

Logical Volume Manager

La gestión tradicional de los discos que consiste en una o distintas particiones por disco, pudiendo contener cada una un sistema de archivos o un espacio de swap, impone límites en términos de flexibilidad de uso y de evolución, en particular:

- ˘ Un sistema de archivos no puede ocupar distintos discos (exceptuando las técnicas de RAID).
- ˘ El tamaño máximo de un sistema de archivos depende del tamaño de la partición que lo contiene y es delicado aumentar el tamaño de una partición y, por lo tanto, de su sistema de archivos.

Estas distintas limitaciones y restricciones han llevado a proponer una nueva herramienta de gestión del espacio de almacenamiento, el **Gestor de volúmenes** lógicos (LVM, *Logical Volume Manager*).

El principio es de implementar una capa lógica intermediaria entre los gestores de sistemas de archivos y el material de almacenamiento. El gestor de sistemas de archivos ve los volúmenes lógicos en lugar de las particiones o discos. Estos volúmenes lógicos son administrados por el gestor LVM, que hace de enlace entre los dispositivos usados. De esta manera, podemos dejar atrás la mayoría de las limitaciones evocadas anteriormente:

- ˘ Un sistema de archivos puede extenderse entre distintos discos físicos.
- ˘ Se puede, mucho más fácilmente, aumentar el tamaño máximo de un sistema de archivos.
- ˘ Se pueden combinar controladores de discos de tecnologías diferentes, o discos locales y remotos dentro de un sistema de archivos.
- ˘ Se puede añadir un disco físico en caliente y utilizarlo para agrandar un sistema de archivos existente.

Se puede combinar perfectamente el RAID de hardware o software con LVM, permitiendo el beneficio de la tolerancia de fallos y la flexibilidad de los volúmenes lógicos.

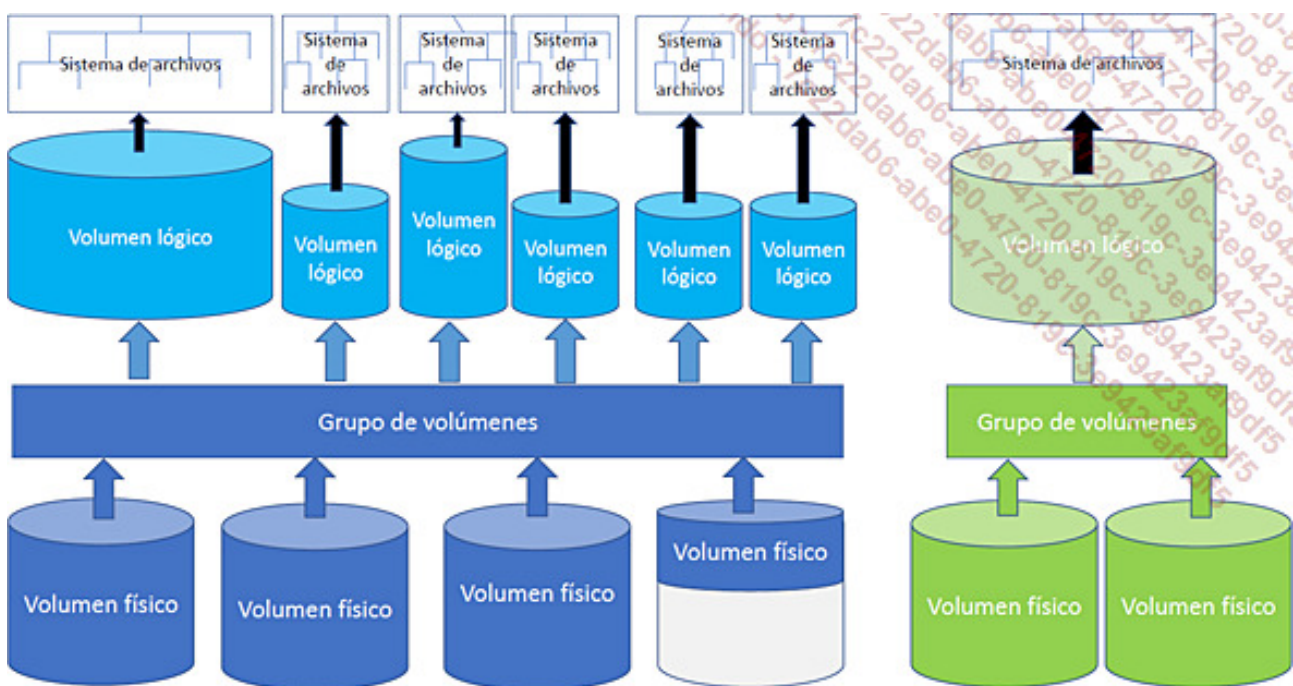
La tecnología LVM, originaria de Linux, se implementó en el núcleo Linux en los años 2000 y se utiliza a gran escala en entornos de producción desde hace años. La versión actual se llama LVM2.



El Gestor de volúmenes lógicos Linux LVM puede gestionar por sí mismo diferentes niveles de RAID de software o apoyarse en el módulo de gestión del RAID visto anteriormente, `md`. En la certificación solamente se pide conocer la gestión de RAID por `md`.

1. Arquitectura de los volúmenes lógicos

Ejemplo de arquitectura LVM con seis discos duros, organizados en dos grupos de volúmenes y siete volúmenes lógicos:



La arquitectura de los volúmenes lógicos LVM está estructurada en tres niveles:

- Los periféricos de almacenamiento, que pueden ser discos enteros (lo más aconsejado) o particiones de discos, gestionados por LVM, se llaman **volúmenes físicos**. Un volumen físico está dividido en conjuntos del mismo tamaño llamados extensiones físicas (PE, *Physical Extents*).
- Un **grupo de volúmenes** agrupa a uno o distintos volúmenes físicos, administra una tabla del conjunto de las extensiones físicas disponibles dentro de los

volúmenes físicos del grupo.

Los **volúmenes lógicos** del grupo de volúmenes. Cuando se crea un volumen lógico, éste está constituido por conjuntos del mismo tamaño llamados extensiones lógicas (LE, *Logical Extents*). El gestor de volúmenes lógicos reparte las extensiones lógicas asignadas a un volumen lógico usando las extensiones físicas de los volúmenes del grupo. Un volumen lógico puede, por lo tanto, estar repartido en uno o distintos volúmenes físicos. Mientras haya extensiones físicas disponibles, se podrán crear nuevos volúmenes lógicos o extender los volúmenes lógicos existentes.

Un volumen lógico está asociado a un archivo especial en modo de bloque, que puede ser utilizado para crear un sistema de archivos (o declararlo espacio de swap).

2. Configuración del gestor LVM

El archivo de configuración por defecto del gestor LVM se encuentra en `/etc/lvm/lvm.conf`.

Aunque el archivo está documentado, su sintaxis es compleja: modificar su contenido debe hacerse prudentemente.

El comando `lvmconfig` permite mostrar o modificar dinámicamente la configuración actual del gestor LVM.

