



Hands On Labs

Managing Logical Volumes With LVM On CentOS 7

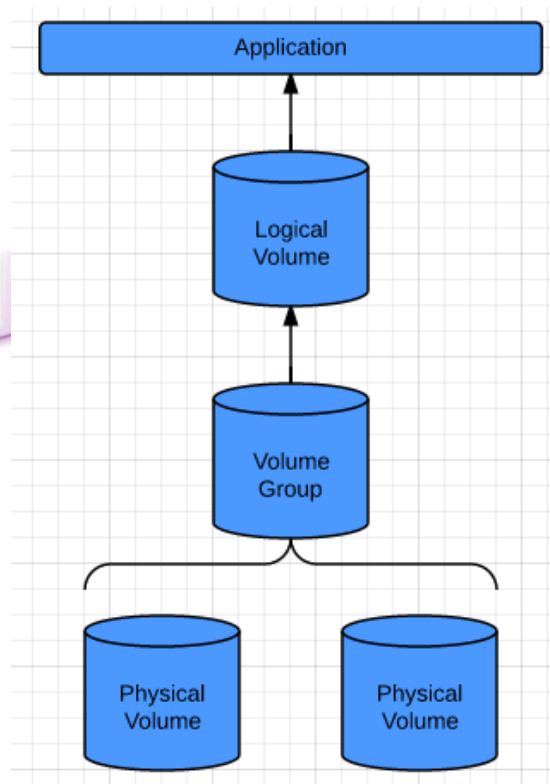
Table of Contents

Introduction	2
Lab Exercises And Setup	3
Physical Volumes	3
Volume Groups	3
Preparing The Physical Disks	4
Creating Physical Volumes For LVM	4
Creating A Volume Group	4
Create The Logical Volume	5
View The Device In /dev	6
Create A File System And Mount The Logical Volume	6
Extending Logical Volumes and Volume Groups	7



Introduction

The Logical Volume Manager, also known as LVM, provides a great amount of flexibility for applications that require expanding disk space. LVM works by creating a virtual layer of storage on top of physical devices. The OS operates against the LVM disk as if it was a regular disk. However, the advantages allow us to expand the disk without having to re-format the file system or create new partitions. If your application starts off needing 50gig and you attach a 50gig disk drive, what happens when your disk fills up? Very simply, with LVM you would attach new device, add it to the volume group and expand your virtual LVM disk. There are other uses of LVM such as the ability to take any combination of physical devices to create an abstract layer that allows that disk space to be combined together with LVM.



Lab Exercises And Setup

In this lab there are three physical disks attached to the lab server provided on the LinuxAcademy.com lab page. This lab is going to go through creating our initial physical volumes, volume groups, and the first Logical Volume. Upon completion we will then learn how to expand or “extend” the logical volume by attaching a new device to the volume group. At this point you should be connected to the LinuxAcademy.com lab server provided from this self paced lab.

The username will be linuxacademy and password 123456.

If you are not acting as the root user, type “sudo su -” and login as the root user to continue. This lab will assume you are either using sudo with each command or are acting as the root user.

Physical Volumes

Each LVM logical volume has an underlying physical storage unit. This is the physical volume component of LVM. A physical volume can either be a partition of a device or the entire disk. In order to use a physical volume, the physical volume must be initialized as a physical volume A.K.A (PV). By doing this a label is placed in the first part of the volume to help identify and provide meta data about the physical volume to our LVM manager. It is placed in the second 512-by sector of the physical volume. 0, 1, or 2 copies of this data can be stored on each physical volume; the default is 1 copy. Once configured, you cannot change the number of copies available. We know the first copy is stored at the start of the device but the second copy is stored at the end of the device. This helps protect against accidental overwriting of data.

Volume Groups

Volume group is a combination of physical volumes that creates a pool of space that the LVM or logical volumes can allocate.

Extents - Inside of a volume group, disk space is provided inside of allocatable fixed units called “extents”. An extent is the smallest unit of space that can be assigned to a volume group. Volume group extents are referred to as physical extents and a logical volume is allocated into sets of logical extents that are the same size as the physical extents. Thus the logical volume extents map to the physical volume extents and that is how a logical volume communicates with the physical volume data.

Preparing The Physical Disks

As we know, a physical disk is just an attached piece of storage. This is normally what we would use for storage on a file system if we were not using LVM.

On your lab server you have three disks attached to the server for this lab. For this portion of the lab we are going to be using two of them. The names will match here in the lab guide.

1. Navigate into the `/dev` directory
 - a. **cd /dev**
2. View the contents of the `/dev` directory. You will notice there is `xvdf` and `xvdg` - those are the two disks we will be using for this lab.
3. We must create a partition on each disk. For the purposes of this lab, we are going to create 1 partition that is the size of the entire disk.
4. `gdisk xvdf`
 - a. `n` - for new partition
 - b. Partition Number: Keep defaults - Press enter
 - c. First sector: Keep defaults - Press enter
 - d. Last sector: Keep defaults - press enter
 - e. Hex code or GUID **8e00** (we need to define what type of volume this is so providing the Linux LVM label for the volume with 8e00 is required)
 - f. Back at the command for `gdisk`, it is time to write the changes to disk - hit **w**, confirm with **Y**
5. Repeat step 4 for the **xvdg** disk.

