

Curso de administración de sistemas GNU/linux  
**LVM2: Logical Volume Manager**

Por Jorge Fuertes  
<http://jorgefuertes.com>  
jorge@jorgefuertes.com

©2009 Jorge Fuertes Alfranca  
Revisado a 15 de mayo de 2009

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Sistema de ficheros y niveles de LVM</b>	<b>3</b>
2.1. Dónde se sitúa LVM	3
2.1.1. Representación gráfica por capas	4
2.2. Niveles de LVM	4
2.2.1. Representación de los niveles	5
<b>3. Manejo práctico de LVM2</b>	<b>5</b>
3.1. Instalando lvm	5
3.2. Creando y destruyendo volúmenes físicos (pv*)	5
3.2.1. Crear: pvcreate	5
3.2.2. Listar: pvdisplay	6
3.2.3. Borrar: pvremove	6
3.3. Creando y destruyendo grupos de volúmenes (vg*)	6
3.3.1. Crear: vgcreate	6
3.3.2. Listar: vgdisplay	6
3.3.3. Borrar: vgrename	7
3.4. Extender y reducir: vgextend y vgsplit	7
3.5. Creando y destruyendo volúmenes lógicos (lv*)	7
3.5.1. Crear: lvcreate	7
3.5.2. Listar: lvdisplay	8
3.5.3. Borrar: lvremove	8
<b>4. Utilizando los volúmenes lógicos</b>	<b>9</b>
4.1. Dando formato a los volúmenes lógicos: mkfs	9
4.2. Montando un sistema de ficheros: mount	9
4.2.1. Montando de forma permanente: fstab	10
<b>5. Aumentando y disminuyendo volúmenes lógicos: lvresize</b>	<b>11</b>
5.1. Reduciendo un volumen	11
5.1.1. Con ext3	11
5.2. Recreciendo un volumen	12
5.2.1. Con xfs	12
<b>6. Copias de seguridad e instantáneas</b>	<b>13</b>
<b>7. Futuro de LVM</b>	<b>13</b>
<b>8. Ejercicios sobre LVM</b>	<b>14</b>
8.1. Enunciados	14
8.2. Soluciones	19
<b>9. Sobre esta unidad didáctica</b>	<b>21</b>
9.1. Notas y advertencias	21
9.2. Derechos	21
9.3. Agradecimientos	21
9.4. Revisiones	21
9.5. Bibliografía y fuentes	22

## 1. Introducción

**LVM2**<sup>1</sup> o *administrador de volúmenes lógicos*, es un sistema de administración de discos que permite una gran potencia y flexibilidad. Una vez que se conoce se convierte en imprescindible en la administración de cualquier servidor ya que frente a las limitadas posibilidades de *fdisk*, permite realizar operaciones tan potentes como:

- Redimensionado de grupos lógicos.
- Redimensionado de volúmenes lógicos
- Instantáneas de lectura y escritura.
- RAID0 de volúmenes lógicos.

Esto quiere decir que LVM nos permitirá añadir discos duros, y cambiar los tamaños de los volúmenes, sin tocar los datos de los usuarios, y de forma independiente a nuestro sistema de ficheros. Los volúmenes lógicos podrán ser: un disco duro completo, una parte de un disco duro y varios discos duros o partes de los mismos juntos.

Un claro beneficio del uso de LVM es el almacenamiento de espacio libre en los grupos de volúmenes, pudiendo hacerlos crecer a nuestro antojo desde nuevos discos duros agregados a nuestro sistema, y utilizar este espacio libre para recrear volúmenes lógicos bajo demanda del sistema, servicios y usuarios.

En esta breve guía veremos como instalar y manejar de forma básica y directa este software. La guía está escrita basándose en la distribución *Debian 5.0/Lenny de GNU/Linux* aunque todo lo relacionado con *LVM2* debería funcionar sin cambios en cualquier distro, no así la instalación de paquetes, que podrá variar. Para instalar los paquetes sobre otra distribución, por favor consulte el manual o la web de dicha distribución.

No se pretende que esto sea un manual exhaustivo de *lvm*, al contrario es una iniciación que nos permitirá empezar a manejarnos sin problemas con este sistema. Hay algunos comandos que ni siquiera se mencionarán aunque todos ellos estarán disponibles desde dentro del mini-shell de *lvm*. Si llamamos a *lvm* y en su prompt tecleamos *help*, obtendremos un listado completo de comandos.