Ejemplo

Configuración por defecto en una distribución CentOS 8:

```
lvmconfig
config {
    checks=1
    abort_on_errors=0
    profile_dir="/etc/lvm/profile"
}
local {
}
dmeventd {
```

```

mirror_library="libdevmapper-event-lvm2mirror.so"
snapshot_library="libdevmapper-event-lvm2snapshot.so"
thin_library="libdevmapper-event-lvm2thin.so"
}
activation {
    checks=0
    udev_sync=1
    udev_rules=1
    verify_udev_operations=0
    retry_deactivation=1
    missing_stripe_filler="error"
    use_linear_target=1
    reserved_stack=64
    reserved_memory=8192
    process_priority=-18
    raid_region_size=2048
    readahead="auto"
    raid_fault_policy="warn"
    mirror_image_fault_policy="remove"
    mirror_log_fault_policy="allocate"
    snapshot_autoextend_threshold=100
    snapshot_autoextend_percent=20
    thin_pool_autoextend_threshold=100
    thin_pool_autoextend_percent=20
    vdo_pool_autoextend_threshold=100
    use_mlockall=0
    monitoring=1
    polling_interval=15
    activation_mode="degraded"
}
global {
    umask=63
    test=0
    units="r"
    si_unit_consistency=1
    suffix=1
    activation=1
    proc="/proc"
    etc="/etc"
    wait_for_locks=1

```

```

locking_dir="/run/lock/lvm"
prioritise_write_locks=1
abort_on_internal_errors=0
metadata_read_only=0
mirror_segtype_default="raid1"
support_mirrored_mirror_log=0
raid10_segtype_default="raid10"
sparse_segtype_default="thin"
event_activation=1
use_lvmlockd=0
system_id_source="none"
use_lvmpolld=1
notify_dbus=1
}
shell {
    history_size=100
}
backup {
    backup=1
    backup_dir="/etc/lvm/backup"
    archive=1
    archive_dir="/etc/lvm/archive"
    retain_min=10
    retain_days=30
}
log {
    verbose=0
    silent=0
    syslog=1
    overwrite=0
    level=0
    command_names=0
    prefix=" "
    activation=0
    debug_classes=["memory","devices","io","activation","allocation",
"metadata","cache","locking","lvmpolld","dbus"]
}
allocation {
    maximise_cling=1
    use_blkid_wiping=1

```

```

wipe_signatures_when_zeroing_new_lvs=1
mirror_logs_require_separate_pvs=0
cache_pool_metadata_require_separate_pvs=0
thin_pool_metadata_require_separate_pvs=0
}
devices {
    dir="/dev"
    scan="/dev"
    obtain_device_list_from_udev=1
    external_device_info_source="none"
    sysfs_scan=1
    scan_lvs=0
    multipath_component_detection=1
    md_component_detection=1
    fw_raid_component_detection=0
    md_chunk_alignment=1
    data_alignment_detection=1
    data_alignment=0
    data_alignment_offset_detection=1
    ignore_suspended_devices=0
    ignore_lvm_mirrors=1
    require_restorefile_with_uuid=1
    pv_min_size=2048
    issue_discards=0
    allow_changes_with_duplicate_pvs=0
}

```

3. Sintaxis general de los comandos LVM

Los comandos de gestión de los objetos LVM (volúmenes físicos, grupos de volúmenes y volúmenes lógicos) tienen un nombre de la forma siguiente:

ObjetoAcción [-Opciones]

o

`[pv|vg|lv]s`

Donde:

| | |
|-----------|--|
| Objeto | Tipo de objeto LVM: <code>pv</code> (<i>physical volume</i>), <code>vg</code> (<i>volume group</i>) o <code>lv</code> (<i>logical volume</i>). |
| Acción | Acción aplicable al objeto: <code>create</code> , <code>display</code> , <code>remove</code> ... |
| -Opciones | Opciones comunes o específicas del objeto. |

Descripción

Los 3 comandos principales tienen una forma simplificada: `pvs`, `vgs` o `lvs`, que muestra el conjunto de las características del tipo de objeto especificado.

En las opciones comunes a los tres comandos, encontramos:

| | |
|---------------------------------|---|
| <code>[-h --help]</code> | Ayuda. |
| <code>[-v --verbose]</code> | Modo detallado. |
| <code>[-q --quiet]</code> | Modo silencioso, sin mostrar nada en la pantalla. |
| <code>[-y --yes]</code> | Responder afirmativamente a todas las preguntas. |
| <code>[-t --test]</code> | Comprobación sin ejecutar nada. |
| <code>[-d --debug]</code> | Modo depuración. |

[Ejemplo](#)

Síntesis de la configuración de tres tipos de objetos LVM, en una distribución CentOS 8:

pvs

```
PV      VG      Fmt Attr PSize  PFree
/dev/sda2 cl      lvm2 a-- <154,09g  0
/dev/sda3 vg-datos lvm2 a-- 137,99g <137,70g
```

El sistema dispone de dos volúmenes físicos en dos grupos de volúmenes.

vgs

```
VG      #PV #LV #SN Attr  VSize  VFree
cl      1  4  0 wz--n- <154,09g  0
vg-datos 1  2  0 wz--n- 137,99g <137,70g
```

El sistema dispone de dos grupos de volúmenes, cada uno de ellos tiene un volumen físico. **cl** tiene cuatro volúmenes lógicos y **vg-datos** tiene dos.

lvs

```
LV VG      Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
home cl    -wi-ao---- <50,18g
root cl    -wi-ao---- 50,00g
swap cl    -wi-ao---- 3,91g
var cl     -wi-ao---- 50,00g
lvol0 vg-datos -wi-a----- 200,00m
lvol1 vg-datos -wi-a----- 100,00m
```

Hay seis volúmenes lógicos.

De la misma manera, se pueden usar comandos de búsqueda de objetos LVM:

```
[pv|vg|lv]scan
```

[Ejemplo](#)

Búsqueda de los 3 tipos de objetos LVM, en una distribución Debian 10:

pvscan

```
PV /dev/sda5 VG debian10-vg lvm2 [<148,81 GiB / <74,14 GiB free]
PV /dev/sdc          lvm2 [<14,46 GiB]
Total: 2 [<163,27 GiB] / in use: 1 [<148,81 GiB] / in no VG: 1 [<14,46 GiB]
```

vgscan

```
Reading all physical volumes. This may take a while...
Found volume group "debian10-vg" using metadata type lvm2
```

lvscan

```
ACTIVE      '/dev/debian10-vg/root' [13,58 GiB] inherit
ACTIVE      '/dev/debian10-vg/var' [<4,84 GiB] inherit
ACTIVE      '/dev/debian10-vg/swap_1' [3,93 GiB] inherit
ACTIVE      '/dev/debian10-vg/tmp' [908,00 MiB] inherit
ACTIVE      '/dev/debian10-vg/home' [<51,26 GiB] inherit
ACTIVE      '/dev/debian10-vg/lvol0' [152,00 MiB] inherit
ACTIVE      '/dev/debian10-vg/lvol1' [28,00 MiB] inherit
```

4. Los volúmenes físicos

Un disco físico (o una partición de disco), para poder ser administrado por LVM debe contener estructuras específicas de gestión de información LVM, que lo definen como un volumen físico LVM. Después, hay que incluirlo en un grupo de volúmenes.



El disco puede ser un disco real o una partición de disco. También se puede tratar de un volumen RAID gestionado por `md`, compuesto de muchos discos; será considerado como un solo volumen físico LVM.

a. Creación de un volumen físico

El comando `pvcreate` permite crear un volumen físico LVM.

