

P1 L3 ISE:

Tras la práctica anterior, el servidor ya alojaba vídeos en el directorio /var. Ahora en esta práctica el cliente solicita mayor seguridad y disponibilidad de datos, ya que la información será sensible. Para llevar esto a cabo:

- . Configuramos RAID1 por software para tener una copia de los datos.
- . Ciframos el volumen lógico donde se ubicará /var para proteger la información ante accesos no autorizados.

INSTALACIÓN ALMALINUX

Partimos de una configuración inicial de Alma Linux con un único disco sda que contiene /boot y el grupo de volúmenes lógicos rl con volúmenes lógicos root y swap.

Vamos a comprobar la estructura con **lsblk**:

```
[angelamg@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   10G  0 disk
├─sda1       8:1    0    1G  0 part /boot
└─sda2       8:2    0    9G  0 part
   └─rl-root 253:0    0    8G  0 lvm  /
      └─rl-swap 253:1    0    1G  0 lvm  [SWAP]
sr0         11:0    1  2.1G  0 rom
```

Podemos observar:

-sda: disco principal de 10GB

 Sda1(1GB): boot

 Sda2 (9 GB): grupo LVM rl

 Rl-root (8GB)

 Rl-swap (1GB)

-sr0: la unidad del CD con la ISO

Para comenzar con la configuración que se nos ha pedido lo primero que vamos a hacer es añadir los dos discos y confirmamos su presencia ejecutando lsblk de nuevo:

```
langelamy@localhost ~1$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   10G  0 disk
├─sda1       8:1    0    1G  0 part /boot
├─sda2       8:2    0    9G  0 part
├─┌r1-root   253:0   0    8G  0 lvm /
│ └─r1-swap  253:1   0    1G  0 lvm [SWAP]
sdb          8:16   0    2G  0 disk
sdc          8:32   0    2G  0 disk
sr0         11:0    1   2.1G  0 rom
```

CREAR PARTICIONES CON FDISK

A continuación, se crean las particiones primarias necesarias en cada disco con la herramienta fdisk, una por disco:

Sudo fdisk /dev/sdb

Sudo fdisk /dev/sdc

Durante la ejecución de fdisk se siguen los pasos indicados en el guion:

N: crea nueva partición

P: mostrar tabla de particiones actual

W: escribir los cambios y salir

Esto se hace para antes de crear el RAID1, es necesario preparar los discos físicos (sdb y sdc) para que el sistema los reconozca como unidades utilizables.

Los discos nuevos están vacíos, sin tabla de particiones ni estructura definida, por eso usamos fdisk que nos permite crear y gestionar particiones.

Nuestro objetivo es definir en cada disco una partición primaria que ocupará todo su espacio.

Posteriormente estas particiones db1 y sdc1 serán las que se usarán para formar el dispositivo RAID1 mediante mdadm, el RAID solo puede

trabajar sobre particiones o discos completos por lo que este paso es imprescindible para estructurar el almacenamiento.

Creación de la partición /dev/sdb1 con fdisk:

```
[angelamy@localhost ~]$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x00d1d99c.

Command (m for help): p
Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00d1d99c

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): p
Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00d1d99c



| Device    | Boot | Start | End     | Sectors | Size | Id | Type  |
|-----------|------|-------|---------|---------|------|----|-------|
| /dev/sdb1 |      | 2048  | 4194303 | 4192256 | 2G   | 83 | Linux |



Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
[ 990.556483] sdb: sdb1
Syncing disks.

[angelamy@localhost ~]$
```

Creación de la partición /dev/sdc1 con fdisk:

```
langelamy@localhost ~l$ sudo fdisk /dev/sdc

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x43b32aa2.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): p
Disk /dev/sdc: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x43b32aa2

Device      Boot Start      End Sectors Size Id Type
/dev/sdc1                2048 4194303 4192256  2G 83 Linux

Command (m for help):
```

Verificación de las particiones creadas con **lsblk**:

```
[angelamg@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   10G  0 disk
├─sda1       8:1    0    1G  0 part /boot
├─sda2       8:2    0    9G  0 part
│   └─r1-root 253:0    0    8G  0 lvm  /
│       └─r1-swap 253:1    0    1G  0 lvm  [SWAP]
sdb          8:16    0    2G  0 disk
└─sdb1       8:17    0    2G  0 part
sdc          8:32    0    2G  0 disk
└─sdc1       8:33    0    2G  0 part
sr0         11:0    1   2.1G  0 rom
```

CREAR RAID1 CON MDADM

Una vez creadas las particiones `/dev/sdb1` y `/dev/sdc1` se procede a configurar RAID1 por software cuyo objetivo es mantener una copia de seguridad de los datos.

