

Introducción al LVM

1. Principio

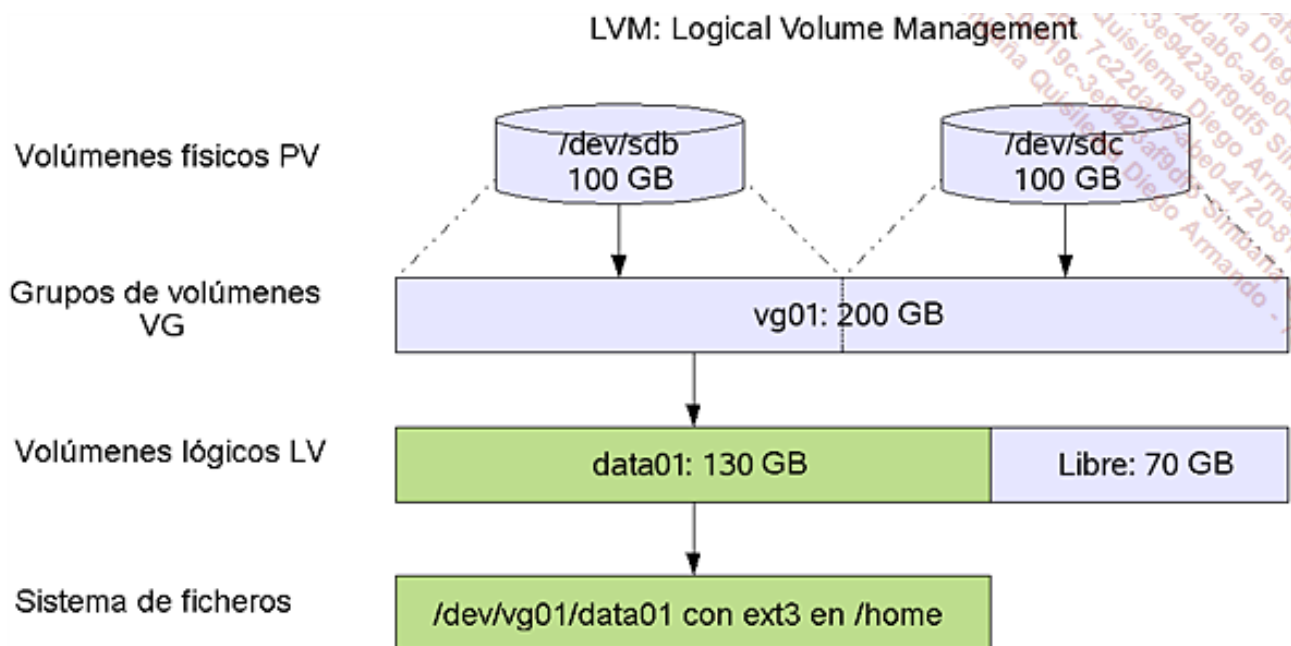
El **Logical Volume Manager** es un sistema de gestión muy perfeccionado de los soportes de almacenamiento. El objetivo es superar, incluso transcender, la gestión física de los discos y su organización lógica básica (las particiones) para extender la capacidad global de los soportes, usando una gestión completamente lógica de ella.

Un LVM permite, del mismo modo que el RAID 0 por ejemplo, crear espacios de datos lógicos en varios discos. Permite también hacer mirroring, como el RAID 1. Pero la comparación no va más allá. El RAID se limita a crear una «partición» en un espacio de almacenamiento definido por el propio RAID (por ejemplo, una partición de 100 GB en un RAID 0 de dos discos de 50 GB).

El LVM agrupa discos físicos o cualquier otro soporte de almacenamiento denominado físico (disco, hardware RAID, software RAID, soporte de almacenamiento procedente de un SAN), que llama a volúmenes físicos PV (*Physical Volume*) en un grupo de volúmenes VG (*Volume Group*). El LVM ve el grupo VG como una especie de metadisco en el cual va a crear volúmenes lógicos LV (*Logical Volume*) a voluntad.

- ˘ Volumen físico PV: un soporte físico de almacenamiento de datos: disco duro, por ejemplo.
- ˘ Grupo de volúmenes VG: una agrupación lógica de 1 a n VG.
- ˘ Volumen lógico LV: un recorte lógico en el interior de un VG.

Se ve el volumen lógico como una partición y se puede utilizar como tal. Puede contener datos; basta con crear un sistema de archivos ordinario (ext4, por ejemplo) y montarlo de manera totalmente clásica.



Al contrario del Software RAID 0, donde la partición de datos debe ocupar todo el espacio, se pueden crear tantos volúmenes lógicos de cualquier tamaño como se desee. Pero va mucho más allá.

El LVM es dinámico. Se pueden añadir y suprimir volúmenes físicos de un grupo de volúmenes. Al añadir volúmenes físicos, la capacidad, y por lo tanto el espacio disponible del grupo, aumenta. El nuevo espacio disponible puede permitir crear nuevos volúmenes lógicos, pero también ampliar un volumen lógico existente.

