Chapter 23 Title - Notes

23.3 Learning Objectives:

- Explain the concepts behind **LVM**.
- Create logical volumes.
- Display logical volumes.
- Resize logical volumes.
- Use **LVM** snapshots.

23.4 LVM

LVM (**L**ogical **V**olume **M**anagement) breaks up one virtual partition into multiple chunks, each of which can be on different partitions and/or disks.

Many advantages to using LVM. Particularly, becomes really easy to change size of logical partitions and filesystems, to add more storage space, rearrange things, etc.

One or more **physical volumes** (disk partitions) grouped together into **volume group**. Then, volume group subdivided into **logical volumes**, which mimic to system nominal physical disk partitions, and can be formatted to contain mountable filesystems.

Variety of command line utilities tasked to create, delete, resize etc. physical and logical volumes. For graphical tool, some Linux distributions use **system-config-lvm**. However, RHEL 7 does not support this tool, and there is currently no graphical tool that works reliably with most recent variations in filesystem types etc. Command line utilities not hard to use, and more flexible anyway.

LVM does impact performance. Definite additional cost that comes from overhead of LVM layer. However, even on non-RAID systems, if using **striping** (splitting of data to more than one disk), can achieve some parallelization improvements.

LVM Components

23.5 LVM and RAID

Like RAID (which will be discussed in next chapter), use of logical volumes = mechanism for creating filesystems which can span more than one physical disk.

Logical volumes created by putting all devices into large pool of disk space (the volume group), then allocating space from pool to create logical volume.

Logical volumes have features similar to RAID devices. Can actually be built on top of RAID device. Would give logical volume redundancy of RAID device with scalability of LVM.

LVM has better scalability than RAID: logical volumes can easily be resized, ie. enlarged or shrunk, as needs require. If more space needed, additional devices can be added to logical volume at any time.

23.6 Volumes and Volume Groups

Partitions converted into physical volumes, and multiple physical volumes grouped into volume groups. Can be more than one volume group on system.

Space in volume group divided into **extents**. 4 MB in size by default,but size can be easily changed when being allocated.

Number of command line utilities used to create/manipulate volume groups, whose name always start with **vg** including:

- **vgcreate**: Create volume groups
- **vgextend**: Adds to volume groups
- **vgreduce**: Shrinks volume groups

Utilities that manipulate what physical partitions enter or leave volume groups start with **pv** and include:

- **pvcreate**: Converts a partition to a physical volume
- **pvdisplay**: Shows the physical volumes being used
- **pvmove**: Moves the data from one physical colume within volume group to others. Might be required if a disk or partition is being removed for some reason. Would then be followed by:
- **pvremove**: Remove a partition from a physical volume

Typing **man lvm** will give full list of LVM utilities.

23.7 Logical Volume Utilities

Number of utilities that manipulate logical volumes. Unsurprisingly, all begin with **lv**. Will discuss most commonly used ones, but short list can be obtained using following command:

```
$ ls -lF /sbin/lv*
```

Note:

- These utilities are in `/sbin`, not `/usr/sbin`, as they may be needed either for boot or repair and recovery
- Most of them symbolically linked to **lvm**, Swiss army knife program that does all the work, but figures out what is being asked to do based on name it is invoked with. Also true for most of **pv*** and **vg*** utilities, as can be verified easily

sbinlvm

23.8 Creating Logical Volumes

lvcreate allocates logical volumes from within volume groups. Size can be specified either in bytes or number of extents (remember, 4 MB by default). Name can be anything desired.

lvdisplay reports on available logical volumes.

Filesystems placed in logical volumes and formatted with **mkfs** as usual.

Starting with possibly creating new volume group, steps involved in setting up and using a new logical volume are:

1. Create partitions on disk drives (type `8e` in **fdisk**)
2. Create physical volumes from partitions
3. Create volume group
4. Allocate logical volumes from volume group
5. Format logical volumes
6. Mount logical volumes (also update `/etc/fstab` as needed)

For example, assuming already created partitions `/dev/sdb1` and `/dev/sdc1`, and given them type `8e`:

```
$ sudo pvcreate /dev/sdb1
$ sudo pvcreate /dev/sdc1
$ sudo vgcreate -s 16M vg /dev/sdb1
$ sudo vgextend vg /dev/sdc1
$ sudo lvcreate -L 50G -n mylvm vg
$ sudo mkfs -t ext4 /dev/vg/mylvm
$ mkdir /mylvm
$ sudo mount /dev/vg/mylvm /mylvm
```

Be sure to add line the line

```
/dev/vg/mylvm /mylvm ext defaults 1 2
```

to `/etc/fstab` to make this persistent mount.