Creamos el dispositivo RAID1 `/dev/md0` con este comando:

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

El parámetro `--level=1` indica que será un RAID1 (copia de datos), y `--raid-devices=2` especifica que el conjunto está formado por dos discos.

```
[angelamg@localhost ~]$ sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1
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
Continue creating array [y/N]? y
mdadm: Defaulting to version 1.2 metadata
[ 3529.867926] md/raid1:md0: not clean -- starting background reconstruction
[ 3529.867944] md/raid1:md0: active with 2 out of 2 mirrors
[ 3529.867960] md0: detected capacity change from 0 to 4188160
mdadm: array /dev/md0 started.
[angelamg@localhost ~]$ [ 3529.879285] md: resync of RAID array md0
[ 3540.154962] md: md0: resync done.
```

Para verificar el estado del RAID usamos:

`sudo mdadm --detail /dev/md0`

```
langelamg@localhost ~]$ sudo mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
    Creation Time : Fri Oct 10 12:34:20 2025
    Raid Level : raid1
    Array Size : 2094080 (2045.00 MiB 2144.34 MB)
    Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Fri Oct 10 12:34:30 2025
    State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

Consistency Policy : resync

    Name : localhost.localdomain:0 (local to host localhost.localdomain)
    UUID : 7695ccc9:f49b0563:c25fcc6a:fc3b5ce9
    Events : 17

    Number Major Minor RaidDevice State
    0      8      17        0      active sync  /dev/sdb1
    1      8      33        1      active sync  /dev/sdc1

langelamg@localhost ~]$ ls /dev/
autofs      disk        hugepages   mem          rfskill     sdc1        stdin       tty15       tty24       tty33       tty42       tty51       tty60       tty
block       dm-0        hwrng       queuec      rl          sg0         stdout      tty16       tty25       tty34       tty43       tty52       tty61       udm
bsg         dm-1        initctl     net          rtc         sg1         tty         tty17       tty26       tty35       tty44       tty53       tty62       uhi
bus         dma_heap    input       null         rtc0        sg2         tty0        tty18       tty27       tty36       tty45       tty54       tty63       uir
cdrom       dri         kmsg        nvram       sda         sg3         tty1        tty19       tty28       tty37       tty46       tty55       tty7        ura
char         fb0         log          port         sda1        shm         tty10       tty2        tty29       tty38       tty47       tty56       tty8        usb
console     fd          loop-control ppp          sda2        snapshot   tty11       tty20       tty3        tty39       tty48       tty57       tty9        usb
core        full        mapper      ptmx         sdb          snd         tty12       tty21       tty30       tty4        tty49       tty58       ttyS0       usb
cpu         fuse        mcelog      pts          sdb1        sr0         tty13       tty22       tty31       tty40       tty5        tty59       ttyS1       use
cpu_dma_latency hpet       md0         random       sdc          stderr      tty14       tty23       tty32       tty41       tty50       tty6        ttyS2       vcs
```

La salida nos muestra información detallada del dispositivo /dev/md0 confirmando que los dos discos están activos como espejo.

CREAR PV DESDE MD0

Una vez creado el dispositivo RAID1 vamos a integrarlo en la gestión de volúmenes lógicos. Para esto vamos a inicializarlo como physical volumen (PV), es decir una cantidad física que puede formar parte de un grupo de volúmenes.

Comando ejecutado:

sudo pvcreate /dev/md0

A continuación, se comprueba el estado con los comandos:

