

## Prácticas COS – Curso 2022-23 – Sesión 4

### Logical Volume Manager (LVM)

LVM es un gestor de volúmenes lógicos. Con LVM es posible redimensionar los volúmenes lógicos una vez creados. Es una forma más flexible de asignar espacio que el particionado tradicional, pues el tamaño de las particiones estándar no se puede modificar.

Entre otras podemos citar las siguientes características: redimensionado de grupos lógicos, redimensionado de volúmenes lógicos, tomar instantáneas/fotos (snapshots) de los volúmenes lógicos e implementar distintos tipos de RAID (raid 0, 1, 5, 6).

En la arquitectura de LVM podemos distinguir tres niveles (ver figura 1):

- **Volúmenes físicos (PV):** Es el soporte final donde se guardan los datos, y pueden ser varios discos completos o particiones de uno o varios discos.
- **Volúmenes de grupo (GV):** Es la agrupación de los volúmenes físicos en un volumen que permite ver todos los discos y particiones como uno solo dispositivo.
- **Volúmenes lógicos (LV):** Son los volúmenes donde irán nuestros sistemas de archivos y por tanto los puntos de montaje (/ , /home, swap). En este nivel es donde se hace el formato del sistema de archivos a usar, y es donde los datos se distribuirán a nivel lógico, similar a las particiones que acostumbramos a usar.

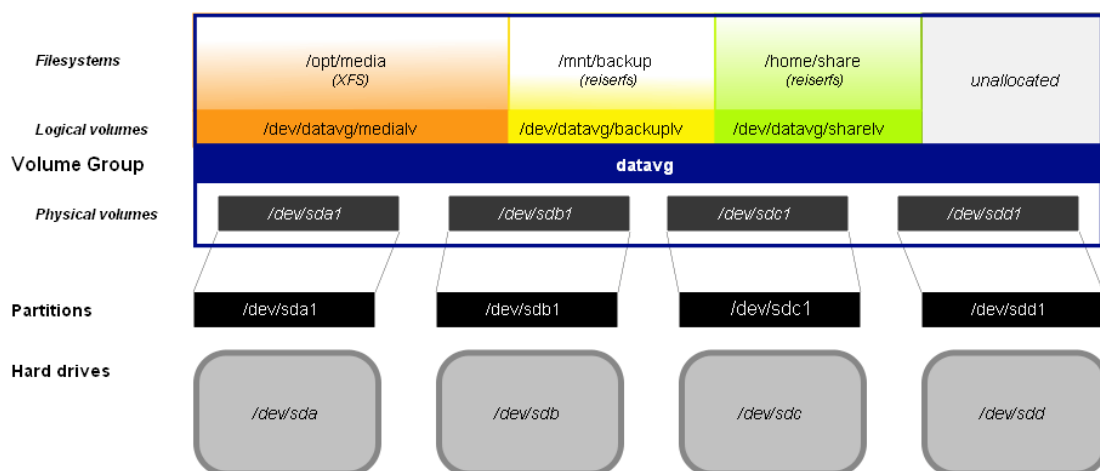


Figura 1. Arquitectura LVM.

Para poder usar LVM primero se deben asignar volúmenes físicos, luego asignar volúmenes de grupo y por último volúmenes lógicos. Los volúmenes físicos (PV) y los grupos de volúmenes (VG) se dividen en unidades llamadas chunks de datos, que también se conoce como PE (Physical extents). Además, un volumen lógico se divide también en LE (logical extents), cuyo tamaño es el mismo para todos los LV del VG.

Trabajaremos sobre la máquina "server1" del clúster de máquinas virtuales que hemos configurado en las sesiones de seminario.

En primer lugar, añadiremos varios discos físicos a la máquina. Mediante la interfaz gráfica de VirtualBox, creamos tres discos de 200 MB: lvm1, lvm2, y lvm3 conectados al controlador SATA de la máquina server1 (discos **vdi** reservados dinámicamente).

Iniciamos sesión como **root** en la máquina virtual **server1**. Durante la instalación de Centos 7 se instala automáticamente el paquete lvm2. Si por algún motivo no se hubiese instalado, ejecutaríamos:

```
# yum -y install lvm2
```

## 1. Particionado de los discos.

Comprobamos que los discos están correctamente conectados con la orden:

```
# lsblk
```

Otra opción es utilizar:

```
# fdisk -l
```

Veremos que los nuevos discos son /dev/sdb, /dev/sdc, y /dev/sdd.

Con la orden **cfdisk** (también puede emplearse **fdisk**), creamos dos particiones (primarias) aproximadamente el mismo tamaño en cada disco del tipo "**8e**" (Linux LVM). Obtendremos las particiones /dev/sdb1, /dev/sdb2, /dev/sdc1, /dev/sdc2, /dev/sdd1 y /dev/sdd2.

En lugar de realizar el particionado de cada disco uno por uno, podríamos clonar la tabla de particiones creada en uno de ellos mediante la orden **sfdisk** (tal como se hizo en el seminario 2).