23.9 Displaying Logical Volumes

In order to display information about LVM, following command line programs available:

- **pvdisplay** shows physical volumes. If you leave off physical volume name, lists all physical volumes:

```
$ pvdisplay
$ pvdisplay /dev/sda5
```

- **vgdisplay** shows volume groups. If you leave off volume group name, lists all volume groups:

```
$ vgdisplay
$ vgdisplay /dev/vg0
```

- **lvdisplay** shows logical volumes. If you leave off logical volume name, lists all logical volumes:

```
$ lvdisplay
$ lvdisplay /dev/vg0/lvm1
```

Without arguments, utilities will display all physical volumes, volume groups, or logical volumes.

23.10 Resizing Logical Volumes

One great advantage of using LVM: easy and quick to change size of logical volume, especially when compared with trying to do so with physical partition that already contains filesystem. When doing this, extents can be added/subtracted from logical volume, and can come from anywhere in volume group. Need not be from physically contiguous sections of disk.

If volume contains filesystem, expanding/shrinking an entirely different operation than changing size of volume. WHen expanding logical volume with filesystem, must first expand volume, then expand filesystem. When shrinking logical volume with filesystem, must first shrink filesystem, then shrink volume.

Best done with **lvresize**:

```
$ sudo lvresize -r -L 20 GB /dev/VG/mylvm
```

where the `-r` option causes resizing of filesystem at same time as volume size change.

Filesystem cannot be mounted when being shrunk. However, some filesystems permit expansion while mounted.

Utilities which change filesystem size -> obviously filesystem dependent. Besides **lvresize**, can also use **lvextend**, **lvreduce** with **resize2fs**.

23.11 Examples of Resizing

To grow logical volume with **ext4** filesystem:

```
$ sudo lvextend -L +500M /dev/vg/mylvm
$ sudo resize2fs /dev/vg/mylvm
```

where the plus sign indicates adding space.

To shrink filesystem:

```
$ sudo umount /mylvm
$ sudo fsck -f /dev/vg/mylvm
$ sudo resize2fs /dev/vg/mylvm 200M
$ sudo lvreduce -L 200M /dev/vg/mylvm
$ sudo mount /dev/vg/mylvm
```

TUrns out that if you have recent version of **lvm** utilities, can do these operations in one step isntead of two, using `-r` option:

```
$ sudo lvextend -r -L +100M /dev/vg/mylvm
$ sudo lvreduce -r -L -100M /dev/vg/mylvm
```

where quantities use plus/minus signs to indicate change required. Uses underlying **fsadm** utility, which can resize any type of filesystem for which you have support. Would be good idea to read **man** page for **fsadm**.

Can also reduce volume group:

```
$ sudo pvmove /dev/ads1
$ sudo vgreduce vg /dev/sdc1
```

23.12 LVM Snapshots

LVM snapshots create exact copy of existing logical volume. Useful for backups, application testing, deploying VMs (Virtual Machines). Original state of snapshot kept as block map.

Snapshots only use space for storing deltas:

- When original logical volume changes, original data blocks copied to snapshot
- If data added to snapshot, stored only there

To create snapshot of existing logical volume:

```
$ sudo lvcreate -l 128 -s -n mysnap dev/vg/mylvm
```

To then make mount point and mount snapshot:

```
$ mkdir /mysnap
$ mount -o ro /dev/vg/mysnap /mysnap
```

To use snapshot and to remove snapshot:

```
$ sudo umount /mysnap
$ sudo lvremove /dev/vg/mysnap
```

Always be sure to remove snapshot when through with it. If you do not remove snapshot and fills up because of changes, automatically disabled. Snapshot with size of original will never overflow.

##

Back to top

Previous Chapter - Table of Contents - Next Chapter