Sintaxis

`pvccreate [Opciones] ArchivoEspecial`

Parámetros principales

| | |
|-----------------|-----------------------------------|
| Opciones | Opciones. |
| ArchivoEspecial | Archivo especial del dispositivo. |

Descripción

El comando escribe las estructuras de gestión LVM necesarias en el espacio de almacenamiento asociado al archivo especial `ArchivoEspecial`. Puede tratarse de un disco entero de una partición o de un volumen RAID gestionado por `md`. El comando escribe una etiqueta particular LVM y un UUID en el espacio de almacenamiento e inicializa los metadatos necesarios.

Numerosas opciones permiten especificar las características técnicas deseadas para el volumen físico.



Si el disco o la partición contiene un sistema de archivos, aunque no esté montado, el comando puede fallar mostrando el error "`Device /dev/xxx excluded by a filter`". En ese caso, hay que suprimir las estructuras del sistema de archivos en el disco, usando por ejemplo el comando `wipefs -a /dev/xxx`.

Ejemplo

Creación de un volumen físico en un disco entero:

Como el disco ya ha servido para diferentes tests, se suprimirá un hipotético sistema de archivos residual:

```
wipefs -a /dev/sdc
```

Se crea el volumen físico:

```
pvcreate /dev/sdc Physical volume "/dev/sdc" successfully created.
```

Se comprueba:

```
blkid /dev/sdc  
/dev/sdc: UUID="J8UKBL-IO35-EUWj-XD4F-8Hc2-aJBv-jC5VPH" TYPE="LVM2_member"
```

El disco se ve como de tipo LVM2.

Se comprueba el conjunto de los volúmenes físicos existentes:

```
pvs  
PV      VG      Fmt Attr PSize  PFree  
/dev/sda2 cl      lvm2 a-- <154,09g  0  
/dev/sda3 vg-datos lvm2 a-- 137,99g <137,70g  
/dev/sdc      lvm2 ---  7,54g  7,54g
```

b. Datos en los volúmenes físicos

El comando `pvdiskdisplay` muestra las características de uno o distintos volúmenes físicos LVM.

Sintaxis

```
pvdiskdisplay [ Opciones ] [ Volfísico ... ]
```

Parámetros principales

| | |
|------------|---|
| Opciones | Opciones. |
| Volfí sico | Archivos especiales del volumen o de los volúmenes físicos. |

Descripción

El comando muestra las características de los volúmenes especificados o de todos los volúmenes físicos reconocidos si no existe el argumento `Volfí sico`. El comando acepta las opciones generales de los comandos LVM, y otros más específicos para los volúmenes físicos.

Ejemplo

Lista detallada de los volúmenes físicos del sistema:

```
pvdisplay
--- Physical volume ---
PV Name           /dev/sda3
VG Name           vg-datos
PV Size           137,99 GiB / not usable 0
Allocatable       yes
PE Size           4,00 MiB
Total PE          35326
Free PE           35251
Allocated PE      75
PV UUID           wd0CH3-KMzU-tYi0-eKtJ-L0on-OweG-I5Jg2B

--- Physical volume ---
PV Name           /dev/sda2
VG Name           cl
PV Size           154,09 GiB / not usable 4,00 MiB
Allocatable       yes (but full)
PE Size           4,00 MiB
Total PE          39447
Free PE           0
```

```

Allocated PE      39447
PV UUID          Jlyv1F-7nhy-3mej-SwLJ-x03U-SOpl-q47hCQ

```

"/dev/sdc" is a new physical volume of "7,54 GiB"

--- **NEW** Physical volume ---

```

PV Name          /dev/sdc
VG Name
PV Size          7,54 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          J8UKBL-IO35-EUWj-XD4F-8Hc2-aJBv-jC5VPH

```

Se puede ver que el sistema tiene tres volúmenes físicos. Los dos primeros, `/dev/sda3` y `/dev/sda2`, son particiones y han sido asignadas a dos grupos de volúmenes distintos, `vg-datos` y `c1`. El tamaño de sus extensiones físicas (**PE**) es **4 MiB**.

El tercer volumen físico, el disco entero `/dev/sdc`, todavía no es miembro de ningún grupo de volúmenes, de momento no se puede explotar.

c. Supresión de un volumen físico

El comando `pvremove` permite suprimir uno o distintos volúmenes físicos LVM.

Sintaxis

```
pvremove [ Opciones ] Volfísico [...]
```

Parámetros principales

`Opciones`

Opciones.

`Volfísico`

Archivos especiales del volumen o volúmenes físicos.

`[...]`

Descripción

El comando suprime las estructuras de gestión LVM del volumen o de los volúmenes físicos especificados. Se perderán los datos.

Para que el comando se ejecute correctamente, ningún volumen lógico debe estar asociado al volumen físico, y tampoco tiene que pertenecer a un grupo de volúmenes (ver el comando `vgreduce`).



Las opciones `-f` y `-ff` permiten forzar la supresión de un volumen físico.

5. Los grupos de volúmenes

Los grupos de volúmenes permiten establecer un enlace entre los volúmenes físicos y los volúmenes lógicos. Deben contener al menos un volumen físico. Se les puede añadir dinámicamente volúmenes físicos, o retirarlos a condición de que el volumen físico no sea asignado y que al menos quede un volumen físico en el grupo.

Se podrán crear o ampliar volúmenes lógicos en un grupo de volúmenes mientras queden extensiones físicas disponibles.

a. Creación de un grupo de volúmenes

Sintaxis

vgcreate [Opciones] NombreGrp Volfísico [Volfísico ...]

Parámetros principales

| | |
|-----------|---|
| Opciones | Opciones. |
| Volfísico | Volumen o volúmenes físicos que se incluirán en el grupo. |

Descripción

Comando para crear un grupo de volúmenes `NombreGrp`, que tiene, al menos, un volumen físico `Volfísico`.



Si el o los espacios de almacenamiento `Volfísico` no son volúmenes físicos, el comando propondrá inicializarlos como volúmenes físicos.

El gestor LVM mantiene una tabla de extensiones físicas y una tabla de extensiones lógicas para el grupo de volúmenes creados. Podrá, de esta manera, asignar extensiones lógicas a cada volumen lógico, que estarán asociadas a extensiones físicas (una extensión lógica puede corresponder a una o a distintas extensiones físicas).

Numerosas opciones, además de las opciones comunes LVM, permiten especificar las características técnicas deseadas para el grupo de volúmenes. Entre ellas:

| | | |
|--------------------------------------|---------------------|-------------------------------------|
| <code>-l --maxlogicalvolumes</code> | <code>Número</code> | Número máximo de volúmenes lógicos. |
| <code>-p --maxphysicalvolumes</code> | <code>Número</code> | Número máximo de volúmenes físicos. |
| <code>-s --physicalextentsize</code> | <code>Tamaño</code> | Tamaño de una extensión física. |

Ejemplo

Creación de un grupo de volúmenes con un volumen físico:

```
vgcreate miaplicacion /dev/sdc
```

```
Volume group "miaplicacion" successfully created
```

Comprobamos la creación del grupo de volúmenes `miaplicacion`:

```
vgs miaplicacion
```

```

VG      #PV #LV #SN Attr  VSize VFree
miaplicacion 1  0  0 wz--n- 7,54g 7,54g

```

Se muestran las características del volumen físico:

```
pvdisk /dev/sdc
```

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

```
PV Name      /dev/sdc
```

```
VG Name      miaplicacion
```

```
PV Size      7,54 GiB / not usable 0
```

```
Allocatable  yes
```

```
PE Size      4,00 MiB
```

```
Total PE     1931
```


| | |
|--------------|--|
| Free PE | 1931 |
| Allocated PE | 0 |
| PV UUID | J8UKBL-IO35-EUWj-XD4F-8Hc2-aJBv-jC5VPH |

Forma parte del grupo de volúmenes, con un tamaño de extensiones físicas de 4 MiB, todas disponibles.

b. Datos de los grupos de volúmenes

El comando `vgdisplay` muestra las características de uno o distintos grupos de volúmenes LVM.

