

Logical Volume Management

The Linux Logical Volume Manager (LVM) is a mechanism to virtualize the disks. It can create "virtual" disk partitions out of one or more physical hard drives, allowing you to grow, shrink, or move those partitions from drive to drive as your needs change.

Components of LVM in Linux:

1. Physical Volumes (PV)
2. Physical Extent (PE)
3. Volume Group (VG)
4. Logical Volume (LV)
5. Logical Extent (LE)

Physical Volume (PV):

It is the standard partition that you add to the LVM. Normally, a physical volume is a standard primary or logical partition with the hex code 8e.

Physical Extent (PE):

It is a chunk of disk space. Every PV is divided into a number of equal sized PEs.

Volume Group (VG):

It is composed of a group of PV's and LV's. It is the organizational group for LVM.

Logical Volume (LV) is composed of a group of LEs. You can format and mount any file system on an LV. The size of these LV's can easily be increased or decreased as per the requirement.

Logical Extent (LE):

It is also a chunk of disk space. Every LE is mapped to a specific PE.

LVM Command	Function
pvs	Displays all the physical volumes
vgs	Displays all volume groups in the system
lvs	Displays all the logical volumes in the system
pvdisk	Displays detailed information on physical volumes
vgdisplay	Displays detailed information on volume groups
lvdisplay	Displays detailed information on logical volumes
pvcreate	Create a new physical volume
vgcreate	Create a new volume group.
lvcreate	Creates a new logical volume
vgextend	Add a new physical disk to a volume group.
lvextend	Extends a logical volume
lvresize	Resizes a logical volume
lvreduce	Reduces a logical volume
pvmove	Moves/migrates data from one physical volume to another
vgreduce	Reduces a volume group by removing a PV from it.
pvremove	Deletes a physical volume
vgremove	Removes /Deletes a volume group
lvremove	Removes /Deletes a logical volume

Creating a Physical Volume (PV):

Note: add a disk with 10GB and scan. Latter follow the below steps.

-> Create a partition using fdisk, and change the hex code of it to 8e.

-> Save and exit the fdisk and update the partition table using partx -a command

```
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/sdd: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 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: 0x1cb3f450

   Device Boot      Start         End      Blocks    Id  System
/dev/sdd1          1         132     1060258+   8e  Linux LVM

Command (m for help): w
```

➔ Create a PV on newly created partition i.e. /dev/sdd1

root@master-server:~

```
[root@master-server ~]# pvs
  PV          VG          Fmt  Attr  PSize  PFree
  /dev/sda2   vg_masterserver lvm2 a-   19.51g    0
[root@master-server ~]# pvcreate /dev/sdd1
Physical volume "/dev/sdd1" successfully created
[root@master-server ~]# pvs
  PV          VG          Fmt  Attr  PSize  PFree
  /dev/sda2   vg_masterserver lvm2 a-   19.51g    0
  /dev/sdd1    vg_masterserver lvm2 a-   1.01g  1.01g
[root@master-server ~]#
```

→

→ Creating a Volume Group (VG):

→ After creating a PV, the next step is to create a Volume Group or VG

→ To create a VG the syntax is

→ # vgcreate <name of the VG> <partitionname>

→ #vgcreate yallavg /dev/sdd1

root@master-server:~

```
[root@master-server ~]# vgcreate yallavg /dev/sdd1
Volume group "yallavg" successfully created
[root@master-server ~]# vgs
  VG          #PV #LV #SN Attr   VSize  VFree
  vg_masterserver    1   2   0 wz--n- 19.51g    0
  yallavg            1   0   0 wz--n-  1.01g  1.01g
[root@master-server ~]#
```

Logical Volume Creation:

→ Once we are ready with a **Volume Group** then it's the time to create a **Logical Volume LV**.

