

```
lvcreate -L [tamaño] -n [nombre_volumen_logico] [nombre_grupo]
```

En nuestro caso, crearemos 2 volúmenes lógicos de 128 MB cada uno, con los nombre *lv1* y *lv2*.

Así que los comando que ejecutaremos son:

```
lvcreate -L 128M -n lv1 lvmttest
lvcreate -L 128M -n lv2 lvmttest
```

Ejecutamos los comandos, que nos indican que los volúmenes lógicos se han creado correctamente

```
[root@localhost ~]# lvcreate -L 128M -n lv1 lvmttest
Logical volume "lv1" created.
[root@localhost ~]# lvcreate -L 128M -n lv2 lvmttest
Logical volume "lv2" created.
```

- **Comprobando los volúmenes lógicos con lvs**

Comprobamos los volúmenes creados con el comando *lvs*, que nos da información sobre los volúmenes lógicos existentes en el sistema.

```
[root@localhost ~]# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv_root VolGroup -wi-ao--- 17,57g
lv_swap VolGroup -wi-a---- 1,94g
lv1     lvmttest -wi-a---- 128,00m
lv2     lvmttest -wi-a---- 128,00m
```

Los atributos que muestra este comando son:

- LV y VG: nombre del volumen lógico y nombre del grupo de volúmenes
- Attr: atributos
- LSize : tamaño del volumen
- Pool: conjunto de bloques libres (para LV más grandes que extensiones)

En nuestro caso, vemos los dos volúmenes lógicos que acabamos de crear, pertenecientes al grupo *lvmtest*, y con un tamaño de 128 MB cada uno.

5. Montando los volúmenes lógicos en el sistema

Ya con los volúmenes lógicos creados y asociados a sus respectivos volúmenes físicos, mediante los grupos de volúmenes, pasamos a montarlos dentro del árbol de directorios de nuestro sistema.

El primer volumen, *lv1*, lo montaremos en */tmp*; y el segundo volumen, *lv2*, lo montaremos en */home*.

- **Desmontando */tmp* y */home***

Los directorios */tmp* y */home* están ocupados por otras particiones, así que los desmontamos usando *umount*

```
[root@localhost ~]# umount /home  
[root@localhost ~]# umount /tmp
```

- **Buscando la unidad asociada a los volúmenes lógicos**

Una vez liberados los directorios, y antes de montar los volúmenes lógicos, debemos crear un sistema de ficheros dentro de ellos, para lo cual tenemos que saber la unidad que tienen asociada.

Para saberlo, usaremos el comando *lsblk*.


```
[root@localhost ~]# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sr0                                11:0    1  1024M  0 rom
sda                                 8:0     0    20G  0 disk
├─sda1                             8:1     0   500M  0 part  /boot
├─sda2                             8:2     0  19,5G  0 part
│   └─VolGroup-lv_root (dm-0) 253:0     0  17,6G  0 lvm    /
│       └─VolGroup-lv_swap (dm-1) 253:1     0    2G  0 lvm
sdb                                 8:16    0   384M  0 disk
├─sdb1                             8:17    0  133,3M  0 part
├─sdb2                             8:18    0  133,4M  0 part
├─sdb3                             8:19    0  109,8M  0 part  [SWAP]
sdc                                 8:32    0     8G  0 disk
├─sdc1                             8:33    0     8G  0 part
│   └─privado (dm-2)          253:2     0     8G  0 crypt
sdd                                 8:48    0     1G  0 disk
sde                                 8:64    0   512M  0 disk
├─sde1                             8:65    0  133,3M  0 part
│   └─md0                         9:0     0   264M  0 raid5
├─sde2                             8:66    0  133,4M  0 part
├─sde3                             8:67    0  133,4M  0 part
│   └─md0                         9:0     0   264M  0 raid5
sdf                                 8:80    0   256M  0 disk
├─sdf1                             8:81    0  133,3M  0 part
│   └─md0                         9:0     0   264M  0 raid5
sdg                                 8:96    0   512M  0 disk
├─sdg1                             8:97    0  509,9M  0 part
│   └─lvmtest-lvm1 (dm-3)      253:3     0   128M  0 lvm
│       └─lvmtest-lvm2 (dm-4) 253:4     0   128M  0 lvm
```

