

RAID

RAID

Gracias a esta tecnología mejora no solo el rendimiento y/o la seguridad utilizando al menos dos discos duros. Los discos duros se convierten en uno solo.

El significado de RAID es Redundant Array of Inexpensive Disk (Matriz redundante de discos independientes).

Tipos de RAID

- **RAID 0** : Se utiliza para doblar el rendimiento y para fusionar todos los discos duros en un sólo disco para aumentar la capacidad de almacenamiento. Es necesario tener 2 discos duros como mínimo. Por ejemplo si tenemos dos discos que funciona a una velocidad alrededor de 20 Mo/s, al poner dos discos se duplicaría la velocidad es decir 40 Mo/s (2x20 Mo/s). Es una partición lógica cuyo tamaño es igual a la suma de los discos integrados en el sistema RAID.
- **RAID 1** : Es utilizado para garantizar la integridad de los datos, en caso de un fallo de uno de los discos duros, es posible continuar las operaciones en el otro disco duro sin ningún problema. No se mejora el rendimiento y no se suman el espacio de los discos como en **RAID 0**. El tipo de **RAID 1** se llama comúnmente “**mirroring**” debido a que éste hace una simple copia del primer disco.
- **RAID 10 y 01** : Garantiza la integridad de los datos y aumenta el rendimiento. Es necesario tener 4 discos como mínimo.
- **RAID 5** : Es como **RAID 10 & 01** pero con mayor capacidad. La seguridad utiliza la paridad en cada disco duro. Se debe tener 3 discos duros como mínimo.

¿ Cómo utilizar los discos duros en RAID ?

El disco maestro debe ser igual o menor el espacio que el disco secundario, como mínimo se debe tener 2 discos, dependiendo del tipo de RAID. Lo recomendable que ambos discos sean del mismo tamaño de capacidad.

Instalación del paquete mdadm

Una de las principales cosas que tenemos que instalar el paquete de **mdadm** que contiene una serie de programas y utilidades para el manejo.

```
# apt-get install mdadm
```

Trabajando con GNU/Linux en RAID

Pasamos primero a crear la partición de tipo **RAID** de dos discos iguales.

```
# fdisk /dev/sda
```

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to sectors (command 'u').

Command (m for help): **p**

Disk /dev/sdd: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x86d799b5

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

Command (m for help): **n**

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): **1**

First cylinder (1-261, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-261, default 261):

Using default value 261

Command (m for help): **t**

Selected partition **1**

Hex code (type L to list codes): **fd**

Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): **p**

Disk /dev/sdd: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x86d799b5

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	261	2096451	fd	Linux raid autodetect

Command (m for help): **w**

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

El segundo disco para realizar el tipo de **RAID 1**.

fdisk /dev/sdd

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to switch off the mode (command 'c') and change display units to sectors (command 'u').

Command (m for help): **p**

Disk /dev/sdd: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x86d799b5

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

Command (m for help): **n**

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): **1**

First cylinder (1-261, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-261, default 261):

Using default value 261

Command (m for help): **t**

Selected partition **1**

Hex code (type L to list codes): **fd**

Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): **p**

Disk /dev/sdd: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x86d799b5

Device	Boot	Start	End	Blocks	Id	System
/dev/sdd1		1	261	2096451	fd	Linux raid autodetect

Command (m for help): **w**

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

Una vez creado las particiones, vamos a crear con el comando **mdadm** el raid.

Creación de un dispositivo raid usando mdadm

Primero que nada instalamos el paquete mdadm.

```
$ apt-get install mdadm
```

Veremos que se instaló el archivo de configuración en **/etc/mdadm/mdadm.conf**, editamos este archivo para configurarlo :

```
$ vi /etc/mdadm/mdadm.conf
```

```
DEVICE=/dev/sd[ad]1
```

```
ARRAY /dev/md0 devices=/dev/sda1,/dev/sdd1
```

```
$ mdadm -C /dev/md0 --level=raid1 --raid-devices=2 /dev/sda1 /dev/sdd1
```

```
mdadm: /dev/sda1 appears to contain an ext2fs file system
```

```
size=2096448K mtime=Thu Jan 1 00:00:00 1970
```

```
Continue creating array? yes
```

```
mdadm: Defaulting to version 1.2 metadata
```

```
mdadm: array /dev/md0 started.
```

Vemos que los dispositivos de tipo RAID son `/dev/mdN`, donde N es el numero de dispositivo, también le indicamos el tipo de raid en este caso **raid0** (`--level=raid0`) y la cantidad de discos que constituyen ese raid (`--raid-devices=2`) y los discos en si (`/dev/sda1` y `/dev/sdd1`).

