

**GETTING STARTED
WITH
LOGICAL VOLUME MANAGEMENT**

Jasmitha Chukka
jasmitha1217@gmail.com

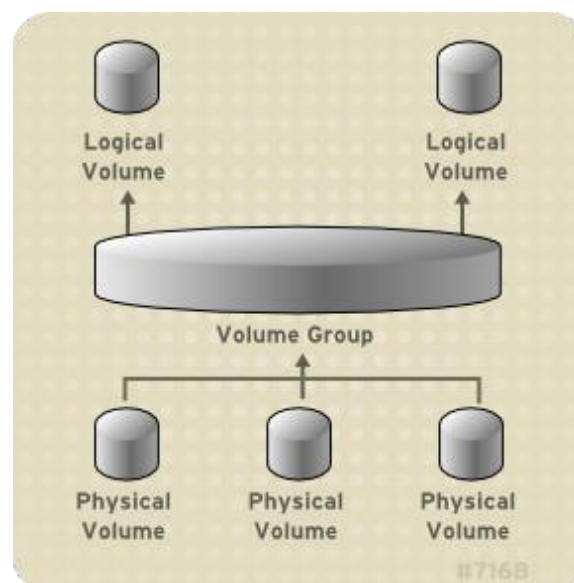
□ LOGICAL VOLUME MANAGEMENT (LVM):

- LVM is a form advanced partition management.
- LVM is a method of allocating hard drive space in to logical volumes that can be **easily resized of partitions**.
- It creates a layer of abstraction over physical storage, which helps you to create logical storage volumes. This provides much greater flexibility in a number of ways than using physical storage directly.

ADVANTAGES OF LVM:

- Flexible capacity
- Convenient device naming
- Resizable storage volumes
- Online data relocation
- Striped Volumes
- RAID volumes
- Volume snapshots
- Cache volumes

LVM ARCHITECTURE:



PHYSICAL VOLUME (PV):

- A **PV** is a partition or whole disk designated for LVM use. To use the device for an LVM logical volume, the device must be initialized as a pv.

VOLUME GROUP (VG):

- A VG is a collection of physical volumes (PVs), which creates a pool of disk space out of which logical volumes can be allocated.
- Within a volume group, the disk space available for allocation is divided into units of a fixed-size called extents.
- An extent is the smallest unit of space that can be allocated. Within a physical volume, extents are referred to as physical extents.

LOGICAL VOLUME (LV):

- A logical volume represents a mountable storage device.
- An administrator can grow or shrink logical volumes without destroying data, unlike standard disk partitions.
- You can lose data if you shrink a logical volume to a smaller capacity than the data on the volume requires.
- Logical partitions which can resize, format and mount...etc.

LVM CONFIGURATION FILES:

- `/etc/lvm/lvm.conf` : Configuration file.
- `/etc/lvm/cache/.cache` : Device name filter cache file (configurable).
- `/etc/lvm/backup/` : Automatic volume group metadata backups.
- `/etc/lvm/archive/` : Automatic volume group metadata archives.

LVM IMPLEMENTATION STEP BY STEP:

- Create physical volume or volumes from the existing hard drives:

Here we are creating three partitions [/dev/nvme0n1p{11,12,13}].

```
#fdisk /dev/nvmeon1
```

change Linux system type to Linux lvm.

```
#partprobe /dev/nvme0n1
```

```
#lsblk
```

CREATING PHYSICAL VOLUMES:

```
#pvcreate /dev/nvme0n1p{11,12,13}
```

→ To verify the physical volumes:

```
#pvdisplay or #pvs
```

→ To scanning physical devices:

```
#pvscan
```

CREATING VOLUME GROUP:

```
#vgcreate india /dev/nvmeon1p{11,12}
```

→ To verify volume group details:

```
#vgdisplay india (or) #vgs
```

→ Scanning disks for volume groups:

```
#vgscan
```

```
#####
```

NOTE: By default, one **physical extend (PE)** size is 4MB.

To change **physical extend (PE)** size:

```
#vgcreate -s 8 india /dev/nvme0n1p{11,12}
```

```
#####
```

→ **To Extend existing volume group:**

```
#vgextend india /dev/nvme0n1p13  
#vgdisplay
```

→ **To reducing Volume Group:**

```
#vgreduce india /dev/nvme0n1p13
```

CREATING LOGICAL VOLUMES:

→ **Hare logical volume size is 2G from the group of India.**

```
#lvcreate -L+2G -n /dev/india/ap
```

→ **To verify Logical volume details:**

```
#lvdisplay /dev/india/ap (or) #lvs
```

→ **Now to verify Volume group details:**

```
#vhdisplay
```

→ **Format logical volume:**

```
#mkfs.xfs -f /dev/india/ap
```

→ **Mounting File system:**

```
#mkdir /lvm-data  
#mount /dev/india/ap /lvm-data  
#df -h
```

→ **To make file system permanent in /etc/fstab file:**

```
/dev/india/ap    /lvm-data    xfs    defaults    0  
0 #systemctl daemon-reload  
#mount /lvm-data  
#df -h
```

EXTEND A LOGICAL VOLUME:

We have seen how to create a logical volume from scratch, but in most cases, you will need to increase the size of an already existing logical volume so it can accommodate more data.

→ **To extend 1GB size to /dev/india/ap file system:**

```
#lvextend -L+1G /dev/india/ap  
#lvdisplay  
#vgdisplay
```

→ **One last step is to resize the filesystem:**

```
#df -h  
#xfs_growfs /dev/india/ap  
#df -h
```

we can see that data blocks have been changed; filesystem has been extended.

REDUCING LOGICAL VOLUME SIZE:

```
#lvreduce -L-1G /dev/india/ap  
#lvdisplay
```

REMOVING LOGICAL VOLUMES:

NOTE: Before removing unmount a file system

```
#lvremove /dev/india/ap  
#lvdiplay
```

REMOVING VOLUME GROUP:

```
#vgremove india  
#vgdisplay
```

REMOVING PHYSICAL VOLUMES:

```
#pvremove /dev/nvme0n1p{11,12,13}  
#pvdipalay
```

NOTE: All done, our Logical Volume has been successfully completed!