sudo pvdisplay

sudo pvs

```
[angelamg@localhost ~]# sudo pvcreate /dev/md0
[sudo] password for angelamg:
Physical volume "/dev/md0" successfully created.
[angelamg@localhost ~]# pvdisplay
WARNING: Running as a non-root user. Functionality may be unavailable.
/run/lock/lvm/P_global:aux: open failed: Permission denied
[angelamg@localhost ~]# sudo pvdisplay
--- Physical volume ---
PV Name               /dev/sda2
VG Name               r1
PV Size               <9.00 GiB / not usable 3.00 MiB
Allocatable           yes (but full)
PE Size               4.00 MiB
Total PE              2303
Free PE               0
Allocated PE          2303
PV UUID               zxzR08-fZwL-HfEi-xu1E-tFyS-fbDY-HF5HAx

"/dev/md0" is a new physical volume of "<2.00 GiB"
--- NEW Physical volume ---
PV Name               /dev/md0
VG Name
PV Size               <2.00 GiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               PxUKU9-3M6L-onXb-eouz-ZnZN-HlrQ-cqqaJF

[angelamg@localhost ~]# sudo pvs
PV          VG Fmt Attr PSize PFree
/dev/md0    lvm2 ---  <2.00g <2.00g
/dev/sda2   r1 lvm2 a--  <9.00g    0
[angelamg@localhost ~]#
```

En la salida de `pvdisk` se observa de `/dev/md0` aparece como un nuevo volumen físico de 2.00 GiB y todavía no pertenece a ningún grupo de volúmenes (VG Name = None).

Esto confirma que el RAID ha sido reconocido por LVM y esta preparado para ser agrupado dentro de su nuevo grupo de volúmenes.

CREAR VG DESDE EL PV

Con el volumen físico `/dev/md0` inicializado se crea un nuevo grupo de volúmenes (VG) llamado `vg_raid1`.

Este grupo será el contenedor que gestione el espacio del almacenamiento disponible en RAID1.

```
[angelamg@localhost ~]$ sudo vgcreate vg_raid1 /dev/md0
[sudo] password for angelamg:
Volume group "vg_raid1" successfully created
[angelamg@localhost ~]$ sudo vgs
VG          #PV #LV #SN Attr   VSize  VFree
rl          1  2  0 wz--n- <9.00g    0
vg_raid1    1  0  0 wz--n- <2.00g <2.00g
[angelamg@localhost ~]$
```

Al visualizar con `vgs` vemos que en la columna de PV Physical Volumes en `rl` hay un PV (`/dev/sda2`) y en `vg_raid1` también hay uno (`/dev/md0`).

En LV (logical volumes) en `rl` hay 2 LV (root y swap) y en `vg_rais1` todavía hay 0 porque aun no hemos creado `new_var`.

El grupo `rl` pertenece al sistema principal y ya tiene dos volúmenes (root y swap), mientras que `vg_raid1` acaba de crearse sobre el RAID `/dev/md0` y está vacío, listo para alojar el nuevo volumen lógico que servirá como `/var`.

CREAR LV NEW_VAR DESDE EL VG

Una vez creas el grupo de volúmenes `vg_raid1` se procede a generar dentro de él un volumen lógico (LV) denominado `new_var` que servirá para alojar el directorio `/var`.

Comando utilizado:

sudo lvcreate -n new_var -L 1.8G vg_raid1

-n new_var: asigna el nombre del volumen lógico.

-L 1.8G: indica el tamaño del volumen, ligeramente inferior al tamaño total del grupo (2 GB) para dejar algo de margen interno.

vg_raid1: nombre del grupo de volúmenes donde se creará.

Para verificar el resultado usamos **sudo lvs**

```
[angelamg@localhost ~]# sudo lvcreate -n new_var -L 1.8G vg_raid1
[sudo] password for angelamg:
Rounding up size to full physical extent 1.80 GiB
Logical volume "new_var" created.
[angelamg@localhost ~]# sudo lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%   Move Log Cpy%Sync Convert
root    rl       -wi-ao---- <8.00g
swap    rl       -wi-ao---- 1.00g
new_var vg_raid1 -wi-a----- 1.80g
[angelamg@localhost ~]#
```

En la salida se observan tres volúmenes lógicos:

-root y swap, pertenecientes al grupo rl,

-new_var, dentro del grupo vg_raid1, con un tamaño de 1.8 GiB.

ENCRYPTAR CON CRYPTSETUP

Para garantizar la confidencialidad de la información almacenada en /var se cifra el volumen lógico creado en LUKS. Esto impide que los datos puedan leerse si se extrae el disco o se accede sin autorización.

Se formatea el volumen lógico con LUKS:

sudo cryptsetup luksFormat /dev/vg_raid1/new_var

Se abre el volumen cifrado asignándole un nombre descriptivo:

sudo cryptsetup luksOpen /dev/vg_raid1/new_var vg_raid1-new_var_crypt

El volumen se mapea en /dev/mapper/vg_raid1-new_var_crypt, que será el dispositivo cifrado donde más adelante se creará el sistema de ficheros y se montará /var.

