Beginner’s Guide to LVM (Logical Volume Management)

thegeekdiary.com/redhat-centos-a-beginners-guide-to-lvm-logical-volume-manager/

admin

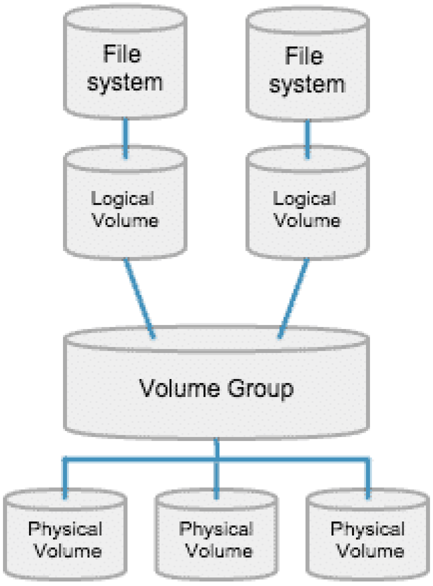
What is LVM

Logical volume manager (LVM) introduces an extra layer between the physical disks and the file system allowing file systems to be :
– resized and moved easily and online without requiring a system-wide outage.
– Using discontinuous space on disk
– meaningful names to volumes, rather than the usual cryptic device names.
– span multiple physical disks

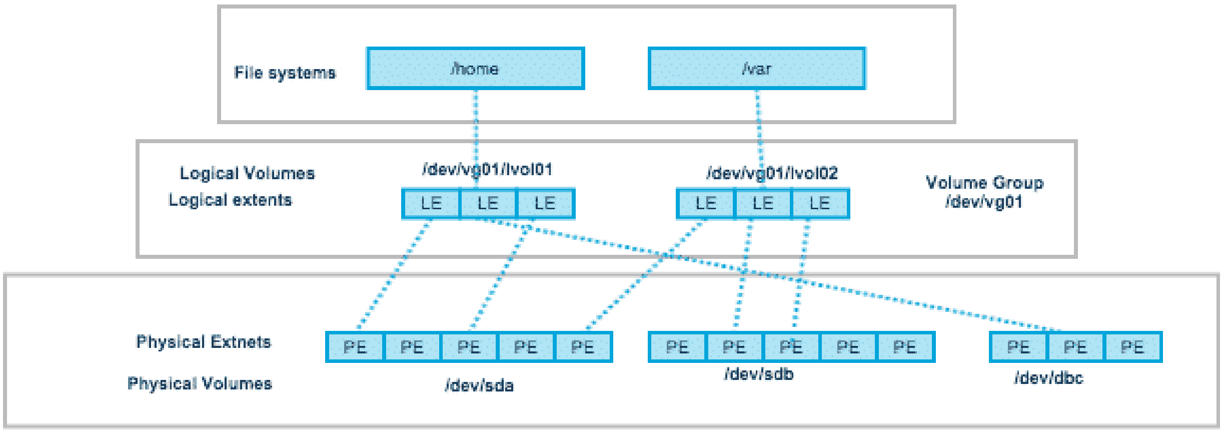
Linux LVM is very similar to HP-UX LVM and provides many other advanced features like snapshots, cluster support (GFS2, OCFS and Lustre).

Concepts

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



The conceptual layers are in turn made up of smaller units like Physical extents(in case of Physical volumes) and Logical extents (in case of Logical Volumes).



Physical Volume (PV)

Each Physical Volume can be a disk partition, whole disk, meta-device, or a loopback file. Use the command **pvcreate** to initialize storage for use by LVM. Initializing a block device as physical volume places a label at the start of the device.

Volume Group (VG)

A Volume Group gathers together a collection of Logical Volumes and Physical Volumes into one administrative unit. Volume group is divided into fixed size physical extents. The command **vgcreate** creates a new volume group using the block special device Physical Volume path previously configured for LVM with pvcreate.
– VGs are made up of PVs, which in turn are made up of physical extents (PEs). The size of PE can differe in different VGs and is defined at the time of creating VG.
– The default size of PE is 4MB, but you can change it to the value you want at the time of VG creation.
– Generally, larger the PE size, better the performance (though less granular control of LV).

Logical Volume (LV)

A Logical Volume is the conceptual equivalent of a disk partition in a non-LVM system. Logical volumes are block devices which are created from the physical extents present in the same volume group. You can use command **lvcreate** to create a logical volume in an existing volume group.

File system

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.

