

# Trabajos prácticos

Aquí se proponen ejercicios para implementar algunos de los puntos abordados en el capítulo. En cada uno de ellos se da un ejemplo comentado de la realización del ejercicio, que deberá adaptar a la configuración de sus sistemas.

## 1. Configuración de un espacio de almacenamiento en RAID 1 (espejo)

Tiene que configurar un nuevo espacio de almacenamiento en la máquina `alpha`, dedicado a una nueva aplicación de programación/desarrollo. Para ello, le va a atribuir un sistema de archivos de tipo `ext4`, de 5 Gb, que tendrá que montarse durante el arranque del sistema en el directorio `/opt/programacion`. Por cuestiones de seguridad, el sistema de archivos de la aplicación usa un volumen RAID de nivel 1 (espejo), administrado por `md`.

### Comandos y archivos útiles

- ✓ `blkid`
- ✓ `fdisk`
- ✓ `pvs`
- ✓ `mdadm`
- ✓ `mkfs`
- ✓ `mkdir`
- ✓ `mount`
- ✓ `/etc/fstab`

### Etapas

1. Identifique los dispositivos de almacenamiento del servidor `alpha` reconocidos por el sistema. Observe la configuración actual (discos, particiones, organización LVM...).
2. Instale dos discos duros (o memorias USB) de capacidad más o menos idéntica, al menos 5GB. Determine su archivo especial en modo de bloque asociado.
3. Cree un volumen RAID `md` de nivel 1 usando estos dos discos, y compruebe después sus características.
4. Cree un sistema de archivos `ext4` en el volumen RAID. Cree el directorio de montaje `/opt/programacion` y monte el sistema de archivos. Cree algunos directorios y archivos en el directorio `/opt/programacion`. Declare el sistema de archivos en montaje automático en el archivo `/etc/fstab`. Compruebe el correcto funcionamiento después de haber reiniciado el sistema.
5. Limpieza: desmonte el sistema de archivos, suprima la línea que creó en `/etc/fstab`, y después suprima el volumen RAID.

### Resumen de los comandos y resultados en pantalla

1. Identifique los dispositivos de almacenamiento del servidor `alpha` reconocidos por el sistema. Observe la configuración actual (discos, particiones, organización LVM...).

El servidor **alpha** corre una distribución Debian 10. Para conocer los dispositivos de almacenamiento reconocidos por el sistema, se puede usar el comando `blkid`:

```
root@alpha:~# blkid
/dev/sda1: UUID="912527a1-daaf-4735-8995-41d2933c1eb6"
TYPE="ext2" PARTUUID="b402ad25-01"
```

```

/dev/sda5: UUID="qat3CM-JYKG-vhND-Yuq3-YOU8-RGd4-xokJiY"
TYPE="LVM2_member" PARTUUID="b402ad25-05"
/dev/mapper/debian10--vg-root: UUID="94ae415d-6667-4984-b715-c938a3922e28"
TYPE="ext4"
/dev/mapper/debian10--vg-swap_1: UUID="cb5da901-a012-40e4-aeb7-09d220bfc575"
TYPE="swap"
/dev/mapper/debian10--vg-var: UUID="c1981b20-01ef-4e48-9f5b-bc3b2bff3326"
TYPE="ext4"
/dev/mapper/debian10--vg-tmp: UUID="e59feb5a-1018-4775-a293-6f0f66230b68"
TYPE="ext4"
/dev/mapper/debian10--vg-home: LABEL="HOME" UUID="9a997fa4-3e9b-4148-
b5e8-8b6374fed370" TYPE="ext4"
/dev/mapper/debian10--vg-lvol0: UUID="efd18095-e45f-44b4-b4b9-4f3ea7091b7a"
SEC_TYPE="ext2" TYPE="ext3"
/dev/mapper/debian10--vg-lvol1: UUID="39283d6c-e6f8-44cc-aa57-46ef454523bf"
SEC_TYPE="ext2" TYPE="ext3"

```

Se ven distintos volúmenes lógicos LVM, todos dentro del grupo de volúmenes `debian10--vg` (creado durante la instalación de la distribución). Por lo menos hay un volumen físico LVM, como le confirma el comando `pvs`:

```

root@alpha:~# pvs
PV      VG      Fmt Attr PSize  PFree
/dev/sda5  debian10-vg lvm2 a--  <148,81g <74,14g

```

La partición nº 5 del disco `/dev/sda` ha sido definida como volumen físico LVM, perteneciente al grupo de volúmenes `debian10--vg`.