**NOTA:** En esta guía aparecerá en repetidas ocasiones el conjunto de símbolos "#>". Son los símbolos que indican que se debe de estar en la línea de comandos de su consola, el *prompt* de su sistema, que puede ser diferente. No debe teclear esos símbolos, son sólo una indicación.

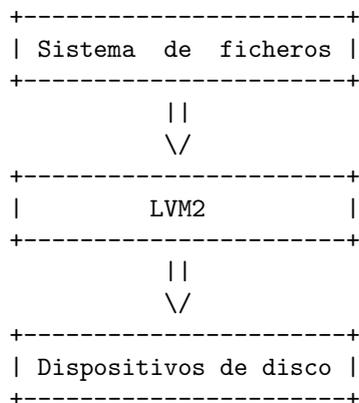
## 2. Sistema de ficheros y niveles de LVM

### 2.1. Dónde se sitúa LVM

El sistema de *LVM* se sitúa en una capa entre el acceso a disco y el sistema de ficheros. Así pues los sistemas de ficheros, como por ejemplo *ext3* o *xf*s no verán directamente los discos, sino los *volúmenes lógicos* creados con *LVM*.

<sup>1</sup><http://es.wikipedia.org/wiki/LVM>

### 2.1.1. Representación gráfica por capas



LVM podrá trabajar directamente con discos, por ejemplo con `/dev/hdc` o podrá trabajar con particiones, por ejemplo con `/dev/hdb2`. De cualquier forma utilizará estos dispositivos "físicos" y podremos empezar a construir sobre ellos.

## 2.2. Niveles de LVM

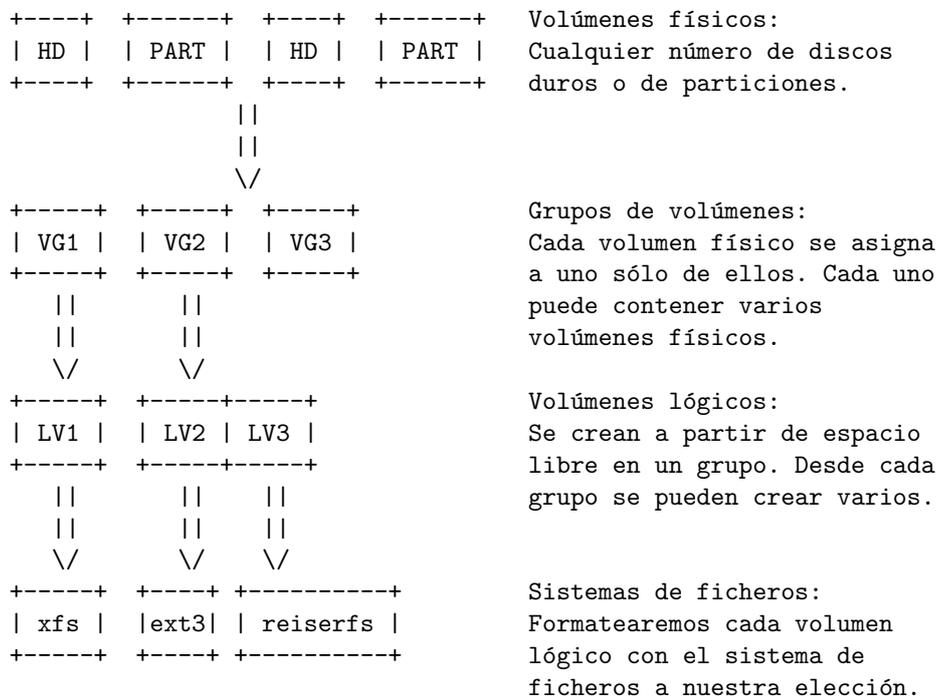
El sistema se basa en tres niveles:

1. Los **volúmenes físicos** o *physical volumes*.
2. Los **grupos de volúmenes** o *volume groups*.
3. Los **volúmenes lógicos** o *logical volumes*.

Habría un cuarto nivel, que no corresponde a LVM, y que sería el sistema de ficheros. Este sistema puede ser cualquiera a nuestra elección, si bien se recomiendan sistemas de ficheros que puedan cambiar de tamaño (aunque sólo sea crecer, como xfs), ya que si bien podremos ampliar el tamaño del dispositivo (volumen lógico), necesitaremos también ampliar el tamaño del sistema de ficheros para aprovechar el crecimiento producido mediante LVM.

