Logical Volume Manager

[**References:**](#_55aycq4krukx) **1**

[**Introduction:**](#_6iw3hjeyn7zs) **2**

[**Logical Volume Manager Procedure:**](#_sytz0drgan69) **3**

[Creating the Physical Volume:](#_rlv1udu4f77t) 3

[Display the physical volumes:](#_6lcys16v3k4) 3

[Create Volume Group](#_jhu4jjt2l0ew) 3

[Display information about VG's:](#_cq3kg9u94orv) 4

[Create Logical Volume](#_5liivmqj3i7l) 4

[Linear Logical Volumes:](#_xfzwle35g70v) 5

[Display Information about Logical Volumes.](#_8if701r37kd2) 5

[Adding more Logical Volumes:](#_g7k8ms8t1bus) 6

[Formatting a logical volume:](#_ogzi7wthprsy) 7

[Formatting with xfs:](#_ynfrtc1g45sf) 7

[Formatting with ext4:](#_vwk0k1hl7idj) 7

[Extending a Logical Volume:](#_9lgwfzperaug) 8

[Mount the volumes:](#_p7mkomonkcx0) 8

[Extend an xfs filesystem:](#_p5aop9ur3ti) 8

[Extend an ext4 filesystem:](#_bbe710rzjo69) 9

[Now recheck the sizes:](#_smcj3cocvxpi) 9

[Displaying Information about All LVM Compatible Block Storage Devices](#_cgc6urstij5n) 9

# References:

<https://linoxide.com/linux-how-to/lvm-configuration-linux/>

<https://www.centos.org/docs/5/html/Cluster_Logical_Volume_Manager/VG_create.html>



# 

# Introduction:

**Logical Volume Management** (LVM) creates a layer of **abstraction** over **physical** **storage**, allowing you to create **logical storage volumes**. With LVM in place, you are not bothered with physical disk sizes because the hardware storage is hidden from the software so it can be resized and moved without stopping applications or unmounting file systems. You can think of LVM as dynamic partitions.

For example, if you are running out of disk space on your server, you can just add another disk and extend the logical volume on the fly.

Below are some advantages of using Logical volumes over using physical storage directly:

* **Resize storage pools:** You can extend the logical space as well as reduce it without reformatting the disks.
* **Flexible storage capacity:** You can add more space by adding more disks and adding them to the pool of physical storage, thus you have a flexible storage capacity.
* **Use of striped, mirrored and snapshot volumes:** Striped logical volume that stripes data across two or more disks can dramatically increase throughput. Mirrored Logical volumes provide a convenient way to configure a mirror for your data. And you can take [device snapshots for backups](https://linoxide.com/linux-how-to/lvm-snapshots-ext4-centos-7/) or to test the effect of changes without affecting the real data.

LVM respects 3 concepts

* **Physical Volume (PV): it** is a whole disk or a partition of a disk
* **Volume Group (VG):** corresponds to one or more PV
* **Logical Volume (LV):** represents a portion of a VG. A LV can only belong to one VG. It’s on a LV that we can create a file system.

# Logical Volume Manager Procedure:

## Creating the Physical Volume:

Physical volume is the actual storage device that will be used in the LVM configuration. It can be an entire disk, a partition on disk or a LUN on the SAN. You can use pvcreate to create the physical volume. In this example I have added one disk /dev/sdb. The **pvcreate** command **initialize** these disk so that it can be a part in forming volume groups.

|  |
| --- |
| pvcreate /dev/sdb |

|  |
| --- |
| root> pvcreate /dev/sdb  Physical volume "/dev/sdb" successfully created. |

### Display the physical volumes:

|  |
| --- |
| pvdisplay |

|  |
| --- |
| root> pvdisplay  --- Physical volume ---  PV Name /dev/sdb  VG Name test-volume  PV Size 372.61 GiB / not usable <2.09 MiB  Allocatable yes  PE Size 4.00 MiB  Total PE 95388  Free PE 95388  Allocated PE 0  PV UUID ku0xCh-oB9K-rjYl-DLXq-h2hf-JGIk-qInVYs |

## Create Volume Group

Physical volumes are combined into volume groups (VGs). It creates a pool of disk space out of which logical volumes can be allocated. The disk space available for allocation in Volume Group is divided into units of a fixed-size called extents. An extent is the smallest unit of storage that can be allocated. Within a physical volume, extents are referred to as physical extents.

|  |
| --- |
| vgcreate test-volume /dev/sdb |

|  |
| --- |
| root> vgcreate test-volume /dev/sdb  Volume group "test-volume" successfully created  You have new mail in /usr/spool/mail/root |

