

Parmanent mounting and LVM

understanding LVM

lvm is short for logical volume manager. it allows to create groups of disks from various drive in to a single filesystem or can be a multiple filesystem

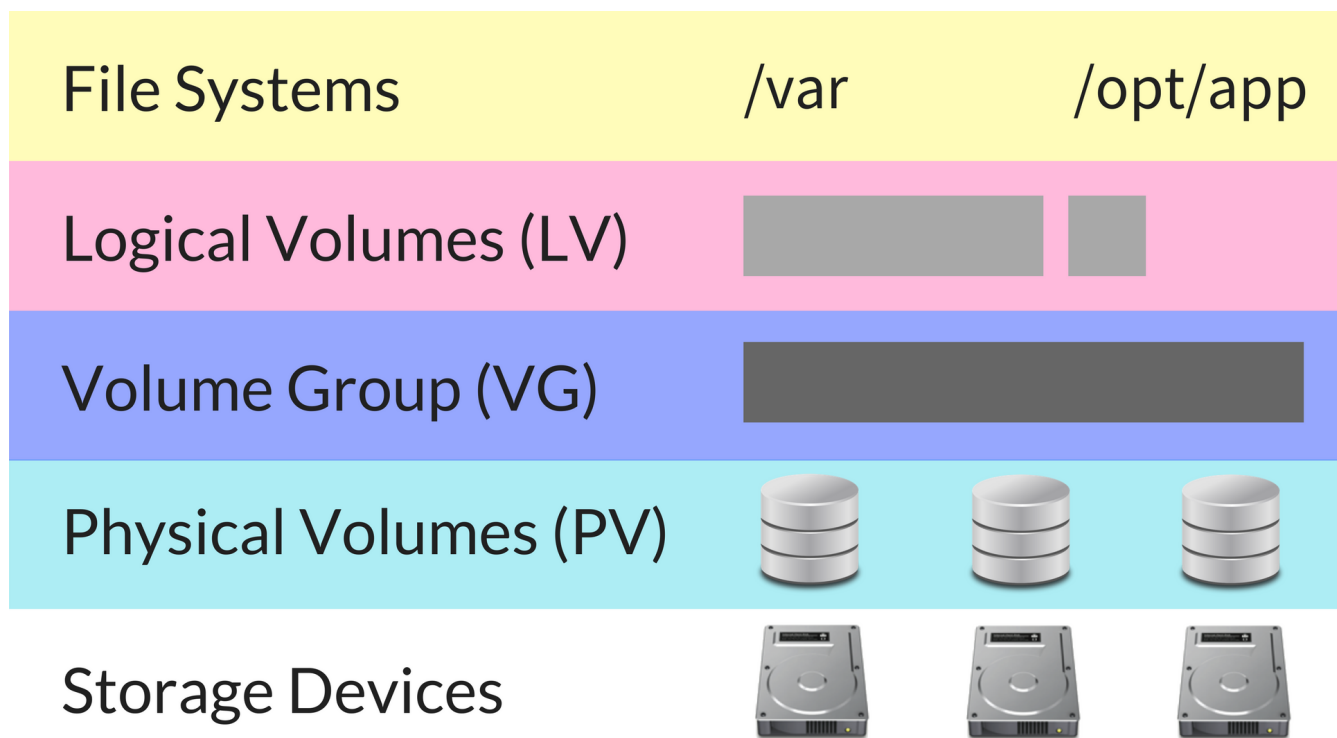
you can lavm for any mount point except for /boot. /boot has to be a regular filesystem. because the GRUB can read a LVM filesystem.

Why LVM:

the first great things about lvm is the flexibility of the system.
We can resize the volume

- 1) we can shrink a volume to reduce unuser space
- 2) we can expand the volume
- 3) we can take snapshots for creating backup

layout of the LVM:



from the picture we can see at the bottom there is storage drive from that we can create the physical volume (PV) which can be as `/dev/sdb` `/dev/sda` `/dev/sdc` which are the actual disk. Above the PV we have the VG or volume group. we can create a group of volume from multiple disk or can be single disk (in picture showing multiple disk). and on the top of our volume group we have our logical volume (LV). this is where we slice our volume group which are like the traditional partition. above the logical volume (LV) we have our actual partition.