Tendremos que pasar por los tres niveles, en orden, para crear un dispositivo lógico que pueda utilizar nuestro sistema de ficheros. En primer lugar crearemos al menos un *volumen físico* a partir de un disco duro o de una partición. Después crearemos un *grupo de volúmenes* agrupando los *volúmenes físicos* creados con anterioridad y pasaremos a crear los *volúmenes lógicos* que deseemos, cogiendo espacio de un *grupo de volúmenes*. Estos *volúmenes lógicos* obtenidos serán los que podamos *formatear* con el sistema de ficheros de nuestra elección.

### 2.2.1. Representación de los niveles



## 3. Manejo práctico de LVM2

Para los ejemplos de las siguientes secciones entenderemos que los discos duros sobre los que trabajaremos serán `/dev/sda` y `/dev/sdb`. Ojo, en su ordenador podrían ser otros y deberá asegurarse. Recuerde que para ver los discos y particiones que su ordenador ya está utilizando podría hacerlo con la orden `mount`, y para ver sus discos y las particiones que contienen puede ejecutar `fdisk -l`. Otros comandos de interés para obtener información son `pvs`, `vgs` y `lvs`.

### 3.1. Instalando lvm

```
#> apt-get install lvm2
```

### 3.2. Creando y destruyendo volúmenes físicos (pv\*)

#### 3.2.1. Crear: pvcreate

Para **crear** un volumen físico, sobre un disco completo, ejecutaremos:

```
#> pvcreate /dev/sda
Physical volume "/dev/sda" successfully created
```

Si todo ha ido correctamente veremos una frase como la que se lee en la línea siguiente de la orden `pvcreate`.

Recuerde que también podría crear un volumen físico sobre una partición en lugar de sobre un disco completo.

### 3.2.2. Listar: pvdisplay

Al solicitar un **listado** de PV, con *pvdisplay* o *pvs*, veremos algo así.

```
#> pvdisplay
"/dev/sda" is a new physical volume of "478,50 MB"
--- NEW Physical volume ---
PV Name           /dev/sda
VG Name
PV Size           478,50 MB
Allocatable       NO
PE Size (KByte)   0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           5GDYAH-qyUm-3NLF-rY8m-N8uR-bVqx-T4G1Cx
```

También podemos obtener un listado rápido de los volúmenes físicos con las órdenes *pvs* y *pvs*.

### 3.2.3. Borrar: pvremove

Para **borrar** un volumen físico (en realidad lo que haremos será quitarle la marca), haremos:

```
#> pvremove /dev/sda
Labels on physical volume "/dev/sda" successfully wiped
```

El PV dejará de estar disponible y no aparecerá en los listados.

## 3.3. Creando y destruyendo grupos de volúmenes (vg\*)

### 3.3.1. Crear: vgcreate

Para **crear** un grupo de volúmenes físicos, es decir, un almacén de espacio libre, haremos:

```
#> vgcreate servidorvg /dev/sda3
Volume group "servidorvg" successfully created
```

Obsérvese la frase de tipo *successfully* que aparece cuando la operación ha sido correcta.

### 3.3.2. Listar: vgdisplay

Puede **comprobar** sus grupos de volúmenes con *vgdisplay*:

```

#> vgdisplay
--- Volume group ---
VG Name          servidorvg
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          476,00 MB
PE Size          4,00 MB
Total PE         119
Alloc PE / Size  0 / 0
Free PE / Size   119 / 476,00 MB
VG UUID          JgT831-sayk-VWD9-Fags-oVHF-s2k0-f6DcDX

```

También podemos obtener un listado rápido de los grupos de volúmenes con las órdenes *vgscan* y *vgs*.

### 3.3.3. Borrar: vgrename

Para **borrar** un grupo de volúmenes (VG), si ya hemos borrado cualquier volumen lógico (LV) que colgase de él, ejecutaremos:

```

#> vgrename servidorvg
Volume group "servidorvg" successfully removed

```

Los volúmenes físicos (PV) quedarán huérfanos y podrán ser borrados o asignados a otro grupo.

## 3.4. Extender y reducir: vgextend y vgsplit

Podemos sacar un volumen físico de un grupo y asignarlo a otro, siempre y cuando dicho volumen no esté en uso, con la orden *vgsplit*. Si por el contrario queremos añadir un nuevo PV a un grupo existente, utilizaremos el comando *vgextend*.