Vemos el detalle del raid que armamos :

```
$ mdadm --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Tue Sep 13 14:51:48 2011
  Raid Level : raid1
  Array Size : 4190208 (4.00 GiB 4.29 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Tue Sep 13 14:51:48 2011
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Chunk Size : 512K

  Name : centrx:0 (local to host centrx)
  UUID : 69dad673:aa9c153d:201cb9f7:4f7a6310
  Events : 0

  Number Major Minor RaidDevice State
    0   8   1   0   active sync  /dev/sda1
    1   8  49   1   active sync  /dev/sdd1
```

Generar un falla de unos de los discos

Generamos la falla con la opción **-f**.

```
$ mdadm /dev/md0 -f /dev/sdd1
mdadm: set /dev/sdd1 faulty in /dev/md0
```

Removemos el disco que se encuentra con falla, para luego poner otro disco, formatearlo y que realice el vuelco de información.

```
$ mdadm /dev/md0 -f /dev/sdd1
mdadm: set /dev/sdd1 faulty in /dev/md0

$ mdadm --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Tue Sep 13 14:51:48 2011
  Raid Level : raid1
  Array Size : 4190208 (4.00 GiB 4.29 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Tue Sep 13 14:51:48 2011
```

State : clean
Active Devices : 1
Working Devices : 1
Failed Devices : 1
Spare Devices : 0

Chunk Size : 512K

Name : centrx:0 (local to host centrx)
UUID : 69dad673:aa9c153d:201cb9f7:4f7a6310
Events : 0

Number	Major	Minor	RaidDevice	State
0	8	1	0	active sync /dev/sda1
1	8	49	1	faulty spare /dev/sdd1

Ahora lo que hacemos es poner remover el dispositivo que funciona mal :

```
$ mdadm /dev/md0 -r /dev/sdd1  
mdadm: hot removed /dev/sdd1 from /dev/md0
```

Ahora podemos poner el nuevo disco, para que sincronice los datos con el disco maestro.

```
$ mdadm /dev/md0 -a /dev/sdd1  
mdadm: re-added /dev/sdd1
```

```
$ mdadm --detail /dev/md0  
/dev/md0:  
Version : 1.2  
Creation Time : Tue Sep 13 14:51:48 2011  
Raid Level : raid1  
Array Size : 4190208 (4.00 GiB 4.29 GB)  
Raid Devices : 2  
Total Devices : 2  
Persistence : Superblock is persistent
```

```
Update Time : Tue Sep 13 14:51:48 2011  
State : clean  
Active Devices : 1  
Working Devices : 1  
Failed Devices : 1  
Spare Devices : 0
```

Chunk Size : 512K

Name : centrx:0 (local to host centrx)
UUID : 69dad673:aa9c153d:201cb9f7:4f7a6310
Events : 0

Number	Major	Minor	RaidDevice	State
0	8	1	0	active sync /dev/sda1
1	8	49	1	spare rebuilding /dev/sdd1

Luego de esto podemos crear el disco tipo lvm.

```
$ pvcreate /dev/md0  
Physical volume "/dev/md0" successfully created
```

```
$ vgcreate vgwww /dev/md0  
Volume group "vgwww" successfully created
```

```
$ lvcreate -L +500M -n lvwww vgwww
Logical volume "lvwww" created
```

```
$ mkfs.ext4 /dev/vgwww/lvwww
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
128016 nodos-i, 512000 bloques
25600 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67633152
63 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 27 montajes o 180 días, lo que suceda primero. Utilice `tune2fs -c o -i` para cambiarlo.