Lets Get Started

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

Create Physical Volumes

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.

```
# lvmdiskscan
.......
  /dev/sdb   [       2.00 GiB]
  /dev/sdc   [       2.00 GiB]
  /dev/sdd   [       2.00 GiB]
  3 disks
  19 partitions
  0 LVM physical volume whole disks
  0 LVM physical volumes
```

Initialize the block devices :

```
# pvcreate /dev/sdb /dev/sdc /dev/sdd
  Physical volume "/dev/sdb" successfully created
  Physical volume "/dev/sdc" successfully created
  Physical volume "/dev/sdd" successfully created
```

Display physical volumes

Use the commands **pvdisplay**, **pvs** and **pvscan** to display the PVs we just created.

```
# pvdisplay
  "/dev/sdb" is a new physical volume of "2.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               2.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               Mt3F7z-a2AV-28Vn-uXe2-QejE-Z6tP-UMlQGM

  "/dev/sdc" is a new physical volume of "2.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdc
  VG Name
  PV Size               2.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               5m1Fuc-yTRn-I2vG-bMfU-6SE7-53EA-s8VQjt

  "/dev/sdd" is a new physical volume of "2.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdd
  VG Name
  PV Size               2.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               1x3e2A-C0Lt-DrUA-tPSM-lsMu-sn70-qg1j8p

# pvscan
  PV /dev/sdb                      lvm2 [2.00 GiB]
  PV /dev/sdc                      lvm2 [2.00 GiB]
  PV /dev/sdd                      lvm2 [2.00 GiB]
  Total: 3 [6.00 GiB] / in use: 0 [0   ] / in no VG: 3 [6.00 GiB]

# pvs
  PV         VG   Fmt  Attr PSize PFree
  /dev/sdb        lvm2 a--  2.00g 2.00g
  /dev/sdc        lvm2 a--  2.00g 2.00g
  /dev/sdd        lvm2 a--  2.00g 2.00g
```

Create a Volume Group

Use the **vgcreate** command to create the new Volume Group **vg01** using the 3 PVs we just created. We can specify the extents with **-s** option and maximum number of PVs and LVs in the VG by using the options **-p** and **-l** respectively. All these option are optional and need not be necessarily used.

```
# vgcreate vg01 /dev/sdb /dev/sdc /dev/sdd
  Volume group "vg01" successfully created
```

The optional options that are used with vgcreate command are :

Option	Meaning
-s	Physical extent size
-p	Max number of PVs
-l	Max number of LVs
–alloc	allocation policy (either contiguous, anywhere, or cling)

Displaying the VG information

The commands vgs and vgdisplay can be used to display the information about the VG we just created :

```
# vgs vg01
  VG   #PV #LV #SN Attr   VSize VFree
  vg01   3   0   0 wz--n- 5.99g 5.99g

# vgdisplay vg01
  --- Volume group ---
  VG Name               vg01
  System ID
  Format                lvm2
  Metadata Areas        3
  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                3
  Act PV                3
  VG Size               5.99 GiB
  PE Size               4.00 MiB
  Total PE              1533
  Alloc PE / Size       0 / 0
  Free  PE / Size       1533 / 5.99 GiB
  VG UUID               Cw7GGz-NH3o-Sax2-5jPv-buZS-938T-tmNKFa
```

Activating and deactivating VGs

The **vgchange** command can be used to activate/deactivate a volume group.
To deactivate a VG :

```
# vgchange -a n vg01
  0 logical volume(s) in volume group "vg01" now active
```

To activate a VG :

```
# vgchange -a y vg01
  1 logical volume(s) in volume group "vg01" now active
```

Create Logical Volume

The Logical volume can now be created in the VG using the **lvcreate command**.
– If you do not specify the LV name in the command, by default the LV is given the name **lvol#**.
– Normally if you do not specify which PV to span the LV, Logical volume will be created on the PV on a next-free basis.
– To create a logical volume lvol01 of size 5 GB :

```
# lvcreate -L 5G -n lvol01 vg01
  Logical volume "lvol01" created
```

Creating a striped volume

To create a striped volume spanning all the 3 PVs we created :

```
# lvcreate -L 5G -I 4096 -i 3 -n lvol01 vg01
  Rounding size (1280 extents) up to stripe boundary size (1281 extents)
  Logical volume "lvol01" created

I - PVs to span while creating striped volume
i - stripe unit
```

Creating mirrored volume

To create a 3 way mirrored volume spanning the 3 PVs (sdb, sdc, sdd) :

```
# lvcreate -L 1G -m 2 -n lvol01 vg01
  Logical volume "lvol01" created
```

We can also specify which devices to be used while creating the mirrored LV. In our case as we had only 3 PVs in the VG, the LV gets created by default on these 3 PVs.

Displaying the LV information

The commands **lvdisplay**, **lvs** and **lvscan** can be used to display the information about the LV we just created.

```
# lvs /dev/vg01/lvol01
  LV     VG   Attr      LSize Pool Origin Data%  Move Log         Cpy%Sync Convert
  lvol01 vg01 mwi-a-m-- 1.00g                         lvol01_mlog   100.00
```

```
# lvdisplay /dev/vg01/lvol01
  --- Logical volume ---
  LV Path                /dev/vg01/lvol01
  LV Name                lvol01
  VG Name                vg01
  LV UUID                ptlmAV-mO42-fWiJ-e2Ml-r9kj-PFcC-MOexxw
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2014-10-22 09:04:25 -0700
  LV Status              available
  # open                 0
  LV Size                1.00 GiB
  Current LE             256
  Mirrored volumes       3
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:4

# lvscan
  ACTIVE            '/dev/vg01/lvol01' [1.00 GiB] inherit
```

Creating File system

The final step is to create a file system on the new LV we just created and mount it on a directory to be able to access it and store data in it. The command **mkfs** can be used to create file system on top of the LV.

```
# mkfs.ext4 /dev/vg01/lvol01
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 37 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

The logical volume can be mounted, once the file system is created. Make sure to add an entry to **/etc/fstab**, so that it is mounted automatically when the system boots.