## 3.5. Creando y destruyendo volúmenes lógicos (lv\*)

### 3.5.1. Crear: lvcreate

Para **crear** un volumen lógico (LV), utilizaremos la orden *lvcreate*:

```

#> lvcreate -n volprueba -l 119 servidorvg
Logical volume "volprueba" created

```

La opción `-n` sirve para indicar el nombre del nuevo volumen lógico, y `-l` indica el tamaño en páginas, en este caso eran 119 todas las que teníamos libres, como se puede ver en el `vgdisplay`, también podríamos haber especificado el tamaño en kilobytes, megabytes, gigabytes, etc., todo ello con la opción `-L` o `-size` y una cifra terminada en una de las siguientes letras: *kKmMgGtTpPeE*.

Un volumen lógico obtiene su espacio de un grupo de volúmenes. Justo después de ser creado el volumen estará disponible para ser utilizado como cualquier otro dispositivo de bloques del servidor.

### 3.5.2. Listar: `lvdisplay`

Podemos ver los volúmenes lógicos presentes con la orden `lvdisplay`:

```
#> lvdisplay
--- Logical volume ---
LV Name                /dev/servidorvg/volprueba
VG Name                servidorvg
LV UUID                FygD7C-TvRG-96gX-yUZv-euDQ-TTfd-tMjBEJ
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                476,00 MB
Current LE             119
Segments               1
Allocation             inherit
Read ahead sectors    auto
- currently set to    256
Block device           254:0
```

Hay que notar especialmente el *path* o camino al nuevo dispositivo que figura en *LV Name* y que en este caso es `/dev/servidorvg/volprueba`. Ese es el dispositivo final que podremos tratar con las utilidades del sistema de ficheros.

También podemos obtener listados con las órdenes `lvmdiskscan`, `lvscan` y `lvs`.

### 3.5.3. Borrar: `lvremove`

Para **destruir** un volumen lógico, siempre que esté desmontado, ejecutaremos:

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

Nótese que hay que indicar el camino completo al volumen, no basta sólo con el nombre ya que podría haber volúmenes lógicos con el mismo nombre pero pertenecientes a distintos grupos de volúmenes.

## 4. Utilizando los volúmenes lógicos

### 4.1. Dando formato a los volúmenes lógicos: mkfs

Una vez creado un volumen lógico (lv), podemos formatearlo con el sistema de ficheros de nuestra elección. Esto es posible mediante la orden **mkfs**:

```
#> mkfs -t ext3 /dev/servidorvg/volprueba
mke2fs 1.41.3 (12-Oct-2008)
Etiqueta del sistema de ficheros=
Tipo de S0: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
121920 nodos-i, 487424 bloques
24371 bloques (5.00\%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67633152
60 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
2032 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
  8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Escribiendo las tablas de nodos-i: hecho
Creating journal (8192 blocks): hecho
Escribiendo superbloques y la información contable del
sistema de ficheros: hecho
```

Este sistema de ficheros se revisará automáticamente cada 39 montajes o 180 días, lo que suceda primero. Utilice `tune2fs -c` o `-i` para cambiarlo.

La opción `-t` sirve para indicar el sistema de ficheros que queremos utilizar, bien sea `ext3` como en el ejemplo, `xf`s, `jf`s o cualquier otro soportado por nuestro sistema.

### 4.2. Montando un sistema de ficheros: mount

Podemos montar un nuevo sistema de ficheros en cualquier parte del árbol de nuestra máquina. Un punto de montaje es un simple directorio. Para este ejemplo crearemos el directorio `/mnt/prueba` y montaremos el volumen lógico `volprueba` en él:

```
#> mkdir /mnt/prueba
#> mount -v -t ext3 /dev/servidorvg/volprueba /mnt/prueba/
/dev/mapper/servidorvg-volprueba on /mnt/prueba type ext3 (rw)
```

En la orden `mount` hemos especificado `-v` para que nos dé información de lo que está haciendo y no sólo los errores, y `-t` para decirle el sistema de ficheros que queremos utilizar. Si no le hubiésemos dicho ningún sistema de ficheros, `mount` probaría con los que conoce hasta poder montar el volumen con uno de ellos. Véase:

```
#> mount -v /dev/servidorvg/volprueba /mnt/prueba/
mount: no ha especificado ningún tipo de sistema de ficheros
para /dev/mapper/servidorvg-volprueba
Se probará con el tipo ext3
/dev/mapper/servidorvg-volprueba on /mnt/prueba type ext3 (rw)
```

Al comando `mount` podemos pasarle diversas opciones de montaje. Todas ellas figuran en el manual (*man mount*). Por ejemplo, si quisiéramos montar un volumen en modo *sólo lectura*, le pasaríamos la opción `-o ro`.

