Step by Step to Create LVM Mirroring in Linux

In the computer world, storage is the most important factor. If data is lost, then everything is lost for the organization or company. Initially, data is stored in normal disks, it does not have the flexibility to manage the data and it is very complicated and disaster for recovery management. To achieve flexibility and goal of the disaster recovery management, RAID is introduced. Using RAID, redundancy can be maintained but still does not have the flexibility to manage filesystem and storage management. Logical Volume Manager (LVM) provides a higher level view of disk storage. LVM is traditionally associated with large installation containing many disks and even very use full in a small environment.

LVM has many features like resizing volume groups online by absorbing new physical volumes (PV) or removing existing PVs and resizing logical volumes (LV) online (https://linoxide.com/how-tos/extend-lvm-partition-online-linux/) by adding or removing physical extents to the existing LV. Using LVM you can create read-only as well as read-write snapshots of LV. It is possible to stripe or mirror whole or parts of LVs across multiple PVs, similar to RAID 0 or RAID 1. It is possible to move online LVs between PVs. The LVM will also work in a shared-storage cluster. We will show steps to create LVM mirroring

1. Create partition

To create LVM mirror, you need to be sure having free space. In our case, we will create two partitions using fdisk command (https://linoxide.com/linux-command/fdisk-commands-manage-partitions-in-linux/) (you can use a partition or whole disk). Before creating partition make sure that how many cylinders has been used and how many are free.

Checking the free space to create partition.

```
# fdisk -l

Disk /dev/sdf: 10.7 GB, 10737418240 bytes
 255 heads, 63 sectors/track, 1305 cylinders
 Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot Start End Blocks Id System
 /dev/sdf1 * 1 765 6144831 83 Linux
 /dev/sdf2 766 892 1020127+ 82 Linux swap / Solaris
```

The above command shows total cylinders are 1305 and used cylinders are 892. It means that still, you have free cylinders, hence we can create a partition.

Creating two new partition using fdisk.

```
# fdisk /dev/sdf
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type:
 p primary (2 primary, 0 extended, 2 free)
 e extended
Select (default p): p
Partition number (3,4, default 3):
First sector (391168-20971519, default 391168):
Using default value 391168
Last sector, +sectors or +size{K,M,G} (391168-20971519, default 20971519): +100MB
Partition 3 of type Linux and of size 95 MiB is set

Command (m for help): t
Partition number (1-3, default 3):
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): n
Partition type:
 p primary (3 primary, 0 extended, 1 free)
 e extended
Select (default e): p
Selected partition 4
First sector (585728-20971519, default 585728):
Using default value 585728
Last sector, +sectors or +size{K,M,G} (585728-20971519, default 20971519): +100MB
Partition 4 of type Linux and of size 95 MiB is set

Command (m for help): t
Partition number (1-4, default 4):
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'
```

```
Command (m for help): p

Disk /dev/sdf: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xabe92155

 Device Boot Start End Blocks Id System
/dev/sdf1 * 1 765 6144831 83  Linux
/dev/sdf2 766 892 1020127+ 82 Linux swap/Solaris
/dev/sdf3 391168 585727 97280 8e Linux LVM
/dev/sdf4 585728 780287 97280 8e Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Make sure that partition ID must be 8e while creating partitions.

2. Create physical volume

Now that we have free partitions, we need to create physical volume with the above partitions.

Create physical volume

```
# pvcreate /dev/sdf[3,4]
 Physical volume "/dev/sdf3" successfully created
 Physical volume "/dev/sdf4" successfully created
```

The above command will initialize partition as LVM partition. Here LVM will assign PV and UUID to the partition only.

3. Create volume group

Now that we have created our physical volume, we need to create a volume group. If you already have a VG based on two or more disk/partition with free space, you don't need to create a new volume group

Create volume group

```
# vgcreate datavg /dev/sdf[3,4]
 Volume group "datavg" successfully created
```

The above command will create volume groups as datavg. LVM creates Volume Group Descriptor Area (VGDA). The VGDA contains information about the volume groups. LVM creates Physical Extent (PE).

4. Creating Mirrored logical volume.

While creating mirrored volumes in particular volume group, you have to specify the number of copies of the data required with `-m` argument of the lvcreate command. For creating 1 mirror copy of data, you have to specify as `-m1`. Specifying `-m1` creates one mirror, which yields two copies of the file system.

Create mirrored lv

```
# lvcreate -L 50M -m1 -n mirrorlv datavg
 Rounding up size to full physical extent 52.00 MB
 Logical volume "mirrorlv" created
```

In the above command, we have created 50MB of the sized LV with 1 copy of data to be maintained. We can check with

```
# lvs
 LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
 root centos -wi-ao---- 17.47g
 swap centos -wi-ao---- 2.00g
 mirrorlv datavg rwi-a-r--- 50MB
```

You can check the status of your mirrored LV with `lvdisplay -v` command. Just search information concerning the logical volume created earlier as below

```
 # lvdisplay -v

--- Logical volume ---
  LV Path                /dev/datavg/mirrorlv
  LV Name                mirrorlv1
  VG Name                datavg
  LV UUID                J2kcdf-YxAn-k2d2-KFGS-pnhb-0Zhz-GRECFr
  LV Write Access        read/write
  LV Creation host, time centos7-srv, 2017-05-08 03:17:28 +0100
  LV Status              available
  # open                 0
  LV Size                52.00 MiB
  Current LE             13
  Mirrored volumes       2
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:18
```

You can see `Mirrored volumes 2` line which indicates the number of disks used for the logical volume and, we can also deduct another information which is the type of our logical volume: **mirrored**.

5. Creating filesystem and mounting it

Now that we have created our logical volume, we need to create a filesystem in order to access it

Create filesystem

```
# mkfs:ext4 /dev/datavg/mirrorlv
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
13328 inodes, 53248 blocks
2662 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=54525952
7 block groups
8192 blocks per group, 8192 fragments per group
1904 inodes per group
Superblock backups stored on blocks:
 8193, 24577, 40961

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done.
```

Now we can Mount the filesystem to a location of our choice. Most commonly used method for mounting filesystem is either manually or by using mount command or by adding entries in `/etc/fstab` so that the filesystem has been mount during boot time.

Mount the filesystem to `/mnt/mirrorlv` location

```
# mount /dev/datavg/mirrorlv /mnt/mirrorlv
```

You can verify by executing the command.

```
df -Ph
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/centos-root 18G 5.2G 13G 30% /
devtmpfs 897M 0 897M 0% /dev
tmpfs 912M 156K 912M 1% /dev/shm
tmpfs 912M 9.1M 903M 1% /run
tmpfs 912M 0 912M 0% /sys/fs/cgroup
/dev/sda1 497M 189M 309M 38% /boot
tmpfs 183M 8.0K 183M 1% /run/user/1000
/dev/mapper/datavg-mirrorlv 51M 842K 47M 2% /mnt/mirrorlv
```

Conclusion

A mirror maintains identical copies of data on different devices. When data is written to one device, it is written to a second device as well, mirroring the data. This provides protection for device failures.

Have anything to say?

Your email address will not be published. Required fields are marked *

Name *

Email *

All comments are subject to moderation.

What is linoxide based on ? Windows or Linux ? Type your answer into the box

Submit Your Comment