Creating Linux LVM Partitions

Tasks

1. Create two 1G partitions on `/dev/xvdb`.

a. Use the `fdisk`command to create two 1G primary partitions on `/dev/xvdb`as shown in the following:

```
# fdisk /dev/xvdb
Welcome to dfisk (util-linux 2.23.2).
...
Command (m for help): n
Partition type:
p primary partition (0 primary, 0 extended, 4 free)
e extended
Select (default p): ENTER
Using default response p
Partition number (1-4, default 1): ENTER
First sector (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): +1G
Partition 1 of type Linux and of size 1 GiB is set
Command (m for help): n
Partition type:
p primary partition (1 primary, 0 extended, 3 free)
e extended
Select (default p): ENTER
Using default response p
Partition number (2-4, default 2): ENTER
First sector (2099200-10485759, default 2099200): ENTER
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-10485759, default
10485759): +1G
Partition 2 of type Linux and of size 1 GiB is set
```

b. Use the "**t**" command to change the system ID on partition 2.

```
Command (m for help): t
Partition number (1,2, default 2): ENTER
Hex code (type L to list codes): L
0 Empty 24 NEC DOS 81 Minix / old Lin ...
1 FAT12 27 Hidden NTFS Win 82 Linux swap / So ...
2 XENIX root 39 Plan 9 83 Linux ...
...
8 AIX 4e QNX4.x 2nd part 8e Linux LVM ...
...
Hex code (type L to list codes): 8e
Changed system type of partition "Linux' 'Linux LVM'
```

c. Print the new partition table.

```
Command (m for help): p
Disk /dev/xvdb: 5368 MB, 5368709120 bytes, 10485760 sectors
...
Device Boot Start End Blocks Id System
/dev/xvdb1 2048 2099199 1048576 83 Linux
/dev/xvdb2 2099200 4196351 1048576 8e Linux LVM
```

d. Save the new partition table.

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

2. Create two 1G partitions on `/dev/xvdd`.

a. Use the `fdisk`command to create two 1G primary partitions on `/dev/xvdd`as shown in the following:

```
# fdisk /dev/xvdd
Welcome to dfisk (util-linux 2.23.2).
...
Command (m for help): n
Partition type:
p primary partition (0 primary, 0 extended, 4 free)
e extended
Select (default p): ENTER
Using default response p
Partition number (1-4, default 1): ENTER
First sector (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): +1G
Partition 1 of type Linux and of size 1 GiB is set
```

```
Command (m for help): n
Partition type:
p primary partition (1 primary, 0 extended, 3 free)
e extended
Select (default p): ENTER
Using default response p
Partition number (2-4, default 2): ENTER
First sector (2099200-10485759, default 2099200): ENTER
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-10485759, default
10485759): +1G
Partition 2 of type Linux and of size 1 GiB is set
```

b. Use the "**t**" command to change the system ID on partition 2.

```
Command (m for help): t
Partition number (1,2, default 2): ENTER
Hex code (type L to list codes): 8e
Changed system type of partition "Linux' 'Linux LVM'
```

c. Print the new partition table.

```
Command (m for help): p
Disk /dev/xvdd: 5368 MB, 5368709120 bytes, 10485760 sectors
...
Device Boot Start End Blocks Id System
/dev/xvdd1 2048 2099199 1048576 83 Linux
/dev/xvdd2 2099200 4196351 1048576 8e Linux LVM
```

d. Save the new partition table.

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

Creating a Logical Volume

Overview

In this practice, you create physical volumes, a volume group, and a logical volume. You also use LVM utilities to display information about the LVM entities.

Tasks

1. Initialize the new partitions for use by LVM (create physical volumes).

a. List the partitions with the Linux LVM (**8e**) system ID.

```
# fdisk -l | grep 8e
/dev/xvdb2 2099200 4196351 1048576 8e Linux LVM
/dev/xvdd2 2099200 4196351 1048576 8e Linux LVM
```

b. Use the `pvcreate`command to create physical volumes on both partitions.

```
# pvcreate -v /dev/xvdb2 /dev/xvdd2
Set up physical volume for "/dev/xvdb2" with 2097152 ...
Zeroing start of device /dev/xvdb2
Writing physical volume data to disk "/dev/xvdb2"
Physical volume "/dev/xvdb2" successfully created
Set up physical volume for "/dev/xvdd2" with 2097152 ...
Zeroing start of device /dev/xvdd2
Writing physical volume data to disk "/dev/xvdd2"
Physical volume "/dev/xvdd2" successfully created
```