Por supuesto al reiniciar la máquina se desmontará el sistema de ficheros, y dejará de estar disponible en lo sucesivo.

#### 4.2.1. Montando de forma permanente: `fstab`

Si queremos que nuestro volumen esté disponible de forma permanente en el sistema, es decir, que se monte automáticamente tras cada reinicio, podemos añadirlo al fichero `fstab`, que es el fichero de configuración del sistema para puntos de montaje. Este fichero está almacenado en el directorio `/etc`, junto al resto de la configuración del sistema.

Un `/etc/fstab` típico sería el siguiente:

```
#> cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/hda7 / ext3 errors=remount-r 0 1
/dev/hda1 /boot ext3 defaults 0 2
/dev/hda2 /home xfs usrquota,grpquota 0 2
/dev/hdb1 /var xfs defaults 0 2
/dev/hda6 none swap sw 0 0
/dev/hda5 /tmp ext3 noatime,noexec,nosuid 0 0
/dev/hdc /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto 0 0
```

En la tercera línea de comentarios vienen especificados los campos, que son:

- **file system:** El volumen que queremos montar.
- **mount point:** El punto de montaje.
- **type:** Tipo de sistema de ficheros, `ext3`, `xfs`, etc...
- **options:** Opciones de montaje. Si es de sólo lectura, si no queremos montarlo automáticamente, si no queremos horas de acceso almacenadas, etc...
- **dump:** Lo utiliza el comando `dump` para determinar que sistemas de ficheros necesitan ser volcados. Si está vacío, `dump` asume que el sistema de ficheros no necesita ser volcado. Lo normal es que esté a "0" y que de hecho, el programa `dump` no esté ni instalado.
- **pass:** Lo usa el programa `fsck(8)` para determinar el orden en el cual se van a chequear los sistemas de ficheros cuando el sistema arranca. Lo normal es que el raíz sea "1", otros estén a "2" y se comprobarán en paralelo

si están en distintos discos. Si este campo está a "0" simplemente no se realizarán comprobaciones periódicas al montar estos sistemas de ficheros.

Para nuestro volumen añadiremos a */etc/fstab* la siguiente línea:

```
/dev/servidorvg/volprueba /mnt/prueba ext3 defaults 0 2
```

Desde este momento podemos montar el punto de montaje directamente, sin decir a que volumen corresponde:

```
#> umount /mnt/prueba
#> mount -v /mnt/prueba
/dev/mapper/servidorvg-volprueba on /mnt/prueba type ext3 (rw)
```

Primero lo hemos desmontado porque estaba montado en los ejemplos anteriores, después ejecutamos la orden *mount* directamente sobre el punto de montaje y el volumen se monta con las opciones que figuran en *fstab*.

## 5. Aumentando y disminuyendo volúmenes lógicos: *lvresize*

Hay que tener en cuenta que por mucho que podamos cambiar a nuestro antojo y *en línea* el tamaño de los volúmenes lógicos, dependemos del sistema de ficheros con el que tengamos funcionando cada volumen. La mayor parte de los sistemas de ficheros permiten cambios de tamaño, pero no todos permiten hacerlo mientras están los volúmenes montados o *en línea*, y **pocos admiten reducción de tamaño**. Normalmente queremos extender un sistema de ficheros, al habernos quedado sin espacio, por tanto eso no suele ser un problema.

Hay que reseñar que *ext2* y *ext3* permiten crecer y reducir si bien la reducción no puede hacerse en línea, y que *afs* y *jfs* no permiten reducciones pero si pueden crecer en línea.