Sintaxis

```
vgdisplay [ Opciones ] [ NombreGrp ... ]
```

Parámetros principales

| | |
|-----------|--------------------------|
| Opciones | Opciones. |
| NombreGrp | Grupo(s) de volumen(es). |

Descripción

El comando muestra las características de los grupos de volúmenes especificados, o de todos si no se han utilizado argumentos `NombreGrp`. El comando acepta las opciones generales de los comandos LVM, y de otros más específicos con respecto a los grupos de volúmenes.

Ejemplo

Datos del grupo de volúmenes creado en el ejemplo anterior:

```

vgdisplay miaplicacion
--- Volume group ---
VG Name          miaplicacion
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         7,54 GiB
PE Size         4,00 MiB
Total PE        1931
Alloc PE / Size 0 / 0
Free PE / Size  1931 / 7,54 GiB
VG UUID         9PP2tD-rK42-nYhh-PqvL-ic2b-5iLC-IYbGTK

```

El campo **MAX LV** con un valor de 0 significa que no hay número máximo de volúmenes configurados, igual que para los volúmenes físicos con **Max PV**.

c. Incorporación/retiro de un volumen físico

Uno de los principales intereses de la gestión LVM del espacio de disco reside en su capacidad evolutiva. Así, se puede añadir dinámicamente un nuevo volumen físico a un grupo de volúmenes existente, lo que nos permitirá crear nuevos volúmenes lógicos o agrandar los ya existentes.

Sintaxis

```
vgextend [ Opciones ] NombreGrp Volfísico [...]
```

Parámetros principales

| | |
|------------------|---|
| Opciones | Opciones. |
| NombreGrp | Grupo de volúmenes que se van a extender. |
| Vol físico [...] | Volumen(es) físico(s) que se incluirán. |

Descripción

El comando añade el/los volumen(es) físico(s) en el grupo de volúmenes, inicializándolos si no han sido creados por el comando `pvcreate`.

Ejemplo

Incorporación de un volumen físico en el grupo de volúmenes creado en el ejemplo anterior:

vgextend miaplicacion /dev/sdd

Physical volume **"/dev/sdd"** successfully created.

Volume **group "miaplicacion"** successfully extended

Como el disco no era un volumen físico, el comando lo ha inicializado como volumen físico y lo ha añadido al grupo de volúmenes.

Se comprueba el nuevo estado del grupo de volúmenes:

vgdisplay miaplicacion

--- Volume group ---

VG **Name** miaplicacion

System ID

Format lvm2

Metadata Areas 2

Metadata Sequence No 2

VG Access **read/write**

VG Status resizable

MAX LV 0

Cur LV 0

```

Open LV      0
Max PV       0
Cur PV      2
Act PV       2
VG Size      22,00 GiB
PE Size      4,00 MiB
Total PE     5632
Alloc PE / Size  0 / 0
Free PE / Size  5632 / 22,00 GiB
VG UUID      9PP2tD-rK42-nYhh-PqvL-ic2b-5iLC-IYbGTK

```

El volumen contiene dos volúmenes físicos. Se pueden mostrar las características del nuevo volumen físico `/dev/sdd`:

```

pvdisplay /dev/sdd
--- Physical volume ---
PV Name      /dev/sdd
VG Name      miaplicacion
PV Size      <14,46 GiB / not usable 2,00 MiB
Allocatable  yes
PE Size      4,00 MiB
Total PE     3701
Free PE      3701
Allocated PE  0
PV UUID      UfR1dg-iURM-wsoX-MhFo-3C2U-HmtV-F37zNo

```

Ha sido creado correctamente.

Por otra parte, si un volumen físico se considera inutilizado, podemos retirarlo del grupo de volúmenes, por ejemplo, para incluirlo en otro grupo.

Sintaxis

```
vgreduce [ Opciones ] NombreGrp Volfísico [...]
```

Parámetros principales

| | |
|------------------|---|
| Opciones | Opciones. |
| NombreGrp | Grupo de volúmenes. |
| Vol físico [...] | Volumen(es) físico(s) que se retirará(n). |

Descripción

El comando retira el/los volumen(es) físico(s) del grupo de volúmenes, a condición de que ninguna extensión física no haya sido asignada a un volumen lógico.

Ejemplo

Retirada de un volumen físico del grupo de volúmenes del ejemplo anterior:

```
vgreduce miaplicacion /dev/sdd
Removed "/dev/sdd" from volume group "miaplicacion"
```

Se comprueba el nuevo estado del grupo de volúmenes:

```
vgdisplay miaplicacion
--- Volume group ---
VG Name          miaplicacion
System ID
Format           lvm2
Metadata Areas    1
Metadata Sequence No 3
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     7,54 GiB
PE Size     4,00 MiB
Total PE    1931
Alloc PE / Size  0 / 0
Free PE / Size  1931 / 7,54 GiB
VG UUID     9PP2tD-rK42-nYhh-PqvL-ic2b-5iLC-IYbGTK

```

El volumen físico ha sido retirado del grupo de volúmenes. Podemos mostrar las características para asegurarnos:

```

pvdisplay /dev/sdd
"/dev/sdd" is a new physical volume of "<14,46 GiB"
--- NEW Physical volume ---
PV Name      /dev/sdd
VG Name
PV Size      <14,46 GiB
Allocatable   NO
PE Size       0
Total PE      0
Free PE       0
Allocated PE   0
PV UUID      UfR1dg-iURM-wsoX-MhFo-3C2U-HmtV-F37zNo

```

d. Supresión de grupos de volúmenes

El comando `vgremove` permite suprimir uno o distintos grupos de volúmenes LVM.

Sintaxis

```
vgremove [ Opciones ] NombreGrp [...]
```

Parámetros principales

| | |
|---------------------|--|
| Opciones | Opciones. |
| NombreGrp [. . .] | Nombre del o de los grupos de volúmenes. |

Descripción

El comando suprime uno o distintos grupos de volúmenes. Si contienen todavía volúmenes lógicos, el comando solicita una confirmación.

6. Los volúmenes lógicos

Un volumen lógico está compuesto por un conjunto de extensiones lógicas (LE, *logical extent*) asignado a un grupo de volúmenes. El nombre del volumen es libre.

Un volumen lógico se utiliza como los otros espacios de almacenamiento. Se puede usar como un espacio de swap, en modo bruto (*raw*), o crear en él un sistema de archivos. Una vez montado en la arborescencia global del sistema de archivos Linux, este último se encuentra a disposición de las aplicaciones.

Los volúmenes lógicos dan mucha flexibilidad y capacidad evolutiva a la gestión del almacenamiento. Proporcionan una capa lógica suplementaria entre las aplicaciones y los dispositivos de almacenamiento. Se pueden extender dinámicamente, incluso en distintos discos.

a. Creación de un volumen lógico

El comando `lvcreate` permite crear un volumen lógico dentro de un grupo de volúmenes.

