Chapter 5 Guided Exercises and Labs

Add a coversheet with your name, module title, and pledge to the beginning of this document
Insert screenshots showing intermediate steps and completion of each of the guided exercises and labs at appropriate locations below and submit a pdf file

Guided Exercise: Add Partitions, File Systems, and Persistent Mounts

In this exercise, you create a partition on a new storage device, format it with an XFS file system, configure it to mount at boot, and mount it for use.

Outcomes

- Use the `parted`, `mkfs.xfs`, and other commands to create a partition on a new disk, format it, and persistently mount it.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

```
[student@workstation ~]$ lab start storage-partitions
```

Procedure 5.1. Instructions

1. Log in to `servera` as the `student` user and switch to the `root` user.

```
student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

2. Create an `msdos` disk label on the `/dev/vdb` device.

```
[root@servera ~]# parted /dev/vdb mklabel msdos
Information: You may need to update /etc/fstab.
```

3. Add a 1 GB primary partition. For proper alignment, start the partition at the sector 2048. Set the partition file-system type to XFS.

1. Use `parted` interactive mode to create the partition.

```
[root@servera ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Partition type?  primary/extended? primary
File system type?  [ext2]? xfs
Start? 2048s
End? 1001MB
(parted) quit
Information: You may need to update /etc/fstab.
```

   Because the partition starts at the sector 2048, the previous command sets the end position to 1001 MB to get a partition size of 1000 MB (1 GB).

   Alternatively, you can perform the same operation with the following non-interactive command: `parted /dev/vdb mkpart primary xfs 2048s 1001 MB`

2. Verify your work by listing the partitions on the `/dev/vdb` device.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  1001MB  1000MB  primary
```

3. Run the `udevadm settle` command. This command waits for the system to register the new partition and returns when it is done.

```
[root@servera ~]# udevadm settle
```

4. Format the new partition with the XFS file system.

```
[root@servera ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=61056 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
```

```
data     =                       bsize=4096   blocks=244224, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=1566, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

5. Configure the new file system to mount onto the `/archive` directory persistently.
    1. Create the `/archive` directory.

        ```
        [root@servera ~]# mkdir /archive
        ```

    2. Discover the UUID of the `/dev/vdb1` device. The UUID in the output is probably different on your system.

        ```
        [root@servera ~]# lsblk --fs /dev/vdb
        NAME   FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
        vdb
        └─vdb1 xfs                 881e856c-37b1-41e3-b009-ad526e46d987
        ```

    3. Add an entry to the `/etc/fstab` file. Replace the UUID with the one that you discovered from the previous step.

        ```
        ...output omitted...
        UUID=881e856c-37b1-41e3-b009-ad526e46d987 /archive xfs defaults  0 0
        ```

    4. Update the `systemd` daemon for the system to register the new `/etc/fstab` file configuration.

        ```
        [root@servera ~]# systemctl daemon-reload
        ```

    5. Mount the new file system with the new entry in the `/etc/fstab` file.

        ```
        [root@servera ~]# mount /archive
        ```

    6. Verify that the new file system is mounted onto the `/archive` directory.

        ```
        [root@servera ~]# mount | grep /archive
        /dev/vdb1 on /archive type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
        ```

6. Reboot `servera`. After the server rebooted, log in and verify that the `/dev/vdb1` device is mounted on the `/archive` directory. When done, log out from `servera`.
    1. Reboot `servera`.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

2. Wait for `servera` to reboot and log in as the `student` user.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

3. Verify that the `/dev/vdb1` device is mounted on the `/archive` directory.

```
[student@servera ~]$ mount | grep /archive
/dev/vdb1 on /archive type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
```

4. Return to the `workstation` machine as the `student` user.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finish

On the `workstation` machine, change to the `student` user home directory and use the `lab` command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish storage-partitions
```

This concludes the section.

Guided Exercise: Manage Swap Space

In this exercise, you create and format a partition for use as swap space, format it as swap, and activate it persistently.

Outcomes

- Create a partition and a swap space on a disk by using the GPT partitioning scheme.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

```
[student@workstation ~]$ lab start storage-swap
```

Procedure 5.2. Instructions

1. Log in to `servera` as the `student` user and switch to the `root` user.

   ```
   [student@workstation ~]$ ssh student@servera
   ...output omitted...
   [student@servera ~]$ sudo -i
   [sudo] password for student: student
   [root@servera ~]#
   ```

2. Inspect the `/dev/vdb` disk. The disk already has a partition table and uses the GPT partitioning scheme. Also, it has an existing 1 GB partition.