```
$ mkdir /www
$ mount /dev/vgwww/lvwww /www
```

LVM2

Características de LVM2

Se encuentran las siguientes características :

- Redimensión de grupos de volúmenes y volúmenes lógicos en línea.
- Crear instantáneas (snapshots) de lectura/escritura del sistema de archivos.
- Constituir los volúmenes lógicos separados en los diferentes volúmenes físicos, de manera similar que RAID 0.
- Mover los volúmenes lógicos entre los diferentes volúmenes físicos.

Conceptos básicos

- **Grupo de volúmenes (VG):** Es el punto de abstracción más alto en LVM. Este concepto define la unión de uno o más PVs en una unidad administrativa. Los PVs pueden comenzar a utilizarse en LVM recién cuando pasan a formar parte de un VG.
- **Volumen físico (PV):** es un disco rígido, una partición o un RAID (físico o por software).
- **Volumen lógico (LV):** es el equivalente a una partición en un sistema que no es LVM. El volumen lógico es visto como un dispositivo de bloque, ya que puede contener un sistema de ficheros.

- **Extensión física (PE):** cada volumen físico está dividido en trozos de datos de tamaño fijo, eso es lo que se conoce como extensión física. Estas extensiones tienen el mismo tamaño que las extensiones lógicas para el grupo de volúmenes. Un volumen físico se divide en múltiples PEs del mismo tamaño. La extensión física por defecto es de 4MB pero puede variar desde 8kB hasta 16GB (usando potencias de 2).
- **Extensión lógica (LE):** cada volumen lógico está dividido en trozos de datos de tamaño fijo, eso es lo que se conoce como extensión lógica. El tamaño de las extensiones es igual para todos los volúmenes lógicos en el grupo de volúmenes. Un volumen lógico se divide en múltiples LEs del mismo tamaño.
- **Device mapper:** es un framework genérico del kernel de Linux que permite realizar un mapeo de un dispositivo de bloques a otro. Es la herramienta fundamental en la que se basa LVM para hacer el mapeo de los dispositivos virtuales con los dispositivos físicos.

Configuración del kernel

En nuestro núcleo (kernel) tenemos que tener habilitado con un (*) para que este incluido dentro del núcleo y no como modulo, lo siguiente :

Soporte RAID y LVM

[] Multiple devices driver support (RAID and LVM)*

Enable LVM support

<> Logical volume manager (LVM) support*

Instalación del paquete lvm2

Una de las principales cosas que tenemos que instalar el paquete de **lvm2** que contiene una serie de programas y utilidades para el manejo.

apt-get install lvm2

Dentro del script de **lvm2 /etc/init.d/lvm2** realiza algunas de los siguientes pasos :

- Carga el modulo **dm-mod**.
- Realiza un **vgscan** comprueba los archivos LVM especiales en **/dev** que se necesitan para activos volúmenes lógicos y crea las que faltan y elimina que no se utilicen.
- Controla la disponibilidad de los volúmenes lógicos del grupo de volúmenes de entrada/salida.

Comandos generales de LVM

En las siguientes tabla veremos los comando para **pv** (volúmenes físicos), **vg** (grupos de volúmenes), **lv** (volúmenes lógicos).

Comandos de volúmenes físicos

Comandos	Descripción
pvchange	Cambia los atributos de un volumen físico.
pvcreate	Crea un volumen físico.
pvdisplay	Muestra información sobre un volumen físico.
pvmove	Mueve extensiones físicas de un volumen físico a otro.
pvscan	Localiza y enumera todos los volúmenes físicos.

Comandos de grupos volumen

Comandos	Descripción
vgcfgbackup	Hace una copia del área del descriptor de grupo volumen.
vgcfgrestore	Recupera el área del descriptor de grupo volumen al disco.
vgchange	Cambia los atributos de un grupo volumen.
vgck	Comprueba la consistencia del área del descriptor de grupo volumen.
vgcreate	Crea un grupo volumen a partir de volúmenes físicos.
vgdisplay	Muestra los atributos de un grupo volumen.
vgexport	Exporta un grupo volumen del sistema.
vgextend	Añade volúmenes físicos a un grupo volumen.
vgimport	Importa un grupo volumen al sistema.
vgmerge	Combina dos grupos volumen.
vgmknodes	Combina un directorio y los archivos especiales de un grupo volumen.
vgreduce	Elimina volúmenes físicos de un grupo volumen.
vgremove	Elimina un grupo volumen.
vgrename	Renombra un grupo volumen.
vgscan	Localiza y enumera todos los grupos volumen del sistema.
vgsplit	Divide un grupo volumen.

Comandos de volúmenes lógicos

Comandos	Descripción
lvchange	Cambia los atributos de un volumen lógico.
lvcreate	Crea un volumen lógico.
lvdisplay	Muestra los atributos de un volumen lógico.
lvextend	Incrementa el tamaño de un volumen lógico.
lvreduce	Reduce el tamaño de un volumen lógico.
lvremove	Elimina un volumen lógico.
lvrename	Renombra un volumen lógico.

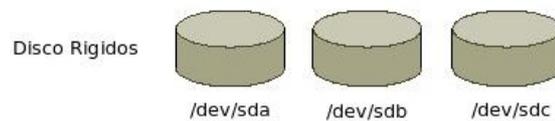
lvscan	Localiza y enumera todos los volúmenes lógicos.
--------	---

Comandos del Administrador de Volúmenes Lógicos (LVM)

Comandos	Descripción
lvchange	Cambia los atributos del sistema LVM.
lvmdiskscan	Localiza y enumera todos los discos disponibles, los dispositivos múltiples y las particiones.
lvmsadc	Recoge información sobre la actividad de LVM.
lvmsar	Muestra información sobre la actividad de LVM.

Empezando con un ejemplo

Tenemos el siguiente esquema :



Lo primero que vamos a realizar es mediante **fdisk** crear una partición tipo **LVM** :

\$ fdisk /dev/sdb

```
Disk /dev/sdb: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xb4c5bd92
```