2. Display information about the physical volumes.

a. Use the `pvdisplay`command to display attributes of the physical volumes.

```
# pvdisplay
"/dev/xvdd2" is a new physical volume of "1.00 GiB"
--- NEW Physical volume ---
PV Name /dev/xvdd2
VG Name
PV Size 1.00 GiB
...
"/dev/xvdb2" is a new physical volume of "1.00 GiB"
--- NEW Physical volume ---
PV Name /dev/xvdb2
VG Name
PV Size 1.00 GiB
...
```

b. Use the `pvs`command to report information in a more condensed form.

```
# pvs
PV VG FmtAttrPSizePFree
/dev/xvdb2 lvm2 a-- 1.00g 1.00g
```

```
/dev/xvdd2 lvm2 a-- 1.00g 1.00g
```
c. Use the `pvscan`command to scan all disks for physical volumes.
```
# pvscan
PV /dev/xvdb2 lvm2 [1.00 GiB]
PV /dev/xvdd2 lvm2 [1.00 GiB]
Total: 2 [2.00 GiB] / in use: 0 [0 ] / in no VG: 2 [2.00 GiB]
```
3. Create a volume group.
Use the `vgcreate`command to create a volume group named **myvolg**from the `/dev/xvdb2` physical volume.
```
# vgcreate -v myvolg /dev/xvdb2
Adding physical volume 'dev/xvdb2' to volume group 'myvolg'
Archiving volume group "myvolg" metadata (seqno 0).
Creating volume group backup "/etc/lvm/backup/myvolg"...
Volume group "myvolg" successfully created
```

4. Display information about the volume group.
a. Use the `vgdisplay`command to display attributes of the volume group.
```
# vgdisplay
--- Volume group ---
VG Name myvolg
System ID
Format lvm2
...
VG Size 1020.00 MiB
...
```
b. Use the `vgs`command to report information in a more condensed form.
```
# vgs
VG #PV #LV #SN AttrVSizeVFree
myvolg 1 0 0 wz--n- 1020.00m 1020.00m
```

c. Use the `vgscan`command to scan all disks for volume groups and rebuild caches.
```
# vgscan
Reading all physical volumes. This may take a while...
Found volume group "myvolg" using metadata type lvm2
```
d. Display information about the physical volumes.
```
# pvs
PV VG FmtAttrPSizePFree
/dev/xvdb2 myvolg lvm2 a-- 1020.00m 1020.00m
/dev/xvdd2 lvm2 a-- 1.00g 1.00g
```
 Note that the `/dev/xvdb2` physical volume is assigned to the **myvolg**volume group.

5. Create a logical volume.
Use the `lvcreate`command to create a 500 MB logical volume named **myvol**from the **myvolg**volume group.

```
# lvcreate -v –L 500m –n myvol myvolg
Setting logging type to disk
Finding volume group “myvolg”
Archiving volume group “myvolg” metadata (seqno 1).
Creating logical volume myvol
...
Logical volume “myvol” created
```

6. Display information about the logical volume.
a. Use the `lvdisplay`command to display attributes of the logical volume.

```
# lvdisplay
--- Logical volume ---
LV Path /dev/myvolg/myvol
LV Name myvol
VG Name myvolg
...
LV Size 500.00 MiB
...
```

b. Use the `lvs`command to report information in a more condensed form.

```
# lvs
LV VG AttrLSize Pool Origin Data% Move Log...
myvolmyvolg -wi-a----- 500.00m
```

c. Use the `lvscan`command to scan all disks for logical volumes.

```
# lvscan
ACTIVE ‘/dev/myvolg/myvol’ [500.00 MiB] inherit
```

d. Display information about the physical volumes.

```
# pvs
PV VG FmtAttrPSizePFree
/dev/xvdb2 myvolg lvm2 a-- 1020.00m 520.00m
/dev/xvdd2 lvm2 a-- 1.00g 1.00g
```

 Note that the free space in the `/dev/xvdb2` physical volume has been reduced.

e. Display information about the volume group.

```
# vgs
VG #PV #LV #SN AttrVSizeVFree
myvolg 1 1 0 wz--n- 1020.00m 520.00m
```

 Note that the free space in the **myvolg**volume group has also been reduced.

Creating a File System and Mounting a Logical Volume

Tasks

1. Display the block device name that was automatically created.

a. List the `/dev` entries for the **myvol** logical volume.

```
# ls -l /dev/myvolg/myvol
lrwxrwxrwx. /dev/myvolg/myvol -> ../dm-0
# ls -l /dev/mapper/myvolg-myvol
lrwxrwxrwx. /dev/mapper/myvolg-myvol -> ../dm-0
```

- Note that two entries were automatically created.
- Note that both entries are symbolic links to `/dev/dm-0`.

b. List the `/dev/dm-0` entry.

```
# ls -l /dev/dm-0
brw-rw----. /dev/dm-0
```

- Note that `/dev/dm-0` is a block device.

