Linux Academy

Live! Lab

Extending Existing Logical Volumes

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

In a previous lab, we learned how to use the Logical Volume Manager (LVM) to create and manage logical volumes. In this lab, we review these concepts and then learn how to extend and move volume groups, physical extents, and logical volumes.

This lab contains three physical disks attached to the lab server that need to be configured with LVM.

Before you begin, log into your server, and switch to the *root* user or prepend the below commands with `sudo` as a superuser.

Setting Up the Volumes

As in previous labs, you need to create the basic physical volumes, volume groups, and logical volumes. If you do not remember how to do this, step-by-step instructions are provided below. More detailed instructions are included with the Managing Logical Volumes on Red Hat Enterprise 7 lab guide.

Create a single partition from the `/dev/xvdf` disk:

```
[root@linuxacademy1 ~]# fdisk /dev/xvdf
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.
Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xc1ab6071.

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039):
Using default value 41943039
Partition 1 of type Linux and of size 20 GiB is set

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

Calling ioctl() to re-read partition table.

Syncing disks.
```

Create a physical volume from the parition:

```
[root@linuxacademy1 ~]# pvcreate /dev/xvdf1
  Physical volume “/dev/xvdf1” successfully created
```

```
[root@linuxacademy1 linuxacademy]# pvdisplay
  “/dev/xvdf1” is a new physical volume of “20.00 GiB”
  --- NEW Physical volume ---
  PV Name               /dev/xvdf1
  VG Name
  PV Size               20.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               pf5uvD-bAHa-o2cL-5VZ0-OWTM-gJD9-f55XrQ
```

Create the volume group, *battlestar*:

```
[root@linuxacademy1 ~]# vgcreate battlestar /dev/xvdf1
  Volume group “battlestar” successfully created
```

Created a logical volume, *galactica*:

```
[root@linuxacademy1 linuxacademy]# lvcreate -n galactica -L 1G battlestar
  Logical volume “galactica” created.
```

Create an XFS file system on the volume just created:

```
[root@linuxacademy1 battlestar]# mkfs -t xfs /dev/battlestar/galactica
meta-data=/dev/battlestar/galactica isize=256    agcount=4, agsize=65536 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=262144, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
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
```

Mount the volume:

```
[root@linuxacademy1 ~]# mkdir -p /mnt/myvolume
[root@linuxacademy1 !]# mount /dev/battlestar/galactica /mnt/myvolume
```

Create two files in the `/mnt/myvolume` directory:

```
[root@linuxacademy1 ~]# cd /mnt/myvolume; touch {file1,file2}
```

Extending a Volume Group

As part of our deployment plan, we need to add another disk to the system. This disk is intended to replace the current *xvdf1* disk, and we want to move the current data to the new disk (these are the files we `touch`ed).

Create a new disk using `gdisk` for the */dev/xvdg* disk; use the entire disk for a single parition:

```
[root@linuxacademy1 myvolume]# gdisk /dev/xvdg
GPT fdisk (gdisk) version 0.8.6

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries.
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-41943006, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-41943006, default = 41943006) or {+-}size{KMGTP}:
Current type is ‘Linux filesystem’
Hex code or GUID (L to show codes, Enter = 8300): 8e00
Changed type of partition to ‘Linux LVM’

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdg.
The operation has completed successfully.
```

Create the physical volume:

```
[root@linuxacademy1 myvolume]# pvcreate /dev/xvdg1
  Physical volume “/dev/xvdg1” successfully created
```

Extend the volume group, and display to ensure it has extended properly:

```
[root@linuxacademy1 myvolume]# vgextend battlestar /dev/xvdg1
  Volume group “battlestar” successfully extended
[root@linuxacademy1 myvolume]# vgdisplay
  --- Volume group ---
  VG Name               battlestar
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               39.99 GiB
  PE Size               4.00 MiB
  Total PE              10238
  Alloc PE / Size       256 / 1.00 GiB
  Free  PE / Size       9982 / 38.99 GiB
```

You may have noticed that in this section we used `gdisk` and not `fdisk`, creating a GPT partition (compared to `fdisk`'s MBR). We can migrate old MBR disks to GPT since we can copy the physical extends from one partition to the other.

Moving Physical Extents

Each physical volume is divided into chunks of data called *extents*, which are covered more in-depth in the initial LVM lab. It is possible to move extents between physical volumes, as long as there is open space.

To move extents off *xvdf1* we can use `pvmove`:

```
[root@linuxacademy1 myvolume]# pvmove /dev/xvdf1
  /dev/xvdf1: Moved: 0.4%
  /dev/xvdf1: Moved: 24.6%
  /dev/xvdf1: Moved: 48.4%
  /dev/xvdf1: Moved: 71.1%
  /dev/xvdf1: Moved: 95.7%
  /dev/xvdf1: Moved: 100.0%
```

We can now remove *xvdf1* from our volume group:

```
[root@linuxacademy1 myvolume]# vgreduce battlestar /dev/xvdf1
  Removed “/dev/xvdf1” from volume group “battlestar”
```

If you navigate to the mounted directory (`/mnt/mymount`) you can see that your files are still there. Although the files were written on the *xvdf1* disk, because we moved them off the initial group, they are still available.

Extending Logical Volumes

We can also extend the *galactica* LVM we created earlier. To extend the volume by 5 GiB, run:

```
[root@linuxacademy1 myvolume]# lvextend -L 5G /dev/battlestar/galactica
  Size of logical volume battlestar/galactica changed from 1.00 GiB (256 extents) to 5.00
GiB (1280 extents).
  Logical volume galactica successfully resized.
```

Now if you perform `df -h` on your system, you can see that the system still has not read the changes we have made. To fix this, we need to resize the file system:

```
[root@linuxacademy1 myvolume]# xfs_growfs /mnt/myvolume
meta-data=/dev/mapper/battlestar-galactica isize=256    agcount=4, agsize=65536 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=262144, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 262144 to 1310720
```

Notice the use of the mount location and not the volume name.

Run `df -h` again to cofirm changes. You have completed this lab!