```
Device Boot   Start    End  Blocks  Id System
```

```
Command (m for help): n
```

```
Command action
```

```
  e  extended
```

```
  p  primary partition (1-4)
```

```
p
```

```
Partition number (1-4): 1
```

```
First cylinder (1-522, default 1):
```

```
Using default value 1
```

```
Last cylinder, +cylinders or +size{K,M,G} (1-522, default 522):
```

```
Using default value 522
```

```
Command (m for help): t
```

```
Selected partition 1
```

```
Hex code (type L to list codes): 8e
```

```
Changed system type of partition 1 to 8e (Linux LVM)
```

```
Command (m for help): p
```

```
Disk /dev/sdb: 4294 MB, 4294967296 bytes
```

```
255 heads, 63 sectors/track, 522 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Sector size (logical/physical): 512 bytes / 512 bytes
 I/O size (minimum/optimal): 512 bytes / 512 bytes
 Disk identifier: 0xb4c5bd92

```
Device Boot Start End Blocks Id System
/dev/sdc1 1 522 4192933+ 8e Linux LVM
```

Command (m for help): w
 The partition table has been altered!

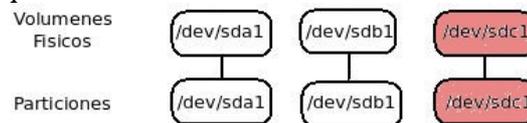
Calling ioctl() to re-read partition table.
 Syncing disks.

En esta primer parte hemos creado una partición una partición de tipo **LVM (8e)**.

Ahora debemos iniciar esta partición como un **PV (physical volume)** :

```
$ pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created
```

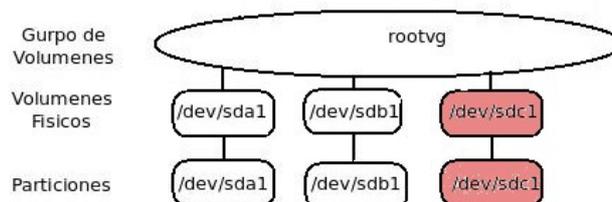
Nos queda el siguiente esquema :



Ahora ya creado el **volumen físico**, ahora pasamos a crear nuestro **grupo de volumen (vg – volume group)**. De esta forma por defecto cuando creamos un **vg** ya automáticamente queda activo.

```
$ vgcreate rootvg /dev/sdc1
Volume group "rootvg" successfully created
```

Quedando de la siguiente forma :



Para ver lo que estuvimos realizando hasta ahora podemos ejecutar los siguientes comandos.

```
$ vgs
```

```
VG #PV #LV #SN Attr VSize VFree
rootvg 1 0 0 wz--n- 4,00g 4,00g
```

Si queremos ver algo mas detallado ejecutamos el **vgs -v**.

```
$ vgs -v
```

```
VG Attr Ext #PV #LV #SN Vsize VFree VG UUID
rootvg wz--n- 4,00m 1 0 0 4,00g 4,00g MiPyOk-hsTI-nRSj-PRKm-TV27-XnFG-NZGQzc
```

Ahí vemos que tenemos el nombre del **volumen group (rootvg)** y un **physical volumen**.

Si queremos ver en mas detalle ejecutamos el siguiente comando :

\$ vgdisplay

```
--- Volume group ---
VG Name          rootvg
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          4,00 GiB
PE Size          4,00 MiB
Total PE         1023
Alloc PE / Size  0 / 0
Free PE / Size   1023 / 4,00 GiB
VG UUID          MiPyOk-hsTI-nRSj-PRKm-TV27-XnFG-NZGQzc
```

Como vemos el **VG size** es el tamaño total del **Volumen Group**.

Para ver mas detalladamente también podemos poner la opción **-v**, lo bueno que con esto nos indica que discos físicos son lo que están en el grupo..

\$ vgdisplay -v

```
--- Volume group ---
VG Name          rootvg
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          4,00 GiB
PE Size          4,00 MiB
Total PE         1023
Alloc PE / Size  0 / 0
Free PE / Size   1023 / 4,00 GiB
VG UUID          MiPyOk-hsTI-nRSj-PRKm-TV27-XnFG-NZGQzc

--- Physical volumes ---
PV Name          /dev/sdc1
PV UUID          iAC40D-IBDC-C23I-zAwy-H8J2-DCxO-9LZ9YB
PV Status        allocatable
Total PE / Free PE 1023 / 1023
```

Para ver los volúmenes físicos podemos ejecutar lo siguiente :

```
$ pvs
```

```
PV      VG      Fmt  Attr  Psize  PFree  
/dev/sdc1 rootvg  lvm2  a-    4,00g  4,00g
```

Mas detalladamente :

```
$ pvdisplay
```

```
--- Physical volume ---  
PV Name      /dev/sdc1  
VG Name      rootvg  
PV Size      4,00 GiB / not usable 2,66 MiB  
Allocatable  yes  
PE Size      4,00 MiB  
Total PE     1023  
Free PE      1023  
Allocated PE  0  
PV UUID      iAC40D-IBDC-C23I-zAwy-H8J2-DCxO-9LZ9YB
```

Ahora podemos crear el **volumen lógico (lv)**.

```
$ lvcreate -L +1G -n lvwww rootvg  
Logical volume "lvwww" created
```

+1G = Esto es el tamaño del volumen lógico puede ser en **G** (gigas), **M** (megas), y **T** (teras).
-n lvwww = **lvwww** es el nombre del volumen lógico.
rootvg = Nombre del grupo del volumen.