Creating Physical Volumes For LVM

Now that our disks have been created correctly, we need to prepare them for use by the LVM manager. In order to do this, we will use the **pvcreate** and **pvdisplay** commands.

1. At the command line we need to define our physical volumes to be used with LVM
 - a. **pvcreate /dev/xvdg1 /dev/xvdf1**
2. We can now view the physical volumes that are available to LVM by typing **pvdisplay**

Creating A Volume Group

A volume group is a group of physical volumes that make up our allocatable disk space for a logical volume. A physical volume is made up of extents; an extent is also mapped from the volume group to the physical volume. This makes it easy to move data from one volume to another or transfer data from an old physical storage to a new physical storage device without interrupting our file system.

1. **vgcreate battlestar /dev/xvdg1 /dev/xvdf1**

Here we created a volume group. It's important to know that we didn't have to use both disks. In fact, we can have multiple volume groups that make up a disk and organize our data. In this lab we are keeping it simple and combining both disks into a single volume group that our LVM volume will choose

from. **Battlestar** is the name of the volume group; this is important because when we create our LVM volume we will have to choose available space from the battlestar volume group. We will likely name our LVM volume “galactica” or another ship in the battlestar fleet. No, this is not required for naming conventions; battlestar was chosen to represent a grouping for our naming convention. You are free to name it whatever you see fit.

1. **vgdisplay**

- a. Look at the output of the **vgdisplay** command. It is showing our volume groups. You can optionally pass the volume group name as a parameter to the command. PE size is the physical extent size. This is the unit of measurement that maps to a 4MiB amount of data on the physical volume. You’ll also notice the VG size which is the size of your volume group. It should be close to 40GB.

Create The Logical Volume

Now that we have prepared our physical volumes and associated them with a volume group, we can now create a logical volume. The logical volume will use disk space available from the volume group, which again maps to the physical volume.

Key knowledge

- **lvcreate** command has several flags associated with it. Please take note to what they are below:
 - **-n** (volume name)
 - **-L** size of the volume in bytes i.e 1M 1G 20G 1M etc.
 - **-l** size in physical extents.
 - If a physical extent is 4MiB (and it is for us) and you want to create an 8MiB LVM volume then **-l** would be 2.

Now it’s time to create the LVM

1. **lvcreate -n galactica -L 20G battlestar**
 - a. galactica is the name of our volume
 - b. 20G is the size of the volume
 - c. battlestar is the volume group we are using.
2. View your created volumes with **lvdisplay**

Now that we have created the volume we can use it just like a regular disk. We need to add a file system and mount it to the operating system. It’s important to know that XFS file system is most commonly used with LVM. This is because XFS allows LVM to scale up in data as well as scale down in data with little effort. EXT file systems do not allow you to shrink the device size like XFS does.

View The Device In /dev

1. `cd /dev`
2. `ls`
 - a. You will notice a new directory called “battlestar” which is the name of our volume group
3. `cd battlestar`
4. `ls`
 - a. You will see galactica, our logical volume. All logical volumes created from the battlestar volume group will be listed here. We format and mount the disk just like we would another device in /dev

Create A File System And Mount The Logical Volume

1. `mkfs -t xfs /dev/battlestar/galactica`
2. `mkdir /mnt/mydir`
3. `mount /dev/battlestar/galactica /mnt/mydir`
4. `df -h`
 - a. You will see the file system mounted at /mnt/mydir and is 20G in size.
 - b. You can optionally now add this to /etc/fstab and it acts just like a regular file system with the advantages of being able to resize the file system and move data and add/remove physical volumes to volume groups easily.

Extending Logical Volumes and Volume Groups

As part of this lab, there is a third physical volume available in /dev. For the rest of this lab we will add that volume to the battlestar volume group and then extend our already existing logical volume with the name of galactica to be 60G in available disk size. You'll notice we are using three physical disks and mounting them as one using the LVM.

1. **gdisk /dev/xvdj**
 - a. If you need help on this section, go to the first part of the lab to follow the instructions on using gdisk to create an LVM labeled disk.
2. **partprobe** - this command will register any new partitions to the kernel if it is not already completed.
3. **pvcreate /dev/xvdj1**
 - a. We need to create a physical volume for LVM using the pvcreate command
4. Next we need to add the new /dev/xvdj1 physical volume to our volume group
 - a. **vgextend battlestar /dev/xvdj1**
 - b. **vgdisplay** will show that the volume group now supports almost 60G in storage space.
5. Now that our volume group is 60GiB, we can extend our current logical volume or create a new logical volume. In this lab we are going to extend it and look at the different ways we can extend it
6. We want to add 39G to our current battlestar LVM so that it is 59G in size. Here is a list of ways you can do so:
 - a. Extend the volume to be a total of 59G
 - i. **lvextend -L 59G /dev/battlestar/galactica**
 - b. Add 39G to the logical volume
 - i. **lvextend -L +39G /dev/battlestar/galactica**
 - c. Please remember you can specify or do this with extents using the -l (lower case L) flag
7. Using **lvdisplay** we can see that the LV size is now 59.00 GiB
8. Using **df -h** we see that the operating system only sees 20G of that. We now need to execute the **xfs_growfs** command in order to resize the file system on the LVM volume.
9. **xfs_growfs /mnt/mydir**
10. **df -h**
 - a. Will show the disk is now 59G in usable disk space!

Great job! That concludes the LVM lab.