

Laboratorio RAID y LVM en Linux — Guía Paso a Paso

Autor: Francisco Javier Morales Ayala

Fecha: Septiembre 2025 | Distro objetivo: Debian/Ubuntu (comandos con sudo)

Resumen

Este manual práctico inicia en RAID 0 y avanza por RAID 1, RAID 5 y LVM. Cada sección incluye pasos, comandos copiables, verificaciones, simulación de fallos, persistencia de la configuración y limpieza final. Está pensado para ejecutarse en una máquina Linux de laboratorio SIN afectar discos reales (usaremos archivos con losetup).

0) Preparación del entorno de laboratorio

Crea una carpeta de trabajo y asegúrate de tener mdadm y lvm2.

```
mkdir -p ~/lab RAID && cd ~/lab RAID
```

```
# Instalar herramientas (Debian/Ubuntu)
sudo apt-get update
sudo apt-get install -y mdadm lvm2
```

```
# Ver versiones
mdadm --version
lvm version
```

Estructura recomendada:

```
mkdir -p {discos,scripts,logs,montajes}
pwd && tree -L 1 || ls -la
```

Nota: si tu distro es RHEL/CentOS/AlmaLinux usa dnf/yum; para initramfs usa dracut en lugar de update-initramfs.

1) Crear discos de prueba con archivos + losetup

Crearemos 'discos' como archivos regulares (200 MiB c/u) y los asociamos a dispositivos loop. Usaremos truncate (rápido) o dd (compatibilidad).

```
cd ~/lab_raid/discos
```

```
# Opción rápida (truncate):
```

```
truncate -s 200M d1.img
```

```
truncate -s 200M d2.img
```

```
truncate -s 200M d3.img
```

```
truncate -s 200M d4.img
```

```
# Alternativa (dd, más lento):
```

```
# dd if=/dev/zero of=d1.img bs=1M count=200 status=progress
```

```
# dd if=/dev/zero of=d2.img bs=1M count=200 status=progress
```

```
# dd if=/dev/zero of=d3.img bs=1M count=200 status=progress
```

```
# dd if=/dev/zero of=d4.img bs=1M count=200 status=progress
```

```
# Asociar a loop
```

```
sudo losetup -fP d1.img
```

```
sudo losetup -fP d2.img
```

```
sudo losetup -fP d3.img
```

```
sudo losetup -fP d4.img
```

```
# Ver loops asignados
```

```
sudo losetup -a | sort
```

Consejo: Anota qué /dev/loopX corresponde a cada imagen (d1..d4).

2) RAID 0 — Rendimiento sin redundancia

Objetivo: crear /dev/md0 con 2 discos (d1, d2), formatear, montar y probar rendimiento.

```
# Crear RAID 0 con dos loops (ajusta los nombres según tu salida real)
sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/loop0 /dev/loop1

# Ver estado
cat /proc/mdstat
sudo mdadm --detail /dev/md0

# Crear sistema de archivos y montar
sudo mkfs.ext4 -F /dev/md0
sudo mkdir -p /mnt/raid0
sudo mount /dev/md0 /mnt/raid0
df -h | grep md0

# Persistir en /etc/fstab (opcional)
# UUID=$(blkid -s UUID -o value /dev/md0); echo "UUID=${UUID} /mnt/raid0 ext4 defaults 0 0" |
etc/fstab
# sudo systemctl daemon-reload
```

Prueba rápida de I/O (no destructiva sobre /mnt/raid0):

```
# Escritura secuencial (1 GiB en bloques de 1M) – cambia count si tu raid es pequeño
cd /mnt/raid0
sudo dd if=/dev/zero of=prueba.bin bs=1M count=1024 status=progress oflag=direct
sudo sync
# Lectura
sudo dd if=prueba.bin of=/dev/null bs=1M status=progress iflag=direct
sudo rm -f prueba.bin
```

Conclusión: RAID 0 acelera, pero la pérdida de un disco implica pérdida total de datos.