Vemos que nuestros volúmenes lógicos tienen asociadas las unidades dm-3 y dm-4, respectivamente

- **Formateando los volúmenes lógicos**

Formateamos los volúmenes como ext4 usando mkfs.

```
[root@localhost ~]# mkfs -t ext4 /dev/dm-3
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
32768 nodos-i, 131072 bloques
6553 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67371008
16 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
2048 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
    8193, 24577, 40961, 57345, 73729

Escribiendo las tablas de nodos-i: hecho
Creating journal (4096 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 37 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

```
[root@localhost ~]# mkfs -t ext4 /dev/dm-4
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de S0: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
32768 nodos-i, 131072 bloques
6553 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67371008
16 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
2048 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729

Escribiendo las tablas de nodos-i: hecho
Creating journal (4096 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 39 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

- **Montando las particiones**

Montamos las particiones usando mount . Montamos dm-3 en /tmp y dm-4 en /home

```
[root@localhost ~]# mount /dev/dm-3 /tmp
[root@localhost ~]# mount /dev/dm-4 /home
```

Comprobamos que se han montado correctamente usando lsblk

```
[root@localhost ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sr0	11:0	1	1024M	0	rom	
sda	8:0	0	20G	0	disk	
├─sda1	8:1	0	500M	0	part	/boot
├─sda2	8:2	0	19,5G	0	part	
│ └─VolGroup-lv_root (dm-0)	253:0	0	17,6G	0	lvm	/
│ └─VolGroup-lv_swap (dm-1)	253:1	0	2G	0	lvm	
sdb	8:16	0	384M	0	disk	
├─sdb1	8:17	0	133,3M	0	part	
├─sdb2	8:18	0	133,4M	0	part	
├─sdb3	8:19	0	109,8M	0	part	[SWAP]
sdc	8:32	0	8G	0	disk	
├─sdc1	8:33	0	8G	0	part	
└─privado (dm-2)	253:2	0	8G	0	crypt	
sdd	8:48	0	1G	0	disk	
sde	8:64	0	512M	0	disk	
├─sde1	8:65	0	133,3M	0	part	
│ └─md0	9:0	0	264M	0	raid5	
├─sde2	8:66	0	133,4M	0	part	
├─sde3	8:67	0	133,4M	0	part	
│ └─md0	9:0	0	264M	0	raid5	
sdf	8:80	0	256M	0	disk	
├─sdf1	8:81	0	133,3M	0	part	
│ └─md0	9:0	0	264M	0	raid5	
sdg	8:96	0	512M	0	disk	
├─sdg1	8:97	0	509,9M	0	part	
│ └─lvmttest-lvm1 (dm-3)	253:3	0	128M	0	lvm	/tmp
│ └─lvmttest-lvm2 (dm-4)	253:4	0	128M	0	lvm	/home

6. Creando ficheros en /home

Nos situamos en el directorio /home, y creamos varios ficheros de texto. Comprobamos que se han creado correctamente mostrando su contenido con more.