Si queremos utilizar todo el tamaño libre del volumen lógico utilizamos la opción **%FREE**.

```
$ lvcreate -L %FREE -n lvwww rootvg
```

Ahora podemos listar los volúmenes lógicos :

```
$ lvs
```

```
LV      VG      Attr  LSize  Origin Snap%  Move Log Copy%  Convert  
lvwww  rootvg  -wi-a- 1,00g
```

```
$ lvs -v
```

```
Finding all logical volumes  
LV      VG      #Seg Attr  LSize  Maj Min KMaj KMin Origin Snap%  Move Copy%  Log Convert LV UUID  
lvwww  rootvg  1 -wi-a- 1,00g  -1 -1 254 6          BANPSh-Yydb-FB1x-VfSR-82iQ-ft0n-emY57G
```

En forma mas detallada :

```
$ lvsdisplay
```

```
--- Logical volume ---  
LV Name      /dev/rootvg/lvwww  
VG Name      rootvg  
LV UUID      BANPSh-Yydb-FB1x-VfSR-82iQ-ft0n-emY57G  
LV Write Access  read/write  
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        254:6
```

/dev/rootvg/lvwww = Directorio completo del dispositivo.

Vemos que dentro del directorio **/dev/mapper** tenemos los nombre del **Volumen del Grupo** – **Volumen Lógico** que es un enlace al **dm-***.

```
$ ls -l /dev/mapper
lrwxrwxrwx 1 root root 7 sep 2 15:38 rootvg-lvwww -> ../dm-6
```

Una vez que terminamos de crear el volumen lógico vamos a formatear la partición, puede ser en **ext4**, **ext3**, **reiserfs**, etc.

```
$ mkfs.ext4 /dev/mapper/rootvg-lvwww
```

```
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
Stride=0 blocks, Stripe width=0 blocks
65536 nodos-i, 262144 bloques
13107 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de ficheros=268435456
8 bloque de grupos
32768 bloques por grupo, 32768 fragmentos por grupo
8192 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376

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 33 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

De esta forma ya podemos montar el volumen :

```
$ mkdir /www
$ mount /dev/rootvg/lvwww /www
```

Verificamos el espacio q tenemos :

```
$ df -h
S.ficheros      Size Used Avail Use% Montado en
/dev/mapper/rootvg-lvwww
    1008M  34M  924M   4% /www
```

Vamos a ampliar el filesystem creado es decir a agrandar el filesystem antes nos fijamos si tenemos espacio en el **vg**, ya que de ahí saca el espacio para agrandar, esto lo podemos hacer en caliente.

```
$ vgs
VG      #PV #LV #SN Attr   Vsize VFree
rootvg  1  1   0 wz--n- 4,00g 3,00g
```

En el ejemplo anterior vemos en nombre del grupo que se llama **rootvg**, tenemos un **PV** (volumen

físico), un **LV** (volumen lógico), con un tamaño total de 4GB y libre 3GB.

Como vemos el tamaño es de **4GB (Vsize)** y tenemos libre **3GB (Vfree)** para seguir expandiendo.

```
$ lvextend -L +1G /dev/rootvg/lvwww
```

0

```
$ lvresize -L +1G /dev/rootvg/lvwww
```

```
Extending logical volume lvwww to 2,00 GiB  
Logical volume lvwww successfully resized
```

Nuevamente vemos si se reflejo la expansión que realizamos.

```
$ df -h
```

```
S.ficheros      Size Used Avail Use% Montado en  
/dev/mapper/rootvg-lvwww  
1008M  34M  924M  4% /www
```

Como vemos aun no se reflejo el nuevo tamaño del disco, para eso tenemos que ejecutar otro comando.