Sintaxis

`lvcreate [Opciones] [-n NombreVol] NombreGrp [Volfísico ... Volfísico]`

Parámetros principales

| | |
|-----------------------------|--|
| Opciones | Opciones. |
| [-n NombreVol] | Nombre del volumen lógico (opcional). |
| NombreGrp | Nombre del grupo de volúmenes. |
| [Volfísico ... Volfísico] | Volumen(es) físico(s) que serán usados para el volumen lógico. |

Descripción

El comando crea un volumen lógico en el grupo de volúmenes `NombreGrp`. Si no se especifica ningún nombre, el comando utilizará un nombre por defecto: `lv01x`, donde `x` es un valor entero que todavía no ha sido usado en el grupo de volúmenes.

El gestor LVM asignará las extensiones lógicas asociándolas a extensiones físicas. Por defecto, puede usar cualquier volumen físico del grupo de volúmenes, pero se pueden especificar los volúmenes físicos que se utilizarán (argumento(s) al final de la línea de `Volfísico ... Volfísico`). Algunas opciones fuerzan el uso de varios volúmenes físicos.

Las numerosas opciones, aparte de las opciones comunes de LVM, permiten especificar las características técnicas deseadas para el volumen lógico. Entre ellas:

| | |
|---|---|
| <code>-L --size Tamaño[k m g]</code> | Tamaño del volumen (en Kb, Mb o Gb). |
| <code>-m --mirrors NúmeroEspejo</code> | Número de espejos además del volumen original. |
| <code>-i --stripes NúmeroVolfísico</code> | Número de volúmenes físicos para las bandas del volumen lógico. |
| <code>--type mirror raid[0 1 5]</code> | Tipo de volumen lógico: espejo, RAID 0, 1, 5. |

Con:

`-L|--size Tamaño [k|m|g]`

Se trata del tamaño útil del volumen lógico, en KB, MB o GB (por defecto, en MB). Si el volumen lógico está en RAID1 o RAID5, el tamaño realmente usado será más importante.

`-m|--mirrors NúmeroEspejo`

El volumen lógico se duplicará en `NúmeroEspejo + 1` ejemplares.

`-i|--stripes NúmeroVolfísico`

El volumen lógico está repartido en bandas de datos en `NúmeroVolfísico` volúmenes físicos.

`--type mirror|raid[0|1|5]`

El volumen lógico es gestionado por LVM en espejo o en RAID de nivel 0, 1 o 5.



El gestor LVM administra otros niveles de RAID, poco habituales.

Ejemplo

Creación de tres volúmenes lógicos en un grupo de volúmenes:

lvcreate -L 100 -n bin miaplicacion

Logical volume "bin" created.

lvcreate -L 1000k -n tmp miaplicacion

Rounding up size to full physical extent 4,00 MiB

Logical volume "tmp" created.

lvcreate -L 1G -n datos miaplicacion

Logical volume "datos" created.

Se comprueba la creación de los volúmenes lógicos:

lvs miaplicacion

| LV | VG | Attr | LSize | Pool | Origin | Data% | Meta% | Move | Log | Cpy% | Sync | Convert |
|-------|--------------|------------|---------|------|--------|-------|-------|------|-----|------|------|---------|
| bin | miaplicacion | -wi-a----- | 100,00m | | | | | | | | | |
| datos | miaplicacion | -wi-a----- | 1,00g | | | | | | | | | |
| tmp | miaplicacion | -wi-a----- | 4,00m | | | | | | | | | |

Se comprueba el grupo de volúmenes:

vgdisplay miaplicacion

--- Volume group ---

VG Name miaplicacion

System ID

Format lvm2

Metadata Areas 1

Metadata Sequence No 6

VG Access read/write

VG Status resizable

```

MAX LV      0
Cur LV     3
Open LV     0
Max PV      0
Cur PV     1
Act PV      1
VG Size     7,54 GiB
PE Size     4,00 MiB
Total PE    1931
Alloc PE / Size  282 / 1,10 GiB
Free PE / Size  1649 / 6,44 GiB
VG UUID     9PP2tD-rK42-nYhh-PqvL-ic2b-5iLC-IYbGTK

```

El grupo de volúmenes tiene tres volúmenes lógicos.

b. Información acerca de los volúmenes lógicos

El comando `lvdisplay` muestra las características de uno o distintos volúmenes lógicos LVM.

Sintaxis

```
lvdisplay [ Opciones ] [ NombreGrp[/NombreLV] ... ]
```

Parámetros principales

| | |
|-----------------------|--|
| Opciones | Opciones. |
| NombreGrp[/NombreLV] | Grupo(s) de volumen(es) o volúmenes lógicos. |

Descripción

El comando muestra las características de los volúmenes lógicos especificados, o de los volúmenes lógicos de los grupos de volúmenes especificados o de todos los volúmenes

lógicos si no hay argumento de tipo `NombreGrp`. El comando acepta las opciones generales de los comandos LVM, y otras más específicas de los volúmenes lógicos.

Ejemplo

Información acerca de los volúmenes lógicos creados en el ejemplo anterior:

lvdisplay miaplicacion

--- Logical volume ---

```
LV Path          /dev/miaplicacion/bin
LV Name          bin
VG Name          miaplicacion
LV UUID          JDiEAV-Bv6U-6oI2-104J-5JiN-6hOy-acv8KT
LV Write Access   read/write
LV Creation host, time beta, 2020-03-18 20:12:31 +0000
LV Status         available
# open           0
LV Size          100,00 MiB
Current LE       25
Segments         1
Allocation        inherit
Read ahead sectors auto
- currently set to 8192
Block device     253:6
```

--- Logical volume ---

```
LV Path          /dev/miaplicacion/tmp
LV Name          tmp
VG Name          miaplicacion
LV UUID          YWlJuc-RQOK-6CCa-Yw1G-rmjb-oMs0-VHABpD
LV Write Access   read/write
LV Creation host, time beta, 2020-03-18 20:13:20 +0000
LV Status         available
# open           0
LV Size          4,00 MiB
Current LE       1
Segments         1
Allocation        inherit
Read ahead sectors auto
- currently set to 256
```

```

Block device      253:7

--- Logical volume ---
LV Path           /dev/miaplicacion/datos
LV Name           datos
VG Name           miaplicacion
LV UUID           KPHekC-el70-Jmkl-W6OR-H0LZ-0rOL-wlQUGT
LV Write Access    read/write
LV Creation host, time beta, 2020-03-18 20:14:04 +0000
LV Status          available
# open            0
LV Size           1,00 GiB
Current LE        256
Segments          1
Allocation         inherit
Read ahead sectors auto
- currently set to 256
Block device      253:8

```

Se observa que los volúmenes lógicos pueden ser designados por los caminos de acceso:

`/dev/miaplicacion/bin` , `/dev/miaplicacion/tmp` y `/dev/miaplicacion/datos` .

Cuando se crea el primer volumen lógico, el gestor LVM crea un directorio con el nombre del grupo de volúmenes, en el directorio `/dev` . Dentro de ese directorio crea un enlace simbólico por cada volumen lógico, con el nombre del volumen lógico. Se puede, por lo tanto, indicar un volumen lógico mediante el camino de acceso siguiente:

`/dev/NombreGrupo/NombreVol`

El enlace simbólico apunta a un archivo especial en modo de bloque cuyo camino de acceso tiene la forma:

`/dev/dm-N`

donde **N** es un número entero.

