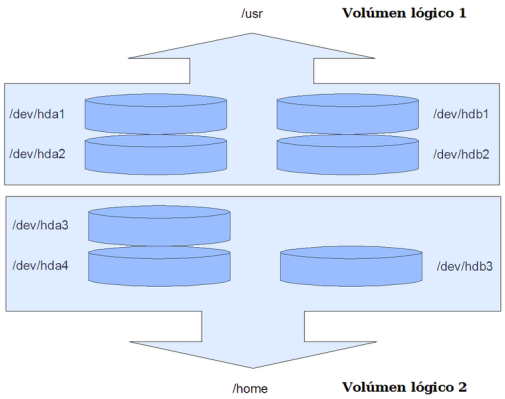
LVM

Teoria conceptual i comandes bàsiques:

LVM, or Logical Volume Management, is a storage device management technology that gives users the power to pool and abstract the physical layout of component storage devices for easier and flexible administration. LVM is used, overall, to combine existing storage devices (disks or regular partitions) into "logical groups" and, then, allocate logical volumes ("LVM partitions") from that combined logical space, as needed. This means that the logical units (which can have a meaningful name like "databases" or "root-backup") could be extended through several disks/regular partitions, thus being, in fact, greater than them. Moreover, another interesting characteristic of LVM logical units is that their size can be extended or shrinked "on the fly" above same (or different!) disks without needing to unmount anything. Also, new physical devices can be added "on the fly" to the "logical group" in order to extend its global size. So, in summary, the main advantages of LVM are increased abstraction, flexibility, and control.

NOTA: LVM es capaz de realizar muchas más cosas. Por emplo, inicialmente LVM no incluía funciones RAID, por lo que en caso de necesitar tolerancia a fallos normalmente se utilizaba encima de volumenes RAID, pero en la actualidad LVM incorpora soporte RAID integrado para los niveles 0, 1, 4, 5, 6 y 10. Otra de las funcionalidades "extra" de LVM es el modo "thin-provisioning", en el que no se reserva espacio físico de almacenamiento hasta que no se realiza la escritura de datos. Otra funcionalidad también interesante es la capacidad de realizar instantáneas del estado del sistema de ficheros en modo lectura y escritura, etc, etc



En la terminología de LVM se utilizan los siguientes conceptos:

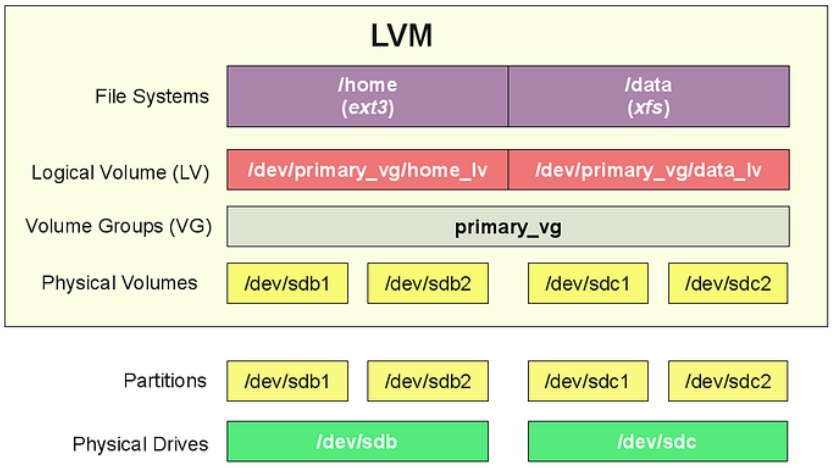
*Volumenes físicos (*PV*): Son discos duros, particiones o cualquier otro dispositivo de bloques (como un volumen RAID, un archivo "loop", etc) sobre los que se "planchará" un grupo VG. LVM los debe preparar previamente.

*Grupos de volúmenes (*VG*): Una colección de volúmenes físicos que representa un sustrato homogéneo sobre el que se definirán los volúmenes lógicos. Volume groups abstract the characteristics of the underlying devices and function as a unified logical device with combined storage capacity of the component physical volumes.

*Volumenes lógicos (*LV*): Es el equivalente a una partición en un sistema tradicional pero con las ventajas de LVM ya comentadas (tamaño dinámico, extendido sobre varios PV, etc). Se ve como un almacén que puede contener un sistema de archivos. They are the primary component that users and applications will interact with.

