

Un pingüino en mi servidor

Historias de un administrador de sistemas GNU/Linux

LVM para torpes (II)

Si [mis dibujos](#) no te dejaron muy claro los conceptos básicos del LVM no te preocupes, ahora lo explicaré desde el principio con los comandos de gestión del LVM. Seguro que quedará más claro.

Empezar de cero a montar LVM en un sistema ya en marcha no es lo más común, ya que lo ideal es hacerlo en el momento de la instalación, pudiendo así incluir todos los discos por completo. Casi todas las distribuciones permiten hacerlo automáticamente, aunque el resultado final no suele ser ideal (ver Consideraciones finales). Cuando comprendas el funcionamiento podrás usar el instalador de tu distribución para crear una estructura LVM personalizada que tenga más sentido que la que propone el instalador sin preguntar.

Entregando dispositivos de almacenamiento al LVM (PV)

El primer paso para usar LVM es decidir que dispositivos de almacenamiento queremos usar. Pueden ser discos, dispositivos RAID por software, particiones, etc. Lo más importante en este paso es que al LVM le daremos el dispositivo para que lo gestione y, salvo raras excepciones, no volveremos a tocar el dispositivo directamente. Nada de darle una partición al LVM y luego crear un sistema de ficheros en ella. Dicho de otra forma, no le toques los PVs al LVM.

La forma de entregar un dispositivo al LVM es marcarlo como un Physical Volumen (PV). Cuando vayas a crear un PV asume que la información que contenga hasta ese momento ya no es tuya, dala por perdida. Así que cuidado con equivocarse con el nombre del dispositivo en el comando *pvcreeate*. Este comando básicamente indica que desde su ejecución el dispositivo indicado como argumento pasa a ser de dominio del LVM. Ejemplos:

```
# Estos comando deben ser ejecutados con el usuario root
# Normanlment los usuarios no pueden jugar con los discos :-)
# Marcar el primer disco, por completo, como PV
pvcreeate /dev/sda
# Marcar varias particiones como PV
pvcreeate /dev/sda3 /dev/sdb1 /dev/sdc1
# Marcar un dispositivo RAID por software como PV
pvcreeate /dev/md0
```

Una vez «marcado» un dispositivo como PV, podemos ver información del mismo con los comandos «*pvs*» y «*pvdisplay*»:

```
# pvcreate /dev/vda3
Writing physical volume data to disk "/dev/vda3"
Physical volume "/dev/vda3" successfully created
# pvs
PV VG Fmt Attr PSize PFree
/dev/vda3 lvm2 a-- 400.00m 400.00m
# pvdisplay /dev/vda3
"/dev/vda3" is a new physical volume of "400.00 MiB"
--- NEW Physical volume ---
PV Name /dev/vda3
VG Name
PV Size 400.00 MiB
Allocatable NO
PE Size 0
Total PE 0
Free PE 0
Allocated PE 0
PV UUID q2noaK-2apE-kkGe-MnNA-IzuE-ICDY-3MiKns
```

En este ejemplo cabe destacar el tamaño del PV, */dev/vda3*, 400M, que están por completo libres y que en la columna «VG» (del comando «*pvs*») o la fila «VG Name» (en «*pvdisplay*») no hay ningún valor. Es decir, el PV está a disposición del LVM pero no pertenece a ningún VG, por lo que por ahora no podemos usarlo (*Allocatable NO*). Por ahora no entraré en la definición de PE, no quiero meter más conceptos y no es necesario a estas alturas.

Creando un Grupo de Volúmenes (VG)

Como comenté en la entrada anterior, un VG es una especie de disco duro virtual. Su tamaño viene dado por la suma del espacio de los PVs que lo componen. Y es del espacio disponible en el VG de donde crearemos los Volúmenes Lógicos (LVs, algo similar a las particiones). Son los LVs los que finalmente usaremos para crear sistemas de ficheros, espacio para swap, discos para máquinas virtuales, etc.

