Linux Academy

Live! Lab

Managing Logical Volumes on Red Hat Enterprise 7

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- The IP address of your server is located on the Live! Lab page
- Username: linuxacademy
- Password: 123456
- Root Password: 123456

Related Courses

Red Hat CSA 7

Related Videos

Partitions on MBR and GPT Disks

Working with Logical Volumes

Extending Logical Volumes, and Swap to a System Non-Destructively

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Introduction

The *Logical Volume Manager* (LVM) is a flexible disk space solution that allows for expanding disk space. In this lab, we use the three provided physical disks attached to the lab server to create our logical volumes, and learn how to extend these volumes when we need more space.

Log in to the server using the credentials provided on the Live! Lab page. The commands below should be run as *root*, or prepended with `sudo` when running as the superuser.

The Logical Volume Manager

The Logical Volume Manager creates a virtual layer of storage on top of the physical hardware. The operating system then reads this *logical volume* as it would a regular disk. Using LVM also allows users to expand their disk space with ease — instead of working with complicated partitions, a new physical disk can be added to the LVM volume group.

Physical Volumes

Physical volumes are the disk hardware used to make a logical volume or volume group: This can be a partition on a disk, or the whole disk itself.

For a physical volume to be used, it must be initialized as one. This is done by adding a metadata label to the second 512-by sector of the volume, identifying it as a physical volume to the manager. This data can also be stored as a copy on the disk once, twice, or not at all. By default, one copy is made and stored at the end of the device.

Volume Groups

A volume group is a combination of physical volumes. These volumes create a pool of space that can be allocated as logical (not physical) volumes.

Extents

Inside of a volume group, disk space is provided in fixed units called *extents*; this is the smallest amount of space that can be assigned to a volume group.

Preparing the Physical Disks

The physical disks we are working with are located in the `/dev` directory, *xvdf* and *xvdg*. Navigate to `/dev` and list the contents.

We are creating LVM partitions on each disk with `gdisk`. For this lab, each partition should be the size of the entire disk.

To partition the *xvdf* disk, run:

```
[root@linuxacademy1 ~]# gdisk xvdf
```

The terminal prompts you to enter a command. Since we are creating a new partition, that command is `n`. Input the letter and press enter.

We want to use the default settings for **Partition number**, **First sector**, and **Last sector**. For the **Hex code or GUID**, we want to set it to *8e00*.

This outputs:

```
Changed type of partition to 'Linux LVM'
```

And then asks for another command. We want to *write* our changes with the `w` command. Confirm the changes.

Repeat this process for the *xvdg* disk.

Initializing the Disk

Although our disks are listed as *Linux LVM* disks, they still need to be initialized as a physical volume for use by the LVM. To do this, we use the `pvcreate` command:

```
[root@linuxacademy1 ~]# pvcreate /dev/xvdf1 /dev/xvdg1
```

To view the physical volumes, run:

```
[root@linuxacademy1 ~]# pvdisplay
```

Creating a Volume Group

A *volume group* is just that — a group of physical volumes that make up the allocated disk space for the logical volume.

We are creating a volume group called *battlestar* that uses both of the disks we just initialized:

```
[root@linuxacademy1 ~]# vgcreate battlestar /dev/xvdf1 /dev/xvdg1
```

Similar to before, you can run `vgdisplay` to see your available volume groups.

Creating the Logical Volume

With our physical volumes prepared and in a volume group, we can now create a logical volume. This uses space from the volume group, which is mapped to the physical disks.

To create the logical volume:

```
[root@linuxacademy1 ~]# lvcreate -n galactica -L 20G battlestar
```

The `-n` flag, above, denotes the name of the logical volume; we choose *galactica* because it fit our naming convention; you do not need to follow the same conventions.

The `-L` flag defines the size to allocate for the volume. *battlestar* is the name of the volume group.

Following the above trend, run `lvdisplay` to display the logical volumes you have.

With this finished, we can use the volume as we would a regular disk.

Viewing Your Work

You can view your finished device in the `/dev` directory. Here you should find a directory called *battlestar*:

```
[root@linuxacademy1 ~]# cd battlestar
[root@linuxacademy1 ~]# ls
```

From here, you can see the *galactica* logical volume.

Mounting the Logical Volume

While still in the `/dev/battlestar` directory, we need to create a file system and mount the disk.

Commonly, when creating a filesystem on LVM disks, we use the *XFS* file system. This allows LVM to scale up and down with data, although it does not allow you to shrink your device size.

Create the file system:

```
[root@linuxacademy1 ~]# mkfs -t xfs /dev/battlestar/galactica
```

We now want to create the `/mnt/mydir` directory and mount our filesystem:

```
[root@linuxacademy1 ~]# mkdir /mnt/mydir
[root@linuxacademy1 ~]# mount /dev/battlestar/galactica /mnt/mydir
```

To see if the logical volume has mounted successfully run:

```
[root@linuxacademy1 ~]# df -h
```

You should see your mounted system; this can also be added to the `/etc/fstab` for it to act as a regular file system.

Extending Logical Volumes and Volume Groups

There is a third available device in the `/dev` directory that we did not use thus far in this lab, *xvdj*. We want to take this disk and add it to our initial volume group (*battlestar*). Then, we want to extend our logical volume (*galactica*) to be 60G in size by using all three physical disks available.

Partition the new disk:

```
[root@linuxacademy1 ~]# gdisk xvdj
```

Run `partprobe` to register the new partition with the kernel.

Create the physical volume:

```
[root@linuxacademy1 ~]# pvcreate /dev/xvdj1
```

We now need to add this to the *battlestar* group. To do this, we use the `vgextend` command:

```
[root@linuxacademy1 ~]# vgextend battlestar /dev/xvdj1
```

Should you now run `vgdisplay`, the VG size is close to 60G, versus the 40 we had initially set.

We also want to expand the *galactica* logical volume to use the whole volume group — all 59.99 GiB. This means we want to *extend* the volume to be 59G in size:

```
[root@linuxacademy1 ~]# lvextend -L 59G /dev/battlestar/galactica
```

Alternately, we can add the 39G of additional storage with the following command:

```
[root@linuxacademy1 ~]# lvextend -L +39G /dev/battlestar/galactica
```

Note the addition of the addition sign (+) to the beginning of the size.

Use `lvdisplay` to confirm the size changes. However, if you run `df -h` you see that the operating system still only recognizes 20G of that. We need to use XFS's grow file system command to extend this:

```
[root@linuxacademy1 ~]# xfs_growfs /mnt/mydir
```

Run `df -h` again to confirm.