Para que el sistema pueda desbloquear automáticamente el volumen cifrado al arrancar, se debe registrar en el fichero `/etc/crypttab`.

De esta forma se añade al archivo una línea con el identificador UUID del volumen cifrado, como indica el guion:

```
vg_raid1-new_var_crypt UUID=<UUID_del_LUKS> none
```

```
[angelamg@localhost ~]$ sudo blkid | grep crypto
/dev/mapper/vg_raid1-new_var: UUID="5d36b49f-2de4-4c1b-af22-e7d71530d612" TYPE="crypto_LUKS"
[angelamg@localhost ~]$
```

```
vg_raid1-new_var_crypt UUID=5b36b49f-2de4-4c1b-af22-e7d71530d612 none
```

MONTAR EL SISTEMA DE FICHEROS EN /VAR

Una vez configurado el volumen cifrado se crea sobre él un sistema de archivos XFS. Esto permitirá que el volumen pueda almacenar los datos de forma estructurada.

Se ejecutan los siguientes comandos:

```
sudo mkfs -t xfs /dev/mapper/vg_raid1-new_var_crypt
```

Formatea el volumen cifrado con el sistema de archivos XFS.

```
sudo mkdir /new_var
```

Crea punto de montaje temporal.

```
sudo mount /dev/mapper/vg_raid1-new_var_crypt /new_var
```

Monta el volumen cifrado para su comprobación.

```
[angelamg@localhost ~]$ sudo mkfs -t xfs /dev/mapper/vg_raid1-new_var_crypt
meta-data=/dev/mapper/vg_raid1-new_var_crypt isize=512    agcount=4, agsize=116992 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1     bigtime=1 inobtcount=1 nrext64=0
data      =                       bsize=4096   blocks=467968, imaxpct=25
         =                       sunit=0       swidth=0 blks
naming    =version 2              bsize=4096   ascii-ci=0, ftype=1
log       =internal log          bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none                  extsz=4096   blocks=0, rtextents=0
[angelamg@localhost ~]$ sudo mkdir /new_var
[angelamg@localhost ~]$ sudo mount /dev/mapper/vg_raid1-new_var_crypt /new_var/
[ 6412.354625] XFS (dm-3): Mounting U5 Filesystem 22ea3b8c-49bf-44ef-b424-6ac92d8bb315b
[ 6412.407687] XFS (dm-3): Ending clean mount
[angelamg@localhost ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0    0  10G  0 disk
├─sda1                              8:1    0   1G  0 part /boot
├─sda2                              8:2    0   9G  0 part
├─┌r1-root                          253:0   0   8G  0 lvm /
│ └─r1-swap                         253:1   0   1G  0 lvm [SWAP]
sdb                                  8:16   0   2G  0 disk
├─sdb1                              8:17   0   2G  0 part
├─┌md0                              9:0    0   2G  0 raid1
│ └─┌vg_raid1-new_var               253:2   0   1.8G  0 lvm
│   └─vg_raid1-new_var_crypt       253:3   0   1.8G  0 crypt /new_var
sdc                                  8:32   0   2G  0 disk
├─sdc1                              8:33   0   2G  0 part
├─┌md0                              9:0    0   2G  0 raid1
│ └─┌vg_raid1-new_var               253:2   0   1.8G  0 lvm
│   └─vg_raid1-new_var_crypt       253:3   0   1.8G  0 crypt /new_var
sr0                                  11:0    1  2.1G  0 rom
```

Verificamos el resultado con lsblk que nos muestra el dispositivo vg_raid1-new_var_crypt montado en el punto /new_var, confirmando que el volumen está operativo y accesible.

