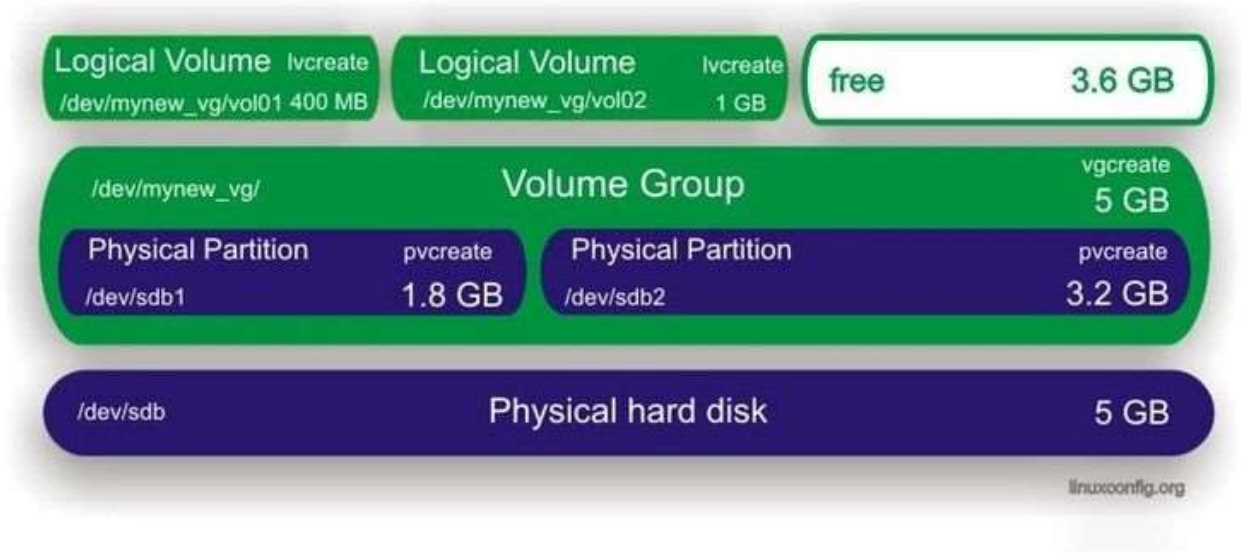
Linux lvm - Logical Volume Manager

Praveen Kumar

Contents

This article describes a basic logic behind a Linux logical volume manager by showing real examples of configuration and usage. Although Debian Linux will be used for this tutorial, you can also apply the same command line syntax with other Linux distributions such as Red Hat, Mandriva, SuSe Linux and others.

