Create Linux Logical Volumes

LVM Storage Setup on Linux.

LVM (logical volume management) technology streamlines the process of handling data storage. Unlike the traditional partitioning model, LVM allows system administrators more leeway in how they manage disk space by virtualizing it. In order for logical volume management to function, the PVs must be partitioned into physical extents (PEs), which are then mapped to logical volumes (LEs).

Afterward, the Logical Extents are categorized by volume (VGs). These new Volume Groups are then integrated into logical volumes (LVs), which serve as the aforementioned fictitious disk partitions. With LVM, it's easy to resize or relocate storage volumes as needed.

Now that we know it, we can move forward with installing LVM. We'll use a flash drive as an example, but the process is the same for any storage medium (hard drive, etc).

Step 1: Find the device that uses disks.

List available devices and partitions using fdisk. From the output, we can see that there is a real device called /dev/sdb.

```
$ sudo fdisk -l Disk /dev/sda: 119.2 GiB, 128035676160 bytes, 250069680 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 Disklabel type: dos Disk identifier: 0xa3bc85b8 Device Boot Start End Sectors Size Id Type /dev/sda1 62517248 250067789 187550542 89.4G 83 Linux /dev/sda2 2048 62517247 62515200 29.8G 83 Linux Partition table entries are not in disk order. Disk /dev/sdb: 14.9 GiB, 15938355200 bytes, 31129600 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 Disklabel type: dos Disk identifier: 0x2a4e70c2 Device Boot Start End Sectors Size Id Type /dev/sdb1 8192 31129599 31121408 14.9G c W95 FAT32 (LBA)
```

Step 2: Create a partition for the LVM device.

Prepare the physical device by formatting it with fdisk, parted or gdisk. We will use the fdisk program. First, we'll delete the ones that are already there and make new ones.

```
$ sudo fdisk /dev/sdb Welcome to fdisk (util-linux 2.32.1). 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 (container for logical partitions) Select (default p): p Partition number (1-4, default 1): First sector (2048-31129599, default 2048): Last sector, +sectors or +size{K,M,G,T,P} (2048-31129599, default 31129599): +7G Created a new partition 1 of type 'Linux' and of size 7 GiB. Command (m for help): t Selected partition 1 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 (1 primary, 0 extended, 3 free) e extended (container for logical partitions) Select (default p): p Partition number (2-4, default 2): First sector (14682112-31129599, default 14682112): Last sector, +sectors or +size{K,M,G,T,P} (14682112-31129599, default 31129599): Created a new partition 2 of type 'Linux' and of size 7.9 GiB. Command (m for help): t Partition number (1,2, default 2): 2 Hex code (type L to list all codes): 8e Changed type of partition 'Linux' to 'Linux LVM'. Command (m for help): w The partition table has been altered. Failed to remove partition 1 from system: Device or resource busy Failed to add partition 1 to system: Device or resource busy Failed to add partition 2 to system: Device or resource busy The kernel still uses the old partitions. The new table will be used at the next reboot. Syncing disks.

Type fdisk -l as follows to check the LVM partitions:

$ sudo fdisk -l [sudo] password for penchant: Disk /dev/sda: 119.2 GiB, 128035676160 bytes, 250069680 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 Disklabel type: dos Disk identifier: 0xa3bc85b8 Device Boot Start End Sectors Size Id Type /dev/sda1 62517248 250067789 187550542 89.4G 83 Linux /dev/sda2 2048 62517247 62515200 29.8G 83 Linux Partition table entries are not in disk order. Disk /dev/sdb: 14.9 GiB, 15938355200 bytes, 31129600 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 Disklabel type: dos Disk identifier: 0x2a4e70c2 Device Boot Start End Sectors Size Id Type /dev/sdb1 2048 14682111 14680064 7G 8e Linux LVM /dev/sdb2 14682112 31129599 16447488 7.9G 8e Linux LVM Partition table entries are not in disk order.

Step 3: Create Physical Volume.

Use pvcreate to make Physical Volumes.

$ sudo pvcreate /dev/sdb1 /dev/sdb2 Physical volume "/dev/sdb1" successfully created. Physical volume "/dev/sdb2" successfully created.

Use the pvdisplay command to verify the physical volumes.

```
$ sudo pvdisplay --- Physical volume --- PV Name /dev/sdb1 VG Name tech PV Size 7.00 GiB / not usable 4.00 MiB Allocatable yes PE Size 4.00 MiB Total PE 1791 Free PE 214 Allocated PE 1577 PV UUID cGnGfI-oVG7-9CcY-kdmK-aR4R-iZY9-O9gD0g --- Physical volume --- PV Name /dev/sdb2 VG Name tech PV Size 7.84 GiB / not usable 3.00 MiB Allocatable yes (but full) PE Size 4.00 MiB Total PE 2007 Free PE 0 Allocated PE 2007 PV UUID UvewNB-Z2d1-T3L1-c92C-rOLa-lcrg-19zuPk
```

Step 4: Create a Volume Group

We can use vgcreate to make a volume group and give it any name we want. We will use the word "tech."

```
$ sudo vgcreate tech /dev/sdb1 /dev/sdb2 Volume group "tech" successfully created
```

Step 5: Create a Logical Volume

We can use lvcreate with a few options and switches to make a logical volume with a name and size that we choose.

With option -n, you can tell the program what the name of the logical volume is.

The size is set by the -L option. It can be in Megabytes (MiB) or Gigabytes (GiB).

```
$ sudo lvcreate -n part1 -L 14GiB tech Logical volume "part1" created.
```

When the command above is run, a device called /dev/tech/part1 will be made. By using the lvdisplay command, we can be sure of this. But this device doesn't have a way to store files.

```
$ sudo lvdisplay --- Logical volume --- LV Path /dev/tech/part1 LV Name part1 VG Name tech LV UUID O1qtcJ-dDAj-gPoL-nZn0-VUMs-rVwe-f31OHq LV Write Access read/write LV Creation host, time computing-pc, 2018-10-14 00:39:25 +0300 LV Status available # open 0 LV Size 14.00 GiB Current LE 3584 Segments 2 Allocation inherit Read ahead sectors auto - currently set to 256 Block device 254:0
```

With vgdisplay, we can also look at the volume groups.

Step 6: Make a Filesystem on a logical volume.

Load a file system that we want into the logical volume that we just made. Let us load xfs file system. Here, we can load ext3, ext4, brtfs, and other file systems as we like.

```
$ sudo mkfs -t xfs /dev/tech/part1 meta-data=/dev/tech/part1 isize=512 agcount=4, agsize=917504 blks = sectsz=512 attr=2, projid32bit=1 = crc=1 finobt=1, sparse=1, rmapbt=0 = reflink=0 data = bsize=4096 blocks=3670016, 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
```

After that, we were able to create a logical volume called `/dev/tech/part1` with the xfs file system.

|----------The End—----------|