Antes de copiar el contenido del directorio /var al nuevo volumen cifrado, se aísla del sistema para evitar conflictos con servicios que estén accediendo a ficheros dentro de /var, esto lo hacemos con el comando:

sudo systemctl isolate rescue

A continuación, comprobamos el estado general del sistema y los servicios activos con **sudo systemctl status**:

```
● localhost.localdomain
  State: running
    Units: 346 loaded (incl. loaded aliases)
      Jobs: 3 queued
    Failed: 0 units
    Since: Fri 2025-10-10 11:35:41 CEST; 4h 8min ago
  systemd: 252-51.e19
  CGroup: /
├─init.scope
│   └─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 31
├─system.slice
│   ├─NetworkManager.service
│   │   └─11680 /usr/sbin/NetworkManager --no-daemon
│   ├─auditd.service
│   │   └─11670 /sbin/auditd
│   ├─chronyd.service
│   │   └─11703 /usr/sbin/chronyd -F 2
│   ├─crond.service
│   │   └─11727 /usr/sbin/crond -n
│   ├─dbus-broker.service
│   │   └─11669 /usr/bin/dbus-broker-launch --scope system --audit
│   │       └─11678 dbus-broker --log 4 --controller 9 --machine-id faf5cd8f57c743949618d01e00b37880 --max-bytes 5368
│   ├─firewalld.service
│   │   └─11684 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
│   ├─rsyslog.service
│   │   └─11735 /usr/sbin/rsyslogd -n
│   ├─sshd.service
│   │   └─11720 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
│   ├─systemd-hostnamed.service
│   │   └─11697 /usr/lib/systemd/systemd-hostnamed
│   ├─systemd-journald.service
│   │   └─583 /usr/lib/systemd/systemd-journald
│   ├─systemd-logind.service
│   │   └─11688 /usr/lib/systemd/systemd-logind
│   └─systemd-udevd.service
│       └─udev
│           └─596 /usr/lib/systemd/systemd-udevd
├─user.slice
│   └─user-1001.slice
│       ├─session-3.scope
│       │   └─11728 "login -- angelamg"
│       │       └─12244 -bash
│       │           └─12271 sudo systemctl status
│       │               └─12275 systemctl status
│       │                   └─12276 less
│       └─user@1001.service
│           └─init.scope
│               └─12235 /usr/lib/systemd/systemd --user
│                   └─12237 "(sd-pam)"
[angelamg@localhost ~]$
```

La salida muestra que el sistema se encuentra en estado running con servicios como NetworkManager, firewalld y sshd activos.

Esta comprobación confirma que el sistema está estable y preparado para realizar la copia de datos de forma segura.

Tras esta verificación, se procede a copiar los datos de /var al nuevo volumen montado en /new_var usando el comando:

```
sudo cp -a /var/. /new_var/
```

La opción -a (archive) conserva los permisos, propietarios, enlaces y atributos SELinux de los archivos originales.

El punto (/var/.) asegura que se copien todos los contenidos, incluidos los archivos ocultos.

Para verificar la copia, se listan los archivos y sus atributos de seguridad:

```
sudo ls -laZ /new_var
```

En la salida se observa que todos los directorios y archivos han sido copiados correctamente, manteniendo sus permisos.

```
langelamg@localhost ~]$ sudo cp -a /var/. /new_var/
langelamg@localhost ~]$ sudo ls -laZ /new_var/
total 16
drwxr-xr-x. 19 root root system_u:object_r:var_t:s0      264 Sep 26 17:12 .
dr-xr-xr-x. 19 root root system_u:object_r:root_t:s0     250 Oct 10 13:21 ..
-rw-r--r--.  1 root root system_u:object_r:etc_runtime_t:s0 208 Sep 26 17:06 .updated
drwxr-xr-x.  2 root root system_u:object_r:var_t:s0        6 Nov  3 2024 adm
drwxr-xr-x.  8 root root system_u:object_r:var_t:s0       88 Sep 26 17:07 cache
drwxr-xr-x.  2 root root system_u:object_r:kdump_crash_t:s0 6 Apr 25 04:50 crash
drwxr-xr-x.  3 root root system_u:object_r:system_db_t:s0  18 Sep 26 17:07 db
drwxr-xr-x.  2 root root system_u:object_r:var_t:s0        6 Nov  3 2024 empty
drwxr-xr-x.  2 root root system_u:object_r:public_content_t:s0 6 Nov  3 2024 ftp
drwxr-xr-x.  2 root root system_u:object_r:games_data_t:s0  6 Nov  3 2024 games
drwxr-xr-x.  3 root root system_u:object_r:var_t:s0       18 Sep 26 17:07 kerberos
drwxr-xr-x. 21 root root system_u:object_r:var_lib_t:s0   4096 Sep 26 17:10 lib
drwxr-xr-x.  2 root root system_u:object_r:var_t:s0        6 Nov  3 2024 local
lrwxrwxrwx.  1 root root system_u:object_r:var_lock_t:s0   11 Sep 26 17:06 lock -> ../run/lock
drwxr-xr-x.  7 root root system_u:object_r:var_log_t:s0   4096 Oct 10 12:33 log
lrwxrwxrwx.  1 root root system_u:object_r:mail_spool_t:s0  10 Nov  3 2024 mail -> spool/mail
drwxr-xr-x.  2 root root system_u:object_r:var_t:s0        6 Nov  3 2024 nis
drwxr-xr-x.  2 root root system_u:object_r:var_t:s0        6 Nov  3 2024 opt
drwxr-xr-x.  2 root root system_u:object_r:var_t:s0        6 Nov  3 2024 preserve
lrwxrwxrwx.  1 root root system_u:object_r:var_run_t:s0    6 Sep 26 17:06 run -> ../run
drwxr-xr-x.  6 root root system_u:object_r:var_spool_t:s0   56 Sep 26 17:07 spool
drwxrwxrwt.  8 root root system_u:object_r:tmp_t:s0      4096 Oct 10 13:23 tmp
drwxr-xr-x.  2 root root system_u:object_r:var_yp_t:s0      6 Nov  3 2024 yp
langelamg@localhost ~]$
```