This is what we are going to do

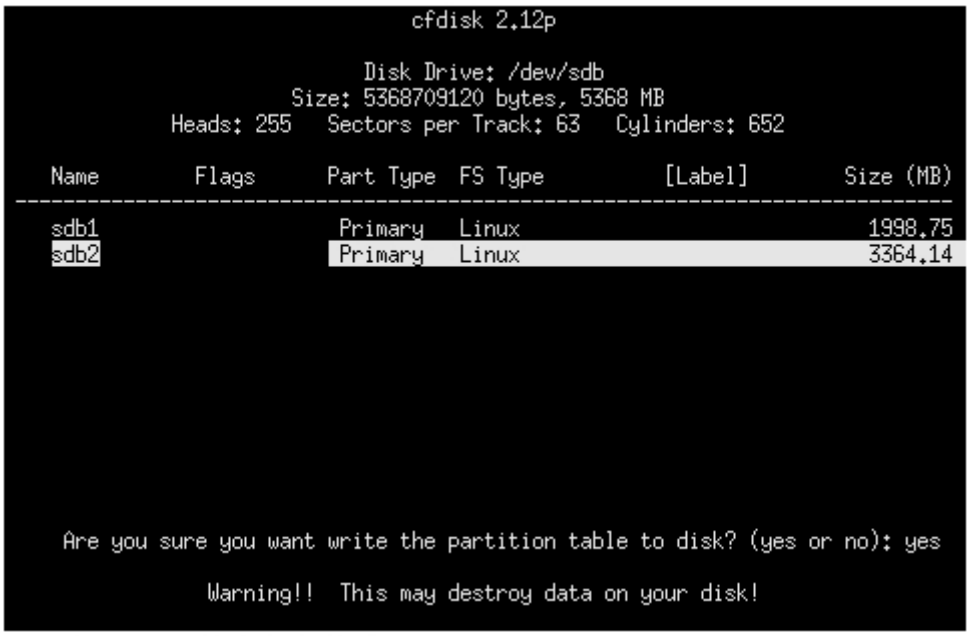


Create Partitions

For this Linux lvm example you need an unpartitioned hard disk /dev/sdb. First you need to create physical volumes. To do this you need partitions or a whole disk. It is possible to run pvcreate command on /dev/sdb, but I prefer to use partitions and from partitions I later create physical volumes.

```
linuxconfig.org# fdisk -l

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          25      200781   83  Linux
/dev/sda2              26         522     3992152+   5  Extended
/dev/sda5              26         217     1542208+  83  Linux
/dev/sda6             218         299      658633+  83  Linux
/dev/sda7             300         327      224878+  82  Linux swap / Solaris
/dev/sda8             328         342      120456   83  Linux
/dev/sda9             343         522     1445818+  83  Linux

Disk /dev/sdb: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdb doesn't contain a valid partition table
linuxconfig.org# cfdisk /dev/sdb
```

Use your preferred partitioning tool to create partitions. In this example I have used cfdisk.




```
page for additional information.
linuxconfig.org# fdisk -l

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          25      200781   83  Linux
/dev/sda2              26         522     3992152+   5  Extended
/dev/sda5              26         217     1542208+  83  Linux
/dev/sda6             218         299      658633+  83  Linux
/dev/sda7             300         327      224878+  82  Linux swap / Solaris
/dev/sda8             328         342      120456   83  Linux
/dev/sda9             343         522     1445818+  83  Linux

Disk /dev/sdb: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         243     1951866   83  Linux
/dev/sdb2             244         652     3285292+  83  Linux
linuxconfig.org#
```

Partitions are ready to use.

Create physical volumes

Use the pvcreate command to create physical volumes.

```
# pvcreate /dev/sdb1
# pvcreate /dev/sdb2
```

The pvdisplay command displays all physical volumes on your system.

```
# pvdisplay
```

Alternatively the following command should be used:

```
# pvdisplay /dev/sdb1
```

```
linuxconfig.org# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
linuxconfig.org# pvcreate /dev/sdb2
  Physical volume "/dev/sdb2" successfully created
linuxconfig.org# pvdisplay
  --- NEW Physical volume ---
  PV Name               /dev/sdb1
  VG Name
  PV Size               1.86 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               rFn4cW-LlQZ-ZfIG-rPxg-Wtw0-8XFf-ROypxN

  --- NEW Physical volume ---
  PV Name               /dev/sdb2
  VG Name
  PV Size               3.13 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               FqDSca-6z8F-RbI7-6apo-31LN-iCRp-7J2vng

linuxconfig.org#
```

Create Virtual Group

At this stage you need to create a virtual group which will serve as a container for your physical volumes. To create a virtual group with the name "mynew_vg" which will include /dev/sdb1 partition, you can issue the following command:

```
# vgcreate mynew_vg /dev/sdb1
```

To include both partitions at once you can use this command:

```
# vgcreate mynew_vg /dev/sdb1 /dev/sdb2
```

```
linuxconfig.org# vgcreate mynew_vg /dev/sdb1
  Volume group "mynew_vg" successfully created
linuxconfig.org# vgdisplay
  --- Volume group ---
  VG Name               mynew_vg
  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               1.86 GB
  PE Size               4.00 MB
  Total PE              476
  Alloc PE / Size       0 / 0
  Free  PE / Size       476 / 1.86 GB
  VG UUID               OtLsp7-dMvt-G60t-qHM0-Ev3W-YgmY-ZjhZOU

linuxconfig.org#
```

Feel free to add new physical volumes to a virtual group by using the vgextend command.

```
# vgextend mynew_vg /dev/sdb2
```

```
linuxconfig.org# vgextend mynew_vg /dev/sdb2
  Volume group "mynew_vg" successfully extended
linuxconfig.org# vgdisplay
  --- Volume group ---
  VG Name               mynew_vg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  2
  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               4.99 GB
  PE Size               4.00 MB
  Total PE              1277
  Alloc PE / Size       0 / 0
  Free  PE / Size       1277 / 4.99 GB
  VG UUID               OtLsp7-dMvt-G60t-qHM0-Ev3W-YgmY-ZjhZOU

linuxconfig.org#
```