```
[root@localhost ~]# cd /home/
[root@localhost home]# echo "fichero1" >> f1
[root@localhost home]# echo "fichero2" >> f2
[root@localhost home]# echo "fichero4" >> f4
[root@localhost home]# echo "fichero3" >> f3
[root@localhost home]# ls
f1 f2 f3 f4 lost+found
[root@localhost home]# more f*
:~::~:
f1
:~::~:
fichero1
:~::~:
f2
:~::~:
fichero2
:~::~:
f3
:~::~:
fichero3
:~::~:
f4
:~::~:
fichero4
```

7. Incrementando el tamaño de un volumen lógico

En este paso, incrementaremos el tamaño de la unidad montada en /home, añadiéndole 100 MB más.

Para ello, incrementaremos el tamaño del volumen lógico, y seguidamente redimensionaremos el sistema de ficheros.

- **Aumentando el volumen lógico**

Para aumentar el tamaño del volumen lógico usaremos el comando `lvextend`, con la siguiente sintaxis:

```
lvextend -L +[incremento] [unidad]
```

En nuestro caso, el incremento es de 100 MB. Para la unidad, necesitamos usar la nomenclatura de LVM, la cual indica el volumen lógico como un fichero dentro de un directorio con el nombre de su grupo, en el directorio /dev. En este caso, la unidad será: /dev/lvmtest/lvm2

De esta forma, el comando quedará como:

```
lvextend -L +100M /dev/lvmtest/lvm2
```

Ejecutamos el comando:

```
[root@localhost ~]# lvextend -L +100M /dev/lvmtest/lvm2
Size of logical volume lvmtest/lvm2 changed from 128,00 MiB (32 extents) to 228,00 MiB (57 extents).
Logical volume lvm2 successfully resized.
```

Tras ejecutar el comando, este nos indica que el volumen ha sido redimensionado correctamente.

Lo comprobamos con `lvs`.

```
[root@localhost ~]# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv_root VolGroup -wi-ao--- 17,57g
lv_swap VolGroup -wi-a---- 1,94g
lvm1    lvmtest -wi-ao--- 128,00m
lvm2    lvmtest -wi-ao--- 228,00m
```

Vemos que, efectivamente, la unidad lógica ahora tiene 228 MB, en lugar de los 128 MB que tenía anteriormente.

- **Redimensionando el sistema de ficheros**

Con el anterior paso hemos logrado aumentar el tamaño del volumen lógico pero el sistema de ficheros sigue con el mismo tamaño.

Para comprobarlo, usaremos el comando `df`, que nos da información sobre el sistema de ficheros de un dispositivo.

```
[root@localhost ~]# df /dev/dm-4 -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/lvmtest-lvm2
                124M  5,6M  113M   5% /home
```

Vemos que, tal como esperábamos, el sistema de ficheros sigue teniendo 124 MB

Para redimensionar el sistema de ficheros, usaremos el comando `resize2fs`, cuya sintaxis es:

```
resize2fs [unidad]
```

Ejecutamos el comando, que nos avisa de que la unidad ha sido redimensionada

```
[root@localhost ~]# resize2fs /dev/lvmtest/lvm2
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/lvmtest/lvm2 is mounted on /home; on-line resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/lvmtest/lvm2 to 233472 (1k) blocks.
El sistema de ficheros en /dev/lvmtest/lvm2 tiene ahora 233472 bloques.
```

Tras ejecutar `df`, comprobamos que el sistema de ficheros ya tiene 204 MB, habiéndose ajustado correctamente al tamaño del volumen lógico.

```
[root@localhost ~]# df /dev/dm-4 -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/lvmtest-lvm2
                221M  6,1M  204M   3% /home
```

Conclusión

La configuración de RAID y LVM es algo mas compleja que el particionamiento tradicional con un sistema de ficheros directamente sobre la partición.

A cambio, nos permite tener una mayor fiabilidad, en el caso de RAID; o flexibilidad, en el caso de LVM.

En esta práctica hemos comprobado la tolerancia a fallos de los RAID, forzando el fallo en uno de los discos que lo componen y comprobando como los ficheros siguen estando accesibles y sin fallos. También hemos comprobado la autoreparación de un RAID mediante la incorporación de un nuevo disco, y la configuración *hot-spare* dentro del mismo RAID.

En LVM, también hemos podido comprobar la flexibilidad que nos ofrece, permitiendonos redimensionar el volumen, y su sistema de ficheros asociado, sin necesidad de desmontarlo físicamente.