Se han encontrado dos particiones nº 1 y 5, en el disco `/dev/sda`. Se puede mostrar su tabla de particiones con el comando `fdisk`:

```

root@alpha:~# fdisk -l /dev/sda
Disco /dev/sda: 149,1 GiB, 160041885696 bytes, 312581808 sectores
Modelo de disco: ST9160821AS
Unidades: sectores de de 1 × 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes

```

Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes

Tipo de etiqueta de disco: dos

Identificador del disco: 0xb402ad25

Disposit.	Inicio	Comienzo	Final	Sectores	Tamaño	Id	Tipo
/dev/sda1	*	2048	499711	497664	243M	83	Linux
/dev/sda2		501758	312580095	312078338	148,8G	5	Extendida
/dev/sda5		501760	312580095	312078336	148,8G	8e	LVM Linux

El disco `/dev/sda` tiene dos particiones útiles: una partición principal n° 1 y un lector lógico n° 5 en la partición extendida n° 2.

2. Instale dos discos duros (o memorias USB) de capacidad más o menos idéntica, al menos 5GB. Determine su archivo especial en modo de bloque asociado.

Como no quedan dispositivos de almacenamiento disponibles, instalaremos, para el ejercicio, dos memorias USB de 16 GB cada una.

Después de haberlas insertado, se usa el comando `blkid` para mostrar los discos `sd*`:

```
blkid | grep sd
/dev/sda1: UUID="912527a1-daaf-4735-8995-41d2933c1eb6"
TYPE="ext2" PARTUUID="b402ad25-01"
/dev/sda5: UUID="qat3CM-JYKG-vhND-Yuq3-YOU8-RGd4-xokJiY"
TYPE="LVM2_member" PARTUUID="b402ad25-05"
/dev/sdb1: LABEL_FATBOOT="STORE N GO" LABEL="STORE N GO"
UUID="8B04-A7A2" TYPE="vfat" PARTUUID="8b04a7a2-01"
/dev/sdc: UUID="E26E-7671" TYPE="vfat"
```

El sistema reconoce las dos memorias USB, una como un disco entero, `/dev/sdc`, y la otra por su partición n°1, `/dev/sdb1`.

Los archivos especiales en modo de bloque asociados a las dos memorias USB han sido creados dinámicamente en el directorio `/dev`:

```
ls -l /dev/sdb /dev/sdc
brw-rw---- 1 root disk 8, 16 marzo 21 10:16 /dev/sdb
brw-rw---- 1 root disk 8, 32 marzo 21 10:16 /dev/sdc
```

3. Cree un volumen RAID `md` de nivel 1 usando dos discos, y compruebe después sus características.

Se creará un volumen RAID de nivel `1` (espejo), combinando las dos memorias USB usando el comando `mdadm`. ¿Se encuentra este comando en el sistema?

```
root@alpha:~# mdadm -V
-bash: mdam: orden no encontrada
```

Hay que instalar el paquete `mdadm`:

```
root@alpha:~# apt-get install mdadm
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
exim4-base exim4-config exim4-daemon-light libgnutls-dane0 libgsasl7
libkyotocabinet16v5 libmailutils5 libntlm0 libunbound8
mailutils mailutils-common
Paquetes sugeridos:
exim4-doc-html | exim4-doc-info eximon4 spf-tools-perl swaks mailutils-mh
mailutils-doc dracut-core
Se instalarán los siguientes paquetes NUEVOS:
exim4-base exim4-config exim4-daemon-light libgnutls-dane0 libgsasl7
libkyotocabinet16v5 libmailutils5 libntlm0 libunbound8
mailutils mailutils-common mdadm
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 5949 ko de archivos.
Se utilizarán 13,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
[...]
```

```
root@alpha:~# mdadm -V
mdadm - v4.1 - 2018-10-01
```

Creación del volumen RAID nivel 1 `/dev/md0` con los dos discos `/dev/sdb` y `/dev/sdc`. Cuidado, el contenido de estos se perderá.

```
root@alpha:~# mdadm -C /dev/md0 -l 1 -n 2 /dev/sdb /dev/sdc
mdadm: partition table exists on /dev/sdb
mdadm: partition table exists on /dev/sdb but will be lost or
      meaningless after creating array
mdadm: Note: this array has metadata at the start and
      may not be suitable as a boot device. If you plan to
      store '/boot' on this device please ensure that
      your boot-loader understands md/v1.x metadata, or use
      --metadata=0.90
mdadm: partition table exists on /dev/sdc
mdadm: partition table exists on /dev/sdc but will be lost or
      meaningless after creating array
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
root@alpha:~#
```

Se comprueba el volume RAID creado:

```
root@alpha:~# mdadm -D /dev/md0
/dev/md0:
  Version: 1.2
  Creation Time: Sat Mar 21 11:06:38 2020
  Raid Level: raid1
  Array Size: 15152128 (14.45 GiB 15.52 GB)
  Used Dev Size: 15152128 (14.45 GiB 15.52 GB)
  Raid Devices: 2
  Total Devices: 2
  Persistence: Superblock is persistent

  Update Time: Sat Mar 21 11:13:29 2020
  State: clean, resyncing
```

Active Devices: 2  
 Working Devices: 2  
 Failed Devices: 0  
 Spare Devices: 0

Consistency Policy: resync

Resync Status: 18% complete

Name: alpha:0 (local to host alpha)  
 UUID: adb224ef:61aa7b4d:eee749c4:5aa5f190  
 Events: 2

Number	Major	Minor	RaidDevice	State
0	8	16	0	active sync /dev/sdb
1	8	32	1	active sync /dev/sdc

El volumen se acaba de crear, los dos discos se están sincronizando.