Create Logical Volumes

From your big cake (virtual group) you can cut pieces (logical volumes) which will be treated as a partitions for your linux system. To create a logical volume, named "vol01", with a size of 400 MB from the virtual group "mynew_vg" use the following command:

- create a logical volume of size 400 MB -L 400

- create a logical volume of size 4 GB -L 4G

```
# lvcreate -L 400 -n vol01 mynew_vg
```

```
linuxconfig.org# lvcreate -L 400 -n vol01 mynew_vg
  Logical volume "vol01" created
linuxconfig.org# lvdisplay
  --- Logical volume ---
  LV Name                /dev/mynew_vg/vol01
  VG Name                mynew_vg
  LV UUID                CVolJV-4oN7-uBga-OeWB-TUOp-dem3-OBxD1G
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                400.00 MB
  Current LE             100
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0

linuxconfig.org#
```

With a following example you will create a logical volume with a size of 1GB and with the name vol02:

```
# lvcreate -L 1000 -n vol02 mynew_vg
```

```
linuxconfig.org# lvcreate -L 1G -n vol02 mynew_vg
  Logical volume "vol02" created
linuxconfig.org# lvdisplay
  --- Logical volume ---
  LV Name                /dev/mynew_vg/vol01
  VG Name                mynew_vg
  LV UUID                CVolJV-4oN7-uBga-OeWB-TUOp-dem3-OBxD1G
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                400.00 MB
  Current LE             100
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0

  --- Logical volume ---
  LV Name                /dev/mynew_vg/vol02
  VG Name                mynew_vg
  LV UUID                RQbUjW-wFuV-hZo9-ZFET-52ks-AJD8-Xe6jzk
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                1.00 GB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:1

linuxconfig.org#
```

Note the free size in virtual group.

```
linuxconfig.org# vgdisplay
  --- Volume group ---
  VG Name               mynew_vg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               4.99 GB
  PE Size               4.00 MB
  Total PE              1277
  Alloc PE / Size       356 / 1.39 GB
  Free  PE / Size       921 / 3.60 GB
  VG UUID               OtLsp7-dMvt-G60t-qHM0-Ev3W-YgmY-ZjhZOU

linuxconfig.org#
```

Create File system on logical volumes

The logical volume is almost ready to use. All you need to do is to create a filesystem.:

```
# mkfs.ext3 -m 0 /dev/mynew_vg/vol01
```

the -m option specifies the percentage reserved for the super-user, set this to 0 if you wish not to waste any space, the default is

```
    mke2fs 1.37 (21-Mar-2005)
    Filesystem label=
    OS type: Linux
    Block size=1024 (log=0)
    Fragment size=1024 (log=0)
    102400 inodes, 409600 blocks
    20480 blocks (5.00%) reserved for the super user
    First data block=1
    50 block groups
    8192 blocks per group, 8192 fragments per group
    2048 inodes per group
    Superblock backups stored on blocks:
            8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

    Writing inode tables: done
    Creating journal (8192 blocks): done
    Writing superblocks and filesystem accounting information: done

    This filesystem will be automatically checked every 31 mounts or
    180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

5%. linuxconfig.org#

Edit /etc/fstab

Add an entry for your newly created logical volume into /etc/fstab

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/sda1       /               ext3    defaults,errors=remount-ro 0       1
/dev/sda9       /home           ext3    defaults        0       2
/dev/sda8       /tmp            ext3    defaults        0       2
/dev/sda5       /usr            ext3    defaults        0       2
/dev/sda6       /var            ext3    defaults        0       2
/dev/sda7       none            swap    sw              0       0
/dev/hdc        /media/cdrom0   iso9660 ro,user,noauto  0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0
/dev/mynew_vg/vol01 /home/foobar ext3   defaults        0       0
~
~
~
~
~
~
~
                                                              13,65         All
```