→ The syntax for creating an LV is

→ #lvcreate -L <size of the LV> -n <name of the LV> <vgname>

→ #lvcreate -L 1G -n yalllv yallavg


root@master-server:~

```
[root@master-server ~]# lvcreate -L 1G -n yallalv yallavg
Logical volume "yallalv" created
[root@master-server ~]# lvs
LV      VG          Attr   LSize   Origin Snap%   Move Log Copy%   Convert
lv_root vg_masterserver -wi-ao 15.60g
lv_swap vg_masterserver -wi-ao  3.91g
yallalv yallavg    -wi-a-  1.00g
[root@master-server ~]#
```

Adding File system to the LV and Mounting it.:

- ➔ As per now we have our VG created so is our LV. In order make it accessible we need to format it with a file system like ext4 or ext3 or vfat.
- ➔ The syntax for formatting an LV is exactly like formatting a normal partition, Instead of /dev/partition name we use the path of LV that will be something like /dev/vg/lv
- ➔ #mkfs.ext4 /dev/vgname/lvname
- ➔ #mkfs.ext4 /dev/yallavg/yallalv

Note: before format check #blkid command. Then format

 root@master-server:~


```
[root@master-server ~]# mkfs.ext4 /dev/yallavg/yallalv
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 38 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@master-server ~]#
```

Mounting:

- ➔ Mounting an LV is exactly same like a normal partition, again the path for mounting will be /dev/vg/lv
- ➔ Create a directory over which the LV should be mounted. #mkdir /yalla-app
- ➔ #mount /dev/vgname/lvname /dirname
- ➔ #mount /dev/yallavg/yallalv/ /yalla-app

 root@master-server:~

```
[root@master-server ~]# mkdir /yalla-app
[root@master-server ~]# pwd
/root
[root@master-server ~]#
```

root@master-server:~

```
[root@master-server ~]# mount /dev/yallavg/yallalv /yalla-app/
[root@master-server ~]# df -h /yalla-app/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/yallavg-yallalv
                          1008M    34M   924M   4% /yalla-app
[root@master-server ~]#
```

➔ Make it a permanent mount by making an entry in /etc/fstab

Note: any changes in configuration file. Take a backup of file.

root@master-server:~

```
[root@master-server ~]# cp /etc/fstab /etc/fstab-bkp
[root@master-server ~]# vi /etc/fstab
[root@master-server ~]# tail -1 /etc/fstab
/dev/yallavg/yallalv    /yalla-app            ext4    defaults    0 0
[root@master-server ~]#
```

Extending a Volume Group:

- ➔ Extending a volume group is actually adding a new PV to the volume group.
- ➔ To extend a volume group we need to create a new partition using fdisk. Don't forget to change its hex code to 8e and update the partition table using partx -a command

Command (m for help): p

Disk /dev/sdd: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 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: 0x1cb3f450

Device	Boot	Start	End	Blocks	Id	System
/dev/sdd1		1	132	1060258+	8e	Linux LVM
/dev/sdd2		133	264	1060290	8e	Linux LVM

Command (m for help): w

➔ Create a PV on the newly created partition using pvcreate command

➔

#pvcreate /dev/sdd2

root@master-server:~

```
[root@master-server ~]# fdisk -ll /dev/sde

Disk /dev/sde: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0x7f4059be

   Device Boot      Start         End      Blocks   Id  System
/dev/sde1            1          132     1060258+   8e  Linux LVM
[root@master-server ~]# pvcreate /dev/sde1
Physical volume "/dev/sde1" successfully created
[root@master-server ~]#
```

- ➔ Add the partition to the VG using vgextend command, the syntax for it is
- ➔ #vgextend <vgname> <partitionname>
- ➔ #vgextend yallavg /dev/sde1

root@master-server:~

```
[root@master-server ~]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
vg_masterserver   1  2  0 wz--n- 19.51g  0
yallavg           1  1  0 wz--n-  1.01g 8.00m
[root@master-server ~]#
[root@master-server ~]# vgextend yallavg /dev/sde1
Volume group "yallavg" successfully extended
[root@master-server ~]#
[root@master-server ~]# vgs
VG                #PV #LV #SN Attr   VSize  VFree
vg_masterserver   1  2  0 wz--n- 19.51g  0
yallavg           2  1  0 wz--n-  2.02g 1.02g
[root@master-server ~]#
```

Increasing the size of a logical volume:

- ➔ Sometimes the file system size may be full, so we need to increase the size of the LV to continue adding the data in it.
- ➔ The size of LV can be increased online, no downtime is required.