3) RAID 1 — Espejo con tolerancia a 1 fallo

Objetivo: crear /dev/md1 con d3 y d4, simular fallo y reconstrucción.

```
# Crear RAID 1
sudo mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2 /dev/loop2 /dev/loop3

# Ver progreso/revisión
watch -n 1 cat /proc/mdstat
# (Ctrl+C para salir)

sudo mdadm --detail /dev/md1

# Formatear y montar
sudo mkfs.ext4 -F /dev/md1
sudo mkdir -p /mnt/raid1
sudo mount /dev/md1 /mnt/raid1
df -h | grep md1
```

Simular un fallo en RAID 1 y reconstruir:

```
# Marcar un miembro FALLIDO
sudo mdadm --fail /dev/md1 /dev/loop2
sudo mdadm --detail /dev/md1
cat /proc/mdstat

# Remover el miembro fallado
sudo mdadm --remove /dev/md1 /dev/loop2

# Reemplazarlo por el mismo loop (simulación) o por otro disponible
sudo mdadm --add /dev/md1 /dev/loop2

# Monitorear la RECONSTRUCCIÓN
watch -n 1 cat /proc/mdstat
```

Nota: en hardware real, reemplaza el disco físico defectuoso y luego 'mdadm --add' con el nuevo.

4) RAID 5 — Rendimiento + Paridad (tolera 1 fallo)

Objetivo: crear /dev/md5 con 3 discos, simular fallo y reconstruir. Opcional: bitmap interno.

```
# Liberar loops si los usaste antes y necesitas 3 nuevos (ejemplo con dl..d3)
# sudo umount /mnt/raid0 || true
# sudo mdadm --stop /dev/md0 || true
```

```
# Crear RAID 5 (usa TUS loops reales)
sudo mdadm --create --verbose /dev/md5 --level=5 --raid-devices=3 /dev/loop0 /dev/loop1 /dev/loop2
bitmap=internal
```

```
# Ver estado
watch -n 1 cat /proc/mdstat
sudo mdadm --detail /dev/md5
```

```
# Preparar FS y montaje
sudo mkfs.ext4 -F /dev/md5
sudo mkdir -p /mnt/raid5
sudo mount /dev/md5 /mnt/raid5
df -h | grep md5
```

Simular fallo en RAID 5:

```
# Marcar fallido un miembro (ajusta el loop)
sudo mdadm --fail /dev/md5 /dev/loop1
sudo mdadm --detail /dev/md5
cat /proc/mdstat
```

```
# Removerlo y añadir un 'nuevo'
sudo mdadm --remove /dev/md5 /dev/loop1
sudo mdadm --add /dev/md5 /dev/loop1
```

```
# Monitorear la reconstrucción
watch -n 1 cat /proc/mdstat
```

Crecimiento de RAID 5 de 3 → 4 discos (opcional):

```
# Añadir un cuarto disco
sudo mdadm --add /dev/md5 /dev/loop3
```

```
# CRECER el arreglo
sudo mdadm --grow /dev/md5 --raid-devices=4
```

```
# Esperar a que finalice el reshape
watch -n 2 cat /proc/mdstat
```

```
sudo mdadm --detail /dev/md5
```

5) Persistir configuración de mdadm y arranque

Registra el arreglo en mdadm.conf y actualiza initramfs (Debian/Ubuntu).

```
# Guardar definición en mdadm.conf
sudo bash -c 'mdadm --detail --scan >> /etc/mdadm/mdadm.conf'
```

```
# Regenerar initramfs (Debian/Ubuntu)
sudo update-initramfs -u
```

```
# Verifica entradas
grep md /etc/mdadm/mdadm.conf
```

En RHEL/AlmaLinux usa dracut:

```
# RHEL-like
sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
sudo dracut -f
```

6) LVM sobre RAID (PV → VG → LV)