Este paso garantiza que el nuevo volumen contiene una réplica exacta del contenido original de /var, lista para sustituirlo de forma segura.

Editamos el fichero `/etc/fstab` y se le añade esta línea:

```

/dev/mapper/vg_raid1-new_var_crypt /var xfs defaults 0 0
#
# /etc/fstab
# Created by anaconda on Fri Sep 26 15:06:20 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rl-root / xfs defaults 0 0
UUID=de692147-056f-4be2-8084-f164c9137d4f /boot xfs defaults 0 0
/dev/mapper/rl-swap none swap defaults 0 0

```

Se actualiza tabla de montajes con **sudo mount -a** y se comprueba el estado antes y después con **lsblk**:

```
[angelamg@localhost ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                8:0      0  10G  0 disk
├─sda1                            8:1      0   1G  0 part  /boot
├─sda2                            8:2      0   9G  0 part
│   └─r1-root                      253:0     0   8G  0 lvm    /
│       └─r1-swap                  253:1     0   1G  0 lvm    [SWAP]
sdb                                8:16     0   2G  0 disk
├─sdb1                            8:17     0   2G  0 part
│   └─md0                         9:0      0   2G  0 raid1
│       └─vg_raid1-new_var         253:2     0  1.8G  0 lvm
│           └─vg_raid1-new_var_crypt 253:3     0  1.8G  0 crypt /new_var
sdc                                8:32     0   2G  0 disk
├─sdc1                            8:33     0   2G  0 part
│   └─md0                         9:0      0   2G  0 raid1
│       └─vg_raid1-new_var         253:2     0  1.8G  0 lvm
│           └─vg_raid1-new_var_crypt 253:3     0  1.8G  0 crypt /new_var
sr0                                11:0     1  2.1G  0 rom

[angelamg@localhost ~]$ mount -a
mount: /var: must be superuser to use mount.
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[angelamg@localhost ~]$ sudo mount -a
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[angelamg@localhost ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                8:0      0  10G  0 disk
├─sda1                            8:1      0   1G  0 part  /boot
├─sda2                            8:2      0   9G  0 part
│   └─r1-root                      253:0     0   8G  0 lvm    /
│       └─r1-swap                  253:1     0   1G  0 lvm    [SWAP]
sdb                                8:16     0   2G  0 disk
├─sdb1                            8:17     0   2G  0 part
│   └─md0                         9:0      0   2G  0 raid1
│       └─vg_raid1-new_var         253:2     0  1.8G  0 lvm
│           └─vg_raid1-new_var_crypt 253:3     0  1.8G  0 crypt /var
│                                   /new_var
sdc                                8:32     0   2G  0 disk
├─sdc1                            8:33     0   2G  0 part
│   └─md0                         9:0      0   2G  0 raid1
│       └─vg_raid1-new_var         253:2     0  1.8G  0 lvm
│           └─vg_raid1-new_var_crypt 253:3     0  1.8G  0 crypt /var
│                                   /new_var
sr0                                11:0     1  2.1G  0 rom

[angelamg@localhost ~]$
```

La salida muestra que el volumen cifrado `vg_raid1-new_var_crypt` está montado en `/new_var` (montaje temporal) y después en `/var`, confirmando la correcta asociación entre el dispositivo cifrado y el nuevo punto de montaje.

