

Implementing Logical Volumes

on Linux-based Dell PowerEdge Servers

Sistina® Logical Volume Manager (LVM) software provides an abstraction layer that allows administrators to work with logical volumes, helping to simplify storage management. This article describes LVM and presents scenarios for using LVM to resize storage volumes on Dell™ PowerEdge™ and PowerVault™ systems running the Red Hat® Linux® operating system.

BY TEFAMARIAM MICHAEL AND JOSHUA GILES

System administrators who manage disk storage in network environments face several challenges: ensuring that data-intensive applications get the data they need quickly, providing and organizing additional storage, preserving data integrity, and backing up data regularly. Scalable storage management software that enables administrators to reconfigure storage allocations dynamically in response to changing enterprise needs can help meet these challenges. One such open source product is the Sistina® Logical Volume Manager (LVM), which is embedded in the Linux® kernel. LVM combines physical disk drives into easily manageable logical volumes and allows online storage administration through tasks that are transparent to the user and the application.

This article discusses the basic architecture and configuration of LVM running under the Red Hat® Linux operating system. It also explains the steps for using LVM to manage storage in Dell™ PowerEdge™ servers and Dell PowerVault™ systems.

Understanding LVM architecture

LVM comprises five basic structures: volume group (VG), logical volume (LV), logical extent (LE), physical extent

LVM combines physical disk drives into easily manageable logical volumes and allows online storage administration through tasks that are transparent to the user and the application.

(PE), and physical volume (PV). A physical volume is typically a hard disk or a partition—for example, a SCSI disk or a RAID abstraction of a hard disk. Each physical volume comprises chunks of data, called physical extents. All physical extents within a volume group are the same size. In Figure 1, the volume group, called Vol_Grp00, is at the top level of abstraction. This group is obtained by

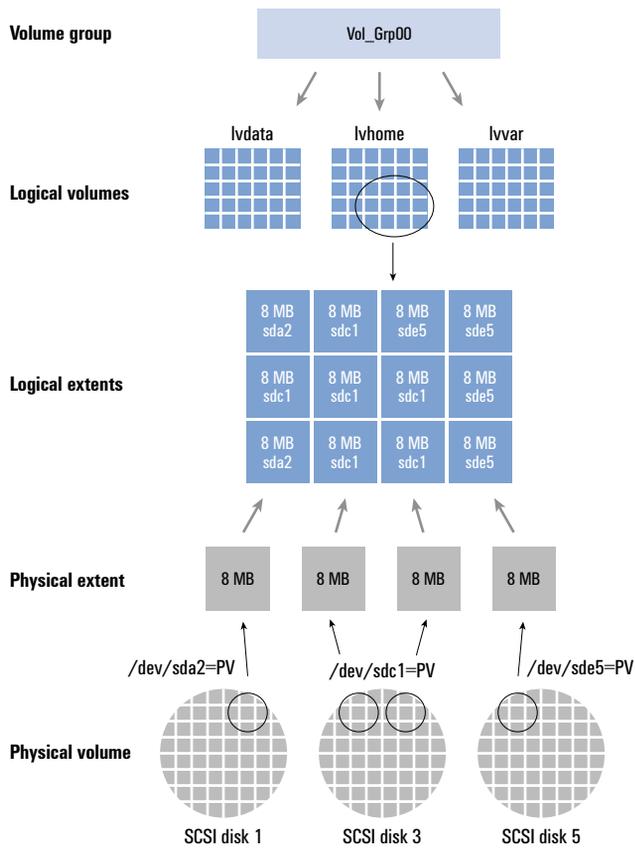


Figure 1. LVM schematic diagram

mapping together the physical volumes /dev/sda2, /dev/sdc1, and /dev/sde5 to create logical volumes that form one manageable group.

In Figure 1, lvdata, lvhome, and lvvar represent logical volumes, which contain the file system and thus can be used as mount points for directories such as /var/ftp, /home, and /usr. Each logical volume is split into chunks of data, called logical extents. Within a volume group, the size of the logical extent is the same as that of the physical extent; that is, a one-to-one mapping exists between the logical extents and the physical extents.