También nos podemos referir a los volúmenes lógicos mediante otro camino, bajo la forma:

`/dev/mapper/NombreGrupo-NombreVol`

`NombreGrupo-NombreVol` es un enlace simbólico que apunta también hacia un archivo especial en modo de bloque, que ya se ha visto anteriormente.

Ejemplo

He aquí los distintos directorios, archivos especiales y enlaces simbólicos después de la creación de los tres volúmenes lógicos del ejemplo anterior:

ls -ld /dev/miaplicacion

```
drwxr-xr-x. 2 root root 100 18 marzo 20:14 /dev/miaplicacion
```

ls -ld /dev/miaplicacion/*

```
lrwxrwxrwx. 1 root root 7 18 marzo 20:12 /dev/miaplicacion/bin -> ../dm-6
```

```
lrwxrwxrwx. 1 root root 7 18 marzo 20:14 /dev/miaplicacion/datos -> ../dm-8
```

```
lrwxrwxrwx. 1 root root 7 18 marzo 20:13 /dev/miaplicacion/tmp -> ../dm-7
```

ls -l /dev/mapper/miaplicacion-*

```
lrwxrwxrwx. 1 root root 7 18 marzo 20:12 /dev/mapper/miaplicacion-bin -> ../dm-6
```

```
lrwxrwxrwx. 1 root root 7 18 marzo 20:14 /dev/mapper/miaplicacion-datos -> ../dm-8
```

```
lrwxrwxrwx. 1 root root 7 18 marzo 20:13 /dev/mapper/miaplicacion-tmp -> ../dm-7
```

ls -l /dev/dm-[678]

```
brw-rw----. 1 root disk 253, 6 18 marzo 20:12 /dev/dm-6
```

```
brw-rw----. 1 root disk 253, 7 18 marzo 20:13 /dev/dm-7
```

```
brw-rw----. 1 root disk 253, 8 18 marzo 20:14 /dev/dm-8
```

c. Extensión de un volumen lógico

Se puede aumentar dinámicamente el tamaño de un volumen lógico mientras no haya extensiones físicas disponibles en los volúmenes físicos de su grupo de volúmenes.

Sintaxis

```
lvextend -L[+]Tamaño [-r|--resizefs] VolLog [ Volfísico ... ]
```

Parámetros principales

| | |
|---------------------------------|---|
| <code>-L[+]Tamaño[k m g]</code> | Tamaño final o que se tendrá que añadir. |
| <code>-r --resizefs</code> | Extiende el sistema de archivos del volumen lógico. |
| <code>VolLog</code> | Volumen lógico que se extenderá. |
| <code>[Volfísico ...]</code> | Volumen(es) físico(s) que será(n) usado(s) para llevar a cabo la extensión. |

Descripción

El comando aumenta el tamaño del volumen lógico, hasta el nuevo tamaño especificado con la opción `-L`, o el número de bytes especificado con la opción `-L+`.

El volumen lógico es un espacio de almacenamiento. Si contiene un sistema de archivos, habrá que extender este último para que pueda usar el nuevo espacio disponible. Si se usa la opción `-r` o `--resizefs`, el sistema de archivos se extiende a la vez que el volumen lógico.

Se puede especificar uno o distintos volúmenes físicos que se usarán para realizar la extensión.

Ejemplo

Se extiende a 100 MB el tamaño del volumen lógico `tmp` del grupo de volúmenes `miaplicacion`:

```
lvextend -L 100m miaplicacion/tmp
```

```
Size of logical volume miaplicacion/tmp changed from 4,00 MiB (1 extents)
```

to 100,00 MiB (25 extents).

Logical volume miaplicacion/tmp successfully resized.

Comprobamos:

lvdisplay miaplicacion/tmp

```

--- Logical volume ---
LV Path                /dev/miaplicacion/tmp
LV Name                tmp
VG Name                miaplicacion
LV UUID                YWlJuc-RQOK-6CCa-Yw1G-rmjb-oMs0-VHABpD
LV Write Access        read/write
LV Creation host, time beta, 2020-03-18 20:13:20 +0000
LV Status              available
# open                 0
LV Size                100,00 MiB
Current LE             25
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:7

```

Ahora el volumen lógico tiene un tamaño de 100 MB.

d. Reducción de un volumen lógico

También se puede disminuir dinámicamente el tamaño de un volumen lógico, con precaución, porque existe el riesgo de perder datos.

Sintaxis

```
lvreduce -L[-]Tamaño [-r|--resizefs] VolLog
```

Parámetros principales

| | |
|-----------------------------------|---|
| <code>-L[-]Tamaño[k m g]</code> | Tamaño final o que tendrá que disminuirse. |
| <code>-r --resizefs</code> | Reduce el sistema de archivos del volumen lógico. |
| <code>VolLog</code> | Volumen lógico que se quiere reducir. |

Descripción

El comando disminuye el tamaño del volumen lógico, hasta el nuevo tamaño especificado con la opción `-L`, o el número de bytes especificado con la opción `-L-`.

El volumen lógico es un espacio de almacenamiento. Si contiene un sistema de archivos habrá que reducirlo previamente para evitar perder datos. Si se usa la opción `-r` o `--resizefs`, el sistema de archivos será reducido a la vez que el volumen lógico.

Ejemplo

Se reduce de 50 MB el tamaño del volumen lógico `tmp` del grupo de volúmenes `miaplicacion`:

```
lvreduce -L 50m miaplicacion/tmp
```

```
Rounding size to boundary between physical extents: 52,00 MiB.
```

```
WARNING: Reducing active logical volume to 52,00 MiB.
```

```
THIS MAY DESTROY YOUR DATA (filesystem etc.)
```

```
Do you really want to reduce miaplicacion/tmp? [y/n]: y
```

```
Size of logical volume miaplicacion/tmp changed from 100,00 MiB (25 extents)
to 52,00 MiB (13 extents).
```

```
Logical volume miaplicacion/tmp successfully resized.
```

Ya que el volumen lógico no estaba utilizado completamente, no había ningún riesgo de pérdida de datos. Se comprueba el nuevo estado del volumen:

```
lvdisplay miaplicacion/tmp
```

```

--- Logical volume ---
LV Path          /dev/miaplicacion/tmp
LV Name          tmp
VG Name          miaplicacion
LV UUID          YWIJuc-RQOK-6CCa-Yw1G-rmjb-oMs0-VHABpD
LV Write Access   read/write
LV Creation host, time beta, 2020-03-18 20:13:20 +0000
LV Status         available
# open           0
LV Size          52,00 MiB
Current LE       13
Segments         2
Allocation        inherit
Read ahead sectors    auto
- currently set to 256
Block device      253:7

```

e. Supresión de volúmenes lógicos

El comando `lvremove` permite suprimir un volumen lógico de un grupo de volúmenes LVM.

Sintaxis

```
lvremove [ Opciones ] NombreVol [...]
```

Parámetros principales

| | |
|-----------|--|
| Opciones | Opciones. |
| NombreVol | Nombre del o de los volúmenes lógicos. |
| [...] | |

Descripción

El comando suprime uno o distintos volúmenes lógicos. Su sistema de archivos no tiene que estar montado. Los datos se perderán.

f. Volúmenes lógicos y sistema de archivos

Los volúmenes lógicos son espacios de almacenamiento, exactamente igual que los discos duros, las particiones o los volúmenes RAID administrados por `md`. Se usan de la misma manera, para almacenar directorios y archivos, como espacio de swap o como espacio bruto gestionado directamente por una aplicación (modo *raw*).