Un volumen lógico es dinámico: se puede ampliar o reducir a voluntad, lo que implica que también debe poderse ampliar un sistema de archivos o reducirlo.

Apunte también que se puede utilizar una matriz RAID como volumen físico.

Las configuraciones de LVM se encuentran repartidas por los archivos y directorios de `/etc/lvm`. El archivo `/etc/lvm/lvm.conf` contiene la configuración global. La configuración de los diferentes volúmenes (físicos, grupos y lógicos) no se encuentra en un archivo, sino en una estructura presente en el interior de los propios periféricos, en sus primeros bloques: son los metadatos de los volúmenes físicos.

2. Los volúmenes físicos

a. Crear un volumen físico

Un volumen físico puede ser un disco completo o una partición clásica en el interior de un disco. En ese caso, la partición debe ser de tipo **0x8e** para un particionado MBR, o **0x8e00** para un particionado GPT.

A continuación, tiene la salida del comando **fdisk** en **/dev/sdb**. Distinga la partición primaria 1 de tipo 8e/8e00 que servirá para los ejemplos siguientes. Las otras particiones se han creado en sdc, sdd...

```
# fdisk -l /dev/sdb
```

Periférico	Inicio	Principio	Fin	Bloques	Id	Sistema
/dev/sdb1	2048	410418	408371	199,4M	8e	LVM Linux

Una vez creadas las participaciones, utilice el comando **pvccreate** en una primera partición (se pueden precisar varias particiones):

```
# pvcreate /dev/sdb1
```

```
Physical volume "/dev/sdb1" successfully created
```

b. Ver los volúmenes físicos

El comando **pvdisplay** permite visualizar el conjunto de los volúmenes físicos accesibles en su sistema. También puede llevar el nombre de volumen específico.

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

```
"/dev/sdb1" is a new physical volume of "<199,40 MiB"
```

```
--- NEW Physical volume ---
```

```
PV Name      /dev/sdb1
```

```
VG Name
```

```
PV Size      <199,40 MiB
```

```
Allocatable  NO
```

```
PE Size      0
```

```
Total PE     0
```

```
Free PE          0
Allocated PE     0
PV UUID         4Z5hDQ-1cj8-2sPm-SA7R-f0er-jtwX-IOGAxo
```

De momento, la información es reducida. El PV no pertenece todavía a ningún grupo de volúmenes (línea `VG Name`). Su tamaño es de 200 MB. Las líneas más interesantes, aunque de momento vacías porque el PV no pertenece a ningún grupo de volúmenes, son aquellas donde está indicado `PE`. PE significa *Physical Extend*, extensión física. Cada VG, y por lo tanto el PV que lo constituye, está dividido en secciones llamadas PE. El PE es la unidad básica de trabajo del LVM.

Si un PE es de 4 MB, significa que se podrá distribuir el espacio en el interior del grupo de volúmenes en secciones de 4 MB. La asignación se hace por PE: por tanto, la creación de un volumen lógico de 500 PE de 4 MB da 2000 MB.

Estos valores pasan a cero en cuanto el PE se integra en un VG.

3. Los grupos de volúmenes

a. Crear un grupo de volúmenes

Para crear un grupo de volúmenes, debe disponer de al menos un volumen físico. Puede crear un grupo de volúmenes con el comando **vgcreate**. Un grupo de volúmenes llevará el nombre que desee. Es el primer argumento del comando. Luego facilite como argumento la lista de los volúmenes físicos que componen el grupo de volúmenes, aquí sólo uno, `/dev/sdb1`.

```
# vgcreate vg01 /dev/sdb1
Volume group "vg01" successfully created
```

b. Propiedades de un grupo de volúmenes

El grupo de volúmenes dispone de numerosas propiedades. Puede listarlas con el

comando **vgdisplay**.

```
# vgdisplay vg01
--- Volume group ---
VG Name          vg01
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           196,00 MiB
PE Size           4,00 MiB
Total PE          49
Alloc PE / Size   0 / 0
Free PE / Size    49 / 196,00 MiB
VG UUID           838Yxp-zGyg-bHcf-ii9i-8tKz-Ghc3-lyMLui
```

Fíjese en las líneas **MAX LV** y **MAX PV**. La primera indica el número máximo de volúmenes lógicos que se podrá crear en este grupo de volúmenes. El valor cero indica un número infinito en teoría. La segunda indica el número máximo de volúmenes físicos que se pueden añadir al grupo de volúmenes. Aquí también el cero indica un número infinito.

Debe fijarse en el hecho de que Linux representa un caso particular en este campo. El número de LV y PV es aquí virtualmente infinito (como, por ejemplo, el volumen lógico lvm2), delo que no ocurre con los demás UNIX, donde los valores MAX LV, MAX PV y PE Size se determinan automáticamente en el momento de la creación de volúmenes, y en función, entre otras, de las propiedades de los volúmenes físicos que componen el grupo.