## 2. Creación de los dispositivos físicos.

Seguidamente, crearemos los dispositivos físicos:

```
# pvcreate /dev/sdb1
# pvcreate /dev/sdb2
...
```

Podemos comprobar los dispositivos físicos disponibles con la orden:

```
# pvscan
```

Y sus detalles con la orden:

```
# pvdisplay
```

## 3. Creación del grupo de volúmenes.

Ahora crearemos un grupo de volumen con los dos primeros discos (cuatro particiones). Hay que elegir el nombre del grupo de volúmenes. Por ejemplo, si nuestro grupo se llama **mvg**, utilizaremos la orden:

```
# vgcreate mvg /dev/sdb1 /dev/sdb2 /dev/sdc1 /dev/sdc2
```

Podemos comprobar los grupos de volúmenes disponibles mediante la orden:

```
# vgscan
```

Y sus detalles mediante la orden:

```
# vgdisplay
```

Se puede desactivar un grupo de volúmenes mediante la orden:

```
# vgchange -a n mvg
```

Y para activarlo (por ejemplo, tras el inicio del sistema):

```
# vgchange -a y mvg
```

Si deseamos eliminar completamente un grupo de volúmenes, primero lo desactivaríamos y después utilizaríamos la orden:

```
# vgremove mvg
```

## 4. Creación de los dispositivos lógicos.

Una vez tenemos creado el grupo de volúmenes, podemos crear dispositivos lógicos sobre éste. En nuestro caso, crearemos dos volúmenes lógicos, que llamaremos lv0 y lv1, de tamaño 120MB y 72 MB, respectivamente. Para ello, utilizaremos la orden:

```
# lvcreate -L 120M -n lv0 mvg  
# lvcreate -L 72M -n lv1 mvg
```

Se puede especificar el tamaño en Megabytes, Gigabytes, Terabytes, Petabytes, o Exabytes, usando M,G,T,P, o E respectivamente.

Podemos ver los volúmenes lógicos creados mediante la orden:

```
# lvscan
```

Y sus detalles mediante la orden:

```
# lvdisplay
```

Ahora podemos formatear (si no se especifica el tipo de sistema de ficheros, por defecto se usa ext4) y montar los dispositivos lógicos, como de costumbre. Por ejemplo, si queremos montar el dispositivo lv0 en /mnt/usuarios y el lv1 en /mnt/db:

```
# mkfs /dev/mvg/lv0  
# mkfs /dev/mvg/lv1  
# mkdir /mnt/usuarios  
# mkdir /mnt/db  
# mount /dev/mvg/lv0 /mnt/usuarios  
# mount /dev/mvg/lv1 /mnt/db  
# df -h
```

Ahora podemos comenzar a utilizar los dispositivos. Copiaremos algunos ficheros (o creamos algunos con contenido aleatorio):

```
# dd if=/dev/urandom of=/mnt/usuarios/diezmegas count=10 bs=1M  
# dd if=/dev/urandom of=/mnt/db/cincomegas count=5 bs=1M  
# ls -la /mnt/usuarios  
# ls -la /mnt/db
```

Podemos borrar dispositivos lógicos, mediante la orden lvremove. El dispositivo debe desmontarse

antes. Hay que tener en cuenta que se perderá toda la información que contenga el dispositivo. Por ejemplo, vamos a crear un dispositivo para borrarlo a continuación:

```
# lvcreate -L 10M -n lv2 mvg
# lvdisplay
# mkfs /dev/mvg/lv2
# mkdir /mnt/borrar
# mount /dev/mvg/lv2 /mnt/borrar
# dd if=/dev/urandom of=/mnt/borrar/cincomegas count=5 bs=1M
# ls -la /mnt/borrar
# umount /mnt/borrar
# lvremove /dev/mvg/lv2
```

## 5. Cambio de tamaño de los dispositivos lógicos.

LVM permite cambiar el tamaño de un volumen lógico. Esto dota de flexibilidad al sistema de almacenamiento, pues adapta las particiones a las necesidades reales del sistema.

En nuestro caso, vamos a aumentar el tamaño del volumen lógico lv0 en 32MB y reducir el del lv1 en 12MB. Una vez realizado el cambio de tamaño, hay que ajustar también el sistema de archivos.

Las órdenes correspondientes a lv0 serían:

```
# lvdisplay
# df -h
# umount /dev/mvg/lv0
# lvextend -L +32M /dev/mvg/lv0
# e2fsck -f /dev/mvg/lv0
# resize2fs /dev/mvg/lv0
# mount /dev/mvg/lv0 /mnt/usuarios
# lvdisplay
# df -h
# ls -l /mnt/usuarios
```

se han añadido las órdenes pertinentes para conocer el tamaño del volumen antes y después de la modificación. Podemos comprobar que los archivos continúan almacenados en el dispositivo.

Las órdenes correspondientes a lv1 serían:

```
# lvdisplay
# df -h
# umount /dev/mvg/lv1
# e2fsck -f /dev/mvg/lv1
# resize2fs /dev/mvg/lv1 60M
# lvreduce -L -12M /dev/mvg/lv1
# mount /dev/mvg/lv1 /mnt/db
# lvdisplay
# df -h
# ls -l /mnt/db
```

Nótese que se debe ajustar el tamaño del sistema de ficheros antes de realizar la reducción del tamaño del volumen lógico.