```
$ resize2fs /dev/rootvg/lvwww
```

```
Filesystem at /dev/rootvg/lvwww is mounted on /www; on-line resizing required  
old desc_blocks = 1, new_desc_blocks = 1  
Performing an on-line resize of /dev/rootvg/lvwww to 524288 (4k) blocks.  
El sistema de ficheros en /dev/rootvg/lvwww tiene ahora 524288 bloques.
```

```
$ df -h
```

```
S.ficheros      Size Used Avail Use% Montado en  
/dev/mapper/rootvg-lvwww  
2,0G  34M  1,9G  2% /www
```

Manejo de Volumen Group

El comando **vgscan** analiza todos los dispositivos del disco compatible en busca de volúmenes físicos **LVM** y grupo de volúmenes. Esto creara la cache en el archivo **/etc/lvm/cache**. Este comando es ejecutado automáticamente cuando se inicia el sistema y en otros momentos durante la operación de **LVM**, por ejemplo cuando se ejecuta **vgcreate** o cuando **LVM** detecta una inconsistencia, si cambiamos el hardware tendremos que ejecutar **vgscan**, dentro del archivo **/etc/lvm/lvm.conf** se pueden definir filtros.

```
$ vgscan
```

```
Reading all physical volumes. This may take a while...  
Found volume group "VolGroup00" using metadata type lvm2  
Found volume group "rootvg" using metadata type lvm2
```

Como dijimos anteriormente cuando creamos un **vg** ya queda activo :

```
$ lvs
```

```
LV VG      Attr LSize  Origin Snap%  Move Log Copy%  Convert  
lvhome VolGroup00 -wi-a- 1,09g  
lvraiz VolGroup00 -wi-a- 1,48g  
lvswap VolGroup00 -wi-a- 244,00m  
lvtmp  VolGroup00 -wi-a- 1,04g  
lvusr  VolGroup00 -wi-a- 856,00m  
lvvar  VolGroup00 -wi-a- 3,21g
```

```
lvwww rootvg -wi-a- 1,49g
```

Acá vemos el carácter (**a**) que indica que esta activado. Si queremos desactivar un **vg** realizamos lo siguiente :

```
$ vgchange -a n rootvg  
0 logical volume(s) in volume group "rootvg" now active
```

```
$ lvs  
LV VG Attr LSize Origin Snap% Move Log Copy% Convert  
lvhome VolGroup00 -wi-a- 1,09g  
lvraiz VolGroup00 -wi-a- 1,48g  
lvswap VolGroup00 -wi-a- 244,00m  
lvtmp VolGroup00 -wi-a- 1,04g  
lvusr VolGroup00 -wi-a- 856,00m  
lvvar VolGroup00 -wi-a- 3,21g  
lvwww rootvg -wi--- 1,49g
```

Para volver activarlo :

```
$ vgchange -a y rootvg  
1 logical volume(s) in volume group "rootvg" now active
```

Si queremos migrar por ejemplo el **vg rootvg** al **vg VolGroup00**, primero tenemos que hacer que el **rootvg** quede desactivado y luego recién ahí poder migrarlo al **VolGroup00**.

```
$ vgs  
VG #PV #LV #SN Attr VSize VFree  
VolGroup00 1 6 0 wz--n- 8,00g 96,00m  
rootvg 2 1 0 wz--n- 4,09g 2,60g
```

```
$ vgchange -a n rootvg  
0 logical volume(s) in volume group "rootvg" now active
```

```
$ vgmerge -v VolGroup00 rootvg  
Checking for volume group "VolGroup00"  
Checking for volume group "rootvg"  
Archiving volume group "rootvg" metadata (seqno 30).  
Archiving volume group "VolGroup00" metadata (seqno 33).  
Writing out updated volume group  
Creating volume group backup "/etc/lvm/backup/VolGroup00" (seqno 34).  
Volume group "rootvg" successfully merged into "VolGroup00"
```

```
$ vgs  
VG #PV #LV #SN Attr VSize VFree  
VolGroup00 3 7 0 wz--n- 12,08g 2,69g
```

Como veremos nos creo una carpeta de backup **/etc/lvm/backup** que contiene los archivos **VolGroup00** y **rootvg** que contiene los metadatos. Podemos manualmente realizar las copias de seguridad de los metadatos con el comando **vgcfbackup**.

Con el siguiente comando veremos que realizamos un backup de todos los **vg**.

```
$ vgcfgbackup -v
```

Si queremos especificar un **vg** en particular realizamos lo siguiente :

```
$ vgcfgbackup -v nombre_vg
```

Supongamos que nos quedamos sin espacio en nuestro **vg** llamado **rootvg** la forma de agrandararlo es

mediante .

Asignar un disco al volumen group

Ahora lo que haremos es asignar un disco a un volumen group, este disco lo tenemos que tener de tipo LVM (8e).