NOTA: Cuando se redimensionan volúmenes lógicos es necesario redimensionar también el sistema de archivos que contienen. Concretamente, al extender el volumen lógico primero se ha de extender el volumen lógico y luego el sistema de archivos contenido; y al reducir el volumen lógico primero se ha de reducir el sistema de archivos y luego el volumen lógico. Afortunadamente, en la mayoría de ocasiones LVM es capaz de realizar estos procesos automáticamente.

In summary, LVM can be used to combine physical volumes into volume groups to unify the storage space available on a system. Afterwards, administrators can segment the volume group into arbitrary logical volumes, which act as flexible partitions.



Each volume within a volume group is segmented into small, fixed-size chunks called **extents**. The size of the extents is determined by the volume group (all volumes within the group conform to the same extent size). The extents on a physical volume are called physical extents, while the extents of a logical volume are called logical extents. A logical volume is simply a mapping that LVM maintains between logical and physical extents. Because of this relationship, the extent size represents the smallest amount of space that can be allocated by LVM. Extents are behind much of the flexibility and power of LVM. The logical extents that are presented as a unified device by LVM do not have to map to continuous physical extents. LVM can copy and reorganize the physical extents that compose a logical volume without any interruption to users. Logical volumes can also be easily expanded or shrunk by simply adding extents to or removing extents from the volume.

Creació d'una estructura LVM:

We will start by walking through a basic procedure that will use two physical disks to form four logical volumes (within one main volume group):

1.-Scan the system for block devices that LVM can see and manage:
`sudo lvmdiskscan`
1BIS.-Mark desired physical devices as physical volumes within LVM (This will write an LVM header to the devices to indicate that they are ready to be added to a volume group):
`sudo pvcreate /dev/sdb /dev/sdc`
1TRIS.-Verify that LVM has registered the physical volumes (devices should be present under the `PV` column):
`sudo pvs`

NOTA: The `sudo lvmdiskscan -l` command only shows known PVs. Another similar command which returns additional information is `sudo pvscan`. But if you need more -human readable- detail, the `sudo pvdisplay` command is a better option (or even a customized `pvs` 's output).
NOTA: To discover the logical extents that have been mapped to each volume, pass in the `-m` option to `pvdisplay`

2.-Create the volume group, give it a name and add both of our physical volumes to it:
`sudo vgcreate miVG /dev/sdb /dev/sdc`
2BIS.-If we check the `sudo pvs` output again, we can see that our PVs are now associated with new VG. We can also see a brief summary of the VG itself (number of PV, number of LV -none yet-, global size, etc) by typing:
`sudo vgs`

NOTA: The `sudo vgscan` scan the system for available volume groups and outputs minimal information about VGs it finds. But if you need more -human readable- detail, the `sudo vgdisplay` command is a better option (or even a customized `vgs` 's output; for example, to show the physical devices and the logical volume path you could execute `sudo vgs -o +devices,lv_path`).
NOTA: Adding the `-v` flag to `vgdisplay` also provides information about the physical volumes the volume group is built upon, and the logical volumes that were created using the volume group
NOTA: Adding the `-s` flag (followed by a size value like "8M" or so) to `vgcreate` is useful to specify the desired extend size to assign to this volume group. By default this size is 4M and it's a reasonable value for the majority of use cases.

3.-To create LVs you only need to supply a name and a size (and the VG where they belong). We'll create four separate logical volumes: a 10G volume called "projects", a 5G volume called "www" , a 20G volume called "db" and a volume that will fill the remaining space called "workspace":
```
sudo lvcreate -L 10G -n projects miVG
sudo lvcreate -L 5G -n www miVG
sudo lvcreate -L 20G -n db miVG
sudo lvcreate -l 100%FREE -n workspace miVG
```
NOTA: The -l option specifies the LV size in terms of number of extents (instead of the -L one, which specifies it in {G,M,K}bytes) The value 100%FREE means "all the necessary extends in order to complete all the free remaining space". Other value is nº%VG
3BIS.-See the logical volumes and their relationship to the volume group by customizing `vgs` output:
`sudo vgs -o +lv_size,lv_name`

NOTA: The `sudo lvscan` scan the system for available logical volumes and outputs minimal information about LVs it finds. But if you need more -human readable- detail, the `sudo lvdisplay` command is a better option (or even `sudo lvs` 's output).
NOTA: Adding the `-m` flag to lv`display` also provides information about how the logical volume is broken down and distributed.

