**LVM stands for Logical Volume Manager.**

With LVM, we can create logical partitions that can span across one or more physical hard drives. First, the hard drives are divided into physical volumes, then those physical volumes are combined together to create the volume group and finally the logical volumes are created from volume group.

Before we start, install the lvm2 package as shown below.

$ yum intall lvm2

To create a LVM, we need to run through the following steps.

* Select the physical storage devices for LVM
* Create the Volume Group from Physical Volumes
* Create Logical Volumes from Volume Group

**Select the Physical Storage Devices for LVM – Use pvcreate, pvscan, pvdisplay Commands**

In this step, we need to choose the physical volumes that will be used to create the LVM. We can create the physical volumes using pvcreate command as shown below.

$ pvcreate /dev/sda6 /dev/sda7

Physical volume "/dev/sda6" successfully created

Physical volume "/dev/sda7" successfully created

As shown above two physical volumes are created – /dev/sda6 and /dev/sda7.

If the physical volumes are already created, you can view them using the pvscan command as shown below.

$ pvscan

PV /dev/sda6 lvm2 [1.86 GB]

PV /dev/sda7 lvm2 [1.86 GB]

Total: 2 [3.72 GB] / in use: 0 [0 ] / in no VG: 2 [3.72 GB]

You can view the list of physical volumes with attributes like size, physical extent size, total physical extent size, the free space, etc., using pvdisplay command as shown below.

$ pvdisplay

--- Physical volume ---

PV Name /dev/sda6

VG Name

PV Size 1.86 GB / not usable 2.12 MB

Allocatable yes

PE Size (KByte) 4096

Total PE 476

Free PE 456

Allocated PE 20

PV UUID m67TXf-EY6w-6LuX-NNB6-kU4L-wnk8-NjjZfv

--- Physical volume ---

PV Name /dev/sda7

VG Name

PV Size 1.86 GB / not usable 2.12 MB

Allocatable yes

PE Size (KByte) 4096

Total PE 476

Free PE 476

Allocated PE 0

PV UUID b031x0-6rej-BcBu-bE2C-eCXG-jObu-0Boo0x

Note : PE – Physical Extents are nothing but equal-sized chunks. The default size of extent is 4MB.

**Create the Volume Group – Use vgcreate, vgdisplay Commands**

Volume groups are nothing but a pool of storage that consists of one or more physical volumes. Once you create the physical volume, you can create the volume group (VG) from these physical volumes (PV).

In this example, the volume group vol\_grp1 is created from the two physical volumes as shown below.

$ vgcreate vol\_grp1 /dev/sda6 /dev/sda7

Volume group "vol\_grp1" successfully created

LVM processes the storage in terms of extents. We can also change the extent size (from the default size 4MB) using -s flag.

vgdisplay command lists the created volume groups.

$ vgdisplay

--- Volume group ---

VG Name vol\_grp1

System ID

Format lvm2

Metadata Areas 2

Metadata Sequence No 1

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 0

Open LV 0

Max PV 0

Cur PV 2

Act PV 2

VG Size 3.72 GB

PE Size 4.00 MB

Total PE 952

Alloc PE / Size 0 / 0

Free PE / Size 952 / 3.72 GB

VG UUID Kk1ufB-rT15-bSWe-5270-KDfZ-shUX-FUYBvR

**LVM Create: Create Logical Volumes – Use lvcreate, lvdisplay command**

Now, everything is ready to create the logical volumes from the volume groups. lvcreate command creates the logical volume with the size of 80MB.

$ lvcreate -l 20 -n logical\_vol1 vol\_grp1

Logical volume "logical\_vol1" created

Use lvdisplay command as shown below, to view the available logical volumes with its attributes.

$ lvdisplay

--- Logical volume ---

LV Name /dev/vol\_grp1/logical\_vol1

VG Name vol\_grp1

LV UUID ap8sZ2-WqE1-6401-Kupm-DbnO-2P7g-x1HwtQ

LV Write Access read/write

LV Status available

# open 0

LV Size 80.00 MB

Current LE 20

Segments 1

Allocation inherit

Read ahead sectors auto

- currently set to 256

Block device 252:0

After creating the appropriate filesystem on the logical volumes, it becomes ready to use for the storage purpose.