Se pueden establecer los valores al crear el grupo de volúmenes usando los siguientes parámetros del comando **pvcreeate**:

-l Número máximo de volúmenes lógicos.

-p Número máximo de volúmenes físicos.

-s Tamaño de las extensiones físicas (con un sufijo k, m, g o t para precisar la unidad).

Las últimas líneas se refieren a las PE (extensiones físicas). El grupo dispone actualmente de 49 extensiones de 4 MB, o sea 196 MB, todas libres. Los volúmenes lógicos ocuparán más tarde un determinado número de estas PE, según su tamaño.

El comando **vgdisplay** acepta el parámetro `-v`, que da más detalles, y en particular la lista de los volúmenes físicos que lo componen.

```
# vgdisplay -v vg01
...
--- Physical volumes ---
PV Name      /dev/sdb1
PV UUID      4Z5hDQ-1cj8-2sPm-SA7R-fOer-jtwX-IOGAxo
PV Status     allocatable
Total PE / Free PE  49 / 49
```

Ahora que el PV `/dev/sdb1` forma parte de un VG, hay más información disponible:

```
# pvdisplay /dev/sdb1
--- Physical volume ---
PV Name      /dev/sdb1
VG Name      vg01
PV Size      <199,40 MiB / not usable <3,40 MiB
Allocatable   yes
PE Size      4,00 MiB
Total PE     49
Free PE      49
Allocated PE  0
PV UUID      4Z5hDQ-1cj8-2sPm-SA7R-fOer-jtwX-IOGAxo
```

4. Los volúmenes lógicos

a. Crear un volumen lógico

Un volumen lógico es una sección de un VG (grupo de volúmenes) o, lo que es lo mismo, una partición en la cual podrá crear un sistema de archivos. Un volumen lógico LV ocupa un determinado número de PE (extensiones físicas) de un VG, contiguas o no. Esto tiene su importancia porque:

- ✓ se puede ampliar un LV mientras quedan PE libres en el VG.
- ✓ se puede reducir un LV, lo que liberará unas PE en el VG, que podrán ser utilizadas para crear nuevos LV o para ampliarlos.

Esto significa que el LVM gestiona una especie de índice y orden de PE para determinar a qué LV pertenece un PE.

Puede crear un volumen lógico con el comando **lvcreate**. Un volumen lógico lleva un nombre, dispone de un tamaño expresado en extensiones lógicas LE (*Logical Extension*) que corresponden a PE en el interior de un LV, ya sea en KB, MB, GB... El comando siguiente crea un volumen lógico llamado **data01** en el interior del VG **vg01**, de un tamaño de 150 MB. La **-L**, precisa que la unidad está en MB (**m**), GB (**g**), TB (Terabyte, **t**), PB (Petabyte) o EB (Exabyte). Para precisar un número de PE, utilice «**-l**».

```
# lvcreate -n data01 -L 150m vg01
Rounding up size to full physical extent 152,00 MiB
Logical volume "data01" created.
```

El sistema muestra un LV como una partición y después de su creación dispone de un archivo periférico asociado. El archivo está en la carpeta **/dev/<nombre_del_vg>/<nombre_del_lv>**. Observe que se trata de un vínculo simbólico hacia un archivo de **/dev/mapper** para mantener la compatibilidad con los demás Unix.

```
# ls -l /dev/vg01/data01
lrwxrwxrwx 1 root root 30 mayo 23 18:05 /dev/vg01/data01 -> ../dm-0
```

b. Propiedades de un volumen lógico

A las propiedades de un volumen lógico se accede con el comando **lvdisplay**:

```
# lvsdisplay /dev/vg01/data01
--- Logical volume ---
LV Path          /dev/vg01/data01
LV Name          data01
VG Name          vg01
LV UUID          EadkGc-f0B8-Cnqc-mAuW-j200-K38i-2mqoFM
LV Write Access   read/write
LV Creation host, time ubuntu, 2020-02-23 18:05:49 +0100
LV Status         available
# open            0
LV Size           152,00 MiB
Current LE        38
Segments          1
Allocation        inherit
Read ahead sectors    auto
- currently set to  256
Block device       253:0
```

Puede añadir los parámetros `-v` y `-m`. En este último caso, **lvsdisplay** muestra también los segmentos ocupados por el volumen lógico en el interior de los diferentes volúmenes físicos y, por tanto, el reparto de las extensiones físicas ocupadas por el volumen lógico en el interior de cada volumen físico. Como, de momento, sólo hay un PV en el interior del VG, puede obtener lo siguiente:

```
# lvsdisplay -m /dev/vg01/data01
...
--- Segments ---
Logical extents 0 to 37:
Type            linear
Physical volume  /dev/sdb1
Physical extents 0 to 37
```

c. Acceso al volumen lógico

Puede crear un sistema de archivos y montar el LV como cualquier partición:


```
# mkfs -t ext4 /dev/vg01/data01
mke2fs 1.45.3 (14-Jul-2019)
Creando un sistema de archivos con 38912 4k bloques y 38912 i-nodos.
UUID del sistema de archivos 4af25d50-5655-4ba8-96b6-171163b405a0
Superbloques de emergencia almacenados en los bloques:
    32768
Asignación de las tablas de grupo: completada
Escritura de las tablas de i-nodos: completado
Creación del log (4.096 bloques): completado
Escritura de los superbloques y de la información de compatibilidad del
sistema de archivos: completado
```