El comando de creación de un VG es extremadamente sencillo. Simplemente le daremos como primer argumento el nombre del VG que estamos creando y a continuación el/los dispositivo(s) (marcados como PVs previamente) que queremos que lo compongan:

```
# Creando un vg llamado "multimedia" con un PV
vgcreate multimedia /dev/vda3
Volume group "multimedia" successfully created
# Creando un vg con más de un PV
```

```
vgcreate produccion /dev/sdb /dev/sdc /dev/sdd  
Volume group "produccion" successfully created
```

Importante en este punto es dar un nombre identificativo al VG. Una de las ventajas del LVM es que vamos a poder dar nombre significativos a nuestro almacenamiento. No pierdas esta ventaja llamando a los VGs algo como «vg00». Yo tengo VGs llamados «raid» y «sinraid» en máquinas que tienen parte de los discos en RAID y otra parte no. Ese nombre me permite, a la hora de crear un LV, saber si mis datos estarán protegidos (y/o acelerados) por el RAID o están en un simple disco. Tengo máquinas que tienen un VG llamado «interno» (formado por los discos físicamente dentro de la máquina) y otro llamado «cabina», porque sus discos están en una cabina externa. Igualmente podríamos tener «producción» y «desarrollo», o «sistema» y «datos». Para gustos los colores, pero aprovecha el poder elegir el nombre 😊 Huelga decir que es mejor evitar símbolos especiales (como espacios), acentos, eñes y esas cosas que podrían dar problemas en nombres de dispositivos.

Una vez creado el VG podemos ver información del mismo con los comandos «vgs» y «vgdisplay»:

```
# vgs  
VG #PV #LV #SN Attr VSize VFree  
multimedia 1 0 0 wz--n- 396.00m 396.00m  
  
# vgdisplay  
--- Volume group ---  
VG Name multimedia  
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 396.00 MiB  
PE Size 4.00 MiB  
Total PE 99  
Alloc PE / Size 0 / 0  
Free PE / Size 99 / 396.00 MiB  
VG UUID xuYff2-maw7-0ldW-PiLv-xZ6F-NFa3-3gNPod
```

Con estos comandos podemos comprobar que nuestro VG se llama «multimedia», está compuesto de un PV, todavía no tiene ningún LV creado y dispone de 396MB libres. Las Physical Extensions (PE) son las unidades de asignación mínima del LVM. Por defecto son de 4MB (aunque se puede cambiar en la creación del

VG). Es decir, cada vez que creamos un LV lo haremos en múltiplos de (por defecto) 4MB. Pero esto no es algo que os deba preocupar por ahora.

A por el producto final, Volúmenes Lógicos (LV)

Una vez tenemos un VG ya podemos crear los dispositivos que realmente usaremos. Los volúmenes lógicos (LV) pertenecen a un VG, del que toman su espacio. Pueden crearse, borrarse y crecer sin necesidad de reiniciar la máquina o parar servicios (bueno, para borrarlos hay que dejar de usarlos primero, obvio).

Un LV puede usar espacio de un solo PV, o de varios. En este último caso puede deberse a que vaya creciendo con el tiempo, y ya no quede espacio en el PV original, o que al crearlo hemos decidido que use varios PVs (por motivos de rendimiento, por ejemplo). Pero esto lo veremos en próximas entradas del blog. Por ahora nos centraremos en la creación de un LV simple.

Para crear un LV, con el comando `lvcreate`, debemos indicarle el VG al que pertenece (como argumento del comando), el tamaño (en PEs con la opción `-l` o en megas/gigas/teras con `-L`) y opcionalmente, pero muy recomendable, el nombre que queremos darle (`-n`). Empleos:

```
# Un LV de 40MB en el VG multimedia para guardar música
lvcreate -L 40M -n musica multimedia

# O especificando el número de PEs (de 4MB por defecto)
lvcreate -l 10 -n musica multimedia
```

Una vez más, aprovecha para darle un nombre representativo del contenido que habrá en él. Desde este momento **la forma de referirse al LV** (para crear el sistema de ficheros, montarlo, o cualquier uso que le vayamos a dar) **será: `/dev/NOMBRE_DEL_VG/NOMBRE_DEL_LV`**. En este ejemplo: **`/dev/multimedia/musica`**. En realidad ese nombre es un enlace al dispositivo creado por el sub-sistema del kernel encargado del LVM, el Device Mapper. El nombre del dispositivo real puede variar y será del tipo `/dev/dm-###`. Por eso es importante usar `/dev/VG/LV`, que siempre apuntará al dispositivo real correcto. También podemos encontrarnos referencias en el sistema al LV como `/dev/mapper/multimedia-musica`, es decir `/dev/mapper/VG-LV`. Mi recomendación es que uses siempre `/dev/VG/LV` y así no hay líos 😊