Se muestra el detalle del archivo especial de bloque creado:

```
root@alpha:~# ls -l /dev/md0
brw-rw---- 1 root disk 9, 0 marzo 21 11:06 /dev/md0
```

El número mayor es el 9, asociado al piloto md.

Se muestran las características de los dos discos espejos:

```
root@alpha:~# blkid /dev/sdb /dev/sdc
/dev/sdb: UUID="adb224ef-61aa-7b4d-eee7-49c45aa5f190" UUID_SUB="426fb624-5e68-7b5e-745b-d20f8823bdb7" LABEL="alpha:0" TYPE="linux_raid_member"
/dev/sdc: UUID="adb224ef-61aa-7b4d-eee7-49c45aa5f190" UUID_SUB="df7f200c-0993-555f-0514-defc41ee624f" LABEL="alpha:0" TYPE="linux_raid_member"
```

El comando mdadm les ha asignado un label, un tipo y un UUID idénticos.

4. Cree un sistema de archivos ext4 en el volumen RAID. Cree el directorio de montaje `/opt/programacion` y monte el sistema de archivos. Cree algunos directorios y archivos en el directorio `/opt/programacion`. Declare el sistema de archivos en montaje automático en el archivo `/etc/fstab`. Compruebe el correcto funcionamiento después de haber reiniciado el sistema.

Se crea un sistema de archivos en el volumen RAID:

```
root@alpha:~# blkid /dev/sdb /dev/sdc
/dev/sdb: UUID="adb224ef-61aa-7b4d-eee7-49c45aa5f190" UUID_SUB="426fb624-5e68-7b5e-745b-d20f8823bdb7" LABEL="alpha:0" TYPE="linux_raid_member"
/dev/sdc: UUID="adb224ef-61aa-7b4d-eee7-49c45aa5f190" UUID_SUB="df7f200c-0993-555f-0514-defc41ee624f" LABEL="alpha:0" TYPE="linux_raid_member"
root@alpha:~# mkfs -t ext4 /dev/md0
mke2fs 1.44.5 (15-Dec-2018)
Creating filesystem with 3788032 4k blocks and 948416 inodes
Filesystem UUID: 9968e68f-646f-4a5d-a490-354f2ce12fe3
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
root@alpha:~#
```

El sistema de archivos creado ocupa todo el espacio del volumen RAID, es decir  $3788032 * 4 \text{ kB}$ , alrededor de 15 GB.

Se comprueba el sistema de archivos, forzando un control completo:

```
root@alpha:~# e2fsck -f /dev/md0
e2fsck 1.44.5 (15-Dec-2018)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
```



Pass 4: Checking reference counts

Pass 5: Checking **group** summary information

/dev/md0: 11/948416 files (0.0% non-contiguous), 86168/3788032 blocks

El sistema de archivos es coherente.

Se crea el directorio de montaje y se monta el sistema de archivos:

```
root@alpha:~# mkdir /opt/programacion
root@alpha:~# mount /dev/md0 /opt/programacion
root@alpha:~# ls -l /opt/programacion
total 16
drwx----- 2 root root 16384 marzo 21 11:24 lost+found
root@alpha:~#
```

Se crean algunos archivos y directorios en el punto de montaje:

```
root@alpha:~# mkdir /opt/programacion/bin /opt/programacion/datos
root@alpha:~# cp /usr/bin/date /opt/programacion/bin/prog
root@alpha:~# echo "Información innovadora" >/opt/programacion/datos/informe.txt
root@alpha:~#
```

Se declara el sistema de archivos en montaje automático, con esta línea en `/etc/fstab`:

```
/dev/md0 /opt/programacion ext4 defaults 0 2
```

Antes de reiniciar, hay que asegurarse de que la línea del archivo `/etc/fstab` es correcta, desmontando el sistema de archivos y volviéndolo a montar con el comando `mount`

`Puntodemontaje` :

```
root@alpha:~# umount /opt/programacion
root@alpha:~# mount /opt/programacion
root@alpha:~# mount -v | grep programacion
```

```
/dev/md0 on /opt/programacion type ext4 (rw,relatime)
root@alpha:~#
```

Se reinicia el sistema, y después se comprueba el montaje:

```
root@alpha:~# shutdown -r 0
[...]
mount | grep prog
/dev/md0 on /opt/programacion type ext4 (rw,relatime)
```

5. Limpieza: desmonte el sistema de archivos, suprima la línea que creó en `/etc/fstab`, y después suprima el volumen RAID.