```
$ vgextend rootvg /dev/sda1  
No physical volume label read from /dev/sda1  
Physical volume "/dev/sda1" successfully created  
Volume group "rootvg" successfully extended
```

Para sacar un disco de un vg

Lo primero que hay que hacer es mover la información del disco antes removerlo del volumen group, sino perdemos la información. Si queremos sacar el disco **/dev/sdd1** realizamos lo siguiente :

```
$ pvmove /dev/sdb1  
/dev/sdd1: Moved: 4,7%  
/dev/sdd1: Moved: 30,0%  
/dev/sdd1: Moved: 42,5%  
/dev/sdd1: Moved: 44,8%  
/dev/sdd1: Moved: 85,6%  
/dev/sdd1: Moved: 88,7%  
/dev/sdd1: Moved: 100,0%
```

Si queremos tener los datos en un volumen físico específico, agregamos como segundo argumento a **pvmove** el disco donde queremos que vaya la información :

```
$ pvmove /dev/sdb1 /dev/sdd1
```

Luego podemos remover el disco físico del volumen group.

```
$ vgreduce rootvg /dev/sdb1  
Removed "/dev/sdb1" from volume group "VolGroup00"
```

Veremos que el disco **sdb1** no está definido a ningún **vg**.

```
$ pvs  
PV          VG      Fmt  Attr  PSize  PFree  
/dev/sdb1           lvm2  a-   1,91g  1,91g
```

Por último, si quiere usar la partición para algo más, y quiere evitar a LVM, piense que esa partición es un volumen físico más:

```
$ pvremove /dev/sdb1  
Labels on physical volume "/dev/sdb1" successfully wiped
```

Si queremos remover todos los volúmenes físicos del grupo de volúmenes :

```
$ vgreduce --all rootvg
```

Renombrar un vg

Para renombrar un **vg** utilizamos el comando **vgrename**, por ejemplo renombramos el **vg VolGroup00** a **vg00**.

```
$ vgrename VolGroup00 vg00
```


rootvg = Nombre del grupo del volumen.

Si queremos utilizar todo el tamaño libre del volumen lógico utilizamos la opción **%FREE**.

```
$ lvcreate -L %FREE -n lvwww rootvg
```

Escaneamos los dispositivos de bloques

Para buscar los dispositivos de bloque que puede usar con volúmenes físicos con el comando **lvmdiskscan**.

```
$ lvmdiskscan  
/dev/loop0 [ 1,11 GiB]  
/dev/dm-0 [ 1,49 GiB]  
/dev/sda1 [ 94,00 MiB] LVM physical volume  
/dev/dm-1 [ 1,09 GiB]  
/dev/sda2 [ 1,91 GiB]  
/dev/dm-2 [ 1,48 GiB]  
/dev/dm-3 [ 244,00 MiB]  
/dev/dm-4 [ 1,04 GiB]  
/dev/dm-5 [ 856,00 MiB]  
/dev/dm-6 [ 3,21 GiB]  
/dev/sdb1 [ 8,00 GiB] LVM physical volume  
/dev/sdc1 [ 4,00 GiB] LVM physical volume  
0 disks  
9 partitions  
0 LVM physical volume whole disks  
3 LVM physical volumes
```

Para ver los discos físicos

```
$ lvmdiskscan -l  
WARNING: only considering LVM devices  
/dev/sda1 [ 94,00 MiB] LVM physical volume  
/dev/sdb1 [ 8,00 GiB] LVM physical volume  
/dev/sdc1 [ 4,00 GiB] LVM physical volume  
0 LVM physical volume whole disks  
3 LVM physical volumes
```

Ver los lv

```
$ lvs  
LV VG Attr LSize Origin Snap% Move Log Copy% Convert  
lvhome VolGroup00 -wi-a- 1,09g  
lvraiz VolGroup00 -wi-a- 1,48g  
lvswap VolGroup00 -wi-a- 244,00m  
lvtmp VolGroup00 -wi-a- 1,04g  
lvusr VolGroup00 -wi-a- 856,00m  
lvvar VolGroup00 -wi-a- 3,21g  
lvwww rootvg -wi--- 1,49g
```

Mostrar en forma mas detallada :

```
$ lvsdisplay  
--- Logical volume ---  
LV Name /dev/rootvg/lvwww  
VG Name rootvg  
LV UUID BANPSh-Yydb-FB1x-VfSR-82iQ-fT0n-emY57G
```

```
LV Write Access    read/write
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      254:6
```

Reducir filesystem

Para poder reducir el filesystem, no debe estar montado, dado que de lo contrario da un error al intentar la ejecución.

```
$ umount /www
```

Chequeamos que todo este andando bien, antes de hacer algo.

```
$ e2fsck -f /dev/rootvg/lvwww
```

```
e2fsck 1.41.12 (17-May-2010)
```

```
Paso 1: Verificando nodos-i, bloques y tamaños
```

```
Paso 2: Verificando la estructura de directorios
```

```
Paso 3: Revisando la conectividad de directorios
```

```
Paso 4: Revisando las cuentas de referencia
```

```
Paso 5: Revisando el resumen de información de grupos
```

```
/dev/rootvg/lvwww: 11/32768 ficheros (0.0% no contiguos), 18833/128000 bloques
```

A continuación reducimos el sistema de ficheros.