La mayoría de las veces se les usa creando un sistema de archivos, que se monta después en la arborescencia para ponerlo a disposición de las aplicaciones.

Para designar un volumen lógico en un comando que espera como argumento la ubicación de un espacio de almacenamiento (`mkfs` por ejemplo), se pueden especificar diferentes caminos de acceso:

- `/dev/NombreGrupo/NombreVol`

Es el camino de acceso más explícito, porque el administrador lo puede asignar al grupo de volúmenes y al volumen lógico de los nombres que corresponden a su uso.

- `/dev/mapper/NombreGrupo-NombreVol`

Este camino de acceso diferente también es relativamente explícito, ya que usa el nombre del grupo de volúmenes y el del volumen lógico, elegidos por el administrador. Si el nombre de grupo o volumen comporta un guion, hay que poner un guion doble para no confundirlo con el guion de separación.

- `/dev/dm-N`

Es el camino de acceso del archivo especial en modo de bloque asociado al volumen lógico, `N` es un número entero único. Los otros caminos de acceso son enlaces simbólicos hacia ese archivo.

Ejemplo

Se crea un sistema de archivos `ext4` en el volumen lógico `bin` del grupo de volúmenes

`miaplicacion` . Se monta manualmente y se crean directorios y archivos:

```
ls -l /dev/miaplicacion/bin /dev/mapper/miaplicacion-bin
```

```
lrwxrwxrwx. 1 root root 7 19 marzo 17:25 /dev/mapper/miaplicacion-bin -> ../dm-6
```

```
lrwxrwxrwx. 1 root root 7 19 marzo 17:25 /dev/miaplicacion/bin -> ../dm-6
```

El volumen lógico puede ser designado como `/dev/miaplicacion/bin` , `/dev/mapper/miaplicacion-bin` o `/dev/dm-6` .

```
mkfs -t ext4 /dev/miaplicacion/bin
```

```
mke2fs 1.44.6 (5-Mar-2019)
```

```
Creating filesystem with 102400 1k blocks and 25688 inodes
```

```
Filesystem UUID:7efe87dd-bbef-49fd-854b-95300e5ba5ce
```

```
Superblock backups stored on blocks:
```

```
8193, 24577, 40961, 57345, 73729
```

```
Allocating group tables: done
```

```
Writing inode tables: done
```

```
Creating journal (4096 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

El sistema de archivos creado ocupa todo el espacio del volumen lógico. Se va a montar en un directorio usando uno de los caminos de acceso del volumen lógico:

```
mkdir /opt/application1
```

```
mount /dev/mapper/miaplicacion-bin /opt/application1
```

El sistema de archivos del volumen lógico está montado. Se crea en él una arborescencia:

```
cd /opt/application1
```

```
mkdir exe lib tools
```

```
cp /usr/bin/vim exe/miaplicacion
```

```
cp /usr/bin/z* tools
```

```
cp /usr/lib64/sss/lib* lib/
```

Se comprueba el porcentaje de ocupación del sistema de archivos:

```
df .
S.ficheros      bloques de 1K  Usados Disponibles Uso% Montado en
/dev/mapper/miaplicacion-bin  95054  9942   77944 12% /opt/application1
```

El comando `df` designa el volumen lógico usando otro de sus caminos de acceso.

Considerando que el volumen lógico es demasiado grande con respecto a lo que se necesitaría, se decide reducirlo, reduciendo también el sistema de archivos que contiene:

```
lvreduce -L 50m --resizefs miaplicacion/bin
Rounding size to boundary between physical extents: 52,00 MiB.
Do you want to unmount "/opt/application1"? [Y|n] n
fsadm: Cannot proceed with mounted filesystem "/opt/application1".
/usr/sbin/fsadm failed: 1
Filesystem resize failed.
```

El comando falla, porque el sistema de archivos está montado. Lo desmontamos y comenzamos de nuevo:

```
cd
umount /dev/miaplicacion/bin
lvreduce -L 50m --resizefs miaplicacion/bin
Rounding size to boundary between physical extents: 52,00 MiB.
fsck de util-linux 2.32.1
/dev/mapper/miaplicacion-bin: 55/25688 files (1.8% non-contiguous),
17288/102400 blocks
resize2fs 1.44.6 (5-Mar-2019)
Resizing the filesystem on /dev/mapper/miaplicacion-bin to 53248 (1k) blocks.
The filesystem on /dev/mapper/miaplicacion-bin is now 53248 (1k) blocks long.

Size of logical volume miaplicacion/bin changed from 100,00 MiB (25 extents)
to 52,00 MiB (13 extents).
Logical volume miaplicacion/bin successfully resized.
```

El volumen lógico ha sido reducido y el sistema de archivos también, se comprueba el sistema de archivos:

```
e2fsck /dev/miaplicacion/bin
e2fsck 1.44.6 (5-Mar-2019)
/dev/miaplicacion/bin: clean, 55/13832 files, 15278/53248 blocks
```

El sistema de archivos está limpio, se vuelve a montar:

```
mount /dev/mapper/miaplicacion-bin /opt/application1
df /opt/application1
S.ficheros      bloques de 1K  Usados  Disponibles  Uso%  Montado en
/dev/mapper/miaplicacion-bin  47400   9430   34244   22% /opt/application1
```

Se ha reducido correctamente el tamaño del sistema de archivos.

Se rellena completamente el sistema de archivos:

```
for nb in 1 2 3 4 5 6 7 8 9 10 11 12 13 14
do
cp /opt/application1/exe/miaplicacion /opt/application1/tools/herramienta$nb done
cp: error al escribir en '/opt/application1/tools/herramienta5': No queda espacio en
el dispositivo
cp: error al escribir en '/opt/application1/tools/herramienta6': No queda espacio en
el dispositivo
cp: error al escribir en '/opt/application1/tools/herramienta7': No queda espacio en
el dispositivo
cp: error al escribir en '/opt/application1/tools/herramienta8': No queda espacio en
el dispositivo
cp: error al escribir en '/opt/application1/tools/herramienta9': No queda espacio en
el dispositivo
cp: error al escribir en '/opt/application1/tools/herramienta10': No queda espacio en
el dispositivo
cp: error al escribir en '/opt/application1/tools/herramienta11': No queda espacio en
el dispositivo
cp: error al escribir en '/opt/application1/tools/herramienta12': No queda espacio en
```

el dispositivo

cp: **error** al escribir en '/opt/application1/tools/herramienta13': **No** queda espacio en el dispositivo

cp: **error** al escribir en '/opt/application1/tools/herramienta14': **No** queda espacio en el dispositivo

df /opt/application1

| S.ficheros | bloques de 1K | Usados | Disponibles | Uso% | Montado en |
|------------------------------|---------------|--------|-------------|------|-------------------|
| /dev/mapper/miaplicacion-bin | 47400 | 46336 | 0 | 100% | /opt/application1 |

El sistema de archivos está ocupado al 100 %, las últimas copias han fallado. Vamos a intentar ampliar el volumen lógico y el sistema de archivos, sin desmontarlo:

lvextend -L 100m --resizefs /dev/miaplicacion/bin

Size of logical volume miaplicacion/bin changed from 52,00 MiB (13 extents) to 100,00 MiB (25 extents).

Logical volume miaplicacion/bin successfully resized.

resize2fs 1.44.5 (15-Dec-2018)

Filesystem at /dev/mapper/miaplicacion-bin is mounted on /opt/application1;

on-line resizing required

old_desc_blocks = 1, new_desc_blocks = 1

The filesystem on /dev/mapper/miaplicacion-bin is now 102400 (1k) blocks long.

Comprobamos:

lvdisplay miaplicacion/bin

--- Logical volume ---

LV Path /dev/miaplicacion/bin

LV Name bin

VG Name miaplicacion

LV UUID JDiEAV-Bv6U-6oI2-104J-5JiN-6hOy-acv8KT

LV Write Access read/write

LV Creation host, time beta, 2020-03-18 20:12:31 +0000

LV Status available

open 1

LV Size 100,00 MiB

```

Current LE      25
Segments       1
Allocation      inherit
Read ahead sectors  auto
- currently set to 8192
Block device    253:6

```

El volumen lógico tiene 100 Mb.