## 6. Adición/eliminación de volúmenes físicos.

LVM permite añadir nuevos dispositivos físicos a un grupo de volúmenes, disponiendo así de más espacio de almacenamiento para los volúmenes lógicos. En nuestro caso, vamos a añadir las dos particiones del tercer disco al grupo de volúmenes que tenemos creado:

```
# vgdisplay
# vgextend mvg /dev/sdd1 /dev/sdd2
# vgdisplay
```

Obsérvese que se dispone de más espacio en el grupo de volúmenes tras la adición del nuevo disco.

LVM también permite eliminar un dispositivo físico de un grupo de volúmenes. Si el disco contiene datos, previamente deberían moverse a otro dispositivo del grupo de volúmenes. En nuestro caso, vamos a eliminar la partición `/dev/sdd2` del grupo de volúmenes. Comprobamos que no está utilizado por ningún volumen lógico (no tiene PE asignados):

```
# pvdisplay /dev/sdd2
```

Y procedemos a su eliminación:

```
# vgreduce mvg /dev/sdd2
# vgdisplay
```

## 7. Migración de datos entre volúmenes físicos.

LVM permite mover datos entre volúmenes físicos. Supongamos que necesitamos sustituir un dispositivo por otro nuevo, y que el dispositivo a sustituir tiene datos almacenados. Con LVM, podemos:

- Si hay espacio disponible en el grupo de volúmenes, simplemente movemos la información desde el dispositivo físico a sustituir hacia otros dispositivos del grupo de volúmenes. Seguidamente, eliminaremos el dispositivo físico, y finalmente añadiremos el nuevo.

En nuestro ejemplo, veamos la secuencia de órdenes adecuada si deseamos sustituir el dispositivo `/dev/sdb1`, que tiene datos almacenados. Podemos comprobar que no está vacío mediante:

```
# pvdisplay /dev/sdb1
```

Movemos sus datos a otros dispositivos del grupo de volúmenes:

```
# pvmove /dev/sdb1
```

Después comprobamos que ya no tiene datos almacenados:

```
# pvdisplay /dev/sdb1
```

Y eliminamos el disco:

```
# vgreduce mvg /dev/sdb1
```

Se puede comprobar que los sistemas de archivos están intactos:

```
# ls -la /mnt/usuarios
# ls -la /mnt/db
```

Ahora añadiríamos el nuevo dispositivo. Tras crear el PV, lo añadiríamos mediante la orden `vgextend` (en este ejemplo hemos supuesto que el nuevo dispositivo es nuevamente `/dev/sdb1`):

```
# vgextend mvg /dev/sdb1
```

- Si no hay espacio disponible en el grupo, añadiremos en primer lugar el nuevo dispositivo físico, seguidamente moveremos los datos desde el dispositivo a eliminar hacia el nuevo, y finalmente lo eliminaremos. En nuestro ejemplo, supondremos que deseamos mover los datos desde `/dev/sdc1` hacia un nuevo dispositivo ubicado en `/dev/sdd2`. La secuencia de órdenes comenzará comprobando si tiene datos almacenados:

```
# pvdisplay /dev/sdc1
```

para añadir el nuevo dispositivo:

```
# vgextend mvg /dev/sdd2
```

para mover los datos:

```
# pvmove /dev/sdc1 /dev/sdd2
```

para comprobar que los datos que tenía almacenados `/dev/sdc1` se han ubicado en `/dev/sdd2`:

```
# pvdisplay /dev/sdc1
```

```
# pvdisplay /dev/sdd2
```

Se puede comprobar que los sistemas de archivos están intactos:

```
# ls -la /mnt/usuarios
```

```
# ls -la /mnt/db
```

## 8. Snapshots.

LVM permite realizar copias instantáneas de volúmenes (*snapshots*) muy fácilmente. Esto puede resultar útil, por ejemplo, para realizar una copia de seguridad de nuestros datos en un instante dado. Los datos de la copia de seguridad serán fiel reflejo de los datos existentes en ese momento, aun cuando se continúe modificando el sistema de archivos. En nuestro caso, supongamos que deseamos sacar una copia instantánea del volumen `lv0`. Sabemos que tiene usados unos 10MB, por lo que a la copia le daremos un valor ligeramente superior para asegurarnos que hay espacio suficiente. En la versión 2 de LVM es posible realizar escrituras.

```
# lvcreate -L 20M -s -n backup /dev/mvg/lv0
```

```
# lvdisplay
```

El nuevo volumen se podría montar y utilizar, como de costumbre:

```
# mkdir /mnt/backup
```

```
# mount /dev/mvg/backup /mnt/backup
```

```
# ls -la /mnt/backup
```

Para destruir el snapshot, cuando ya no se necesite:

```
# umount /mnt/backup
```

```
# lvremove /dev/mvg/backup
```