Each physical extent has a unique identification number on a physical volume but does not necessarily have a unique identification number on a logical volume because several different physical volumes can constitute one logical volume. Therefore, the logical extent identification numbers also identify the associated physical extents. Whenever the storage area is accessed, the address or the identification number of the logical extent is used to actually perform I/O on the physical storage.

The volume group descriptor area (VGDA) is stored at the beginning of each physical volume and functions similarly to the

partition table for LVM. The VGDA contains one volume group descriptor, one physical volume descriptor, one logical volume descriptor, and several physical extent descriptors. When the system is booted up, the volume group and logical volumes are activated and the VGDA is loaded into memory. The VGDA allows LVM to identify where the logical volumes are actually stored. The one-to-one mapping of physical to logical volumes is necessary to access the physical location and to perform I/O operations.

Conceptualizing LVM processes

LVM comprises lvm-mod, a kernel module (that is, a device driver) under General Public License (GPL), and applications that use the module to perform storage-related management processes. In a typical scenario, administrators use an LVM command such as pvcreate or vgscan to perform storage management functions. For example, Figure 2 represents the execution flow of an LVM task to retrieve the status of a particular storage device, as follows:

Because storage management is one of the main challenges system administrators regularly face, most major Linux distributions have included LVM with their installers, simplifying the task of LVM deployment.

1. LVM commands invoke the LVM kernel module, lvm-mod, to perform the operating system-level tasks that are required to service the request.
2. The device driver executes the request on the hardware storage device, and the LVM kernel module configures the device dynamically through the /proc file system.
3. LVM dynamically obtains information on the storage devices by using the /proc file system.

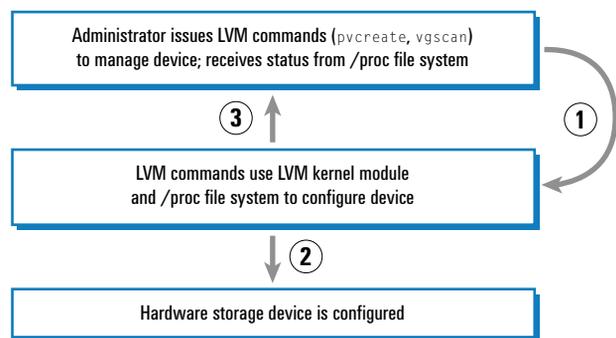


Figure 2. LVM command process

File system mount point	Size (MB)	File system type	LVM group*	Logical volume name
/	6144	ext3	None	None
/boot	128	ext3	None	None

Logical volume mount point	Size (MB)	File system type	LVM group	Logical volume name
/home	4096	ext3	lvm00	lvm_home
/usr	6144	ext3	lvm00	lvm_usr
/var	6144	ext3	lvm00	lvm_var
/data	4096	ext3	lvm00	lvm_data

*Neither the root nor boot partitions use LVM.

Figure 3. Example partition values

Implementing LVM during a Linux installation

Because storage management is one of the main challenges system administrators regularly face, most major Linux distributions have included LVM with their installers, simplifying the task of LVM deployment. For instance, Red Hat Linux included LVM starting with its 8.0 release. Although LVM can be introduced during system installation or any time after, this article discusses only the case of deploying LVM during Red Hat Linux installation and modifying storage volumes afterward. Migration of an already installed system to LVM can be complex, especially if it involves moving any file systems required for system boot, such as root and boot, to an LVM volume.

An important problem administrators must address when installing Linux is how best to partition and allocate system storage. Predicting how these partitions will be used during the lifetime of the system can be difficult. Even though storage on Linux systems can be expanded by adding more disks or storage enclosures, expanding or modifying existing partitions is very risky, if not impossible. LVM greatly simplifies the task of expanding or modifying partitions.

For data protection and integrity, Dell strongly suggests using a Dell PowerEdge Expandable RAID Controller (PERC) or Dell Cost-Effective RAID Controller (CERC) implementing a RAID level of 1, 5, or higher, depending on system needs.

Planning storage partitions and allocations

Before installing Linux, administrators should plan how to partition and allocate their storage. They should also evaluate how much risk and complexity they are willing to handle in order to recover a failed system. For example, creating root and boot partitions outside LVM makes system recovery easier by eliminating the need to activate LVM when rescuing a failed system. Also, those using Linux Loader (LILO) as the preferred boot loader should be aware that it

does not support LVM. Administrators using LILO must put the boot partition outside LVM.