Create a logical volume:

first we should know that we can resize the logical volume between the size of the disk. we can do multiple disk lvm but in the example we create lvm in one physical volume

Commands are

1) first we have to create a disk and we have give id “8e” which is the id of the linux lvm.

```
[vagrant@localhost ~]$ sudo fdisk /dev/sdc  
Welcome to fdisk (util-linux 2.23.2).
```

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Command (m for help): p

Disk /dev/sdc: 32.2 GB, 32212254720 bytes, 62914560 sectors

Units = sectors of 1 * 512 = 512 bytes

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

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

Disk label type: dos

Disk identifier: 0x73e2fe56

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

Command (m for help): n

Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

Select (default p): p

Partition number (1-4, default 1):
First sector (2048-62914559, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-62914559, default 62914559): +20G
Partition 1 of type Linux and of size 20 GiB is set

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'

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

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

```
[vagrant@localhost ~]$  
[vagrant@localhost ~]$ sudo partprobe /dev/sdc  
[vagrant@localhost ~]$
```

now disk create complete

2) we have to create a physical volume
command is

```
[vagrant@localhost ~]$ sudo pvcreate /dev/sdc1  
Physical volume "/dev/sdc1" successfully created
```

3) we can use the command 'pvdisplay' to see the physical volume.

4) now we have to create the upper portion which is the volume group

command syntax is

`vgcreate <name> <path>`

```
[vagrant@localhost ~]$ sudo vgcreate new_vg /dev/sdc1  
Volume group "new_vg" successfully created
```

5) now we create the lvm on the volume group 'new_vg'

command syntax is

`lvcreate -L <length> <path_of_volume_group>`

```
[vagrant@localhost ~]$ sudo lvcreate -L 4G /dev/new_vg  
Logical volume "lvol0" created.
```

It will automatically give name to the logical volume

6) we have to create the filesystem

```
[vagrant@localhost ~]$ sudo mkfs.ext4 /dev/new_vg/lvol0
```

work done now we have to mount it to see if it is working

```
[vagrant@localhost ~]$ sudo mkdir /lv_folder  
[vagrant@localhost ~]$ sudo mount /dev/new_vg/lvol0  
/lv_folder/  
[vagrant@localhost ~]$ cd /lv_folder/  
[vagrant@localhost lv_folder]$ ls  
lost+found
```

done.....

extending the size:

to resize we must first unmount the file system.

```
[vagrant@localhost /]$ sudo umount lv_folder/  
[vagrant@localhost /]$
```

to extend the existing size of the partition
the command syntax is

```
lvextend -L +<size> <path_of_the_lvm>
```

```
[vagrant@localhost /]$ sudo lvextend -L +500MB  
/dev/new_vg/lvol0
```

**Size of logical volume new_vg/lvol0 changed from 4.00 GiB (1024 extents) to 4.49 GiB (1149 extents).
Logical volume lvol0 successfully resized**

then we have to check the new added space

```
[vagrant@localhost /]$ e2fsck -f /dev/new_vg/lvol0
```

after that we have to do the filesystem thing

```
[vagrant@localhost /]$ sudo resize2fs /dev/new_vg/lvol0  
resize2fs 1.42.9 (28-Dec-2013)  
Resizing the filesystem on /dev/new_vg/lvol0 to 1176576  
(4k) blocks.  
The filesystem on /dev/new_vg/lvol0 is now 1176576  
blocks long.
```

Lvm shrink:

To reduce the command is

```
lvreduce -L -<size> <location>
```

```
[vagrant@localhost /]$ sudo lvreduce -L 200MB  
/dev/new_vg/lvol0
```

**WARNING: Reducing active logical volume to 200.00 MiB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lvol0? [y/n]: y**

**\Size of logical volume new_vg/lvol0 changed from 4.49
GiB (1149 extents) to 200.00 MiB (50 extents).
Logical volume lvol0 successfully resized
[vagrant@localhost ~]\$**

then we have to mount it to a folder

permanent mounting:

all the mounting we have done will be unmounted automatically after the reboot .to make it permanent we have to add it to the */etc/fstab* file

[vagrant@localhost ~]\$ sudo vim /etc/fstab