```
# mkdir /data01
# mount /dev/vg01/lvol01 /data01

# vi /etc/fstab
/dev/vg01/lvol01        /data01                 ext4    defaults        0 0

# df -h /data01
Filesystem                  Size  Used  Avail  Use%  Mounted on
/dev/mapper/vg01-lvol01     1008M  34M  924M   4%    /data01
```

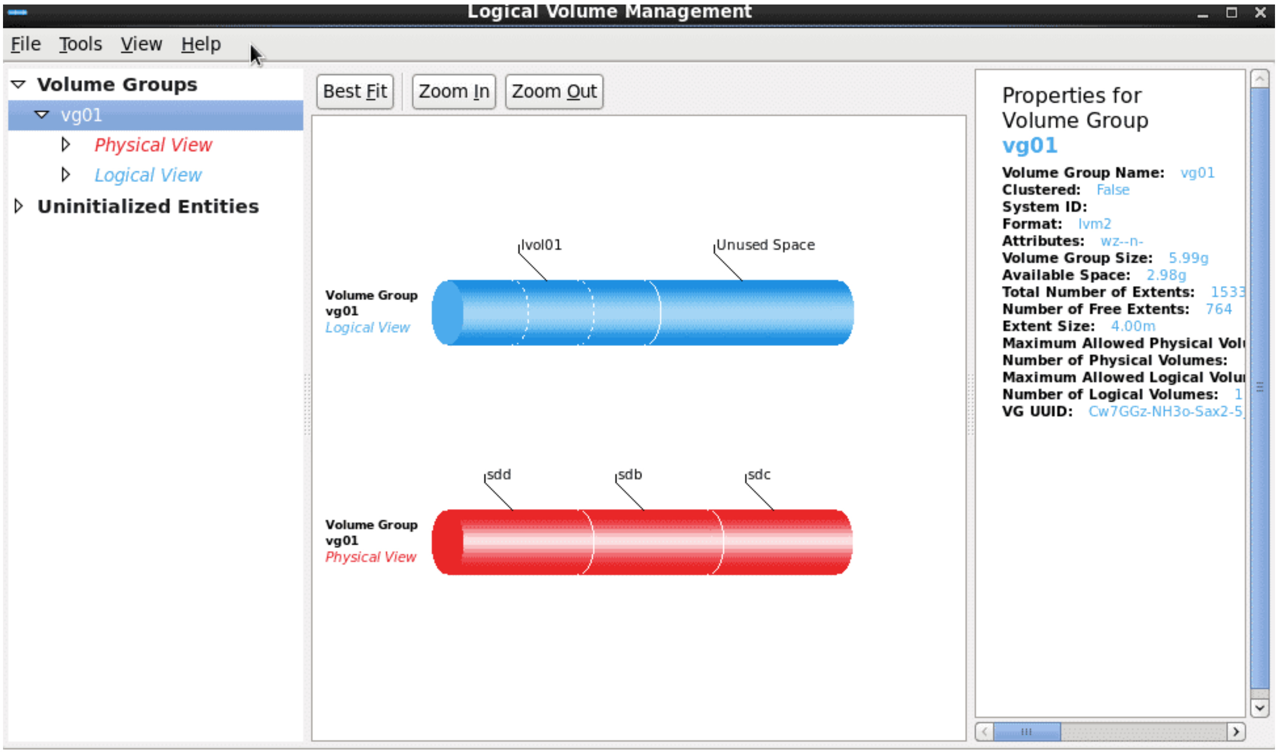
The Graphical Tool to manage LVM

There is a cool graphical tool available (**system-config-lvm**) in case you want to use. If not already installed on the system, install it using yum:

```
# yum install system-config-lvm
```

To start the Graphical LVM administration tool, fire the command :

```
# system-config-lvm
```



Properties for
Volume Group
vg01

Volume Group Name: vg01
Clustered: False
System ID:
Format: lvm2
Attributes: wz--n-
Volume Group Size: 5.99g
Available Space: 2.98g
Total Number of Extents: 1533
Number of Free Extents: 764
Extent Size: 4.00m
Maximum Allowed Physical Volu
Number of Physical Volumes:
Maximum Allowed Logical Volu
Number of Logical Volumes: 1
VG UUID: Cw7GGz-NH3o-Sax2-5