2. Create a file system on the logical volume.

a. Create an **ext4** file system on the **myvol** logical volume.

```
# mkfs.ext4 /dev/mapper/myvolg-myvol
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
...
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

b. Create a `/myvol` mount point.

```
# mkdir /myvol
```

c. Mount the file system.

```
# mount /dev/mapper/myvolg-myvol /myvol
```

d. Display the mounted file systems.

```
# df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-myvol
477M 2.3M 445M 1% /myvol
```

3. Update the file systems mount table.

Use the `vi` editor to add the following line to `/etc/fstab`.

```
/dev/mapper/myvolg-myvol /myvol ext4 defaults 0 0
```

Creating a Thinly Provisioned Logical Volume

Overview

In this practice, you create a thin pool and two thinly provisioned logical volumes. You then monitor the allocated pool data and extend the size of the thin pool.

Tasks

1. Remove the logical volume.

a. Unmount the **myvol**logical volume.

```
# umount /myvol
```

b. Use the `lvremove`command to remove the **myvol**logical volume.

- Answer **`y`** when asked, "Do you really want to ..."

```
# lvremovemyvolg/myvol
Do you really want to remove active logical volume myvol? [y/n]:
y
Logical volume "myvol" successfully removed
```

c. Remove the `/myvol`mount point.

```
# rmdir /myvol
```

d. By using the `vi`editor, remove the following line from `/etc/fstab`.

```
# vi /etc/fstab
/dev/mapper/myvolg-myvol /myvol ext4 defaults 0 0
```

2. Remove a physical volume from a volume group.

a. Use the `vgs`command to display information about the volume groups.

```
# vgs
VG #PV #LV #SN AttrVSizeVFree
myvolg 2 0 0 wz--n- 1.99g 1.99g
```

- Note that the **myvolg**volume group has two physical volumes (PV).

b. Use the `pvs`command to display information about the physical volumes.

```
# pvs
PV VG FmtAttrPSizePFree
/dev/xvdb2 myvolg lvm2 a-- 1020.00m 1020.00m
/dev/xvdd2 myvolg lvm2 a-- 1020.00m 1020.00m
```

- Notice that both physical volumes are allocated to the **myvolg**volume group.

c. Use the `vgreduce`command to remove a physical volume from a volume group.

- Remove `/dev/xvdd2` from **myvolg**.

```
# vgreducemyvolg /dev/xvdd2
Removed "/dev/xvdd2" from volume group "myvolg"
```

d. Use the `vgs`command to display information about the volume groups.

```
# vgs
VG #PV #LV #SN AttrVSizeVFree
```

```
myvolg 1 0 0 wz--n- 1020.00m 1020.00m
```
 Note that the **myvolg**volume group now has one physical volume.
e. Use the `pvs`command to display information about the physical volumes.
```
# pvs
PV VG FmtAttrPSizePFree
/dev/xvdb2 myvolg lvm2 a-- 1020.00m 1020.00m
/dev/xvdd2 lvm2 a-- 1.00g 1.00g
```
 Note that only the `/dev/xvdb2` physical volume is allocated to the **myvolg**volume group.

3. Create a thin pool.

a. Use the `lvcreate`command to create a 100 MB thin pool named **mythinpool**in the **myvolg**volume group.
 Note that you must specify the size of the pool because you are creating a pool of physical space.
```
# lvcreate -v -L 100m -T myvolg/mythinpool
...
Logical volume "mythinpool" created
```

b. Use the `lvs`command to display information about the logical volumes.
```
# lvs
LV VG AttrLSize Pool Origin Data% Move...
mythinpoolmyvolgtwi-a-tz-- 100.00m 0.00
```

c. List the contents of the `/dev/myvolg`directory.
```
# ls -l /dev/myvolg
ls: cannot access /dev/myvolg: No such file or directory
```
 Note that there is no entry in the `/dev`directory for the **myvolg**volume group.

4. Create two thinly provisioned volumes.
a. Use the `lvcreate`command to create a 200 MB thin volume named **thinvol1** in the **myvolg/mythinpool**thin pool.
 Note that you are specifying a virtual size for the thin volume that is greater than the pool that contains it.
```
# lvcreate -v -V 200m -T myvolg/mythinpool -n thinvol1
...
Logical volume "thinvol1" created
```
b. Use the `lvcreate`command to create another 200 MB thin volume named **thinvol2** in the **myvolg/mythinpool**thin pool.
 Do not include the `-v` (verbose) option.
```
# lvcreate -V 200m -T myvolg/mythinpool -n thinvol2
Logical volume "thinvol2" created
```

c. Use the `lvs`command to display information about the logical volumes.