Una vez comprobado que el nuevo volumen cifrado se monta en /var se procede a liberar el espacio del antiguo /var para dejar únicamente el nuevo sistema. Para ello:

1. Comentamos la línea de /etc/fstab que añadimos antes:

```
#/dev/mapper/vg_raid1-new_var_crypt /var xfs defaults 0 0
#
# /etc/fstab
# Created by anaconda on Fri Sep 26 15:06:20 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rl-root / xfs defaults 0 0
UUID=de692147-056f-4be2-8084-f164c9137d4f /boot xfs defaults 0 0
/dev/mapper/rl-swap none swap defaults 0 0
```

2. Montar antiguo /var y desmontar el volumen cifrado:

sudo mount -a

sudo umount /dev/mapper/vg_raid1-new_var_crypt

De esta forma el antiguo /var vuelve a estar disponible para su gestión.

3. Renombramos el antiguo /var

sudo mv /var /var_old

Se mantiene una copia de seguridad de su contenido por si fuera necesario restaurarlo más adelante.

4. Descomentamos la línea de /etc/fstab

Una vez movido el antiguo directorio, se vuelve a editar /etc/fstab para eliminar el # al inicio de la línea del volumen cifrado:

sudo vi /etc/fstab

5. Crear un nuevo /var vacío y montar el volumen cifrado:

sudo mkdir /var

sudo mount -a

```
[angelamg@localhost ~]$ sudo mkdir /var
[angelamg@localhost ~]$ mount -a
mount: /var: must be superuser to use mount.
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[angelamg@localhost ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                  8:0      0   10G  0 disk
├─sda1                              8:1      0    1G  0 part  /boot
├─sda2                              8:2      0    9G  0 part
│   └─r1-root                       253:0      0    8G  0 lvm    /
│   └─r1-swap                       253:1      0    1G  0 lvm    [SWAP]
sdb                                  8:16     0    2G  0 disk
├─sdb1                              8:17     0    2G  0 part
│   └─md0                           9:0      0    2G  0 raid1
│       └─vg_raid1-new_var           253:2      0   1.8G  0 lvm
│           └─vg_raid1-new_var_crypt 253:3      0   1.8G  0 crypt /new_var
sdc                                  8:32     0    2G  0 disk
├─sdc1                              8:33     0    2G  0 part
│   └─md0                           9:0      0    2G  0 raid1
│       └─vg_raid1-new_var           253:2      0   1.8G  0 lvm
│           └─vg_raid1-new_var_crypt 253:3      0   1.8G  0 crypt /new_var
sr0                                  11:0     1   2.1G  0 rom
```

El nuevo /var queda ahora montado directamente sobre el volumen cifrado /dev/mapper/vg_raid1-new_var_crypt.

6. Verificación:

Se comprueba que los contextos de seguridad del nuevo /var son correctos:

ls -laZ /var

```
[angelamg@localhost ~]$ sudo lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                 8:0      0   10G  0 disk
├─sda1                             8:1      0    1G  0 part  /boot
├─sda2                             8:2      0    9G  0 part 
│   └─r1-root                       253:0    0    8G  0 lvm    /
│   └─r1-swap                       253:1    0    1G  0 lvm    [SWAP]
sdb                                 8:16     0    2G  0 disk
├─sdb1                             8:17     0    2G  0 part 
│   └─md0                           9:0      0    2G  0 raid1 
│       └─vg_raid1-new_var           253:2    0    1.8G  0 lvm 
│           └─vg_raid1-new_var_crypt 253:3    0    1.8G  0 crypt  /var
sdc                                 8:32     0    2G  0 disk
├─sdc1                             8:33     0    2G  0 part 
│   └─md0                           9:0      0    2G  0 raid1 
│       └─vg_raid1-new_var           253:2    0    1.8G  0 lvm 
│           └─vg_raid1-new_var_crypt 253:3    0    1.8G  0 crypt  /var
sr0                                11:0     1    2.1G  0 rom
```

Si se detectan diferencias, pueden restaurarse con:

sudo restorecon -R /var