Se desmonta el sistema de archivos, se suprime la declaración en `/etc/fstab`.

```
root@alpha:~# umount /dev/md0
root@alpha:~# vi /etc/fstab
#/dev/md0 /opt/programacion ext4 defaults 0 2
```

Se para el volumen RAID y se quitan las estructuras RAID de las dos memorias USB:

```
mdadm -S /dev/md0
root@alpha:~# mdadm -S /dev/md0
mdadm: stopped /dev/md0
root@alpha:~# mdadm --zero-superblock /dev/sdb /dev/sdc
root@alpha:~# blkid /dev/sdb /dev/sdc
/dev/sdb: UUID="E26E-7671" TYPE="vfat"
/dev/sdc: PTUUID="8b04a7a2" PTTYPE="dos"
root@alpha:~#
```

## 2. Implementación de un grupo de volúmenes

Necesita configurar un nuevo espacio de almacenamiento en la máquina `beta`, dedicado a los datos de cuatro usuarios. Para ello, debe usar un disco de al menos 13 GB. Para que cada usuario tenga sus datos en un espacio personal, decide crear un grupo de volúmenes que contendrá un volumen lógico de 3 GB para cada usuario. Contendrán cada uno un sistema de archivos ext4, que se montará en cuatro directorios en `/var`.

### Comandos y archivos útiles

- ~ `blkid`
- ~ `fdisk`
- ~ `pvcreate`
- ~ `vgcreate`
- ~ `lvcreate`
- ~ `pvs`, `lvs`, `vgs`, `pvdisplay`, `vgdisplay`, `lvdisplay`
- ~ `mkfs`
- ~ `mkdir`
- ~ `mount`
- ~ `/etc/fstab`
- ~ `Lvremove`, `vgremove`, `pvremove`

### Etapas

1. Detecte los dispositivos de almacenamiento del servidor `beta` reconocidos por el sistema. Observe la configuración actual (discos, particiones, organización LVM ...). Elija un espacio de disco disponible (disco, partición o memoria USB) de al menos 16 GB.
2. Cree un volumen físico LVM en el espacio de almacenamiento seleccionado y compruebe sus características.
3. Cree un grupo de volúmenes LVM `data-user` integrando este volumen físico y compruebe sus características.
4. Cree cuatro volúmenes lógicos, de `user1` a `user4`, de 3 GB, en el grupo de volúmenes LVM `data-user`, y compruebe sus características.
5. Cree un sistema de archivos ext4 en cada volumen lógico. Cree cuatro directorios de montaje, de `user1` a `user4`, en `/var/datos-user` y monte los sistemas de archivos.
6. Declare los cuatro sistemas de archivos en montaje automático usando el camino de acceso `/dev/NombreGrp/NombreVol`. Compruebe que se montan correctamente gracias a la información de `/etc/fstab`.
7. Limpieza: desmonte los sistemas de archivos, suprima las líneas implicadas en este procedimiento en `/etc/fstab` y suprima los objetos LVM creados.

### Resumen de los comandos y resultados en pantalla

1. Detecte los dispositivos de almacenamiento del servidor `beta` reconocidos por el sistema. Observe la configuración actual (discos, particiones, organización LVM ...). Elija un espacio de disco disponible (disco, partición o memoria USB) de al menos 16 GB.

El servidor `beta` corre una distribución CentOS 8. Para conocer los dispositivos de almacenamiento reconocidos por el sistema, se puede usar el comando `blkid`:

```
[root@beta ~]# blkid | sort
/dev/mapper/cl-home: LABEL="HOME" UUID="01010101-0a0a-0b0b-0c0c-000000000001"
TYPE="xfs"
/dev/mapper/cl-root: UUID="64a55ea9-3628-452c-8e67-6a1c35e30713" TYPE="xfs"
/dev/mapper/cl-swap: UUID="33519596-768d-4364-a202-90bf82a5b524" TYPE="swap"
/dev/mapper/cl-var: UUID="eaef606f-e454-4166-afc9-a4a5433a8237" TYPE="xfs"
/dev/sda1: UUID="26bfdc97-1e14-437a-9fb2-a6016aa4a347" TYPE="ext4"
PARTUUID="12deb3a0-01"
/dev/sda2: UUID="Jlyv1F-7nhy-3mej-SwLJ-x03U-SOpl-q47hCQ" TYPE="LVM2_member"
PARTUUID="12deb3a0-02"
/dev/sda3: PARTUUID="12deb3a0-03"
/dev/sda4: UUID="493a1a1c-0163-4193-a55e-2488445ec96a" TYPE="ext4"
PARTUUID="12deb3a0-04"
/dev/sdd: PTUUID="033ac21d" PTTYPE="dos"
```

Se ven cuatro volúmenes lógicos LVM, todos dentro del grupo de volúmenes `cl` (creado durante la instalación de la distribución). El disco `/dev/sda` tiene cuatro particiones, de `1` a `4`, una de tipo LVM, dos con un sistema de archivos ext4 y una sin sistema de archivos. También hay un disco `/dev/sdd`, sin partición.