```
# lvs
LV VG AttrLSize Pool Origin Data...

mythinpoolmyvolgtwi-a-tz-- 100.00m 0.00
thinvol1myvolgVwi-a-tz-- 200.00m mythinpool 0.00
thinvol2myvolgVwi-a-tz-- 200.00m mythinpool 0.00
```

- Note that the “Data%” column values are 0.00.

d. List the contents of the `/dev/myvolg`and `/dev/mapper` directories.

```
# ls -l /dev/myvolg
lrwxrwxrwx. thinvol1 -> ../dm-4
lrwxrwxrwx. thinvol2 -> ../dm-5
# ls -l /dev/mapper
...
lrwxrwxrwx. myvolg-thinvol1 -> ../dm-4
lrwxrwxrwx. myvolg-thinvol2 -> ../dm-5
```

- In this example, the files that represent the thin volumes are symbolic links to `dm-4` and`dm-5`.

5. Create a file system on the thin volume.

a. Create an **ext4** file system on the **thinvol1** thin volume.

```
# mkfs.ext4 /dev/myvolg/thinvol1
...
Writing superblocks and filesystem accounting information: done
```

b. Create an **ext4** file system on the **thinvol2** thin volume.

```
# mkfs.ext4 /dev/myvolg/thinvol2
...
Writing superblocks and filesystem accounting information: done
```

c. Create the `/myvol1` and `/myvol2` directories.

```
# mkdir /myvol1 /myvol2
```

d. Mount the file systems.

- Mount `/dev/myvolg/thinvol1` on `/myvol1`.
- Mount `/dev/myvolg/thinvol2` on `/myvol2`.

```
# mount /dev/myvolg/thinvol1 /myvol1
# mount /dev/myvolg/thinvol2 /myvol2
```

e. Display the mounted file systems.

```
# df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-thinvol1
190M 1.6M 175M 1% /myvol1
/dev/mapper/myvolg-thinvol2
190M 1.6M 175M 1% /myvol2
```

- Note that the `df`command shows the size of the file system as 190M. This is an over-allocation of the actual available storage in the thin pool.

f. Use the `lvs`command to display information about the logical volumes.

```
# lvs
LV VG AttrLSize Pool Origin Data...
mythinpoolmyvolgtwi-a-tz-- 100.00m 22.12
thinvol1myvolgVwi-a-tz-- 200.00m mythinpool 5.53
thinvol2myvolgVwi-a-tz-- 200.00m mythinpool 5.53
```

- This shows that you have used 22.12% of the allocated pool data (100 MB).
- This also shows that each thin volume has used 5.53% of 200 MB.

6. Copy files to a thin volume and monitor usage.

a. Copy `/boot/vmlinuz*` to `/myvol1`.

```
# cp /boot/vmlinuz* /myvol1
```

b. Run the `sync` command and then run the `lvs`command to display information about the logical volumes.

```
# sync
# lvs
LV VG AttrLSize Pool Origin Data...
mythinpoolmyvolgtwi-a-tz-- 100.00m 40.38
thinvol1myvolgVwi-a-tz-- 200.00m mythinpool 14.66
thinvol2myvolgVwi-a-tz-- 200.00m mythinpool 5.53
```

- This shows that you have used 40.38% of the allocated pool data (100 MB).
- This also shows that the thin volume mounted on `/myvol1`, **thinvol1**, has used 14.66% of 200 MB.

c. Use the `lvextend`command to increase the size of the **myvolg/mthinpool**thin pool to 500 MB.

```
# lvextend -L 500m myvolg/mythinpool
Extending logical volume mythinpool_tdata to 500.00 MiB
Logical volume mythinpool successfully resized
```

d. Use the `lvs`command to display information about the logical volumes.

```
# lvs
LV VG AttrLSize Pool Origin Data...
mythinpoolmyvolgtwi-a-tz-- 500.00m 8.07
thinvol1myvolgVwi-a-tz-- 200.00m mythinpool 14.66
thinvol2myvolgVwi-a-tz-- 200.00m mythinpool 5.53
```

- Note that the size of the thin pool is 500 MB and the percentage used is 8.07%.