### Display information about VG's:

|  |
| --- |
| vgdisplay |

|  |
| --- |
| root> vgdisplay  --- Volume group ---  VG Name test-volume  System ID  Format lvm2  Metadata Areas 1  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 1  Act PV 1  VG Size <372.61 GiB  PE Size 4.00 MiB  Total PE 95388  Alloc PE / Size 0 / 0  Free PE / Size 95388 / <372.61 GiB  VG UUID 7MkvEE-B2RH-iYoJ-hwU1-ooCQ-EsPn-30g6jQ |

## Create Logical Volume

A volume group is divided up into logical volumes. So if you have created vg-01 earlier then you can create logical volumes from that VG. The amount of space you want to allocate depend on your requirement. You might want to create LV of 200MB, 1GB etc. There are three type of logical volumes. We will only look at Linear Volumes. The others are discussed in the references above.

* Linear Volume
* Striped Volume
* Mirrored Volume

### **Linear Logical Volumes:**

A linear volume aggregates space from one or more physical volumes into one logical volume. For example, if you have two 2GB disks, you can create a 4GB logical volume. The physical storage is concatenated. The application will see one device that is 4 GB in size. By default we create linear volumes in LVM. Consider the following examples:

|  |
| --- |
| root> lvcreate -L 10G -n home test-volume |

|  |
| --- |
| root> lvcreate -L 10G -n home test-volume  Logical volume "home" created. |

### Display Information about Logical Volumes.

|  |
| --- |
| root> lvdisplay |

|  |
| --- |
| root> lvdisplay  --- Logical volume ---  LV Path /dev/test-volume/home  LV Name home  VG Name test-volume  LV UUID BeyEez-DCKN-1Bl5-OZoq-hR3v-DhJ3-rEEgNj  LV Write Access read/write  LV Creation host, time hendrix.colorado.edu, 2018-05-13 14:24:35 -0600  LV Status available  # open 0  LV Size 10.00 GiB  Current LE 2560  Segments 1  Allocation inherit  Read ahead sectors auto  - currently set to 256  Block device 253:0 |

### Adding more Logical Volumes:

|  |
| --- |
| lvcreate -L 10G -n data test-volume  lvcreate -L 10G -n local test-volume |

Output and display the logical volumes:

|  |
| --- |
| root> lvcreate -L 10G -n data test-volume  Logical volume "data" created.  root> lvcreate -L 10G -n local test-volume  Logical volume "local" created.  root> lvdisplay  --- Logical volume ---  LV Path /**dev/test-volume/home**  LV Name **home**  VG Name test-volume  LV UUID BeyEez-DCKN-1Bl5-OZoq-hR3v-DhJ3-rEEgNj  LV Write Access read/write  LV Creation host, time hendrix.colorado.edu, 2018-05-13 14:24:35 -0600  LV Status available  # open 0  LV Size 10.00 GiB  Current LE 2560  Segments 1  Allocation inherit  Read ahead sectors auto  - currently set to 256  Block device 253:0    --- Logical volume ---  LV Path **/dev/test-volume/data**  LV Name **data**  VG Name test-volume  LV UUID NY8K0u-P04k-GwhO-r3uD-c2pd-1HEc-YedRac  LV Write Access read/write  LV Creation host, time hendrix.colorado.edu, 2018-05-13 14:33:09 -0600  LV Status available  # open 0  LV Size 10.00 GiB  Current LE 2560  Segments 1  Allocation inherit  Read ahead sectors auto  - currently set to 256  Block device 253:1    --- Logical volume ---  LV Path **/dev/test-volume/local**  LV Name **local**  VG Name test-volume  LV UUID 0iwmIi-XZNj-oJhd-xQt1-s1dS-IYpw-WNOOgP  LV Write Access read/write  LV Creation host, time hendrix.colorado.edu, 2018-05-13 14:33:20 -0600  LV Status available  # open 0  LV Size 10.00 GiB  Current LE 2560  Segments 1  Allocation inherit  Read ahead sectors auto  - currently set to 256  Block device 253:2 |

### Formatting a logical volume:

#### Formatting with xfs:

|  |
| --- |
| root> mkfs.xfs /dev/test-volume/home  meta-data=/dev/test-volume/home isize=512 agcount=4, agsize=655360 blks  = sectsz=512 attr=2, projid32bit=1  = crc=1 finobt=0, sparse=0  data = bsize=4096 blocks=2621440, imaxpct=25  = sunit=0 swidth=0 blks  naming =version 2 bsize=4096 ascii-ci=0 ftype=1  log =internal log bsize=4096 blocks=2560, version=2  = sectsz=512 sunit=0 blks, lazy-count=1  realtime =none extsz=4096 blocks=0, rtextents=0 |