Although LVM can be used with several file system types, such as ext2, ext3, and ReiserFS, this article addresses only ext3, the default Red Hat Linux file system. Figure 3 shows an example partitioning scheme; the installation steps described in the following sections use these values.

Using Disk Druid for manual partitioning

Implementing a partitioning scheme such as the one shown in Figure 3 during installation of Red Hat Linux 9 Professional is straightforward. The LVM feature is available only in the graphical user interface (GUI) installation mode. To use this feature, boot the system by using the Red Hat Linux 9 CD 1. At the boot prompt, press the Enter key to install in GUI mode. At the Disk Partitioning Setup screen, select “Manually partition with Disk Druid,” followed by “Next.”

When the Disk Druid menu is displayed, administrators can use the New button to create both boot and root partitions on the first hard drive, /dev/sda. Then, selecting the remaining free space of /dev/sda to edit and setting the file system to LVM enables administrators to create physical volumes on the remaining space of the first drive. This step can be repeated for all the remaining hard drives. Selecting the LVM button on the Disk Druid screen launches the Make LVM Volume Group window.

From this window, administrators can add, edit, and delete logical volume groups. Other possible actions include setting the volume group name and physical extent size (the default is 4 MB) and choosing the hard drive on which to create the logical volume. The Add and Edit buttons allow administrators to set the mount point, size, file system type, and name of the planned logical volumes (see Figure 4).

If implemented using the default setting of 4 MB per physical extent, a volume group is limited to 255 GB because only 65,534 physical or logical extents are allowed in a volume group. To provide more than 256 GB in a volume group, administrators must create the physical extents in chunk sizes larger than 4 MB. Physical extents must be a power of 2 and can range anywhere between 8 KB

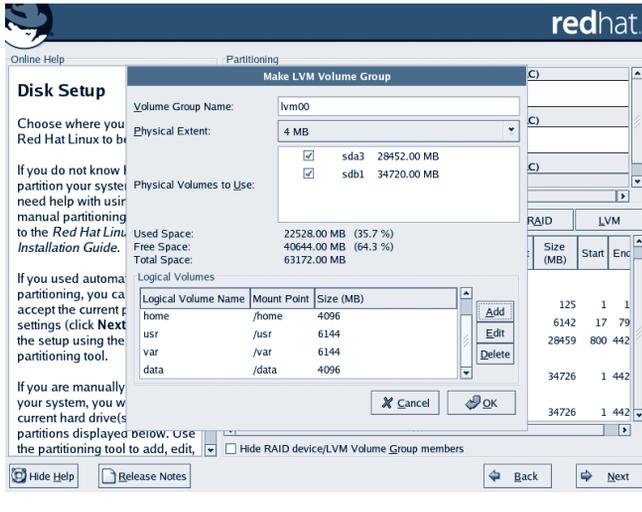


Figure 4. Make LVM Volume Group window in Red Hat Linux installer

and 512 MB. However, the installer limits this range to between 1 MB and 64 MB.

Managing and maintaining system storage

Once LVM is implemented during installation, managing and maintaining system storage is the next task. If any logical volumes in the system become full, more space can be added by extending the logical volume using several methods, including:

- Add more space from the free unallocated space in the same volume group (see Scenario A in this article)
- Reduce another logical volume (see Scenario B)
- Add more hard drives or Dell PowerVault disk enclosures to the system (see Scenario C)

In the following scenarios that detail these approaches, /home is the logical volume to be expanded. Although LVM does not require these safety measures, administrators should perform a full system backup and bring the system to runlevel 1 (init 1) before starting the expansion; in some approaches, the logical volume being modified should be taken offline. Dell also recommends performing these operations during off-peak hours. For more information about the LVM commands used in these scenarios, see “LVM commands for managing and maintaining system storage” in this article.

