

## ¿Qué es LVM? O cómo hacer que tus discos sean de chicle

**LVM** significa *Logical Volume Manager* (Gestor de Volúmenes Lógicos). Si RAID era una forma de "pegar" discos para que parecieran uno, LVM es como tener una gran piscina de almacenamiento (**Volume Group**) de la que puedes ir sacando cubos de agua (**Logical Volumes**) del tamaño que te dé la gana y cuando te dé la gana.

La gracia de LVM es que te independiza del hardware físico. Puedes crear, redimensionar, mover y hacer instantáneas de tus "particiones" sin tener que sudar la gota gorda.

### Los Conceptos Clave (El Trío Calavera)

1. **Physical Volume (PV) - El Ladrillo:** Es el componente básico. Coges un disco duro entero (o una partición) y lo marcas como "apto para LVM". Es como ponerle una etiqueta que dice: *"Estoy disponible para la piscina común"*. Nuestros `/dev/sdb`, `/dev/sdc`, etc., se convertirán en PVs.
2. **Volume Group (VG) - La Piscina:** Es la suma de todos tus Physical Volumes. Si juntas tres discos (PVs) de 1TB, creas una gran piscina (VG) de 3TB. Es tu reserva total de espacio.
3. **Logical Volume (LV) - El Cubo:** Del gran VG (la piscina), puedes crear "particiones" lógicas o volúmenes lógicos del tamaño que necesites. ¿Necesitas 20GB para `/home`? ¡Hecho! ¿Que mañana necesitas 30GB? ¡Lo agrandas en caliente! Estos LVs son los que formateas con un sistema de ficheros (`ext4`, `xfs`...) y montas en tu sistema. Son los que realmente usas, como `/dev/sda1` o `/dev/sda2` en un sistema tradicional.

### La diferencia con Thin LVM (El Mago del "Ya te lo daré")

El LVM normal es "thick provisioning". Si pides un cubo (LV) de 100GB, se reservan esos 100GB de la piscina (VG) al momento, los uses o no.

**Thin LVM** es más pillo. Es "thin provisioning" o aprovisionamiento fino. Tú creas una "Thin Pool" (una sub-piscina) dentro de tu VG. Luego, desde esa pool, creas volúmenes "thin". Puedes crear, por ejemplo, 10 volúmenes de 100GB cada uno (1000GB en total) aunque tu pool solo tenga 200GB.

¿El truco? Los volúmenes thin solo ocupan el espacio que realmente están usando. Van pidiendo bloques a la pool a medida que escriben datos. Es fantástico para máquinas virtuales o contenedores, donde creas muchos sistemas de ficheros pero no todos se llenan a la vez. ¡Pero ojo! Hay que vigilar que la pool no se llene de verdad, o tendrás un problema más gordo que el de la parte contratante de la primera parte.

---

## Ejercicio Práctico: ¡Más madera! ¡Es la guerra (de los discos)!

Vamos a usar los mismos 5 discos de antes (3 SATA: `sdb`, `sdc`, `sdd` y 2 NVME: `nvme0n1`, `nvme0n2`) para montar un LVM. Usaremos un disco adicional (`sdf`) para el paso de ampliación.

### Paso 1: Preparando los Ladrillos (Physical Volumes)

Primero, marcamos todos los discos que vamos a usar como PVs. *Asegúrate de que no tienen datos importantes. ¡Este paso borra lo que haya!*

```
# ¡Conviértete en root o usa sudo para todo!  
sudo pvcreate /dev/sdb /dev/sdc /dev/sdd /dev/nvme0n1 /dev/nvme0n2
```

Puedes comprobar que se han creado correctamente con `sudo pvs` o `sudo pvdisplay`.

## **Paso 2: Construyendo la Piscina (Volume Group)**

Ahora juntamos todos esos ladrillos en una única y gran piscina. La llamaremos `vg_maestro`.

```
sudo vgcreate vg_maestro /dev/sdb /dev/sdc /dev/sdd /dev/nvme0n1 /dev/nvme0n2
```

Verifica la creación con `sudo vgs` o `sudo vgdisplay vg_maestro`. Verás todo el espacio sumado.

