

Notes

1. For the 1st objective if we enter `lsblk` command, we can see `/dev/sdb` , `/dev/sdc` , `/dev/sdd` and `/dev/sde` is the available disks. These are set of commands I used to set the partition.

```
Sudo fdisk /dev/sdb
```

```
n
```

```
p (default and press enter)
```

```
partition number (default 1 enter)
```

```
first sector (default 2048 enter)
```

```
last sector (default 119999999 enter)
```

```
t
```

```
HexCode = 8e
```

```
w
```

After this it will show partition table is altered.

2. For the 2nd objective created a raid array for 4 disks using this command `sudo mdadm --create /dev/md0 --level=5 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1`. RAID 5 is chosen here because it offers a good balance between performance, storage efficiency, and fault tolerance, which is ideal for setups requiring reliable storage without sacrificing too much capacity. RAID 5 provides redundancy by distributing parity information across all drives in the array. This means if one drive fails, the data can still be reconstructed from the remaining drives. This is crucial in environments where data integrity and uptime are essential. Unlike RAID 1 (mirroring), which duplicates all data, RAID 5 only dedicates one drive's worth of space to parity (regardless of the number of drives). This provides better storage efficiency than other redundant RAID levels. With four drives in RAID 5, you effectively get the capacity of three drives (since one drive's space is used for parity). RAID 5 offers improved read speeds because data is spread across multiple disks, allowing parallel access. Although write speeds are somewhat slower (due to parity calculations), RAID 5 still delivers reasonable performance for general-purpose applications.

3. Created Physical Volume using command `sudo pvcreate /dev/md0`

Created a volume group using command `vgcreate savg /dev/md0`

4. Created Logical Volume `tmp` using command `sudo lvcreate-n tmp -L 1G savg` and creating an ext4 file on it.

```
Sudo mdks.ext4 /dev/savg/tmp
```

5. To create a logical volume that fills 80% of the remainder of the newly added storage and create a xfs file is by using command:-

```
sudo vgdisplay savg
```

```
sudo lvcreate -n home -L 12.9G savg
```

```
sudo mkfs.xfs /dev/savg/home
```

6. To permanently add the /tmp and /home filesystems with specified mount options I had to edit the /etc/fstab by using command:-

```
sudo nano /etc/fstab
```

```
/dev/savg/tmp /tmp ext4 defaults,nodev,nosuid,noexec 0 2
```

```
/dev/savg/home /home xfs defaults,nodev 0 2
```

Then created a directory `sudo mkdir -p /mnt/tmp`

```
Sudo mkdir -p /mnt/home
```

```
mount /dev/savg/home /mnt/home
```

```
df -h /mnt/home
```

```
sudo sync -ax HAX /home/ /mnt/home
```

```
cd /mnt/home
```

```
ls -al
```

```
fuser -km /mnt/home
```

```
umount /mnt/home
```

```
umount /mnt/home
```

```
mount -a
```

```
systemctl daemon-reload
```

```
cd /home
```

```
ls -al
```

```
df -h /home
```

```
df -h /tmp
```

To enable quotas in the /etc/fstab

```
/dev/mapper/savg-home /home xfs defaults,usrquota,grpquota 0 0
```

```
sudo mount -o remount /home
```

To set the the soft and hart limit to each user use command:-

```
sudo xfs_quota -x -c 'limit bsoft=1000M bhard=1200M mscott' /home
```

Change the username and set the soft and hard limit for each user.

Set the soft and hard limit to managers, sales and accounting group.

```
sudo xfs_quota -x -c 'limit -g bsoft=1000M bhard=1200M managers' /home
```

For setting the grace period

```
sudo setquota -t 86400 86400 /home
```

It took me 4 hours to finish the assignment.