- ➔ Check the current size of the LV by using `#df -h` command.
- ➔ Increase the size of the LV by using `lvextend` or `lvresize` command, the syntax for it is
- ➔ `#lvextend -L +additional size /dev/vgname/lvname` (syntax for `lvresize` is also same)
- ➔ `#lvextend -L +1G /dev/yallavg/yallalv`

```

root@master-server:~
[root@master-server ~]# lvextend -L +1G /dev/yallavg/yallalv -r
Extending logical volume yallalv to 2.00 GiB
Logical volume yallalv successfully resized
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/mapper/yallavg-yallalv is mounted on /yalla-app; on-line resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/mapper/yallavg-yallalv to 524288 (4k) blocks.
The filesystem on /dev/mapper/yallavg-yallalv is now 524288 blocks long.

[root@master-server ~]# █

```

Note: with `-r` option we can resize the file system at a time.

`#df -h /yalla-app`

```

root@master-server:~
[root@master-server ~]# df -h /yalla-app/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/yallavg-yallalv
                        2.0G       34M   1.9G    2% /yalla-app

[root@master-server ~]# █

```

Reducing the size of an LV:

- ➔ LV size cannot be reduced online, it requires a down time i.e. unmounting the file system.
- ➔ And also check before unmounts any user is accessing the file system by using below commands
- ➔ `#lsof /filesystem name` ; `#fuser -cu /filesystemname`
- ➔ Check the size of the lv using `df -h` command

```

root@master-server:~
[root@master-server ~]# df -h /yalla-app/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/yallavg-yallalv
                        2.0G       34M   1.9G    2% /yalla-app

[root@master-server ~]# █

```

- ➔ Unmount the LV using `umount` command

Note: before unmounts check any one is accessing the file system

By using below commands

```
root@master-server:~  
[root@master-server ~]# lsof /yalla-app/  
[root@master-server ~]#  
[root@master-server ~]# fuser -cu /yalla-app/  
[root@master-server ~]#
```

Note: so with above output we come to know that no one is accessing the file system (/yalla-app)

So now you can umount the filesystem

#umount /yalla-app

```
root@master-server:~  
[root@master-server ~]# df -h /yalla-app/  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/yallavg-yallalv  
                2.0G   34M  1.9G   2% /yalla-app  
[root@master-server ~]#  
[root@master-server ~]# umount /yalla-app/  
[root@master-server ~]# df -h /yalla-app/  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/vg_masterserver-lv_root  
                16G   2.5G   13G  18% /  
[root@master-server ~]#
```

- ➔ Organize the data in LV by using e2fsck command
- ➔ #e2fsck -f /dev/vgname/lvname
- ➔ #e2fsck -f /dev/yallavg/yallalv

```
root@master-server:~  
[root@master-server ~]# e2fsck -f /dev/yallavg/yallalv  
e2fsck 1.41.12 (17-May-2010)  
Pass 1: Checking inodes, blocks, and sizes  
Pass 2: Checking directory structure  
Pass 3: Checking directory connectivity  
Pass 4: Checking reference counts  
Pass 5: Checking group summary information  
/dev/yallavg/yallalv: 11/131072 files (0.0% non-contiguous), 16812/524288 blocks  
[root@master-server ~]#
```

- ➔ Update the file system by using resize2fs command

- ➔ #resize2fs /dev/vgname/lvname 300M (where 300M is the approximate total size of LV after reduction)

root@master-server:~

```
[root@master-server ~]# resize2fs /dev/yallavg/yallalv 300M
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/yallavg/yallalv to 76800 (4k) blocks.
The filesystem on /dev/yallavg/yallalv is now 76800 blocks long.

[root@master-server ~]#
```

- ➔ Now reduce the size by using # lvreduce -L -200M /dev/vgname/lvname command
- ➔ #lvreduce -L -200M /dev/yallavg/yallalv

root@master-server:~

```
[root@master-server ~]# lvreduce -L -200M /dev/yallavg/yallalv
WARNING: Reducing active and open logical volume to 1.80 GiB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce yallalv? [y/n]: y
Reducing logical volume yallalv to 1.80 GiB
Logical volume yallalv successfully resized
[root@master-server ~]#
```

- ➔ Mount the LV and run the command df-h, to verify the change in the size of LV
- ➔ #mount /dev/vgname/lvname /fs
- ➔ #mount /dev/yallavg/yallalv /yalla-app

root@master-server:~

```
[root@master-server ~]# mount /dev/yallavg/yallalv /yalla-app/
[root@master-server ~]# df -h /yalla-app/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/yallavg-yallalv
                           294M      33M   247M  12% /yalla-app

[root@master-server ~]#
```

END OF LVM