Scenario A: Expanding from unallocated space in the same volume group

The “Make LVM Volume Group” window in Figure 4 shows that the volume group lvm00 still has 40 GB of active, unallocated space that can be used to expand any of the logical volumes. The

following steps describe how to expand the logical volume /home by 10 GB from this unallocated space:

1. Confirm that the desired logical volume is not being used and take it offline:

```
umount /logical_volume_mounting_point
```

For this scenario, enter:

```
umount /home
```

2. To scan for physical and then logical volumes, display all existing physical volumes with information about the physical device (such as /dev/sda3) to volume group mappings, the volume group to which the device belongs, and the present size status of the device:

```
pvscan
```

```
lvscan
```

3. To determine the unallocated free space that can be used to expand the logical volumes, display all volume groups and their size allocation status:

```
vgdisplay
```

4. Expand the logical volume from an active but unallocated physical volume:

```
lvextend -L +X /dev/lvm_name/logical_volume_name  
/dev/physical_volume
```

where X is the expansion size. The results from steps 2 and 3 provide the available size for this expansion. The -L parameter extends the logical volume size. In this scenario, the unallocated physical volume is /dev/sda3. Expand the logical volume /home by 10 GB by entering:

```
lvextend -L +10G /dev/lvm00/home /dev/sda3
```

5. Verify that the logical volume was extended by comparing the current size to that found in step 2:

```
pvscan
```

```
lvscan
```

These commands should report the updated space information.

6. Resize the file system:

```
e2fsck -f /dev/logical_volume_name  
resize2fs /dev/lvm_name/logical_volume_name
```

For this scenario, enter:

```
e2fsck -f /dev/lvm00/home  
resize2fs /dev/lvm00/home
```

7. Bring the logical volume online:

```
mount /logical_volume_mounting_point
```

For this scenario, enter:

```
mount /home
```

8. Verify that the added space is accounted for:

```
df -h
```

Administrators can also use the `e2fsadm` command to expand the logical volume and resize its file system; this command combines the `lvextend`, `e2fsck`, and `resize2fs` commands.

Scenario B: Reducing another logical volume

Administrators should expand logical volumes by reducing another only when the other volume has sufficient additional space. For instance, if a data volume uses only a fraction of its allocated space and this usage is expected to continue, administrators can safely reduce that volume by some amount and assign that space to another logical volume, `/home` in this scenario. Administrators should not

reduce the logical volume before reducing the file system, because doing so can corrupt the file system on the reduced logical volume. Again, Dell recommends backing up the system before following the approach in this scenario. Performing the following procedure expands one logical volume by reducing another—in this case, by adding 2 GB from `/data` to `/home`:

1. Identify a logical volume to shrink that has sufficient free space. The command `df -h` reports information about the system storage usage.

2. Bring the system down to runlevel 1:

```
init 1
```

3. Take both logical volumes offline:

```
umount /logical_to_shrink
umount /logical_to_grow
```

For this scenario, enter:

```
umount /data
umount /home
```

4. Reduce the selected logical volume and its associated file system:

```
e2fsadm -L -X /dev/lvm_name/logical_volume_name
```

where `X` is the size by which the volume should be reduced. Take care not to reduce the size beyond its free space; doing so can cause data loss—or corrupt the file system.

For this scenario, enter the following command to reduce both the file system and the logical volume `/data` by 2 GB:

```
e2fsadm -L -2G /dev/lvm00/data
```

5. Expand the logical volume by the same amount the other volume was reduced:

```
e2fsadm -L +X /dev/lvm_name/logical_volume_name
```

where `X` is the expansion size. For this scenario, expand `/home` by 2 GB by entering:

```
e2fsadm -L +2G /dev/lvm00/home
```

6. Mount both file systems and bring the system up to runlevel 3 or 5:

```
mount -a
init desired_runlevel
```

7. Verify the changes:

```
df -h
```

Administrators can also use the `resize2fs`, `lvreduce`, and `lvextend` commands in place of `e2fsadm` to achieve these results.

LVM COMMANDS FOR MANAGING AND MAINTAINING SYSTEM STORAGE

Figure A shows various LVM commands used in resizing and managing logical volumes. Detailed information about each command is available from the command's man page.