```

df /opt/application1
S.ficheros      bloques de 1K Usados Disponibles Uso% Montado en
/dev/mapper/miaplicacion-bin  95054 46848  42678 53% /opt/application1

```

Se ha podido ampliar el sistema de archivos, sin tener que desmontarlo. Se pueden terminar las copias anteriores:

```

for nb in 11 12 13 14
do cp /opt/application1/exe/miaplicacion /opt/application1/tools/
herramienta$nb
done
ls -l /opt/application1/tools/ut*
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta1
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta10
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:27 /opt/application1/tools/
herramienta11
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:27 /opt/application1/tools/
herramienta12
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:27 /opt/application1/tools/
herramienta13
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:27 /opt/application1/tools/
herramienta14
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta2
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/

```



```

herramienta3
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta4
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta5
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta6
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta7
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta8
-rwxr-xr-x. 1 root root 3522560 19 marzo 19:14 /opt/application1/tools/
herramienta9

```

La copia se ha efectuado correctamente.

7. Instantánea LVM (snapshot)

Una instantánea (*snapshot*) LVM es un volumen lógico que permite mantener una copia exacta del estado de otro volumen lógico en el momento de la creación de la instantánea.

Al principio, el volumen lógico instantáneo no contiene casi ningún dato. El gestor LVM registrará en este volumen lógico, poco a poco, las modificaciones efectuadas en el volumen lógico destino. De esta manera, siempre será posible reconstituir el estado inicial del volumen lógico a partir de su contenido y del diferencial almacenado en el volumen instantáneo.

Se puede usar un volumen lógico instantáneo para respaldar el estado de un volumen lógico. El volumen lógico instantáneo se usa como un volumen lógico clásico, se puede montar en la arborescencia global del sistema de archivos de Linux, donde aparecerá como un conjunto de directorios y de archivos. Se podrá usar como respaldo, gracias a las herramientas clásicas de respaldo, permitiendo continuar la explotación del volumen lógico original durante este respaldo, mientras se optimiza el espacio de disco consumido.

a. Creación de un volumen lógico instantáneo

Para crear un volumen lógico instantáneo, se usa la opción `-s` del comando `lvcreate`:

Sintaxis

```
lvcreate -L Tamaño -s -n NombreVolInst VolOriginal
```

Parámetros principales

| | |
|-------------------------------|--|
| <code>-L Tamaño[k m g]</code> | Tamaño del volumen lógico instantáneo. |
| <code>-s</code> | Creación de una instantánea (<i>snapshot</i>). |
| <code>-n NombreVolInst</code> | Nombre del volumen lógico instantáneo. |
| <code>VolOriginal</code> | Volumen lógico original. |

Descripción

El comando crea un volumen lógico instantáneo `NombreVolInst`, de un tamaño `Tamaño`, "copia" del volumen lógico `VolOriginal`.

Ejemplo de uso de un volumen lógico instantáneo

Se crea un volumen lógico instantáneo:

```
lvcreate -L 20m -s -n miaplicacion/bin-snap miaplicacion/bin  
Logical volume "bin-snap" created.
```

Se muestran las características del volumen lógico instantáneo:

```

lvdisplay miaplicacion/bin-snap
--- Logical volume ---
LV Path                /dev/miaplicacion/bin-snap
LV Name                bin-snap
VG Name                miaplicacion
LV UUID                8QBeHW-ZtLR-YMiW-9Gsl-42wi-gW10-eyCGta
LV Write Access        read/write
LV Creation host, time beta, 2020-03-19 19:54:55 +0000
LV snapshot status     active destination for bin
LV Status               available
# open                 0
LV Size                100,00 MiB
Current LE             25
COW-table size         20,00 MiB
COW-table LE           5
Allocated to snapshot  0,06%
Snapshot chunk size     4,00 KiB
Segments               1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:11

```

El volumen lógico instantáneo es una imagen del volumen lógico `bin`, consume 0,06 % del tamaño de `bin`. Su tamaño (`COW-table size`, Copy On Write) es 20 MB.

Se pueden modificar distintos objetos en el sistema de archivos del volumen lógico original, y después mostrar las características del volumen lógico instantáneo:

```

rm -f /opt/application1/tools/z*
cp /usr/bin/d* /opt/application1/tools
echo "Read the Fine Manual" > /opt/application1/lib/HOWTO
lvdisplay miaplicacion/bin-snap
--- Logical volume ---
LV Path                /dev/miaplicacion/bin-snap
LV Name                bin-snap
VG Name                miaplicacion
LV UUID                8QBeHW-ZtLR-YMiW-9Gsl-42wi-gW10-eyCGta

```

```

LV Write Access    read/write
LV Creation host, time beta, 2020-03-19 19:54:55 +0000
LV snapshot status active destination for bin
LV Status          available
# open            0
LV Size            100,00 MiB
Current LE         25
COW-table size     20,00 MiB
COW-table LE       5
Allocated to snapshot 39,26%
Snapshot chunk size 4,00 KiB
Segments          1
Allocation         inherit
Read ahead sectors auto
- currently set to 256
Block device       253:11

```

La ocupación del volumen lógico instantáneo ha aumentado.

Se monta el volumen lógico instantáneo y se observa su contenido:

```

mount /dev/miaplicacion/bin-snap /mnt
ls /mnt
exe lib lost+found tools
ls /mnt/tools/z*
/mnt/tools/zcat /mnt/tools/zfgrep /mnt/tools/zipgrep /mnt/tools/zmore
/mnt/tools/zcmp /mnt/tools/zforce /mnt/tools/zipinfo /mnt/tools/znew
/mnt/tools/zdiff /mnt/tools/zgrep /mnt/tools/zipnote /mnt/tools/zsoelim
/mnt/tools/zegrep /mnt/tools/zip /mnt/tools/zipsplit
/mnt/tools/zenity /mnt/tools/zipcloak /mnt/tools/zless
ls /mnt/lib/HOWTO
ls: no se puede acceder a '/mnt/lib/HOWTO': No existe el fichero o el directorio
du -ks /mnt
56623 /mnt

```

Las modificaciones efectuadas en el volumen lógico original no han sido efectuadas en el volumen lógico instantáneo. Una vez montado, corresponde a 56 623 KB de datos, mientras

que el tamaño del volumen lógico solamente es 20 MB.