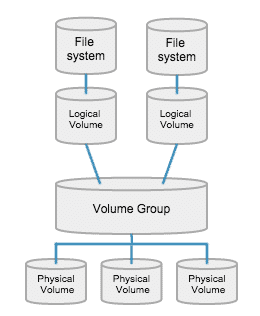
**What is LVM**

LVM or Logical Volume Management is a disk management tool that makes it easier to manage disk space. With the help of LVM, we can easily manage our partitions, we can extend our partitions, reduce them, and replace a failing disk or adding new disk to increase the capacity.

**Concepts**

### LVM comprises of few conceptual layers such as physical volume, logical volume and file systems.

### 



**How does LVM work?**

Before we learn to create LVMs , we will briefly discuss how does it work & will learn some of the terms associated with LVM.

* To create LVM, we will have to first create simple partition with partition ID 8e using fdisk command. For this tutorial, we are using two disks i.e. **/dev/sdb** & **/dev/sdc,**
* Next, we will create **Physical Volumes** (aka **PV**) of these newly partitioned disks. Each physical volume can be a disk partition or even a whole disk.
* After creation of PVs, we will create **Volumes Groups** or**VG**s from the Physical volumes. VGs are collection of partitions into an administrative unit
* Lastly, **Logical Volumes** or **LV**s will be created from Volume Groups. LV is equivalent to a disk partition.
* File systems are built on top of logical volumes. The command mkfs can be used to create file system on top of a logical volume. Once the file system is created we can mount the logical volume as per our need.
* Now that we have some idea about LVM, we will now move onto creating a LVM.

**Let’s Get Started**

First ensure lvm2 packages is installed?

rpm –qa|grep lvm2

if not do

#yum install lvm2

**The example**  
In the example below we would :  
1. Create 2 Physical volumes from 2 physical disks (**/dev/sdb, /dev/sdc**).  
2. Create Volume group from these 2 PVs (**/dev/vg01**).  
3. Create a Logical Volume in this VG (**/dev/vg01/lvol01**).  
4. Create a File system on this LV and mount it (**/data01**).

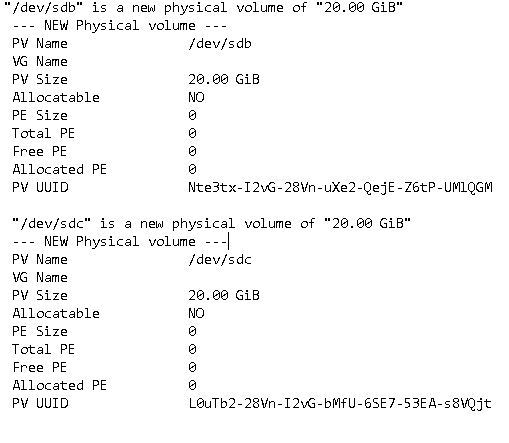
**Creating Physical Volume:**

The **pvcreate** command is used to initialize the PV for use by LVM. Before creating the PV, make sure the disk is visible in the OS. To scan the block devices to be used as PVs, use the **lvmdiskscan** command.

To create PV, we will use ‘**pvcreate**’ command,

**$ pvcreate /dev/sdb /dev/sdc**

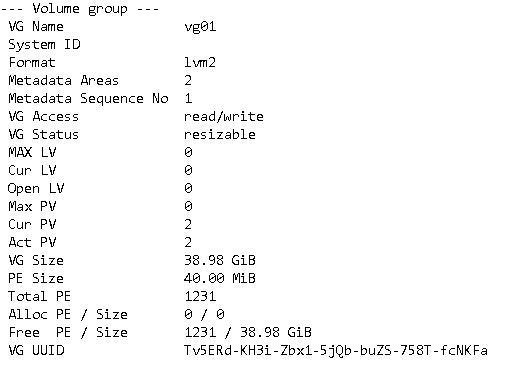
You will get confirmation that Physical Volume has been created. We can see & confirm the newly created Physical volume using ‘**pvdisplay**’



**Creating Volume Group**

Next we will create a VG of the PVs using ‘**vgcreate**’ command,

**$ vgcreate vg01 /dev/sdb /dev/sdc**

Here, ‘vg01’ is the name of volume group. To confirm the creation of VG, we will use ‘**vgdisplay**’ command **vgdisplay vg01**

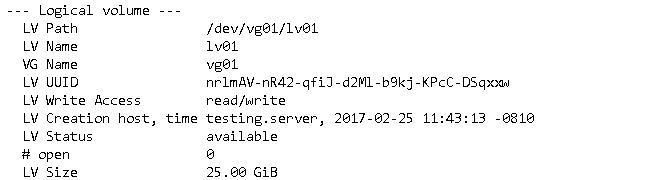
**Creating Logical Volume**

Lastly we will create Logical Volume from Volume group ‘vg01’ using **‘lvcreate**’ command

**$ lvcreate –l 25G –n lv01 vg01**

Here, ‘**-l 25G**’ means size of logical volume, ‘-**n lv01**’ means the name of LV is lv01 & ‘**vg01**’ is the name of Volume Group. To see the information about the new LV, use ‘**lvdisplay**’ command

**$ lvdsplay /dev/vg01/lv01**



Now our LV has been created but before we can use it we need to assign it a file system & mount it to a point.

**Assigning Filesystem & Mounting LV**

We are now going to assign a filesystem to LV ‘lvo1’. We will assigning ‘**ext4**’ to it using ‘**mkfs**’,

**$ mkfs.ext4 /dev/vg01/lv01**

Our LV is now formatted with ext4 filesystem. Next we will create a mounting point for mounting lv01,

**$ mkdir /data**

& now we will mount the lv01 to /data

**$ mount /dev/vg01/lv01 /data**

But this is a temporary mount & will be unmounted if our system reboots. To permanently mount it, we need to append the /etc/fstab file with the following entry

**How to expand the / partition:**

For getting the available Physical size run

# vgdisplay

#lvextend -l +1024 <logical\_vol\_name-withFullPath>

Use **+** to add the more space. After Extending, we need to re-size the file-system using.

# resize2fs <logical\_vol\_name-withFullPath>

Now let’s see the size of re-sized logical volume using.

# lvdisplay

**Reducing Logical Volume (LVM)**

* Before starting, it is always good to backup the data, so that it will not be a headache if something goes wrong.
* To Reduce a logical volume there are 5 steps needed to be done very carefully.
* While extending a volume we can extend it while the volume under mount status (online), but for reduce we must need to unmount the file system before reducing.

Let’s see what are the 5 steps below.

1. unmount the file system for reducing.
2. Check the file system after unmount.
3. Reduce the file system.
4. Reduce the Logical Volume size than Current size.
5. Recheck the file system for error.
6. Remount the file-system back to stage.

First unmount the mount point.

#umount -v <Mount name>

Then check for the file-system error using following command.

#e2fsck -f <logical\_vol\_name-withFullPath>

( -f Force checking even if the file system seems clean. )

Next, reduce the file-system.

# lvreduce -L -1G <logical\_vol\_name-withFullPath>

Re-size the file-system back, In this step if there is any error that means we have messed-up our file-system.

# resize2fs <logical\_vol\_name-withFullPath>

Mount the file-system back to same point.

Check the size of partition and files.