Crearemos un PV sobre /dev/md5, un VG y uno o más LV; montaremos y luego ampliaremos.

```
# Crear PV
sudo pvcreate /dev/md5
sudo pvs

# Crear VG
sudo vgcreate vg_lab /dev/md5
sudo vgs

# Crear LV de 100M
sudo lvcreate -L 100M -n lv_datos vg_lab
sudo lvs

# Formatear y montar
sudo mkfs.ext4 -F /dev/vg_lab/lv_datos
sudo mkdir -p /mnt/datos
sudo mount /dev/vg_lab/lv_datos /mnt/datos
df -h | grep vg_lab
```

Extender el LV (y FS) en caliente:

```
# Ampliar 150M adicionales
sudo lvextend -L +150M -r /dev/vg_lab/lv_datos
# (-r redimensiona el sistema de archivos automáticamente)
df -h | grep vg_lab
```

Creación de un snapshot (lectura/escritura) y restauración:

```
# Snapshot de lv_datos (tamaño 100M de reserva)
sudo lvcreate -s -L 100M -n snap_datos /dev/vg_lab/lv_datos
sudo lvs

# Montar snapshot para pruebas
sudo mkdir -p /mnt/snap
sudo mount /dev/vg_lab/snap_datos /mnt/snap

# (Realiza validaciones, luego desmonta y elimina)
sudo umount /mnt/snap
sudo lvremove -y /dev/vg_lab/snap_datos
```


7) PVResize tras crecer el RAID (opcional)

Si incrementaste /dev/md5 (p.ej., añadiendo discos y mdadm --grow), ejecuta pvresize para que LVM vea el nuevo tamaño.

```
# Redetectar tamaño del PV
sudo pvresize /dev/md5
sudo pvs
sudo vgs
```

Luego puedes extender LVs o crear nuevos:

```
# Extender el LV existente 200M
sudo lvextend -L +200M -r /dev/vg_lab/lv_datos
lvs -a -o +devices
```

8) Pruebas y benchmarking básico

Usa dd para pruebas secuenciales. Si tienes fio instalado, úsalo para cargas aleatorias.

```
# Escribir 2 GiB en bloques de 4M (ajusta al tamaño de tu arreglo)
cd /mnt/raid5
sudo dd if=/dev/zero of=test.bin bs=4M count=512 oflag=direct status=progress
sudo sync
```

```
# Leer
sudo dd if=test.bin of=/dev/null bs=4M iflag=direct status=progress
sudo rm -f test.bin
```

```
# (Opcional) con fio si está instalado:
# sudo apt-get install -y fio
# fio --name=seqwrite --filename=/mnt/raid5/fio.bin --rw=write --bs=1M --size=1G --iodepth=32
# fio --name=seqread --filename=/mnt/raid5/fio.bin --rw=read --bs=1M --size=1G --iodepth=32
# rm -f /mnt/raid5/fio.bin
```

9) Verificación, monitoreo y salud del arreglo

Comandos clave para revisar estado, eventos y logs.

```
# Estado general
cat /proc/mdstat

sudo mdadm --detail /dev/md0 2>/dev/null || true
sudo mdadm --detail /dev/md1 2>/dev/null || true
sudo mdadm --detail /dev/md5 2>/dev/null || true

# Monitoreo en tiempo real del journal (Debian/Ubuntu / systemd)
sudo journalctl -f | grep mdadm

# Forzar verificación de consistencia (scrub) en RAID 5
echo check | sudo tee /sys/block/md5/md/sync_action
watch -n 2 'cat /sys/block/md5/md/sync_action; cat /proc/mdstat'
```