Se pueden listar los volúmenes físicos existentes con el comando `pvs`:

```
[root@beta ~]# pvs
PV      VG Fmt Attr PSize  PFree
/dev/sda2 cl lvm2 a-- <154,09g 0
```

La partición n° `2` del disco `/dev/sda` ha sido definida como un volumen físico LVM, que pertenece al grupo de volúmenes `cl`.

Se puede mostrar la tabla de particiones de `/dev/sda` con el comando `fdisk`:

```
[root@beta ~]# fdisk -l /dev/sda
Disco /dev/sda: 298,1 GiB, 320072933376 bytes, 625142448 sectores
```

Unidades: sectores de  $1 \times 512 = 512$  bytes  
 Tamaño de sector (lógico/físico): 512 bytes / 4096 bytes  
 Tamaño de E/S (mínimo/óptimo): 4096 bytes / 4096 bytes  
 Tipo de etiqueta de disco: dos  
 Identificador de disco: 0x12deb3a0

Disposit.	Inicio	Comienzo	Final	Sectores	Tamaño	Id	Tipo
/dev/sda1	*	2048	2099199	2097152	1G	83	Linux
/dev/sda2		2099200	325257215	323158016	154,1G	8e	LVM Linux
/dev/sda3		325257216	614649855	289392640	138G	83	Linux
/dev/sda4		614649856	625125375	10475520	5G	c W95	FAT32 (LBA)

El disco `/dev/sda` tiene cuatro particiones principales.

Se decide usar el disco disponible `/dev/sdd` para el desarrollo del ejercicio. Se comprueba su tamaño y su configuración con `fdisk`:

```
[root@beta ~]# blkid /dev/sdd
/dev/sdd: PTUUID="033ac21d" PTYPE="dos"
[root@beta ~]# fdisk -l /dev/sdd
Disco /dev/sdd: 14,5 GiB, 15525216256 bytes, 30322688 sectores
Unidades: sectores de 1 × 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 4096 bytes
Tamaño de E/S (mínimo/óptimo): 4096 bytes / 4096 bytes
Tipo de etiqueta de disco: dos
Identificador de disco: 0x033ac21d
```

El disco tiene una capacidad útil de 14,5 GB y no contiene ninguna partición.

2. Cree un volumen físico LVM en el espacio de almacenamiento seleccionado y compruebe sus características.

Se crea el volumen físico LVM en el disco `/dev/sdd`:

```
[root@beta ~]# pvcreate /dev/sdd
WARNING: dos signature detected on /dev/sdd at offset 510. Wipe it? [y/n]: y
Wiping dos signature on /dev/sdd.
Physical volume "/dev/sdd" successfully created.
```

Se muestra su estado y sus características:

```
[root@beta ~]# pvdisplay /dev/sdd
"/dev/sdd" is a new physical volume of "<14,46 GiB"
--- NEW Physical volume ---
PV Name          /dev/sdd
VG Name
PV Size          <14,46 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          Oblvn2-qJXm-qw9i-luQS-gW4o-aShr-YofkKQ
```

El volumen físico todavía no pertenece a ningún grupo de volúmenes. No hay ninguna extensión física definida.

El comando le ha asignado un UUID y un tipo LVM:

```
[root@beta ~]# blkid /dev/sdd
/dev/sdd: UUID="Oblvn2-qJXm-qw9i-luQS-gW4o-aShr-YofkKQ" TYPE="LVM2_member"
```

3. Cree un grupo de volúmenes LVM `data-user` integrando este volumen físico y compruebe sus características.

Se crea el grupo de volúmenes LVM:

```
[root@beta ~]# vgcreate data-user /dev/sdd
Volume group "data-user" successfully created
```

Se muestra su estado y sus características:

```
[root@beta ~]# vgdisplay data-user
--- Volume group ---
VG Name          data-user
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No 1
VG Access        read/write
VG Status        resizable
MAX LV          0
Cur LV         0
Open LV         0
Max PV          0
Cur PV         1
Act PV         1
VG Size         <14,46 GiB
PE Size         4,00 MiB
Total PE        3701
Alloc PE / Size 0 / 0
Free PE / Size  3701 / <14,46 GiB
VG UUID         xWdU5b-f8kT-TnNn-kVUk-mJmG-OOZk-IJSvOo
```

El grupo de volúmenes admite un número ilimitado de volúmenes lógicos (`MAX LV = 0`) y de volúmenes físicos (`Max PV = 0`). Este administra un volumen físico, con un tamaño de extensión física de 4 MB (`PE Size`).

Se muestra el estado y las características del volumen físico:

```
[root@beta ~]# pvdisplay /dev/sdd
--- Physical volume ---
PV Name          /dev/sdd
VG Name          data-user
PV Size          <14,46 GiB / not usable 2,00 MiB
```



```

Allocatable    yes
PE Size        4,00 MiB
Total PE       3701
Free PE        3701
Allocated PE   0
PV UUID        Oblvn2-qJXm-qw9i-luQS-gW4o-aShr-YofkKQ

```

El volumen físico está bien integrado en el grupo de volúmenes, y sus extensiones físicas están definidas.