   ```
   [root@servera ~]# parted /dev/vdb print
   Model: Virtio Block Device (virtblk)
   Disk /dev/vdb: 5369MB
   Sector size (logical/physical): 512B/512B
   Partition Table: gpt
   Disk Flags:

   Number  Start   End     Size    File system  Name  Flags
    1      1049kB  1001MB  1000MB               data
   ```

3. Add a new partition of 500 MB for use as swap space. Set the partition type to `linux-swap`.
   1. Create the `myswap` partition. Because the disk uses the GPT partitioning scheme, you must give a name to the partition. Notice that the start position, 1001 MB, is the end of the existing first partition. The `parted` command ensures that the new partition immediately

follows the previous one, without any gap. Because the partition starts at the 1001 MB position, the command sets the end position to 1501 MB to get a partition size of 500 MB.

```
[root@servera ~]# parted /dev/vdb mkpart myswap linux-swap \
1001MB 1501MB
Information: You may need to update /etc/fstab.
```

2. Verify your work by listing the partitions on the `/dev/vdb` disk. The size of the new partition is not exactly 500 MB. The difference in size is because the `parted` command must align the partition with the disk layout.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name   Flags
 1      1049kB  1001MB  1000MB               data
 2      1001MB  1501MB  499MB                myswap swap
```

3. Run the `udevadm settle` command. This command waits for the system to register the new partition and returns when it is done.

```
[root@servera ~]# udevadm settle
```

4. Initialize the new partition as swap space.

```
[root@servera ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 476 MiB (499118080 bytes)
no label, UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328
```

5. Enable the new swap space.
    1. Verify that creating and initializing the swap space does not yet enable it for use.

```
[root@servera ~]# swapon --show
```

    2. Enable the new swap space.

```
[root@servera ~]# swapon /dev/vdb2
```

    3. Verify that the new swap space is now available.

```
[root@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 476M   0B   -2
```

4. Disable the swap space.

```
[root@servera ~]# swapoff /dev/vdb2
```

5. Confirm that the swap space is disabled.

```
[root@servera ~]# swapon --show
```

6. Enable the new swap space at system boot.
   1. Use the `lsblk` command with the `--fs` option to discover the UUID of the `/dev/vdb2` device. The UUID in the output will be different on your system.

```
[root@servera ~]# lsblk --fs /dev/vdb2
NAME FSTYPE FSVER LABEL UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
vdb2 swap   1           762735cb-a52a-4345-9ed0-e3a68aa8bb97
```

   2. Add an entry to the `/etc/fstab` file. In the following command, replace the UUID with the one that you discovered from the previous step.

```
...output omitted...
UUID=762735cb-a52a-4345-9ed0-e3a68aa8bb97  swap  swap  defaults  0 0
```

   3. Update the `systemd` daemon for the system to register the new `/etc/fstab` file configuration.

```
[root@servera ~]# systemctl daemon-reload
```

   4. Enable the swap space by using the entry in the `/etc/fstab` file.

```
[root@servera ~]# swapon -a
```

   5. Verify that the new swap space is enabled.

```
[root@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 476M   0B   -2
```

7. Reboot the `servera` machine. After the server reboots, log in and verify that the swap space is enabled. When done, log out from `servera`.

1. Reboot the `servera` machine.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

2. Wait for `servera` to reboot and log in as the `student` user.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

3. Verify that the swap space is enabled.

```
[student@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 476M   0B   -2
```

4. Return to the `workstation` machine as the `student` user.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finish

On the `workstation` machine, change to the `student` user home directory and use the `lab` command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish storage-swap
```

This concludes the section.

Lab: Manage Basic Storage

In this lab, you create several partitions on a new disk, formatting some with file systems and mounting them, and activating others as swap spaces.

Outcomes

- Display and create partitions with the `parted` command.
- Create file systems on partitions and persistently mount them.
- Create swap spaces and activate them at boot.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise.

This command prepares your environment and ensures that all required resources are available.

```
[student@workstation ~]$ lab start storage-review
```

Procedure 5.3. Instructions

1. The `serverb` machine has several unused disks. On the first unused disk, create a 2 GB GPT `backup` partition. Because it is difficult to set an exact size, a size between 1.8 GB and 2.2 GB is acceptable. Configure the `backup` partition to host an XFS file system.
2. Format the 2 GB `backup` partition with an XFS file system and persistently mount it to the `/backup` directory.
3. On the same disk, create two 512 MB GPT partitions called `swap1` and `swap2`. A size between 460 MB and 564 MB is acceptable. Configure the file-system types of the partitions to host swap spaces.
4. Initialize the two 512 MiB partitions as swap spaces and configure them to activate at boot. Set the swap space on the `swap2` partition to be preferred over the other.
5. To verify your work, reboot the `serverb` machine. Confirm that the system automatically mounts the first partition onto the `/backup` directory. Also, confirm that the system activates the two swap spaces.

Evaluation

As the `student` user on the `workstation` machine, use the `lab` command to grade your work. Correct any reported failures and rerun the command until successful.

```
[student@workstation ~]$ lab grade storage-review
```

Finish

On the `workstation` machine, change to the `student` user home directory and use the `lab` command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish storage-review
```

This concludes the section.