Sólo queda montar el nuevo sistema de archivos.

```
# mount -t ext4 /dev/vg01/data01 /mnt/data01
# df /mnt/data01
Sis. de fich.    1K-blocs   Ocupado Disponible Capacidad Montado en
/dev/mapper/vg01-data01
        6192704  143488  5734644  3%    /mnt/data01
```

5. Ampliaciones y reducciones

a. Los grupos de volúmenes

Hasta aquí todo sigue el guión esperado. La fuerza del LVM reside en su dinamismo. Vayamos ahora a sacar partido de la situación: supongamos que necesita crear un nuevo LV de 150 MB llamado **data02** en el interior del VG **vg01**. A continuación presentamos el estado actual de **vg01**:

```
--- Volume group ---
VG Name          vg01
System ID
Format           lvm2
Metadata Areas    1
```

```

Metadata Sequence No 2
VG Access      read/write
VG Status      resizable
MAX LV         0
Cur LV        1
Open LV        1
Max PV         0
Cur PV        1
Act PV         1
VG Size        196 MiB
PE Size        4,00 MiB
Total PE       49
Alloc PE / Size 38 / 152,00 MiB
Free PE / Size  11 / 44,00 MiB
VG UUID        838Yxp-zGyg-bHcf-ii9i-8tKz-Ghc3-lyMLui

```

No hay espacio suficiente, sólo 44 MB (11 PE). Hay que añadir un nuevo volumen físico en el interior de este VG. Esto se hace con el comando **vgextend**, que funciona de la misma manera que **vgcreate**: indique el nombre del VG seguido del PV o de los PV que quiere añadir.

```

# pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created
# vgextend vg01 /dev/sdc1
Volume group "vg01" successfully extended

```

A continuación se muestra el nuevo estado de **vg01**. Observe que el VG contiene ahora dos PV, y que juntos suman un tamaño de 240 MB (60 PE).

```

# vdisplay vg01 |grep Free
Free PE / Size  60 / 240,00 MiB

```

Sólo queda crear el LV **data02** de 150 MB:

```

# lvcreate -n data02 -L 6g vg01

```

Logical volume "data02" created

Cuando crea un LV, el LVM busca optimizar el uso de las PE, de manera que sean lo más contiguas posible, y si se puede, en un mismo PV. Esto se ve con el comando **lvdisplay** y el parámetro **-m**, en las líneas **Segments** y la lista de los segmentos propuestos.

```
# lvdisplay -m /dev/vg01/data02
--- Logical volume ---
LV Path          /dev/vg01/data02
LV Name          data02
VG Name          vg01
LV UUID          1WzTUu-rb4B-m3tT-tynZ-cs3C-pBhn-BktXR2
LV Write Access   read/write
LV Creation host, time ubuntu, 2020-02-23 18:09:55 +0100
LV Status        available
# open           0
LV Size          152,00 MiB
Current LE       38
Segments       1
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device     253:1

--- Segments ---
Logical extents 0 to 37:
Type            linear
Physical volume  /dev/sdc1
Physical extents 0 to 37
```

Los comandos siguientes crean el sistema de archivos en el volumen lógico y lo montan:

```
# mkfs -t ext4 /dev/vg01/data02
# mount -t ext4 /dev/vg01/data02 /mnt/data02
```

b. Ampliar un volumen lógico

Resulta que el LV **data01** de 150 MB es demasiado pequeño. Hay que añadirle 80 MB, lo que es posible porque quedan 88 MB (22 PE) en el grupo de volúmenes **vg01**:

```
# vgdisplay vg01 | grep Free
Free PE / Size    22 / 88,00 MiB
```

La ampliación de un volumen lógico se hace en este orden:

- ˆ Ampliación del LV con el comando **lvextend**.
- ˆ Ampliación del sistema de archivos con **resize2fs** (ext), **btrfsctl** (btrfs) o **xfs growfs** (xfs).

Ampliación del LV

El comando **lvextend** autoriza los parámetros **-l** (número de extensiones lógicas LE) o **-L** como para **lvcreate**. A continuación, indique el nuevo tamaño del LV o, si añade un prefijo **+**, el tamaño adicional deseado. También puede indicar, como último argumento, el nombre del PV en el cual forzar la extensión del LV (también es posible con **lvcreate**). Sólo funcionará si el PV o los PV indicados disponen de bastante PE.