### 5.1. Reduciendo un volumen

#### 5.1.1. Con *ext3*

En primer lugar desmontaremos el volumen:

```
#> umount /mnt/prueba
```

A continuación lo reduciremos:

```
#> resize2fs -f /dev/servidorvg/volprueba 200M
resize2fs 1.41.3 (12-Oct-2008)
Resizing the filesystem on /dev/servidorvg/volprueba to 204800
(1k) blocks.
El sistema de ficheros en /dev/servidorvg/volprueba tiene ahora
204800 bloques.
```

Es ahora cuando podemos reducir el volumen (el disco sobre el que está el sistema de ficheros):

```
#> lvresize --size 200M /dev/servidorvg/volprueba
WARNING: Reducing active logical volume to 200,00 MB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce volprueba? [y/n]: y
```

Ya podemos montarlo de nuevo, o comprobarlo con un *fsck*.

Como verá hemos hecho la reducción, pero las utilidades se han estado quejando. Lo habitual es que no tengamos que reducir, que es algo peligroso, si no aumentar o recrecer un volumen.

## 5.2. Recreando un volumen

### 5.2.1. Con xfs

En primer lugar debemos aumentar el tamaño del volumen (disco sobre el que está nuestro sistema de ficheros). Evidentemente deberá haber espacio libre en el *volume group* al que pertenece:

```
#> lvresize --size +200M /dev/servidorvg/volprueba
Extending logical volume volprueba to 400,00 MB
Logical volume volprueba successfully resized
```

Ahora indicaremos al sistema de ficheros que crezca hasta ocupar todo el espacio disponible en el volumen:

```
#> xfs_growfs /mnt/prueba
meta-data=/dev/mapper/servidorvg-volprueba isize=256   agcount=4,
          agsize=12800 blks
          =                               sectsz=512   attr=2
data      =                               bsize=4096   blocks=51200, imaxpct=25
          =                               sunit=0     swidth=0 blks
naming    =version 2                       bsize=4096
log       =internal                        bsize=4096   blocks=1200, version=2
          =                               sectsz=512   sunit=0 blks, lazy-count=0
realtime  =none                             extsz=4096   blocks=0, rtextents=0
data blocks changed from 51200 to 102400
```

Obsérvese que lo hemos hecho con el volumen montado, sin producir ninguna molestia a los usuarios y sin parar ningún servicio.

Comprobaremos el nuevo tamaño con *df*:

```
#> df -h
S.ficheros                               Tamaño Usado  Disp Uso% Montado en
/dev/mapper/servidorvg-volprueba 396M 4,3M 392M  2% /mnt/prueba
```

El ejemplo es aplicable a cualquier sistema de ficheros que permita crecer, en lugar de *xfs\_growfs* deberemos utilizar la utilidad apropiada en cada caso. Hay más información disponible en <http://tldp.org/HOWTO/LVM-HOWTO/extendlv.html>.

## 6. Copias de seguridad e instantáneas

Para tomar una instantánea de un volumen, podemos hacerlo de la siguiente forma:

```
#> lvcreate -s -n pruebabackup --size 50M /dev/servidorvg/volprueba
Rounding up size to full physical extent 52,00 MB
Logical volume "pruebabackup" created
```

Opciones utilizadas:

- **-s**: Snapshot. Indicamos que estamos tomando una instantánea.
- **-n**: Nombre del volumen-instantánea.
- **-size**: El tamaño del volumen-instantánea. En este tamaño se van a ir guardando los cambios que se produzcan tanto en el original como en este si lo utilizamos. Por tanto si tenemos pensado utilizarlo o el original cambia mucho y vamos a necesitar esta instantánea durante mucho tiempo, será necesario darle mayor tamaño. Si el volumen-instantánea se llenase, sería desconectado (*dropped*) y quedaría inútil.
- Ruta y nombre del volumen a crear.

Con este volumen recién creado podemos realizar todo tipo de operaciones como si fuese el original. Podemos montarlo y hacer todo tipo de modificaciones, o montarlo y hacer una copia de seguridad con *tar* obteniendo todos los datos del original tal y como estaban en el momento de la instantánea.

Cuando ya no lo necesitemos podemos destruirlo como un volumen normal.

## 7. Futuro de LVM

LVM es un sistema estable y con demostrado funcionamiento y utilidad a nivel empresarial. Sin embargo hay sistemas que lo superan en prestaciones, especialmente aquellos que combinan el sistema de ficheros y el manejador de volúmenes todo en uno.

El más famoso de estos "todo en uno" es **ZFS**<sup>2</sup> de *Sun Microsystems*, y es el sistema de ficheros por defecto en *Solaris* y otras empresas como *Apple* ya le han echado el ojo para sus sistemas. Se están haciendo esfuerzos para portarlo a *GNU/Linux* y el autor cree que no pasará mucho tiempo antes de que esté disponible.