10) Limpieza completa del laboratorio

Desmonta, elimina LVs, VGs, PVs, detén arreglos y libera loop devices.

```
# Desmontar
sudo umount -R /mnt/raid0 2>/dev/null || true
sudo umount -R /mnt/raid1 2>/dev/null || true
sudo umount -R /mnt/raid5 2>/dev/null || true
sudo umount -R /mnt/datos 2>/dev/null || true

# LVM
sudo lvremove -y vg_lab 2>/dev/null || true
sudo vgremove -y vg_lab 2>/dev/null || true
sudo pvremove -y /dev/md5 2>/dev/null || true

# Detener arreglos md
sudo mdadm --stop /dev/md0 2>/dev/null || true
sudo mdadm --stop /dev/md1 2>/dev/null || true
sudo mdadm --stop /dev/md5 2>/dev/null || true

# Desasociar loops
for lp in $(losetup -a | cut -d: -f1); do
    sudo losetup -d "$lp" || true
done

# Borrar imágenes (;cuidado! estamos en ~/lab_raid/discos)
cd ~/lab_raid/discos && rm -f *.img
```

Checklist de validación por etapa

- Creación de imágenes y loops: losetup -a muestra d1..d4 asociados.
- RAID 0 (/dev/md0) activo: /proc/mdstat y mdadm --detail OK; montaje en /mnt/raid0.
- RAID 1 (/dev/md1) activo: simular fallo y reconstrucción completada.
- RAID 5 (/dev/md5) activo: simular fallo y reconstrucción; opcional grow a 4 discos.
- LVM: pvcreate/vgcreate/lvcreate OK; lvextend -r aplicado y visible en df -h.
- Persistencia: mdadm.conf actualizado y (opcional) initramfs regenerado.
- Limpieza: sin md activos, sin loops asociados, sin montajes residuales.

Errores comunes y cómo resolverlos

1) 'device or resource busy' al desmontar o detener mdadm:

```
# Identifica procesos usando el punto de montaje
sudo lsof +f -- /mnt/raid5
sudo fuser -vm /mnt/raid5
# Cierra procesos y reintenta umount/mdadm --stop
```

2) Los loops no aparecen:

```
# Reasocia explícitamente
sudo losetup -D                # libera todo (con cuidado)
sudo losetup -fP d1.img
sudo losetup -fP d2.img
sudo losetup -fP d3.img
sudo losetup -fP d4.img
losetup -a
```

3) UUID / fstab incorrecto:

```
# Relee el UUID y corrige fstab
blkid /dev/md5
sudoedit /etc/fstab
```

4) Reduce de LV/FS:

Reducir ext4 requiere desmontar y usar e2fsck + resize2fs. No lo hagas en caliente. Ejemplo:

```
sudo umount /mnt/datos
sudo e2fsck -f /dev/vg_lab/lv_datos
sudo resize2fs /dev/vg_lab/lv_datos 150M
sudo lvreduce -L 150M /dev/vg_lab/lv_datos
sudo mount /dev/vg_lab/lv_datos /mnt/datos
```

Apéndice A — Salidas esperadas (ejemplos)

Salida típica de /proc/mdstat durante reconstrucción:

```
Personalities : [raid6] [raid5] [raid4] [raid1] [raid0]
md5 : active raid5 loop2[2] loop1[1] loop0[0]
      409536 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [U_U]
      [>.....] recovery = 3.1% (6400/204768) finish=1.2min speed=28000K/sec
```

Salida de mdadm --detail /dev/md5:

```
/dev/md5:
    Version : 1.2
  Creation Time : Thu Sep 11 12:34:56 2025
    Raid Level : raid5
    Array Size : 409536 (400.00 MiB 419.43 MB)
  Used Dev Size : 204768 (200.00 MiB 209.72 MB)
    Raid Devices : 3
  Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Thu Sep 11 12:36:01 2025
      State : clean, degraded
Active Devices : 2
Working Devices : 2
Failed Devices : 1
Spare Devices : 0
    Name : lab:5
   UUID : 11111111:2222:3333:4444:555555555555
  Events : 25
```

Number	Major	Minor	RaidDevice	State	
0	7	0	0	active sync	/dev/loop0
1	7	1	1	faulty	/dev/loop1
2	7	2	2	active sync	/dev/loop2

Cierre

Con esta guía paso a paso puedes impartir un laboratorio completo: desde la creación de arreglos RAID con dispositivos loop, hasta su integración con LVM, pruebas de fallo y crecimiento en línea, todo validado y con limpieza segura al final.