## **Paso 3: Sirviendo los Cubos (Logical Volumes) con sus Sistemas de Ficheros**

Vamos a crear tres LVs desde nuestra piscina `vg_maestro`. Uno para datos, otro para webs y otro para... bueno, para lo que sea. Les daremos tamaños diferentes y, para divertirnos, sistemas de ficheros distintos.

```
# 1. LV de 20G para datos, con el clásico ext4  
sudo lvcreate -n lv_datos -L 20G vg_maestro  
sudo mkfs.ext4 /dev/vg_maestro/lv_datos  
  
# 2. LV de 15G para webs, con el robusto xfs  
sudo lvcreate -n lv_webs -L 15G vg_maestro  
sudo mkfs.xfs /dev/vg_maestro/lv_webs  
  
# 3. LV de 10G para backups, con el moderno btrfs  
sudo lvcreate -n lv_backups -L 10G vg_maestro  
sudo mkfs.btrfs /dev/vg_maestro/lv_backups
```

Ahora, si quieres, puedes montarlos para usarlos:

```
sudo mkdir /mnt/datos /mnt/webs /mnt/backups  
sudo mount /dev/vg_maestro/lv_datos /mnt/datos  
sudo mount /dev/vg_maestro/lv_webs /mnt/webs  
sudo mount /dev/vg_maestro/lv_backups /mnt/backups
```

---

## **Ampliando el chiringuito: ¡Que entre más gente!**

Aquí viene la magia de LVM. Imagina que te quedas sin espacio. ¡Pues compramos más discos!

## **Paso 4: Agregar otro disco al Volumen (VG)**

Supongamos que has añadido un nuevo disco a tu máquina virtual: `/dev/sdf`.

### **1. Conviértelo en un PV:**

```
sudo pvcreate /dev/sdf
```

## 2. Añádalo a la piscina existente:

```
sudo vgextend vg_maestro /dev/sdf
```

¡Listo! Si ahora ejecutas `sudo vgdisplay vg_maestro`, verás que el tamaño total de la piscina ha aumentado. "Disculpen si les llamo caballeros, pero es que no les conozco muy bien".

## Paso 5: Aumentar el espacio de un Sistema de Ficheros existente

El departamento de datos te llora porque sus 20GB no son suficientes. ¡Démosles 10GB más!

### 1. Extiende el Logical Volume:

```
# Añadimos 10GB más al LV 'lv_datos'
sudo lvextend -L +10G /dev/vg_maestro/lv_datos
```

### 2. Redimensiona el sistema de ficheros: El LV es más grande, pero el sistema de ficheros que vive dentro aún no se ha enterado. Hay que "estirarlo".

```
# Para ext4 (el de lv_datos)
sudo resize2fs /dev/vg_maestro/lv_datos
```

*Si fuera XFS, el comando sería `sudo xfs_growfs /mnt/datos` (se hace sobre el punto de montaje). Y sí, ¡ambos se pueden hacer "en caliente", sin desmontar el volumen!*

---

## El Reto Final: Engordar el Sistema Raíz (/)

Esta es la operación más delicada. Meterle mano a la raíz del sistema es como intentar operar a un paciente mientras corre una maratón. Si tu sistema de ficheros raíz / está en un LV (lo cual es una práctica excelente), el procedimiento es similar al anterior, pero con una precaución clave.

**¡ADVERTENCIA!** Agrandar la partición raíz es seguro, pero reducirla es buscarse un disgusto. Procede con cuidado.

**Asumimos que tu raíz (/) está en un LV, por ejemplo `/dev/mapper/vg_sistema-lv_root`, y que tienes espacio libre en el `vg_sistema`.**

### 1. Extender el Logical Volume raíz: Supongamos que quieres añadirle 50GB.

```
sudo lvextend -L +50G /dev/mapper/vg_sistema-lv_root
```

### 2. Extender el sistema de ficheros raíz: Aquí el comando es el mismo que antes, dependiendo del sistema de ficheros. Como la mayoría de las raíces son `ext4`, sería:

```
sudo resize2fs /dev/mapper/vg_sistema-lv_root
```

**¿Se puede hacer en caliente?** Sí, `ext4`, `xfs` y `btrfs` soportan el redimensionamiento en caliente (online). Por tanto, no necesitas reiniciar ni usar un Live CD para *agrandar* el sistema raíz.

Simplemente ejecuta los dos comandos y, como por arte de magia, tendrás más espacio. Puedes verificarlo con `df -h /`.

Y con esto y un bizcocho... ¡ya eres un maestro del LVM! Has aprendido a ser más flexible que un contorsionista en una caja de cerillas. ¡Ahora ve y gestiona esos volúmenes!