**Creating the Physical Volumes**

Before the LVM tools can be used to create physical volumes, you need to create a

partition marked as the LVM partition type. If you are using an MBR disk,

the partition type is 8e. If you are using a GUID disk, use the partition type 8300.

In this exercise, you create a physical volume. To do this exercise, you need a hard

disk that has free (unpartitioned) disk space available. The recommended method to

make disk space available is by adding a new hard disk in your virtual machine environment.

In this exercise, I use a clean /dev/vdb device to create the partition. You

may have to change the device name to match your configuration. If you do not have

a dedicated hard disk available to create this configuration, you might want to consider

attaching a USB key to your machine.

**1.** Open a root shell and type **fdisk /dev/vdb** .

**2.** Type **n** to create a new partition. Select **p** to make it a primary partition, and

use the partition number that is suggested as a default. If you are using a clean

device, this will be partition number 1.

**3.** Press **Enter** when asked for the first sector and type **+100M** to accept the last

sector.

**4.** Once you are back on the fdisk prompt, type **t** to change the partition type.

Because there is one partition only, fdisk does not ask which partition to use this

partition type on. You may have to select a partition if you are using a different

configuration.

**5.** The partitioner asks for the partition type you want to use. Type **8e** . Then,

press **w** to write changes to disk and quit fdisk. Listing 15.2 shows an overview

of all commands that have been used so far. If you are getting a message that the

partition table could not be updated while writing the changes to disk, reboot

**6.** Now that the partition has been created, you need to flag it as an LVM physical

volume. To do this, type **pvcreate /dev/vdb1** . You should now get this prompt:

Physical volume “/dev/vbd1” successfully created.

**7.** Now type **pvs** to verify that the physical volume has been created successfully.

The output may look like Listing 15.3 . Notice that in this listing another physical

volume already exists; that is because RHEL uses LVM by default to organize

storage.

**Creating the Volume Groups**

Now that the physical volume has been created, you can assign it to a volume group.

This is a simple one-command procedure.

Just type **vgcreate** followed by the name of the volume group you want to create

and the name of the physical device you want to add to it. So, if the physical volume

name is /dev/vdb1, the complete command is **vgcreate vgdata /dev/vdb1** .

In this procedure, you learned how to create a volume group in a two-step procedure

where first the physical volume is created with the **pvcreate** command, after

which the volume group is added using the **vgcreate** command. You can do this in

a one-step procedure as well (where using a separate **pvcreate** command will not be

necessary). If you are adding a partition to the volume group, however, it must be

marked as partition type 8e already.

If you want to add the disk **/** dev/

sdc, for instance, just type **vgcreate vgdata /dev/sdc** to create a volume group

vgdata that contains the /dev/sdc device. When you are doing this to add a device

that has not been marked as a physical volume yet, the vgcreate utility will automatically

flag it as a physical volume.

using the **vgs** command for a short summary, or the **vgdisplay** command to get

more information

**Creating the Logical Volumes and File Systems**

Now that the volume group has been created, you can start creating logical volumes

from it.

For instance, use **lvcreate -n lvvol1 -L 100M vgdata** to create a logical volume with the name lvvol1 and add that to the vgdata volume group.

**Resizing Volume Groups**

The main part of LVM flexibility sits in the fact that it is so easy to resize the volume

groups and the logical volumes that are using disk space from the volume

group. The **vgextend** command is used to add storage to a volume group, and the

**vgreduce** command is used to take physical volumes out of a volume group (which

can lead to some additional complications). For the RHCSA test, you need to know

how to extend the available storage in volume groups. This procedure is relatively

easy:

**1.** Make sure that a physical volume or device is available to be added to the

volume group.

**2.** Use **vgextend** to extend the volume group. The new disk space will show

immediately in the volume group.

Most file system resizing

operations can even be done online, without any need to unmount the file system.

To grow the logical volume size, use **lvresize** , followed by the **-r** option to resize

the file system used on it. Then, specify the size you want the resized volume to be.

The easiest and most intuitive way to do that is by using **-L** followed by a **+** sign and

the amount of disk space you want to add, as in **lvresize -L +1G -r /dev/vgdata/**

**lvdata** . An alternative way to resize the logical volume is by using the **-l** option. This

option is followed by the number of extents that are added to the logical volume

or by the absolute or relative percentage of extents in the volume group that will

be used. You can, for example, use the following commands to resize the logical

volume:

■ **lvresize -r -l 75%VG /dev/vgdata/lvdata** This resizes the logical volume

so that it will take 75% of the total disk space in the volume group.

■ **lvresize -r -l +75%VG /dev/vgdata/lvdata** This tries to add 75% of the

total size of the volume group to the logical volume. (Notice the difference

with the previous command.)

■ **lvresize -r -l +75%FREE /dev/vgdata/lvdata** This adds 75% of all free

disk space to the logical volume.

■ **lvresize -r -l 75%FREE /dev/vgdata/lvdata** This resizes the logical volume

to a total size that equals 75% of the amount of free disk space. (Notice

the difference with the previous command.)

**NOTE** The size of an XFS file system cannot be decreased; it can only be

increased. If you need a file system that can be shrunk in size, use Ext4, not XFS.

LVM(LOGICAL VOLUME MANAGER)  
        -----------------------------------------  
  
1>first create partion  
  
#partprobe -s  
  
#pvcreate /dev/sda?  
  
#pvcreate /dev/sda?  
  
#vgcreate vg0 /dev/sda? /dev/sda?  
  
#vgdisplay  
  
#pvdisplay  
  
#lvdisplay  
  
#lvcreate -n lvm0 -L 150M vg0   (where -n---->name, -L--->length)  
  
#lvdisplay  
  
#mkfs.ext3 /dev/vg0/lvm0  
  
#mkdir /mnt/lvm1  
  
#mount /dev/vg0/lvm0 /mnt/lvm0  
  
#vim /etc/fstab  
/dev/vg0/lvm0 /mnt/lvm0 ext3 defaults 0 0  
  
#mount -a  
  
--------------------------------------------------------------------------------  
  
Steps To Extend The LVM  
-----------------------------  
  
#lvextend -L 300M /dev/vg0/lvm0  
         (OR)  
#lvextend -L +150M /dev/vg0/lvm0  
  
#lvdisplay  
  
#df -h  
  
#resize2fs /dev/vg0/lvm0 (To give the information abt changing the size to the system)  
  
#df -h  
--------------------------------------------------------------------------------  
  
Steps To Reduce The Size Of LVM  
-----------------------------------  
  
#umount /mnt/lvm0  
  
#e2fsck -f /dev/vg0/lvm0   (forcefully check the file system type)  
  
#resize2fs /dev/vg0/lvm0 200M  
  
#lvreduce -L 200M /dev/vg0/lvm0  
  
#lvdisplay /dev/vg0/lvm0  
  
#mount /dev/vg0/lvm0 /mnt/lvm0  
  
#df -h  
--------------------------------------------------------------------------------  
  
To Extend Existing VG  
---------------------------  
1> create a partion  
  
#partprobe -s  
  
#pvcreate /dev/sda?  
  
#vgextend vg0 /dev/sda?  
  
#vgdispaly