```
$ lvreduce -L 800M /dev/rootvg/lvwww
```

```
WARNING: Reducing active logical volume to 800,00 MiB
```

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

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

```
Reducing logical volume lvwww to 800,00 MiB
```

```
Logical volume lvwww successfully resized
```

Ahora vamos reducir el tamaño del volumen físico, el nuevo tamaño sera de **800MB**.

```
$ resize2fs /dev/rootvg/lvwww 800M
```

```
resize2fs 1.41.12 (17-May-2010)
```

```
Resizing the filesystem on /dev/rootvg/lvwww to 204800 (4k) blocks.
```

```
El sistema de ficheros en /dev/rootvg/lvwww tiene ahora 204800 bloques.
```

Volvemos a chequear que todo este andando bien, antes de hacer algo.

```
$ e2fsck -f /dev/rootvg/lvwww
```

```
e2fsck 1.41.12 (17-May-2010)
```

```
Paso 1: Verificando nodos-i, bloques y tamaños
```

```
Paso 2: Verificando la estructura de directorios
```

```
Paso 3: Revisando la conectividad de directorios
```

```
Paso 4: Revisando las cuentas de referencia
```

```
Paso 5: Revisando el resumen de información de grupos
```

```
/dev/rootvg/lvwww: 11/32768 ficheros (0.0% no contiguos), 18833/128000 bloques
```

Cambiando parámetros de un lv

```
$lvchange -pr rootvg/lvwww  
Logical volume "lvwww" changed
```

```
$lvs  
LV VG Attr LSize Origin Snap% Move Log Copy% Convert  
lvhome VolGroup00 -wi-a- 1,09g  
lvraiz VolGroup00 -wi-a- 1,48g  
lvswap VolGroup00 -wi-a- 244,00m  
lvtmp VolGroup00 -wi-a- 1,04g  
lvusr VolGroup00 -wi-a- 856,00m  
lvvar VolGroup00 -wi-a- 3,21g  
lvwww rootvg -ri-a- 1,49g
```

En este caso vemos que cambiamos el **lv** a solo lectura (r). Para volver a poner lo como escritura :

```
$lvchange -prw rootvg/lvwww  
Logical volume "lvwww" changed
```

Renombrar un lv

```
$ lvrename vg00/lvwww vg00/lvwww-2  
Renamed "lvwww" to "lvwww-2" in volume group "vg00"
```

Remover un lv

Primero tenemos que desmontar el filesystem donde tengamos montado **lvwww-2**.

```
$ lvremove vg00/lvwww-2  
Do you really want to remove active logical volume lvwww-2? [y/n]: y  
Logical volume "lvwww-2" successfully removed
```

Ver los lv activos

```
$ lvscan  
ACTIVE /dev/vg00/lvraiz' [1,48 GiB] inherit  
ACTIVE /dev/vg00/lvswap' [244,00 MiB] inherit  
ACTIVE /dev/vg00/lvtmp' [1,04 GiB] inherit  
ACTIVE /dev/vg00/lvusr' [856,00 MiB] inherit  
ACTIVE /dev/vg00/lvvar' [3,21 GiB] inherit
```

Extender un lv

En el siguiente ejemplo

```
$ lvextend -L +1G /dev/rootvg/lvwww  
o  
$ lvresize -L +1G /dev/rootvg/lvwww  
  
Extending logical volume lvwww to 2,00 GiB  
Logical volume lvwww successfully resized
```

Crear un snapshot lv

Creamos el lv.

```
$ lvcreate -L +1G lvwww vg00
```

Logical volume "lvwww" created

Ahora creamos el snapshot.

```
$ lvcreate --size 100M --snapshot --name lvwww-snap /dev/vg00/lvwww
Logical volume "lvwww-snap" created
```

Para verificar realizamos el siguiente comando :

```
$ lvs
LV      VG Attr LSize  Origin Snap% Move Log Copy% Convert
lvraiz  vg00 -wi-a- 1,48g
lvswap  vg00 -wi-a- 244,00m
lvtmp   vg00 -wi-a- 1,04g
lvusr   vg00 -wi-a- 856,00m
lvvar   vg00 -wi-a- 3,21g
lvwww   vg00 owi-a- 1,00g
lvwww-snap vg00 swi-a- 100,00m lvwww 0,01
```

Para mas detalle :

```
$ lvsdisplay /dev/vg00/lvwww

--- Logical volume ---
LV Name           /dev/vg00/lvwww
VG Name           vg00
LV UUID           FZmaZL-iGyY-EmiK-V37H-cT13-klWo-1UCRK8
LV Write Access   read/write
LV snapshot status source of
                  /dev/vg00/lvwww-snap [active]

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      254:0
```