Mount logical volumes

Before you mount do not forget to create a mount point.

```
# mkdir /home/foobar
```

```
linuxconfig.org# mkdir /home/foobar
linuxconfig.org# mount -a
linuxconfig.org# cd /home/foobar/
linuxconfig.org# df -h .
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/mynew_vg-vol01
                      388M  8.1M  360M   3% /home/foobar
linuxconfig.org#
```

Extend logical volume

The biggest advantage of logical volume manager is that you can extend your logical volumes any time you are running out of the space. To increase the size of a logical volume by another 800 MB you can run this command:

```
# lvextend -L +800 /dev/mynew_vg/vol01
```

```
linuxconfig.org# lvextend -L +800 /dev/mynew_vg/vol01
  Extending logical volume vol01 to 1.17 GB
  Logical volume vol01 successfully resized
linuxconfig.org# df -h .
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/mynew_vg-vol01
                      388M  8.1M  360M   3% /home/foobar
linuxconfig.org#
```

The command above does not actually increase the physical size of volume, to do that you need to:

```
# resize2fs  /dev/mynew_vg/vol01
```

Look at the figure below to see what problems you may encounter when extending a volume:

```
linuxconfig.org# df -h .
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/mynew_vg-vol01
                      388M  8.1M  360M   3% /home/foobar
linuxconfig.org# lvextend -L +800 /dev/mynew_vg/vol01
  Extending logical volume vol01 to 1.17 GB
  Logical volume vol01 successfully resized
linuxconfig.org# df -h .
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/mynew_vg-vol01
                      388M  8.1M  360M   3% /home/foobar
linuxconfig.org# resize2fs /dev/mynew_vg/vol01
resize2fs 1.37 (21-Mar-2005)
/dev/mynew_vg/vol01 is mounted; can't resize a mounted filesystem!

linuxconfig.org# cd /
linuxconfig.org# umount /home/foobar/
linuxconfig.org# resize2fs /dev/mynew_vg/vol01
resize2fs 1.37 (21-Mar-2005)
Please run 'e2fsck -f /dev/mynew_vg/vol01' first.

linuxconfig.org# e2fsck -f /dev/mynew_vg/vol01
e2fsck 1.37 (21-Mar-2005)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/mynew_vg/vol01: 11/102400 files (0.0% non-contiguous), 21167/409600 blocks
linuxconfig.org# resize2fs /dev/mynew_vg/vol01
resize2fs 1.37 (21-Mar-2005)
Resizing the filesystem on /dev/mynew_vg/vol01 to 1228800 (1k) blocks.
The filesystem on /dev/mynew_vg/vol01 is now 1228800 blocks long.

linuxconfig.org# mount -a
linuxconfig.org# cd /home/foobar/
linuxconfig.org# df -h .
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/mynew_vg-vol01
                      1.2G  8.1M  1.1G   1% /home/foobar
linuxconfig.org#
```

Remove logical volume

The command lvremove can be used to remove logical volumes. Make sure that before you attempt to remove logical volumes your logical volume does not have any valuable data stored on it, moreover, make sure the volume is unmounted.

```
# lvdisplay
```

```
linuxconfig.org# lvdisplay
  --- Logical volume ---
  LV Name                /dev/mynew_vg/vol02
  VG Name                mynew_vg
  LV UUID                RQbUjW-wFuV-hZo9-ZFET-52ks-AJD8-Xe6jzk
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                1.00 GB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:1

  --- Logical volume ---
  LV Name                /dev/mynew_vg/vol01
  VG Name                mynew_vg
  LV UUID                Z0NBw5-1Muk-InQl-7WcR-wPin-WaPd-sDCKNV
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                1.17 GB
  Current LE             300
  Segments               3
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0

linuxconfig.org#
```

```
# lvremove /dev/mynew_vg/vol02
```

```
linuxconfig.org# lvremove /dev/mynew_vg/vol02
Do you really want to remove active logical volume "vol02"? [y/n]: y
  Logical volume "vol02" successfully removed
linuxconfig.org# lvdisplay
  --- Logical volume ---
  LV Name                /dev/mynew_vg/vol01
  VG Name                mynew_vg
  LV UUID                Z0NBw5-1Muk-InQl-7WcR-wPin-WaPd-sDCKNV
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                1.17 GB
  Current LE             300
  Segments               3
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0

linuxconfig.org#
```