Al igual que en los pasos anteriores, podemos ver información sobre el LV creado con los comandos `«ls»` y `«lvdisplay»`:

```
# ls
LV VG Attr LSize Pool Origin Data% Move Log Copy% Convert
musica multimedia -wi-a--- 40.00m

# lvdisplay -m
--- Logical volume ---
LV Path /dev/multimedia/musica
```

```

LV Name musica
VG Name multimedia
LV UUID Kttf7H-z5kt-68f6-8B8r-cNk0-OUBa-2cRFfe
LV Write Access read/write
LV Creation host, time curso, 2014-10-30 18:35:31 +0100
LV Status available
# open 0
LV Size 40.00 MiB
Current LE 10
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 256
Block device 253:0

--- Segments ---
Logical extent 0 to 9:
Type linear
Physical volume /dev/vda3
Physical extents 0 to 9

```

Podemos ver el tamaño de nuestro LV (en megas o en LEs), el nombre que tiene, el VG al que pertenece, y si en «*lvdisplay*» añadimos la opción «-m» veremos en donde se encuentra localizado físicamente el LV (que PV(s) a usado el LVM para obtener el espacio del LV, destacado en negrita en el ejemplo anterior).

Ya está listo para usar. Podemos crear un sistema de ficheros en él y montarlo como cualquier otro dispositivo de almacenamiento:

```

# mkfs.ext4 /dev/multimedia/musica
mke2fs 1.42.5 (29-Jul-2012)
....
# mount /dev/multimedia/musica /media/
# df /media/
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/mapper/multimedia-musica 39663 4587 33028 13% /media

```

Consideraciones finales

Algo que considero fundamental para poder aprovechar plenamente la flexibilidad del LVM es mantener el máximo de espacio libre en el VG. Es decir, no dar espacio «a lo loco» a los LVs antes de que éstos lo necesiten. Me mata ver como en una intalación nueva todo el espacio del VG es asignado a uno o varios LVs que apenas usan dicho espacio. De nada sirve tener gigas y gigas de espacio libre en un LV. Y un VG sin espacio sin asignar tampoco sirve para mucho (al menos hasta que se aumente con un PV nuevo). Todo espacio libre que pueda estar en el VG nos va a permitir crecer los LVs que lo necesiten o crear LVs nuevos (aunque

sólo los necesitemos temporalmente). Si tenemos un VG de 1T y hoy sólo necesitamos un LV que contendrá 100G de información, no le des mucho más (tal vez 150G o 200G), pero deja el resto en el VG para cuando sea necesario. Crear un LV es sencillo y no causa interrupciones en el servicio, encojerlo es otra historia...

\$ exit

Esta entrada se publicó en Administración de sistemas y está etiquetada con almacenamiento, introducción, lvm, sistema de ficheros, sysadmin en 31 octubre, 2014 [<https://blog.inittab.org/administracion-sistemas/lvm-para-torpes-ii/>] .

26 comentarios en “LVM para torpes (II)”



wacos

23 enero, 2015 a las 22:54

excelente amigo.. muy bien explicado!!



Lyiorch

30 marzo, 2015 a las 19:18

Padrísimo!!!! muchas gracias, me ha sido de gran utilidad, por favor sigue haciendo este tipo de blogs. Uno muy bueno, sería como hacer arreglos de disco y otro como monitorear en tiempo real el uso del lvm.

Gracias.

Pingback: [LVM para torpes \(tomado de inittab.org\)](https://blog.inittab.org/administracion-sistemas/lvm-para-torpes-ii/) | [Blog de un informatico](#) 



Alejandro

1 abril, 2016 a las 12:21

Hola,

Tengo varias preguntas sobre LVM, tras leer tus articulos. Espero no aburrirte..

- 1- Seria accesible desde Windows un LV, suponiendo que el sistema de ficheros este soportado por Windows?
- 2- Me he quedado con la intriga sobre como encojer un LV. De verdad es tan complicado?
- 3- Tengo un disco duro, el cual tengo planeado utilizar integro para crear un PV, y ahí un VG y varios LV para instalar Gentoo. Me imagino que puedo meter la particion boot en un LV sin problema y hacer un arranque dual Windows-linux. (Windows estaria en otro disco duro en el que no habria ningun PV).

Muchas gracias.



Alberto Gonzalez Iniesta Autor

1 abril, 2016 a las 13:27

Buenas,

- 1.- No. Independientemente del sistema de ficheros. Aunque con **Microsoft y Ubuntu** nunca se sabe...
- 2.- Bueno, no tanto. Lo primero es que no se puede hacer «en caliente». Hay que: desmontar sistema de ficheros (FS), hacer un chequeo *forzoso* del FS, encoger el FS a un tamaño **menor** que el final deseado (margen de seguridad), encoger el LV, estirar el FS para usar lo que quede del margen de seguridad y volver a montar 😊
- 3.- Sí, si usas GRUB para gestionar el arranque de la máquina, puedes tener /boot en un LV.



