8.3.2 Using LVM

Click one of the buttons to take you to that part of the video.

Use LVM 0:00-0:22

In this demonstration, we're going to practice working with Logical Volume Manager, LVM. In order to create an LVM volume on our system here, we need to first define our physical volumes, then we need to create a volume group from those physical volumes, and finally within that volume group, define one or more logical volumes.

Create Physical Volumes 0:23-2:33

In this system, I have three hard disks currently set up. I have sda, sdb, and sdc. sda is my system volume where my operating system resides, where my home directories are, and so on.

We're not going to touch sda. In fact, sda already has several LVM logical volumes defined on it. They were created when the system was initially installed, and you'll see those as we go through the demonstration.

However, sdb and sdc are new hard disks, so they're blank--there's nothing on them. We want to use them to create LVM logical volumes. The first thing we need to do is create physical volumes out of sdb and sdc. To do this, I first need to switch to my root user account, because I have to have root level privileges to manage the storage in the system.

Let's create a physical volume out of sdb. This is done using the pvcreate command. We simply enter 'pvcreate', and then the name of the device file, '/dev/sdb'. We're told that we have a new physical volume created for LVM out of sdb. Let's do the same thing for sdc.

At this point, we should have two physical volumes defined for LVM: sdb and sdc. Let's verify this using the 'pvscan' command. Notice that it tells us we have three physical volumes defined.

The first one is one that was created during the system installation on sda. We're not going to be concerned with that one. We are going to be concerned with these two physical volumes here: sdb and sdc. Notice that each one was about 20 GB in size.

By defining these as LVM physical volumes, we can now aggregate the space from these two storage devices and then create a logical volume on them. Currently, we can't yet, because they are still running independently of each other.

In order to define logical volumes on these two devices, we need to first add them to a single volume group. We currently have one volume group defined on the system. It's fedora_fS1. Again, this was created during system installation. We're not going to work with that volume group.

Create Volume Groups 2:34-4:12

Instead, what we want to do is create a new one, so we do the 'vgcreate' command, and then we specify the name of the volume group we want to create. Let's call it 'data', and then we can specify which physical volume we want to initially add to that volume group. Let's add '/dev/sdb'.

At this point, we've created the volume group name data, and we've added one of the physical volumes to that group. It's important to note that the vgcreate command can be used only to create a volume group; it cannot be used to modify a volume group, such as to add another physical volume to it. If we want to add another physical volume to an existing volume group, we need to use a different command.

This command is vgextend. We run 'vgextend', we specify the name of the volume group--the existing volume group, because we are modifying it, we're not creating it now. Then we specify the name of the physical volume that we want to add to that existing volume group, '/dev/sdc'.

At this point we should have a single volume group defined, and the two physical volumes should be members of the same volume group. Let's run the 'pvscan' command again and verify that this is the case, and it is.

We have a volume group named data now, and sdb and sdc are both members of that volume group. With the volume group defined, we can now finally define logical volumes from within that volume group. We could define just one logical volume, or we could define many logical volumes.

Create Logical Groups 4:13-6:40

For our purposes, let's define two different logical volumes within the data volume group. We'll do a volume named shared, and a volume named private. The shared volume will be used for users to share data back and forth, while the private volume will contain information that's proprietary in nature, and access will be restricted.

To define these two logical volumes, we use the lvcreate command. We enter 'lvcreate', and then '-L', followed by the size of the volume we want to create. The shared volume will have lots of users' data on it, so it will need to be a little bit bigger in size.

Let's specify that it be '25G' in size. We have to provide a name for the new volume. Again, we said it would be called 'shared'. Then we have to specify which volume group we want to create this volume within. In this case we're going to use the 'data' volume group that we just defined up here using our two physical volumes.

Hit Enter, the logical volume shared has been created. We can verify this using the 'lvscan' command. Notice here we have a logical volume now defined, /dev, and then the name of the volume group data, and then the name of the logical volume shared--25 GB in size.

These other two volume groups were defined when the system was created. Notice they're in a totally different volume group; they're in the fedora_fS1 volume group. We have a swap partition, and we have our root partition defined. Within that volume group, but again we're not working with that one today, we're dealing strictly with the data volume group.

Let's go ahead and run the same command again, this time to create the private volume that we talked about earlier. Let's make it a little smaller; let's just make it '10G' in size. We'll change the name to 'private', but we will create it within the same volume group: 'data'. And the logical volume private is created.

We'll run 'lvscan' again, and we see that we have two logical volumes defined within the same volume group: shared and private.

If we look back up here real quick, notice that sdb is 20 GB in size and sdc is 20 GB in size, but down here, we have a 25 GB volume defined named shared and a 10 GB volume defined.

Review Advantages of LVM 6:41-7:31

One of the great things about LVM is the fact that all of the space on these two volumes up here have been aggregated into basically just a big pool of available space. Within that pool we defined these two volumes, and as we did so, the physical disk boundaries associated with the actual hard disk drives were completely irrelevant.

It didn't matter that we created a 25 GB volume with two hard disk drives that are only 20 GB in size. With LVM, the volume can easily span two different physical hard disk drives. Not a problem, it's very happy to work that way.

At this point, our logical volumes have been created; we would now go ahead and create file systems on those logical volumes just like we would a standard disk partition, and we would mount them in the file system, just like we would a standard disk partition as well. We'll cover those in separate demonstrations.

Summary 7:32-7:42

In this demonstration, we first used LVM to define physical volumes from two hard disk drives installed in the system, we then defined a volume group from those physical volumes, and then within that volume group, we defined two logical volumes.