Otro "todo en uno" todavía más potente es **Zumastor**<sup>3</sup>, actualmente en desarrollo. Nativo de *GNU/Linux* promete ser mejor y más rápido que el aclamado *ZFS*. Muy pronto parece que habrá versiones estables incluidas en las distribuciones.

Pero por el momento... LVM ha demostrado sobradamente hacer lo que promete, y hacerlo con total seguridad.

---

<sup>2</sup>[http://es.wikipedia.org/wiki/ZFS\\_\(sistema\\_de\\_archivos\)](http://es.wikipedia.org/wiki/ZFS_(sistema_de_archivos))

<sup>3</sup><http://zumastor.org>

## 8. Ejercicios sobre LVM

### 8.1. Enunciados

1. En la configuración de su máquina virtual (*VirtualBox*) cree cinco discos duros nuevos, con los siguientes tamaños: 50Mb, 100Mb, 200Mb, 300Mb y 400Mb. Asígnelos a la máquina virtual de Debian.
2. Desde Debian, obtenga un listado de los discos duros presentes en su sistema. Especifique cuales de ellos están montados y donde, y cuales están libres.
3. Utilizando la herramienta *fdisk*, divida en dos partes iguales el disco de 400Mb.
4. Cree el grupo de volúmenes *grupo1vg* e introduzca en él los discos de 50, 100 y 200Mb.
5. Cree el grupo de volúmenes *grupo2vg* e introduzca en él el disco de 300Mb.
6. A partir del grupo *grupo1vg* cree un volumen lógico con nombre *vol1*, de tamaño 150Mb, formateéelo con *ext3* y móntelo en */mnt/vol1*.

7. A partir del grupo *grupo1vg* cree un volumen lógico con nombre *vol2*, de tamaño 150Mb, formateélo con *xf*s y móntelo en */mnt/vol2*.
  
8. A partir del grupo *grupo1vg* cree un volumen lógico con nombre *vol3*, con todo el espacio disponible, formateélo con *xf*s y móntelo en */mnt/vol3*.
  
9. A partir del grupo *grupo2vg* cree un volumen lógico de 50Mb con nombre *vol4*, formateé con *xf*s y móntelo siguiendo la misma nomenclatura que hasta ahora.
  
10. A partir del grupo *grupo2vg* cree un volumen lógico de 200Mb con nombre *vol5*, formateé con *xf*s y móntelo siguiendo la misma nomenclatura que hasta ahora.
  
11. Haga un *tar.bz* de su */etc* guardándolo en */mnt/vol1/backup*.
  
12. Extraiga todo el contenido del tar generado en el ejercicio anterior en */mnt/vol2/etc.bk*. Por favor, no pregunte al profesor si debe crear los directorios necesarios para extraer.
  
13. Apague su máquina virtual y tire un snap-shot de la misma. Enciéndala de nuevo.



20. Cree un script que dispare una instantánea de */mnt/vol1* cada vez que es llamado. El nombre de dicha instantánea debe contener la fecha y la hora en el formato "*ddmmyyyghhmmss*". La instantánea debe tener reservado un espacio de 5Mb para cambios.
  
21. Modifique el script anterior para monte automáticamente la instantánea en */mnt/vol1-snap*, recuerde que deberá hacerlo cada vez, así que tendrá que desmontar la anterior si está montada.
  
22. Añana *logging* a su script. Cada operación debe quedar reflejada en */var/log/snaps-vol1.log*, con fecha y hora.

23. **Avanzado**<sup>4</sup>: Consiga que el script conserve sólo las dos últimas instantáneas y que borre todas las anteriores cada vez que es llamado.

24. **Avanzado**: Haga que el script sólo dispare una nueva instantánea si han pasado más de  $x$  minutos desde la anterior. La cantidad de minutos debe ser configurable mediante una variable al principio del script.