4.-Now that we have logical volumes, we can use them as normal block devices. The logical volumes are available within the /dev directory just like other storage devices. You can access them in two equivalent ways: using the "/dev/VGname/LVname" path or the "/dev/mapper/VGname-LVname" path. So to format our four logical volumes with the Ext4 filesystem, you can type:
```
sudo mkfs.ext4 /dev/miVG/projects  &&  sudo mkfs.ext4 /dev/miVG/www
sudo mkfs.ext4 /dev/miVG/db &&  sudo mkfs.ext4 /dev/miVG/workspace
```

5.-After formatting, you should create mount points...:

```
sudo mkdir -p /mnt/{projects,www,db,workspace}
```

5BIS...and then mount the logical volumes to the appropriate location:

```
sudo mount /dev/miVG/projects /mnt/projects && sudo mount /dev/miVG/www /mnt/www
sudo mount /dev/miVG/db /mnt/db && sudo mount /dev/miVG/workspace /mnt/workspace
```

6.-To make the mounts persistent, add them to /etc/fstab just like you would with normal block devices:

/dev/miVG/projects /mnt/projects ext4 defaults,nofail 0 0
/dev/miVG/www /mnt/www ext4 defaults,nofail 0 0
/dev/miVG/db /mnt/db ext4 defaults,nofail 0 0
/dev/miVG/workspace /mnt/workspace ext4 defaults,nofail 0 0

Tasques bàsiques de manteniment de LVMs (redimensionar PVs, VGs o LVs):

Aunque no es corriente que un dispositivo “físico” de almacenamiento pueda crecer, este caso se puede dar cuando se trata de una máquina virtualizada, en la que podríamos aumentar el tamaño su disco, o también en una máquina en la que ampliemos una partición tradicional usada como PV. En cualquier caso, si estamos en esta situación donde el dispositivo físico ha aumentado de tamaño, es necesario indicar a LVM que redimensione su correspondiente PV hasta alcanzar el mismo tamaño final del dispositivo físico subyacente. Para ello hay que ejecutar (suponiendo que el dispositivo físico sea /dev/sdb):

```
sudo pvresize /dev/sdb
```

NOTA: Esta operación (la ampliación del PV), como las que siguen, se puede realizar sin problemas con todo el sistema corriendo, sin tener que parar ningún servicio o desmontar sistema de ficheros.

Obviamente, el hecho de redimensionar un PV implica de forma automática que el VG al que pertenece será redimensionado en la misma cantidad de tamaño. No obstante, existe otra forma de redimensionar un VG, que es simplemente concatenándole un nuevo (o más) PVs adicionales, así:

```
sudo vgextend miVG /dev/sdd /dev/sde
```

Una vez tengamos un VG ampliado, para ocupar este nuevo espacio mediante cualquier LV existente (no importa qué otros LVs tenga "delante" o "detrás" porque estos conceptos en LVM no existen) deberemos redimensionarlo también convenientemente, así (en este ejemplo se está aumentando en 20G el tamaño actual del LV indicado):

```
sudo lvresize -r -L +20G /dev/miVG/www
```

NOTA: El valor del parámetro -L también puede ser un valor absoluto (-L 200G) para indicar el nuevo tamaño del LV
NOTA: El paràmetro -r sirve para redimensionar también, a la vez que el LV, el sistema de ficheros contenido en él. Si no se indicara, ese paso se debería hacer manualmente (lo que generalmente implica desmontar previamente el sistema de ficheros para volverlo a montar una vez concluido el proceso) y dependería del sistema de ficheros en cuestión (por ejemplo, en sistemas Ext4 el comando a usar es *resize2fs /dev/miVG/www* , en sistemas XFS es *xfs_growfs /dev/miVG/www*)
NOTA: Existe otro comando con prácticamente idénticos parámetros llamado *lvextend* , el cual, no obstante, únicamente es capaz de aumentar el tamaño del LV pero no disminuirlo

En el caso de querer reducir el tamaño de un LV, hay que ir con cuidado porque primero deberemos comprobar cuándo espacio ocupado actualmente tenemos en él (mediante `df -h`, por ejemplo) ya que el tamaño final no podrá ser nunca inferior a este bajo el peligro de perder datos. Seguidamente, deberemos desmontar el sistema de ficheros: a diferencia de las ampliaciones, las reducciones de tamaño no se pueden realizar "en caliente".

```
sudo umount /dev/miVG/www
```

NOTA:After unmounting, it's good idea to check the filesystem to ensure that everything is in working order with `fsck` command. Specifically, pass in the filesystem type with the -t option and use -f to check even when the filesystem appears okay, like this: `sudo fsck -t ext4 -f /dev/miVG/www`

Now you should reduce the filesystem size using the filesystem's native tools. For Ext4 filesystems, this would be the *resize2fs* command. Pass in the final size for the filesystem:

```
sudo resize2fs -p /dev/miVG/www 3G
```

Once the operation is complete, resize the logical volume by passing the same size to the *lvresize* command:

```
sudo lvresize -L 3G /dev/miVG/www
```

After the logical volume has been reduced, check the filesystem again with same `fsck` command and, if everything is functioning correctly, you can remount the filesystem using your usual mount command.