#### Formatting with ext4:

|  |
| --- |
| root> mkfs.ext4 /dev/test-volume/data  mke2fs 1.42.9 (28-Dec-2013)  Filesystem label=  OS type: Linux  Block size=4096 (log=2)  Fragment size=4096 (log=2)  Stride=0 blocks, Stripe width=0 blocks  655360 inodes, 2621440 blocks  131072 blocks (5.00%) reserved for the super user  First data block=0  Maximum filesystem blocks=2151677952  80 block groups  32768 blocks per group, 32768 fragments per group  8192 inodes per group  Superblock backups stored on blocks:  32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632  Allocating group tables: done  Writing inode tables: done  Creating journal (32768 blocks): done  Writing superblocks and filesystem accounting information: done |

### Extending a Logical Volume:

To extend a logical volume you simply tell the lvextend command how much you want to increase the size. You can specify how much to grow the volume, or how large you want it to grow it.

#### Mount the volumes:

|  |
| --- |
| root> mkdir /mnt/home /mnt/data  root> mount /dev/test-volume/home /mnt/home  root> mount /dev/test-volume/data /mnt/data  root> df -h /mnt/home /mnt/data  Filesystem Size Used Avail Use% Mounted on  /dev/mapper/test--volume-home 10G 33M 10G 1% /mnt/home  /dev/mapper/test--volume-data 9.8G 37M 9.2G 1% /mnt/data |

Now we extend the logical volumes and then resize the file systems.

#### Extend an xfs filesystem:

|  |
| --- |
| root> **lvextend -L50G /dev/test-volume/home**  Size of logical volume test-volume/home **changed from 10.00 GiB** (2560 extents) **to 50.00 GiB** (12800 extents).  **Logical volume test-volume/home successfully resized.**  root> **xfs\_growfs /mnt/home**  meta-data=/dev/mapper/test--volume-home isize=512 agcount=4, agsize=655360 blks  = sectsz=512 attr=2, projid32bit=1  = crc=1 finobt=0 spinodes=0  data = bsize=4096 blocks=2621440, imaxpct=25  = sunit=0 swidth=0 blks  naming =version 2 bsize=4096 ascii-ci=0 ftype=1  log =internal bsize=4096 blocks=2560, version=2  = sectsz=512 sunit=0 blks, lazy-count=1  realtime =none extsz=4096 blocks=0, rtextents=0  data blocks changed from 2621440 to 13107200 |

#### Extend an ext4 filesystem:

|  |
| --- |
| root> **lvextend -L+50G /dev/test-volume/data**  Size of logical volume test-volume/data changed **from** **10.00 GiB** (2560 extents) **to 60.00 GiB** (15360 extents).  Logical volume test-volume/data successfully resized.  root> **resize2fs /mnt/data/**  resize2fs 1.42.9 (28-Dec-2013)  open: Is a directory while opening /mnt/data/  root> resize2fs /dev/test-volume/data  resize2fs 1.42.9 (28-Dec-2013)  Filesystem at /dev/test-volume/data is mounted on /mnt/data; on-line resizing required  old\_desc\_blocks = 2, new\_desc\_blocks = 14  The filesystem on /dev/test-volume/data is now 28835840 blocks long. |

### Now recheck the sizes:

|  |
| --- |
| root> df -h /mnt/home /mnt/data  Filesystem Size Used Avail Use% Mounted on  /dev/mapper/test--volume-home 50G 33M 50G 1% /mnt/home  /dev/mapper/test--volume-data 109G 60M 104G 1% /mnt/data |

### Displaying Information about All LVM Compatible Block Storage Devices

It is important to be able to get information about the various LVM components in your system easily. Fortunately, the LVM tool suite provides an abundant amount of tools for displaying information about every layer in the LVM stack.

|  |
| --- |
| lvmdiskscan |

|  |
| --- |
| root> lvmdiskscan  /dev/test-volume/home [ 50.00 GiB]  /dev/sda1 [ 4.88 GiB]  /dev/test-volume/data [ 110.00 GiB]  /dev/sda2 [ 15.62 GiB]  /dev/test-volume/local [ 10.00 GiB]  /dev/sda3 [ <14.65 GiB]  /dev/sda5 [ <9.77 GiB]  /dev/sda6 [ 1.95 GiB]  /dev/sda7 [ 418.88 GiB]  /dev/sdb [ 372.61 GiB] LVM physical volume  3 disks  6 partitions  1 LVM physical volume whole disk  0 LVM physical volumes |