$ mkfs.ext3 /dev/vol\_grp1/logical\_vol1

**LVM resize: Change the size of the logical volumes – Use lvextend Command**

We can extend the size of the logical volumes after creating it by using lvextend utility as shown below. The changes the size of the logical volume from 80MB to 100MB.

$ lvextend -L100 /dev/vol\_grp1/logical\_vol1

Extending logical volume logical\_vol1 to 100.00 MB

Logical volume logical\_vol1 successfully resized

We can also add additional size to a specific logical volume as shown below.

$ lvextend -L+100 /dev/vol\_grp1/logical\_vol1

Extending logical volume logical\_vol1 to 200.00 MB

Logical volume logical\_vol1 successfully resized

## PV related commands in Linux LVM :

**Directories and Files :**  
  
## Directories  
/etc/lvm - default lvm directory location  
/etc/lvm/backup - where the automatic backups go  
/etc/lvm/cache - persistent filter cache  
/etc/lvm/archive - where automatic archives go after a volume group change  
/var/lock/lvm - lock files to prevent metadata corruption  
  
# Files  
/etc/lvm/lvm.conf - main lvm configuration file  
$HOME/.lvm - lvm history   
  
 **Diagnostics :**  
  
#lvmdump  
#lvmdump -d   
  
# dmsetup [info|ls|status]   
  
// Note: by default the lvmdump command creates a tar ball

## PHYSICAL VOLUME

**Display :**  
  
# pvdisplay -v  
# pvs -v  
# pvs -a   
  
**Scanning :**   
  
#pvscan -v  
  
Note: scans for disks for non-LVM and LVM disks   
  
 **Add / Remove / Check PV:**  
  
# pvcreate /dev/sdb1  
# pvremove /dev/sdb1   
#pvck -v /dev/sdb1   
  
**Change physical attributes:**   
  
## do not allow allocation of extents on this drive  
  
#pvchange -x n /dev/sdb1  
  
- Common Attributes that you may want to use:  
  
--addtag add a tag  
-x allowed to allocate extents  
-u change the uuid   
  
**Move PV** :   
  
# pvmove -v /dev/sdb2 /dev/sdb3  
  
**Note:** moves any used extents from this volume to another volume, in readiness to remove that volume. However you cannot use this on mirrored volumes, you must convert back to non-mirror using "lvconvert -m 0"

**Moving a VG to another server:**  
  
To do this we use the vgexport and vgimport commands.  
  
vgexport and vgimport is not necessary to move disk drives from one server to another. It is an administrative policy tool to prevent access to volumes in the time it takes to move them.  
  
**1. Unmount the file system**  
First, make sure that no users are accessing files on the active volume, then unmount it  
  
# unmount /appdata  
  
**2.Mark the volume group inactive**  
Marking the volume group inactive removes it from the kernel and prevents any further activity on it.  
  
# vgchange -an appvg  
vgchange -- volume group "appvg" successfully deactivate  
  
  
**3. Export the volume group**  
  
It is now must to export the volume group. This prevents it from being accessed on the old server and prepares it to be removed.  
  
# vgexport appvg  
vgexport -- volume group "appvg" successfully exported  
  
Now, When the machine is next shut down, the disk can be unplugged and then connected to it's new machine  
  
**4. Import the volume group**  
  
When it plugged into the new server, it becomes /dev/sdc (depends).  
  
so an initial pvscan shows:  
  
# pvscan  
pvscan -- reading all physical volumes (this may take a while...)  
pvscan -- inactive PV "/dev/sdc1" is in EXPORTED VG "appvg" [996 MB / 996 MB free]  
pvscan -- inactive PV "/dev/sdc2" is in EXPORTED VG "appvg" [996 MB / 244 MB free]  
pvscan -- total: 2 [1.95 GB] / in use: 2 [1.95 GB] / in no VG: 0 [0]  
  
We can now import the volume group (which also activates it) and mount the file system.  
  
If you are importing on an LVM 2 system, run:  
  
# vgimport appvg  
Volume group "vg" successfully imported  
  
**5. Activate the volume group**  
  
You must activate the volume group before you can access it.  
  
# vgchange -ay appvg  
  
Mount the file system  
  
# mkdir -p /appdata  
# mount /dev/appvg/appdata /appdata  
  
The file system is now available for use.