4. Cree cuatro volúmenes lógicos, de `user1` a `user4`, de 3 GB, en el grupo de volúmenes LVM `data-user`, y compruebe sus características.

Se crean los cuatro volúmenes lógicos LVM:

```

[root@beta ~]# lvcreate -L 3g -n user1 data-user
Logical volume "user1" created.
[root@beta ~]# lvcreate -L 3g -n user2 data-user
Logical volume "user2" created.
[root@beta ~]# lvcreate -L 3g -n user3 data-user
Logical volume "user3" created.
[root@beta ~]# lvcreate -L 3g -n user4 data-user
Logical volume "user4" created.

```

Se muestra el estado y las características de los volúmenes lógicos del grupo de volúmenes:

```

[root@beta ~]# lvdisplay data-user
--- Logical volume ---
LV Path          /dev/data-user/user1
LV Name          user1
VG Name          data-user
LV UUID          6xDWtl-by3t-9nmV-ET49-qKUv-7Qwp-mGOXTV
LV Write Access   read/write
LV Creation host, time beta, 2020-03-21 17:20:43 +0000

```

```

LV Status      available
# open         0
LV Size        3,00 GiB
Current LE     768
Segments       1
Allocation     inherit
Read ahead sectors  auto
- currently set to 8192
Block device   253:2

```

--- Logical volume ---

```

LV Path        /dev/data-user/user2
LV Name        user2
VG Name        data-user
LV UUID        qJOCqW-ctg1-skTu-vYAW-o1y9-dsUi-sCagH7
LV Write Access read/write
LV Creation host, time beta, 2020-03-21 17:20:50 +0000
LV Status      available
# open         0
LV Size        3,00 GiB
Current LE     768
Segments       1
Allocation     inherit
Read ahead sectors  auto
- currently set to 8192
Block device   253:3

```

--- Logical volume ---

```

LV Path        /dev/data-user/user3
LV Name        user3
VG Name        data-user
LV UUID        dhZOFw-hjLQ-aAci-dNwe-QAHD-pdYV-rHv4om
LV Write Access read/write
LV Creation host, time beta, 2020-03-21 17:20:56 +0000
LV Status      available
# open         0
LV Size        3,00 GiB
Current LE     768
Segments       1
Allocation     inherit
Read ahead sectors  auto

```

```

- currently set to 8192
Block device      253:6

--- Logical volume ---
LV Path           /dev/data-user/user4
LV Name           user4
VG Name           data-user
LV UUID           borV7j-SXcK-clTo-h3yc-vp5B-kQtl-HGNS5S
LV Write Access    read/write
LV Creation host, time beta, 2020-03-21 17:21:01 +0000
LV Status          available
# open             0
LV Size            3,00 GiB
Current LE         768
Segments           1
Allocation          inherit
Read ahead sectors  auto
- currently set to 8192
Block device      253:7

```

Cada volumen tiene 3 GB y está identificado por un UUID específico.

Se muestran los diferentes caminos de acceso al archivo especial de bloque asociado a cada volumen lógico:

```

[root@beta ~]# ls -ld /dev/data-user
drwxr-xr-x. 2 root root 120 21 marzo 17:21 /dev/data-user
[root@beta ~]# ls -ld /dev/data-user/*
lrwxrwxrwx. 1 root root 7 21 marzo 17:20 /dev/data-user/user1 -> ../dm-2
lrwxrwxrwx. 1 root root 7 21 marzo 17:20 /dev/data-user/user2 -> ../dm-3
lrwxrwxrwx. 1 root root 7 21 marzo 17:20 /dev/data-user/user3 -> ../dm-6
lrwxrwxrwx. 1 root root 7 21 marzo 17:21 /dev/data-user/user4 -> ../dm-7
[root@beta ~]# ls -l /dev/mapper/data--user-*
lrwxrwxrwx. 1 root root 7 21 marzo 17:20 /dev/mapper/data--user-user1 -> ../dm-2
lrwxrwxrwx. 1 root root 7 21 marzo 17:20 /dev/mapper/data--user-user2 -> ../dm-3
lrwxrwxrwx. 1 root root 7 21 marzo 17:20 /dev/mapper/data--user-user3 -> ../dm-6
lrwxrwxrwx. 1 root root 7 21 marzo 17:21 /dev/mapper/data--user-user4 -> ../dm-7
[root@beta ~]# ls -l /dev/dm-[2367]
brw-rw----. 1 root disk 253, 2 21 marzo 17:20 /dev/dm-2
brw-rw----. 1 root disk 253, 3 21 marzo 17:20 /dev/dm-3

```

```
brw-rw----. 1 root disk 253,6 21 marzo 17:20 /dev/dm-6
brw-rw----. 1 root disk 253,7 21 marzo 17:21 /dev/dm-7
```

Para el camino que usa `/dev/mapper`, el carácter guion está escrito doble en el nombre del grupo, para evitar la confusión con el guion de separación entre el nombre del grupo y el del volumen.