Alejandro

1 abril, 2016 a las 14:33

Gracias maquina.

He googleado un poco y parece ser que ya hay una solucion para manejar ext4 y lvm2 en Windows.

<https://sourceforge.net/projects/ext2read/>



Anton

1 mayo, 2016 a las 16:08

Perdon por offtop, pero que risa, googleador con experiencia)))) vaya palabra ahahah =)

**Guilherme**

29 noviembre, 2016 a las 13:45

Una pequeña duda que me surgió, ¿es recomendable hacer una tabla de particiones sobre el disco (GPT por ejemplo) antes de asignar un PV o mejor así «a cañón» y simplemente se formatea después con lo que quieras?

Muchas gracias.

**Alberto Gonzalez Iniesta** Autor

29 noviembre, 2016 a las 14:51

Hola Guilherme, pues es más tema de gustos personales. Si el disco *sólo* lo vas a usar como PV, yo no me molestaba en tabla de particiones. Saludos.

**Mario**

20 junio, 2017 a las 15:52

Hola!

Lo primero darte las gracias por tu post, es muy útil, tanto la primera parte como la segunda.

Tengo un problemilla con el pvcreate. En mi servidor tengo /dev/xvda1 pero no aparece en pvdisplay. Cuando hago pvcreate /dev/xvda1 me dice que tengo montado un filesystem en /dev/xvda1. Esto es posible si no está declarado como pv ni vg ni lv ?

Mario

**Alberto Gonzalez Iniesta** Autor

20 junio, 2017 a las 15:59

Hola Mario,

Que no aparezca en pvdisplay es normal si no lo has «marcado» antes con pvcreate.

Ahora, si te dice que /dev/xvda1 tiene un filesystem montado es que definitivamente no quieres que xvda1 sea un PV, pues contiene información que además se está usando. Y por su nombre, yo diría que tienes el

raíz de tu instalación en él. Vamos, que si lo marcas como PV posiblemente sea lo último que hagas (en esa instalación 😊)

Necesitas un disco o partición que no se esté usando para nada. Saludos!

**Mario**

21 junio, 2017 a las 08:44

Efectivamente, tengo el raíz ahí, así que lo que voy a hacer es crearme dos vg en el otro disco que tiene la máquina.

Necesito tener root_vg de 15GB y data_vg de 130GB. Tengo un disco de 8GB (con el /dev/xvda1) y un disco de 160GB que está vacío. En principio entiendo que bastaría con crear los dos VG en el segundo disco de 160GB sin hacer partición, verdad?

Gracias!

Mario

**Mario**

21 junio, 2017 a las 11:27

Nada, ya he creado las dos particiones y los correspondientes VGs y ya estoy creando los LVs

Gracias !!

**Istvan**

28 noviembre, 2017 a las 19:09

Hola Alberto, excelente aporte!

En resumen, el LV creado es de 4MB, es decir, que no ocupa todo el «espacio» del VG, sólo una pequeña porción.

Una consulta: En mi trabajo, he visto que un LV ocupa todo un VG. Entiendo que esta configuración viene de la instrucción `lvcreate -l 100%VG -n «nombre_LV» «nombre_VG»` (es decir, le digo al LV que ocupe el 100% del VG).

Esto tiene sentido en términos de mejores prácticas? . Por lo que veo, «quien lo hizo» no estaba muy claro de las ventajas que te da LVM.

Gracias de antemano!



Alberto Gonzalez Iniesta Autor

25 diciembre, 2017 a las 16:50

Hola,

Pues todo depende, si estás seguro que no vas a necesitar el espacio del VG para *nada* más... Te ahorras vigilar el espacio libre del LV e ir aumentándolo. Pero pierdes la posibilidad de hacer un snapshot o un LV para algo temporal... En mi opinión pierdes flexibilidad por un poco de comodidad. Saludos.