El comando siguiente añade 20 LE (44x20=80 MB) en **data01**:

```
# lvextend -l +20 /dev/vg01/data01
Size of logical volume vg01/data01 changed from 152,00 MiB (38 extents)
to 232,00 MiB (58 extents).
Logical volume vg01/data01 successfully resized.
```

Mire ahora en qué PV están ubicados los datos:

```
# lvsdisplay -m /dev/vg01/data01
--- Logical volume ---
LV Path          /dev/vg01/data01
LV Name          data01
VG Name          vg01
LV UUID          EadkGc-f0B8-Cnqc-mAuW-j200-K38i-2mqoFM
LV Write Access   read/write
```

```

LV Creation host, time ubuntu, 2020-02-23 18:05:49 +0100
LV Status          available
# open            1
LV Size           232,00 MiB
Current LE        58
Segments          2
Allocation         inherit
Read ahead sectors auto
- currently set to 256
Block device      253:0

```

```

--- Segments ---

```

```

Logical extents 0 to 48:

```

```

Type              linear
Physical volume    /dev/sdb1
Physical extents   0 to 48

```

```

Logical extents 49 to 57:

```

```

Type              linear
Physical volume    /dev/sdc1
Physical extents   38 to 46

```

El volumen lógico data01 ocupa, en efecto, 230 MB, en dos segmentos de PE, que están en los PV `/dev/sdb1` y `/dev/sdc1`. El LVM asignó por lo tanto un espacio en el conjunto de los PV del VG.

Extensión del sistema de archivos

Sólo el volumen lógico ha sido ampliado. De momento, el tamaño del sistema de archivos contenido en `data01` no ha cambiado:

```

# df -h /mnt/data01
Sis. de fich.    Tam.  Oc.  Disp. %Oc.  Montado en
/dev/mapper/vg01-data01
                132M 168K 121M   1% /mnt/data01

```

El comando **resize2fs** permite reducir y ampliar un sistema de archivos. El primer argumento es el sistema de archivos; el segundo, el tamaño, con un eventual sufijo K (KB),

M (MB) o G (GB). Sin sufijo, se indica el número de bloques del sistema de archivos. Si el tamaño está ausente, el sistema de archivos se adaptará al tamaño de la partición o del LV.

Se puede utilizar el comando **resize2fs** en caliente, es decir, con el sistema de archivos montado, para las ampliaciones. En cambio, habrá que desmontar el sistema de archivos para reducirlo.

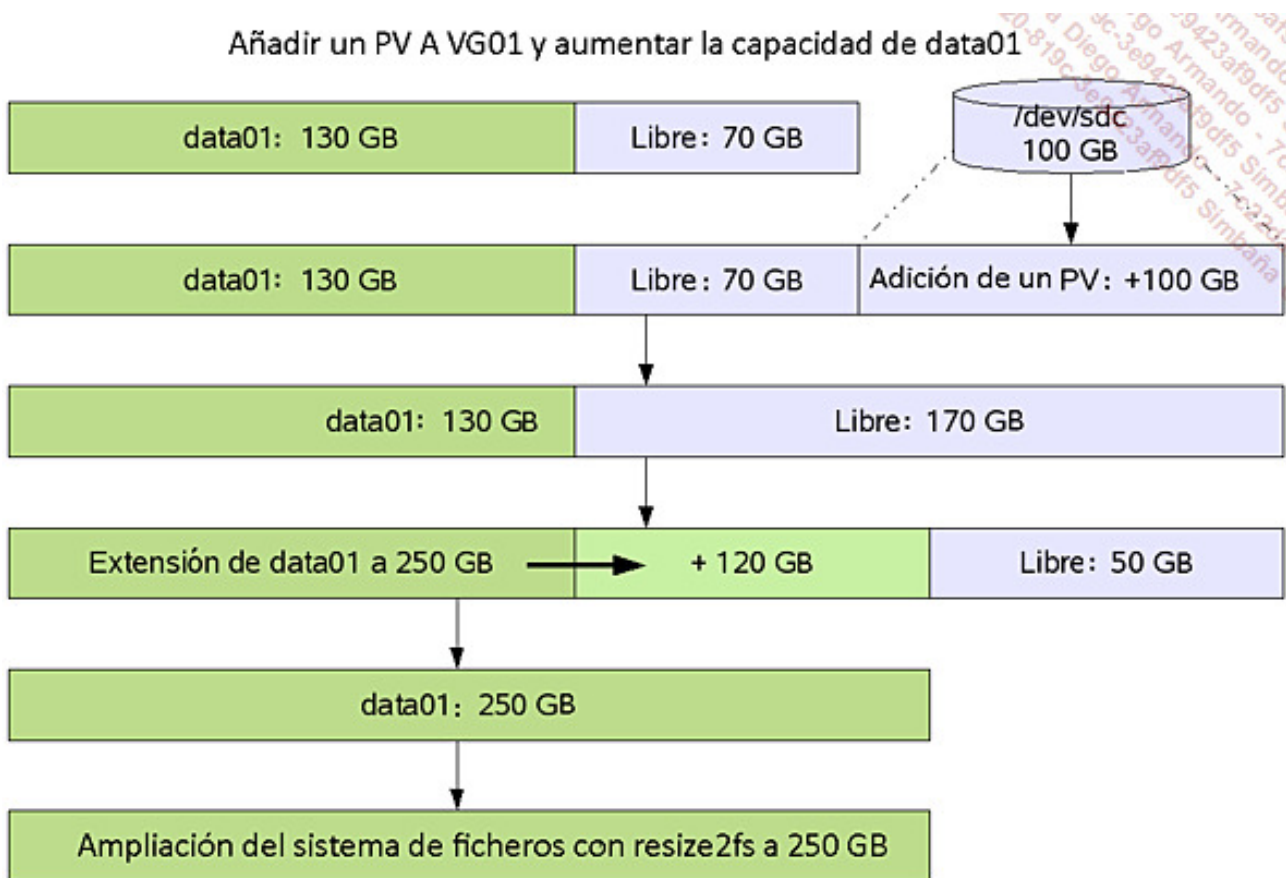
```
# resize2fs /dev/vg01/data01
resize2fs 45.3 (14-Jul-2019)
El Sistema de archivos de /dev/vg01/data01 está montado en /mnt/data01;
El cambio de tamaño se debe efectuar en línea
old desc_blocks = 1, new_desc_blocks = 1
El sistema de archivos /dev/vg01/data01 tiene ahora un tamaño
de 59392 bloques.
```