Se muestra el estado del grupo de volúmenes y el del volumen físico:

```
[root@beta ~]# vgdisplay data-user
--- Volume group ---
VG Name          data-user
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No 5
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          4
Open LV          0
Max PV           0
Cur PV          1
Act PV           1
VG Size          <14,46 GiB
PE Size          4,00 MiB
Total PE         3701
Alloc PE / Size  3072 / 12,00 GiB
Free PE / Size   629 / <2,46 GiB
VG UUID          xWdU5b-f8kT-TnNn-kVUk-mJmG-OOZk-IJSvOo
[root@beta ~]# pvdisplay /dev/sdd
--- Physical volume ---
PV Name          /dev/sdd
VG Name          data-user
PV Size          <14,46 GiB / not usable 2,00 MiB
Allocatable      yes
PE Size          4,00 MiB
Total PE         3701
Free PE          629
Allocated PE     3072
```

PV UUID      Oblvn2-qJXm-qw9i-luQS-gW4o-aShr-YofkKQ

5. Cree un sistema de archivos `ext4` en cada volumen lógico. Cree cuatro directorios de montaje, de `user1` a `user4`, en `/var/datos-user` y monte los sistemas de archivos.

Para poder explotar los cuatro volúmenes lógicos LVM, se implementan sistemas de archivos que se montarán en la arborescencia.

Creación de un sistema de archivos en los volúmenes `user1`:

```
[root@beta ~]# mkfs -t ext4 /dev/data-user/user1 /dev/data-user/user2
/dev/data-user/user3 /dev/data-user/user4
mke2fs 1.44.6 (5-Mar-2019)
mkfs.ext4: Invalid blocks on device '/dev/data-user/user2'
/dev/data-user/user1
[root@beta ~]# mkfs -t ext4 /dev/data-user/user1
mke2fs 1.44.6 (5-Mar-2019)
Creating filesystem with 786432 4k blocks and 196608 inodes
Filesystem UUID: c393c106-e65a-4f6d-ac5d-dade61f96aab
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

También se puede usar directamente el comando `mke2fs`:

```
[root@beta ~]# mke2fs -t ext4 /dev/data-user/user2
mke2fs 1.44.6 (5-Mar-2019)
mke2fs 1.44.6 (5-Mar-2019)
Creating filesystem with 786432 4k blocks and 196608 inodes
Filesystem UUID: c393c106-e65a-4f6d-ac5d-dade61f96aab
```

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912

Allocating group tables: done

Writing inode tables: done

Creating journal (16384 blocks): done

Writing superblocks and filesystem accounting information: done

Para los dos últimos, se añade la opción en modo silencioso en pantalla `-q`:

```
[root@beta ~]# mke2fs -q -t ext4 /dev/data-user/user3
```

```
[root@beta ~]# mke2fs -q -t ext4 /dev/data-user/user4
```

Se crean los directorios de montaje:

```
[root@beta ~]# mkdir -p /var/datos-user/user1 /var/datos-user/user2
```

```
/var/datos-user/user3 /var/datos-user/user4
```

Se pueden montar los sistemas de archivos de los volúmenes lógicos, usando los caminos de acceso

`/dev/datos-user/user[1-4]` :

```
[root@beta ~]# mount /dev/data-user/user1 /var/datos-user/user1
```

```
[root@beta ~]# mount /dev/data-user/user1 /var/datos-user/user2
```

```
[root@beta ~]# mount /dev/data-user/user1 /var/datos-user/user3
```

```
[root@beta ~]# mount /dev/data-user/user1 /var/datos-user/user4
```

Los volúmenes lógicos se encuentran a disposición de los usuarios (a condición de que los derechos de accesos estén definidos correctamente). Se pueden crear directorios y archivos:

```
[root@beta ~]# echo "Bienvenido a LVM" > /var/datos-user/user1/hello.txt
```

```
[root@beta ~]# echo "Bienvenido a LVM" > /var/datos-user/user2/hello.txt
```

```
[root@beta ~]# echo "Bienvenido a LVM" > /var/datos-user/user3/hello.txt
```

```
[root@beta ~]# echo "Bienvenido a LVM" > /var/datos-user/user4/hello.txt
```

```
[root@beta ~]# ls -lR /var/datos-user/
```

```
/var/datos-user/:
```

```
total 16
```

```
drwxr-xr-x. 3 root root 4096 21 marzo 17:54 user1
```

```
drwxr-xr-x. 3 root root 4096 21 marzo 17:54 user2
```

```
drwxr-xr-x. 3 root root 4096 21 marzo 17:54 user3
```

```
drwxr-xr-x. 3 root root 4096 21 marzo 17:54 user4
```

```
/var/datos-user/user1:
```

```
total 20
```

```
-rw-r--r--. 1 root root 19 21 marzo 17:55 hello.txt
```

```
drwx-----. 2 root root 16384 21 marzo 17:41 lost+found
```

```
/var/datos-user/user1/lost+found:
```

```
total 0
```

```
/var/datos-user/user2:
```

```
total 20
```

```
-rw-r--r--. 1 root root 19 21 marzo 17:55 hello.txt
```

```
drwx-----. 2 root root 16384 21 marzo 17:41 lost+found
```

```
/var/datos-user/user2/lost+found:
```

```
total 0
```

```
/var/datos-user/user3:
```

```
total 20
```

```
-rw-r--r--. 1 root root 19 21 marzo 17:55 hello.txt
```

```
drwx-----. 2 root root 16384 21 marzo 17:41 lost+found
```

```
/var/datos-user/user3/lost+found:
```

```
total 0
```

```
/var/datos-user/user4:
```

```
total 20
```

```
-rw-r--r--. 1 root root 19 21 marzo 17:55 hello.txt
```

```
drwx-----. 2 root root 16384 21 marzo 17:41 lost+found
```

```
/var/datos-user/user4/lost+found:
```

```
total 0
```

6. Declare los cuatro sistemas de archivos en montaje automático usando el camino de acceso `/dev/NombreGrp/NombreVol` . Compruebe que se montan correctamente gracias a la información de `/etc/fstab` .

Se declaran los cuatro sistemas de archivos en `/etc/fstab` , con las líneas:

```
/dev/data-user/user1 /var/datos-user/user1 ext4 defaults 0 2
/dev/data-user/user2 /var/datos-user/user2 ext4 defaults 0 2
/dev/data-user/user3 /var/datos-user/user3 ext4 defaults 0 2
/dev/data-user/user4 /var/datos-user/user4 ext4 defaults 0 2
```

Se desmontan los cuatro sistemas de archivos con el comando `umount` (es preferible usar el directorio de montaje en lugar del camino de acceso, ya que a veces el comando no lo gestiona bien):

```
umount /var/datos-user/user1 /var/datos-user/user2 /var/datos-user/user3
/var/datos-user/user4
```

Se montan todos los sistemas de archivos configurados en modo automático en `/etc/fstab` y no montados:

```
[root@beta ~]# mount -a
[root@beta ~]# mount | grep user
tmpfs on /run/user/42 type tmpfs (rw,nosuid,nodev,relatime,seclabel,
size=380316k,mode=700,uid=42,gid=42)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,
size=380316k,mode=700,uid=1000,gid=1000)
/dev/mapper/data--user-user1 on /var/datos-user/user1
type ext4 (rw,relatime,seclabel)
/dev/mapper/data--user-user2 on /var/datos-user/user2
type ext4 (rw,relatime,seclabel)
/dev/mapper/data--user-user3 on /var/datos-user/user3
type ext4 (rw,relatime,seclabel)
/dev/mapper/data--user-user4 on /var/datos-user/user4
type ext4 (rw,relatime,seclabel)
```



```
[root@beta ~]#
```

Los cuatro sistemas de archivos están montados. Se puede comprobar que el comando `mount` usa el camino de acceso `/dev/mapper/NombreGrp-NombreVol`, y no el especificado en `/etc/fstab`.

7. Limpieza: desmonte los sistemas de archivos, suprima las líneas implicadas en este procedimiento en `/etc/fstab`, suprima los objetos LVM creados.

Se desmontan los cuatro sistemas de archivos:

```
[root@beta ~]# umount /var/datos-user/user1 /var/datos-user/user2
/var/datos-user/user3 /var/datos-user/user4
```

Se suprimen las cuatro líneas del archivo `/etc/fstab`.

Y después se suprimen los cuatro volúmenes lógicos:

```
[root@beta ~]# lvremove data-user/user1 data-user/user2 data-user/user3
data-user/user4
Do you really want to remove active logical volume data-user/user1? [y/n]: y
Logical volume "user1" successfully removed
Do you really want to remove active logical volume data-user/user2? [y/n]: y
Logical volume "user2" successfully removed
Do you really want to remove active logical volume data-user/user3? [y/n]: y
Logical volume "user3" successfully removed
Do you really want to remove active logical volume data-user/user4? [y/n]: y
Logical volume "user4" successfully removed
```

Se suprime el grupo de volúmenes:

```
[root@beta ~]# vgremove data-user
Volume group "data-user" successfully removed
```

Se suprime el volumen físico:

```
[root@beta ~]# pvremove /dev/sdd
Labels on physical volume "/dev/sdd" successfully wiped.
```

Se muestra el conjunto de los objetos LVM del sistema:

```
[root@beta ~]# vgs
VG #PV #LV #SN Attr VSize VFree
cl 1 4 0 wz--n- <154,09g 0
[root@beta ~]# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
home cl -wi-ao---- <50,18g
root cl -wi-ao---- 50,00g
swap cl -wi-ao---- 3,91g
var cl -wi-ao---- 50,00g
[root@beta ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda2 cl lvm2 a-- <154,09g 0
```