Rogelio

7 febrero, 2018 a las 16:34

Ojalá todos los blogs linuxeros estuviesen tan bien explicados como este artículo en su dos entregas en vez de espantar a los usuarios novatillos y que hacerlos volver al mundo Windows con el rabo entre las piernas, como tristemente pasa tan a menudo en otros sitios. :-/. Enhorabuena. 😊

Pero hay algo que no me queda claro: Si creo un VL formado por dos v. físicos que no siempre están conectados, o no a la vez (por ejemplo, un disco interno que se monta manualmente cuando hace falta volver contenidos de una cámara, y uno externo que lo mismo, la mayor parte del tiempo está desmontado y desconectado y sólo de vez en cuando se pone en uso?, ¿cómo se las apaña LVM en estos casos. No hay problemas? ¿LVM reparte los archivos mediante alguna lógica que haga que si falta un VF simplemente no se muestre lo que contiene ese VF o Iso reparte al «tuntún» de manera que medio archivo puede estar en un un volumen y el otro medio en otro? Porque si es esto último supongo que habrá que tener mucho cuidado de que todos los volúmenes de un grupo estén funcionando cuando se vaya a usar dicho grupo o el desastre estará asegurado, no?

Perdón si mi pregunta es muy elemental, es lo que tiene la ignorancia en ciertos temas ?

Un saludo.



Alberto Gonzalez Iniesta Autor

7 febrero, 2018 a las 17:11

Hola Rogelio, gracias por tu comentario. No es buena idea montar un VG con discos que no van a estar presentes siempre 😊 Perderás datos, todos o casi todos. Será una matanza de datos.. 😊 Es como si arrancaras un trozo a un disco duro. Si hay un LV con datos en los dos nunca volverá a ser el mismo. Si puedes montar un VG en un USB, sólo en él, y usar LVM perfectamente (mientras esté pinchado). Saludos.

**Facundo Bertran**

12 marzo, 2018 a las 01:29

Hola, tienes idea de como obtener datos de la capacidad usada de un PV via SNMP?

**Alberto Gonzalez Iniesta** Autor

12 marzo, 2018 a las 10:15

Hola, pues no sabría decirte. Pero la capacidad de un PV es algo que sólo cambia si aumentas algún LV con él, con lo que mientras no realices esa tarea, no se va a ver alterado. A diferencia de un sistema de ficheros donde puede haber usuarios o servicios escribiendo. Saludos, Alberto

**vxdas**

14 marzo, 2018 a las 16:19

Gracias por tu aporte.....creo estar entendiendo un poco el funcionamiento de LVM, pero no estoy seguro de como implementarlo en mi vps ya corriendo centos 7.

Tengo dos particiones /dev/vda y /dev/vdb de 50 gigas cada una y me gustaria poderlas fusionar

**Alberto Gonzalez Iniesta** Autor

15 marzo, 2018 a las 10:07

Hola! No tengo muy claro si ya estás usando los dos discos que mencionas (vda y vdb). Si ya tienes el sistema instalado, y no estás usando LVM todavía, va a ser MUY complejo ponerlo en marcha ahora. Habría que ver que particiones tienen los discos y cuales se están usando. En función de eso podrías tirar por LVM,

o por el viejo truco/chapucilla de montar particiones en algunos directorios (por ejemplo dejar vdb1 para /var). Pero con el sistema ya funcionando es más complicado todo. Saludos!

**cristopher**

19 junio, 2018 a las 06:58

Todo bien , pero cuando voy a montar me aparece el mensaje » El punto de montaje no existe»
Acabo de crear la ubicacion con mkdir y la veo al realizar un ll , pero no me deja montar nada
Ayuda por favor

**cristopher**

19 junio, 2018 a las 07:14

Solucionado , tuve que poner la ruta absoluta . No tengo idea porque pero aca esta por si le sirve a alguien

```
[root@localhost~]# mount /dev/usuarios/usuario1 /root/usuario1
```

**Zero**

26 septiembre, 2018 a las 22:28

Muhas gracias por la info.
El tutorial es super sencillo y bien explicado.
Como debe ser.

Gracias.!!

**Carlos Mata**

6 mayo, 2021 a las 21:52

Buenas tardes. Una duda que tengo. Por lo que he visto, LVM provee de raid por software. Si esto es así, es recomendable usar raid a nivel de hardware y sobre eso LVM tambien en raid?



Hola, Carlos. Si tienes RAID por hardware, éste suele ser más rápido que el que se hace por software (con el único inconveniente de que dependes de la controladora de disco, es decir no te puedes llevar el RAID a otra máquina con diferente controladora). Si no tienes RAID por hardware, yo prefiero usar el RAID software por software nativo de Linux (mdadm) antes que usar el de LVM. Pero es más por costumbre que por otra cosa. Saludos.