Mire el estado del sistema de archivos: ocupa ahora 12 GB:

```
# df -h /mnt/data01
Sis. de fich.    Tam.  Oc.  Disp. %Oc. Montado en
/dev/mapper/vg01-data01
                212M 168K 201M  1% /mnt/data01
```

Ahora puede ver la potencia del LVM: adición de nuevos volúmenes físicos y ampliación de volúmenes lógicos en caliente y de manera dinámica. ¿Ya no queda espacio? No importa: basta con añadir un nuevo disco, transformarlo en PV, añadirlo en el VG y volver a dimensionar el LV que no tiene bastante espacio, sin que sea necesario particionar de nuevo, recrear sistemas de archivos, hacer backups, etc.

Añadir un PV A VG01 y aumentar la capacidad de data01



Un sistema de archivos btrfs también puede aumentar o reducir su tamaño. Al contrario de un sistema de archivos de tipo ext, la reducción se puede hacer en caliente, sin desmontar el sistema de archivos. Para aumentar su sistema de archivos a la capacidad máxima del LV utilice el comando **btrfsctl**. Observe que se le indica el punto de montaje del sistema de archivos:

```
# btrfsctl -r max /mnt/data01
```

c. Reducir un volumen lógico

Para reducir el tamaño de un volumen lógico, tiene que proceder según el orden siguiente:

- Comprobación del sistema de archivos que hay que reducir con **fsck**, para sistemas de archivos ext.
- Reducción del sistema de archivos contenidos en el volumen lógico con **resize2fs** (ext) o **btrfsctl** (btrfs). Un sistema de archivos xfs no se puede reducir.

Reducción del volumen lógico con el comando **lvreduce**.

Va a reducir el LV **data01** a 30 MB. Sólo es posible cuando los datos ocupan menos de 30 MB. En primer lugar, compruebe el tamaño actual del sistema de archivos. En este caso, está casi vacío:

```
# df -h /mnt/data01
Sis. de fich.    Tam.  Oc.  Disp. %Oc.  Montado en
/dev/mapper/vg01-data01
                212M 168K 201M   1%  /mnt/data01
```

Sólo se puede reducir el sistema de archivos si no está montado; desmóntelo:

```
# umount /mnt/data01
```

Compruebe el sistema de archivos de tipo ext:

```
# fsck -f /dev/vg01/data01
fsck de util-linux 2.34
e2fsck 1.45.3 (14-Jul-2019)
Paso 1: verificación de los i-nodos,
de los bloques y de los tamaños
Paso 2: verificación de la estructura de los directorios
Paso 3: verificación de la conectividad de los directorios
Paso 4: verificación de los contadores de referencia
Paso 5: verificación de la información del sumario del grupo

/dev/mapper/vg01-data01 : 11/38912 archivos (0.0% no contiguos),
5362/59392 bloques
```

Vuelva a dimensionar el sistema de archivos a 4 GB:

```
# resize2fs /dev/vg01/data01 30 Mo
resize2fs 1.45.3 (14-Jul-2019)
Redimensionando el sistema de archivos /dev/vg01/data01 a 7680 (4k) bloques.
```


El sistema de archivos /dev/vg01/data01 tiene ahora un tamaño de 7680 bloques (4k).