Command	Description
<code>pvcreate</code>	Creates a physical volume by initializing a disk or a partition
<code>pvscan</code>	Scans all drives for physical volumes and reports drive usage, including available space
<code>lvscan</code>	Scans all disks for logical volumes
<code>lvextend</code>	Expands the size of a logical volume by adding a given amount of space from a given physical volume
<code>lvreduce</code>	Reduces the size of a logical volume (if a file system exists on the volume, the file system must be reduced using <code>resize2fs</code> before running this command; <code>lvreduce</code> does not permit reducing a logical volume size below its used space)
<code>vgextend</code>	Extends a volume group by adding physical volumes
<code>lvdisplay</code>	Displays information about the attributes of a logical volume
<code>vgdisplay</code>	Displays information about the attributes of volume groups
<code>e2fsadm</code>	Expands or reduces an ext2 file system and logical volume together
<code>resize2fs</code>	Expands or reduces an ext2 or ext3 file system
<code>e2fsck</code>	Checks a Linux ext2

Figure A. LVM commands

Scenario C: Adding more storage

If a Dell PowerEdge server lacks unallocated or unused storage space, administrators must add more storage to expand volumes. This approach requires adding more hard drives or a Dell disk enclosure such as a PowerVault 220S. For background on expanding storage for PowerEdge servers, see “Expanding Storage on Linux-based Servers” by Matt Domsch and Tesfamariam Michael in *Dell Power Solutions*, February 2003 (http://www.dell.com/us/en/esg/topics/power_ps1q03-michael.htm). The article is particularly helpful because the expansion procedure can be very complex, especially for RAID storage, the recommended type. To expand a volume by adding more storage, administrators can follow these steps:

1. Partition the added drive or RAID volume and set its partition ID to 8e:

```
fdisk /dev/new_added_drive
```

For this scenario, to add the drive /dev/sdc, enter:

```
fdisk /dev/sdc
```

2. Create a physical volume on the new partition:

```
pvcreeate /dev/partition_of_new_added_drive
```

For this scenario, enter:

```
pvcreeate /dev/sdc1
```

3. Take the logical volume to be expanded offline:

```
umount /logical_volume_mounting_point
```

For this scenario, enter:

```
umount /home
```

4. Expand the volume group:

```
vgextend volume_group_name physical_volume_name
```

For this scenario, enter:

```
vgextend lvm00 /dev/sdc1
```

5. Expand the size of the logical volume and its file system:

```
e2fsadm -L +X logical_volume_name
```

where *X* is the expansion size.

LVM eases storage management in Linux installations by empowering Linux system administrators to expand and reduce storage at an abstracted level.

For this scenario, to expand /home by 10 GB, enter:

```
e2fsadm -L +10G /dev/lvm00/home
```

Alternatively, use the `lvextend` and `resize2fs` commands instead of `e2fsadm`. In this case, enter:

```
lvextend -L +10G lvm00
resize2fs /dev/lvm00/home 10G
```

6. Mount the expanded logical volume:

```
mount /logical_volume_mounting_point
```

For this scenario, enter:

```
mount -a
```

7. Bring the system back to the desired runlevel (3 or 5):

```
init desired_runlevel
```

8. Verify the changes:

```
df -h
```

Achieving more manageable storage environments

LVM eases storage management in Linux installations by empowering Linux system administrators to expand and reduce storage at an abstracted level. Using the techniques described in this article, administrators can upgrade, reallocate, and modify storage resources as needed, helping to create a nimble, scalable storage environment with minimal complexity. 

Tesfamariam Michael (tesfamariam_michael@dell.com) is a software engineer on the Linux Development Team of the Dell Product Group, which tests Linux on all Dell PowerEdge servers. Tesfamariam has an M.S. in Computer Science from Clark Atlanta University, a B.S. in Electrical Engineering from the Georgia Institute of Technology, and a B.S. in Mathematics from Clark Atlanta University. His areas of interest include operating systems and I/O devices.

Joshua Giles (joshua_giles@dell.com) is a Linux systems management software engineer on the Linux Development Team of the Dell Product Group. His interests include operating systems, grammar- and automata-based programming, and Support Vector Machine (SVM) learning. Joshua has a B.S. from the New Mexico Institute of Mining and Technology.

FOR MORE INFORMATION

Dell and Linux: <http://www.dell.com/linux>

LVM HowTo page: <http://tldp.org/HOWTO/LVM-HOWTO>

Red Hat Linux: <http://www.redhat.com>

Sistina LVM: http://www.sistina.com/products_lvm.htm