---

<sup>4</sup>Sólo para alumnos que ya manejen bash-scripting a nivel de medio

## 8.2. Soluciones

### Script demostración ejercicios 20-24:

```
#!/bin/bash

# Este script toma instantáneas de un volumen y sólo permite tomarlas
# si se ha cumplido el tiempo programado.
#
# Las monta y desmonta las anteriores, y conserva sólo las dos últimas.

FECHA=$(date +%d%m%Y%H%M%S')
FICHERO=$FECHA-snap

# Configuración:
MINTIME=2 # Tiempo mínimo entre instantáneas. En minutos.

# Funciones:
function log {
    echo $1
    echo "[$(date +%d%m%Y@%H%M%S')] ${1}" >> /var/log/snap.log
}

if [ -f /tmp/snap.last ]
then
    LASTSNAP=$(cat /tmp/snap.last)
else
    LASTSNAP=0
fi

TS=$(date +%s")
SECLAST=$(expr $TS - $LASTSNAP)
#echo $LASTSNAP - $TS = $SECLAST
MINLAST=$(expr $SECLAST / 60)
#echo $SECLAST / 60 = $MINLAST

if [ $MINLAST -lt $MINTIME ]
then
    log "No han transcurrido más de $MINTIME minutos."
    log "No se hará nada."
    exit 0
fi

log "Guardando timestamp ${TS}"
echo $TS > /tmp/snap.last

log "Creando instantánea ${FICHERO}"
lvcreate -s --size 1M -n $FICHERO /dev/servidorvg/volprueba

log "Creando /mnt/snap"
mkdir -p /mnt/snap
```

```

log "Desmontando /mnt/snap"
umount /mnt/snap &> /dev/null

log "Montando última instantánea"
mount -t xfs /dev/servidorvg/$FICHERO /mnt/snap

log "Borrando instantáneas viejas"
NUM=$(ls -C1 --color=none /dev/servidorvg/*-snap|wc -l)
CONT=1
LASTTWO=$(echo "$NUM - 1"|bc)
echo "$NUM / $LASTTWO"
for i in $(ls -C1 --color=none /dev/servidorvg/*-snap)
do
    echo -n "$CONT $i..."
    if [ $CONT -lt $LASTTWO ]
    then
        echo "borrando."
        lvremove -f $i
    else
        echo "conservando."
    fi
    let CONT++
done

# Fin del script.
exit 0

```

## 9. Sobre esta unidad didáctica

### 9.1. Notas y advertencias

**Debian:** Esta guía está basada en el sistema *Debian GNU/Linux*, podría haber pequeños cambios si se aplica a otras distribuciones de *GNU*, pero en su mayor parte funcionará bien con la excepción de lo referido al sistema de paquetería de programas, los comandos que empiezan por *apt*, ya que otras *distros* no basadas en *Debian* podrían incorporar sistemas diferentes para el manejo de sus paquetes.

### 9.2. Derechos

Esta guía se cede bajo contrato Coloriuris. Sólo puede ser utilizada previa aceptación del contrato de cesión sito en:

- <http://www.coloriuris.net/contratos/ef5af6aaa441ab9c213273fade56dca1>

Dicho contrato garantiza que estoy cediendo los derechos de uso y modificación sin ánimo de lucro.

### 9.3. Agradecimientos

El autor quiere reflejar su agradecimiento a todas las páginas de Internet que ponen a disposición de todo el mundo sus contenidos, así como a todo aquél que publica artículos, manuales y experiencias en Internet, ya que eso favorece a la difusión del conocimiento y al desarrollo humano. *La información quiere ser libre.*

Un agradecimiento muy especial a toda la comunidad del Software Libre. Sin ellos el autor viviría en la oscuridad: Programadores, traductores, asociaciones, hacktivistas, webmasters, etc...

También quiero agradecer muy especialmente su ayuda a mis alumnos y lectores, por tomarse la molestia de comunicarme las erratas y por darme ideas para mejorar los ejercicios.

### 9.4. Revisiones

El autor irá eventualmente publicando revisiones de esta unidad en su página personal, y estará encantado de recibir sugerencias y dudas en la misma o en su email:

- <http://jorgefuertes.com>.
- [cursos@jorgefuertes.com](mailto:cursos@jorgefuertes.com).

Por supuesto se puede contactar con el autor para contratarle para hacer nuevas unidades, adaptaciones, modificaciones, cursos, etc...

## 9.5. Bibliografía y fuentes

Para la realización de esta guía el autor ha leído diversas fuentes, entre las que se encuentran la documentación oficial de LVM2, así como sus *páginas man*, y las siguientes páginas web:

- <http://es.wikipedia.org/wiki/LVM>.
- <http://www.websecurity.es>.
- <http://www.tldp.org/HOWTO/LVM-HOWTO>.
- <http://trucolinux.com>.