Para un sistema de archivos btrfs, haga lo siguiente:

```
# btrfsctl -r 30m /mnt/data01
```

Compruebe el nuevo tamaño del sistema de archivos. 4096*7680 dan, efectivamente, 30 MB. Para btrfs, simplemente ejecutando df obtendrá el resultado.

```
# dumpe2fs -h /dev/vg01/data01 | grep ^Block
dumpe2fs 1.45.3 (14-Jul-2019)
Block count:      7680
Block size:       4096
Blocks per group: 32768
```

Para terminar, vuelva a dimensionar el LV a 4 GB. La sintaxis de **lvreduce** es la misma que **lvextend**, excepto que no es posible precisar un PV. Tenga cuidado de no equivocarse aquí: si ha reducido mal el sistema de archivos, puede destruirlo. Conteste **y** a la pregunta sobre si está seguro.

```
# lvreduce -L 30 Mo /dev/vg01/data01
Rounding size to boundary between physical extents: 32,00 MiB.
WARNING: Reducing active logical volume to 32,00 MiB.
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce vg01/data01? [y/n]: y
Size of logical volume vg01/data01 changed from 52,00 MiB (13 extents)
to 32,00 MiB (8 extents).
Logical volume vg01/data01 successfully resized.
```

Esta etapa no es necesaria para un sistema de archivos btrfs. Si su sistema de archivos es del tipo ext, monte de nuevo el sistema de archivos:

```
# mount -t ext4 /dev/vg01/data01 /mnt/data01
```

```
# df -h /mnt/data01
```

```
Sis. de archivos    Tam.  Oc.  Disp. %Oc Montado en
/dev/mapper/vg01-data01 12M 96K 10M 1% /mnt/data01
```

d. Mover el contenido de un volumen físico

En entornos empresariales es corriente mover un PV hacia otro. Puede ser para sustituir un disco que contiene el PV por otro (para ampliar, por ejemplo). En este caso, puede desplazar el contenido de un PV hacia otro, incluso varios PE de un LV hacia otro PV, o también determinados PE precisos. Sin precisar el destino, el LVM va a mover todos los PE del PV en los otros PV del grupo de volúmenes. Cuidado: los volúmenes físicos deben ser del mismo grupo de volúmenes.

El comando **pvmove** permite mover los PE de un PV hacia otro. Tendrá que desplazar el contenido del PV **/dev/sdc1** a **/dev/sdb1**. **/dev/sdc1** contiene 38 PE en uso: todos los LE del LV **data02**.

```
--- Physical volume ---
PV Name      /dev/sdc1
VG Name      vg01
PV Size      <199,40 MiB / not usable <3,40 MiB
Allocatable  yes
PE Size      4,00 MiB
Total PE     49
Free PE      11
Allocated PE 38
PV UUID      xSxLvq-ytss-u9Kp-hcZo-Z6Yc-0qDL-qUDeDj
--- Physical Segments ---
Physical extent 0 to 37:
  Logical volume /dev/vg01/data02
  Logical extents 0 to 37
  Physical extent 38 to 48:
  FREE
```

Compruebe si el volumen físico **/dev/sdb1** dispone de bastante espacio para recibir el contenido de **/dev/sdc1**. Quedan 41 PE en este último, por lo tanto es posible.

```

--- Physical volume ---
PV Name      /dev/sdb1
VG Name      vg01
PV Size      <199,40 MiB / not usable <3,40 MiB
Allocatable  yes
PE Size
             4,00 MiB
Total PE
             49
Free PE     41
Allocated PE      8
PV UUID          4Z5hDQ-1cj8-2sPm-SA7R-f0er-jtwX-IOGAxo

```

Mueva el PV `/dev/sdc1` hacia el PV `/dev/sdb1`. Puede utilizar el parámetro `-v` para seguir el progreso. Observe que la operación se efectúa cuando no hay ningún sistema de archivos desmontado:

```

# pvmove -v /dev/sdc1 /dev/sdb1
Executing: /sbin/modprobe dm-mirror
Cluster mirror log daemon not included in build.
Archiving volume group "vg01" metadata (seqno 8).
Creating logical volume pvmove0
activation/volume _list configuration setting not defined:
Checking only host tags for vg01/data02.
Moving 38 extents of logical volume vg01 /data02.
activation/volume
_list configuration setting not defined:
Checking only host tags for vg01/data02.
Creating vg01-pvmove0
Loading table for vg01-pvmove0 (253:2).
Loading table for vg01-data02 (253:1).
Suspending vg01-data02 (253:1) with device flush
Resuming vg01-pvmove0 (253:2).
Resuming vg01-data02 (253:1).
Creating volume group backup "/etc/lvm/backup/vg01" (seqno 9).
activation/volume _list configuration setting not defined:
Checking only host tags for vg01/pvmove0.
Checking progress before waiting every 15 seconds.