Tasques bàsiques de manteniment de LVMs (eliminar PVs, VGs o LVs):

If you no longer need a logical volume, you can remove it. To do so, first you should unmount that LV if it is currently mounted...:

```
sudo umount /dev/miVG/www
```

...and then, you should remove the LV by typing:

```
sudo lvremove /dev/miVG/www
```

To remove an entire volume group, first you should usually remove all its logical volumes using the procedure above. Afterwards, you can delete the entire volume group by typing:

```
sudo vgremove miVG
```

NOTA:You will be prompted to confirm that you wish to remove the volume group. If you have any logical volumes still present, you will be given individual confirmation prompts for those before removing.

If you wish to remove a physical volume from LVM management, the procedure you will need depends on whether the device is currently being used by LVM. If the physical volume is in use, first you will have to move the physical extents located on the device to a different location. This requires the volume group to have enough other physical volumes to handle the physical extents. If it is the case, move them off of the physical volume you wish to remove by typing:

```
sudo pvmove /dev/sda
```

Once the extents have been relocated to peer volumes, you can remove the physical volume from the volume group by typing:

```
sudo vgreduce miVG /dev/sda
```

After this is complete, you can remove the physical volume marker from the storage device by typing:

```
sudo pvremove /dev/sda
```

You should now be able to use the removed storage device for other purposes or remove it from the system entirely.

Usar RAID integrat:

Logical volumes can be created with some RAID level, which must be specified as a value of the *--type* parameter in *lvcreate* command. Some of these values are:

***"**linear": The default type (not RAID at all). The underlying physical devices used (if more than one) will simply be appended to each other, one after the other.

*"striped": Similar to RAID 0, the striped topology divides data into chunks and spread in a round-robin fashion across the underlying physical volumes. This type requires a minimum of two physical volumes and the adding of the *-i* parameter, which specifies the exact number of stripes that should be maintained (the minimum is 2). The size of each stripe can be specified by *-I* parameter (in KB).

*"raid1": Creates a mirrored RAID 1 volume. By default, the mirror will have two copies, but more can be specified by the *-m* parameter (this option specifies the number of additional copies of data to keep so a value of "1" specifies that one additional copy is maintained, for a total of two sets of data). Requires a minimum of two physical volumes (-m=1).

*"raid5": Creates a RAID 5 volume. Requires a minimum of three physical volumes.

*"raid6": Creates a RAID 6 volume. Requires a minimum of four physical volumes.

For instance, to create a striped volume:

```
sudo lvcreate --type striped -i 2 -L 10G -n striped_vol miVG
```

To create a mirrored volume:

```
sudo lvcreate --type raid1 -m 2 -L 20G -n mirrored_vol miVG
```

Usar snapshots:

Una snapshot es un volumen lógico que representa la instantánea en un momento dado del contenidor de otro volumen lógico. A partir de su creación, no obstante, solo almacenará los cambios que se vayan produciendo respecto el volumen lógico original, por lo que su tamaño puede ser mucho más reducido que el de este. Para crear un snapshot de un determinado LV se ha de ejecutar el siguiente comando (notar el uso del parámetro -s y de que como último valor no se está indicando ningún VG sino el LV base):

```
sudo lvcreate -s -L 10G -n www_cambios miVG/www
```

Una vez creado el snapshot, se deberá montar como cualquier otro LV para poder ser utilizado:

```
sudo mount /dev/miVG/www_cambios /mnt/www_cambios
```

A partir de este momento, en "/mnt/www_cambios" aparece lo mismo que en "/mnt/www" pero a medida que se realizan cambios su contenido se irá distanciando, sin que los cambios en un volúmen lógico afecten al otro.

NOTA: You can bring the base LV back to the state when the snapshot was taken simply by typing:
`sudo lvconvert --merge /dev/miVG/www_cambios`