```

```
/dev/sdc1: Moved: 36,84%
/dev/sdc1: Moved: 100,00%
Polling finished successfully.
```

Ahora compruebe el estado del grupo de volúmenes:

```
# vgdisplay -v vg01 | grep -A 100 "Physical"
PV Name          /dev/sdb1
PV UUID          4Z5hDQ-1cj8-2sPm-SA7R-fOer-jtwX-IOGAxo
PV Status        allocatable
Total PE / Free PE  49 / 3
PV Name          /dev/sdc1
PV UUID          xSxLvg-ytss-u9Kp-hcZo-Z6Yc-0qDL-qUDeDj
PV Status        allocatable
Total PE / Free PE  49 / 49
```

El segundo PV del `vg01` está totalmente vacío. Por lo tanto, ya se puede suprimir del VG.

e. Reducir un grupo de volúmenes

El comando **vgreduce** permite retirar uno o varios PV de un grupo de volúmenes. Para ello, hace falta primero que los PV en cuestión estén vacíos: sus PE deben estar totalmente libres. Es el caso de `/dev/sdc1`, que va a retirar del VG `vg01`:

```
# vgreduce vg01 /dev/sdc1
Removed "/dev/sdc1" from volume group "vg01"
```

Controle que el VG ya no contiene ese PV:

```
# vgdisplay -v vg01 | grep -A 100 "Physical"
--- Physical volumes ---
PV Name          /dev/sdb1
PV UUID          4Z5hDQ-1cj8-2sPm-SA7R-fOer-jtwX-IOGAxo
PV Status        allocatable
```

Total PE / Free PE 49 / 3

6. Suprimir un grupo de volúmenes

a. Etapas

Para suprimir un grupo de volúmenes, debe seguir las etapas siguientes:

- ˆ Desmontar todos los sistemas de archivos de los LV asociados.
- ˆ Suprimir todos los volúmenes lógicos con **lvremove**.
- ˆ Retirar todos los volúmenes físicos del VG con **lvreduce**.
- ˆ Destruir el VG con **vgremove**.

Va a destruir el grupo de volúmenes **vg01**.

b. Suprimir un volumen lógico

Desmunte **data01** y **data02** :

```
# umount /mnt/data0{1,2}
```

Suprima los volúmenes lógicos con **lvremove**:

```
# lvremove /dev/vg01/data01 /dev/vg01/data02
Do you really want to remove active logical volume "data01"? [y/n]: y
Logical volume "data01" successfully removed
Do you really want to remove active logical volume "data02"? [y/n]: y
Logical volume "data02" successfully removed
```

c. Retirar todos los volúmenes físicos

Utilice el comando **vgreduce** con el parámetro **-a** para **ALL**:

```
# vgreduce -a vg01
Can't remove final physical volume "/dev/sdb2" from volume group
"vg01"
```

Observe que el comando **vgreduce** siempre deja como mínimo un PV en el VG, porque hace falta al menos un PV para constituir un VG.

d. Destruir un grupo de volúmenes

Utilice el comando **vgremove** para destruir un grupo de volúmenes:

```
# vgremove vg01
Volume group "vg01" successfully removed
```

Compruebe que los archivos y directorios asociados han desaparecido:

```
# ls /dev/vg01
ls: no puede acceder a /dev/vg01: Ningún archivo o carpeta de este tipo
```

Para terminar, el comando **vgdisplay** ya no devuelve nada:

```
# vgdisplay
```

e. Suprimir un volumen físico

Ahora se pueden destruir los dos volúmenes físicos, puesto que ya no están en uso. Puede destruir la información contenida en el volumen con el comando **pvremove**. Sin embargo, si destruye la partición vía **fdisk** o crea un sistema de archivos encima, el efecto es el mismo.

```
# pvremove /dev/sdb1
```

Labels on physical volume **"/dev/sdb1"** successfully wiped.

```
# pvremove /dev/sdc1
```

Labels on physical volume **"/dev/sdc1"** successfully wiped.

7. Comandos adicionales

Sólo hemos echado un vistazo rápido a las posibilidades del LVM. Tenga presente que cuenta con otros comandos, como por ejemplo:

- ✧ **pvchange**: modifica el estado de un volumen físico, por ejemplo para prohibir la asignación de extensiones físicas en este volumen.
- ✧ **pvresize**: vuelve a dimensionar un volumen físico si su partición o disco de origen ha sido ampliado o reducido.
- ✧ **pvscan**: busca todos los volúmenes presentes en todos los soportes de almacenamiento del sistema.
- ✧ **vgchange**: modifica los atributos de un grupo de volúmenes, para activarlo o desactivarlo por ejemplo, pero también para modificar sus valores máximos de PV y de PE, o para prohibir su ampliación o su reducción.
- ✧ **vgscan**: busca todos los grupos de volúmenes en todos los soportes.
- ✧ **vgrename**: renombra un grupo de volúmenes.
- ✧ **vgmerge**: ensambla dos grupos de volúmenes en uno único.
- ✧ **lvresize**: vuelve a dimensionar un volumen lógico. Equivale tanto a **lvextend** como a **lvreduce**.
- ✧ **lvchange**: modifica los atributos de un volumen lógico.
- ✧ **lvrename**: renombra un volumen lógico.