

Assignment – Disk Partitioning, File System Creation, Management and Mounting

Launch virtual machine in the cloud, attach 20 GB EBS volume

Important Note:- Do not try partition, filesystem creation on your local desktop or laptop instead use virtual machine to do practice. These operations are destructive, chances of system crash. Work carefully.

Create partition on newly attached disk as per below instructions -

- a) Create 2 primary partitions of 3 GB each
- b) Create 2 logical partitions of 6 GB each
- c) Format all 4 partitions and create ext4 filesystem on that
- d) Create 4 folders inside root (/) folder name it as Data1, Data2, Data3, Data4
- e) Mount all formatted partitions on the respective folders
- f) Create empty file inside each folders of size 2 GB, 2GB, 4 GB and 4 GB respectively using command - `dd - "convert and copy a file"`
- g) Go inside /Data1 and run command - `while(true); do sleep 5s; done` , do ctrl-z
- h) Check disk utilization of each mount point
- i) Unmount all partitions /Data1, /Data2, /Data3 and /Data4

Note:- All partitions should be automatically mounted post reboot.

After identifying the attached 20GB EBS volume (e.g., /dev/nvme1n1) using `lsblk`, the `fdisk` utility was used to create two primary partitions of 3 GiB each, as shown below.

```
Command (m for help): p
Disk /dev/nvme1n1: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: Amazon Elastic Block Store
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x900b28f7

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943039, default 41943039): +3G

Created a new partition 1 of type 'Linux' and of size 3 GiB.

Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (6293504-41943039, default 6293504):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (6293504-41943039, default 41943039): +3G

Created a new partition 2 of type 'Linux' and of size 3 GiB.
```

The above image shows the creation of two primary partitions of 3 GiB each on the 20GB disk /dev/nvme1n1 using the 'fdisk' utility. The 'p' command prints existing partition details, and 'n' is used to create new partitions. The user selects p for primary type, accepts default sector values, and uses +3G to specify the size. Both partitions are successfully created.

```
Command (m for help): n
Partition type
   p   primary (2 primary, 0 extended, 2 free)
   e   extended (container for logical partitions)
Select (default p): e
Partition number (3,4, default 3):
First sector (12584960-41943039, default 12584960):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (12584960-41943039, default 41943039): +14G
Value out of range.
Last sector, +/-sectors or +/-size{K,M,G,T,P} (12584960-41943039, default 41943039):

Created a new partition 3 of type 'Extended' and of size 14 GiB.

Command (m for help): n
All space for primary partitions is in use.
Adding logical partition 5
First sector (12587008-41943039, default 12587008):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (12587008-41943039, default 41943039): +6G

Created a new partition 5 of type 'Linux' and of size 6 GiB.
```

This image shows the creation of an extended partition and a logical partition using 'fdisk'. Entered 'n' to create a new partition and selects 'e' for extended. Then, the added a logical partition by running 'n' again—since primary partition slots are full, 'fdisk' automatically creates logical partition 5. It is assigned 6 GiB using +6G.

```

Command (m for help): p
Disk /dev/nvme1n1: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: Amazon Elastic Block Store
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x900b28f7

Device            Boot      Start         End Sectors  Size Id Type
/dev/nvme1n1p1                2048     6293503   6291456    3G 83 Linux
/dev/nvme1n1p2           6293504   12584959   6291456    3G 83 Linux
/dev/nvme1n1p3           12584960   41943039  29358080   14G  5 Extended
/dev/nvme1n1p5           12587008   25169919   12582912    6G 83 Linux
/dev/nvme1n1p6           25171968   37754879   12582912    6G 83 Linux

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

ubuntu@ip-172-31-32-198:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0         7:0    0 26.3M  1 loop /snap/amazon-ssm-agent/9881
loop1         7:1    0 73.9M  1 loop /snap/core22/1748
loop2         7:2    0 73.9M  1 loop /snap/core22/1802
loop3         7:3    0 44.4M  1 loop /snap/snapd/23545
loop4         7:4    0 44.4M  1 loop /snap/snapd/23771
nvme0n1       259:0    0   8G  0 disk
├─nvme0n1p1   259:2    0   7G  0 part /
├─nvme0n1p15  259:4    0 106M  0 part /boot/efi
└─nvme0n1p16  259:5    0  913M  0 part /boot
nvme1n1       259:1    0  20G  0 disk
├─nvme1n1p1   259:3    0   3G  0 part
├─nvme1n1p2   259:6    0   3G  0 part
├─nvme1n1p3   259:7    0    1K  0 part
├─nvme1n1p5   259:8    0   6G  0 part
└─nvme1n1p6   259:9    0   6G  0 part

```

Then run the 'p' command again to verify the final partition table. It now shows two 3 GiB primary partitions (nvme1n1p1, p2), a 14 GiB extended partition (p3), and two 6 GiB logical partitions inside it (p5, p6). The 'w' command is then used to write these changes to the disk. After exiting 'fdisk', the 'lsblk' command confirms that all five partitions have been successfully created on /dev/nvme1n1.

```
ubuntu@ip-172-31-32-198:~$ sudo mkfs.ext4 /dev/nvme1nlp2
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 786432 4k blocks and 196608 inodes
Filesystem UUID: 37f46772-ed48-4dcf-ac10-d3b5de3a07cf
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

ubuntu@ip-172-31-32-198:~$ sudo mkfs.ext4 /dev/nvme1nlp5
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 1572864 4k blocks and 393216 inodes
Filesystem UUID: 2f1609f2-0b70-43da-a252-33653e02008f
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

ubuntu@ip-172-31-32-198:~$ sudo mkfs.ext4 /dev/nvme1nlp6
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 1572864 4k blocks and 393216 inodes
Filesystem UUID: 3bc1d1aa-a8a9-4fc2-a253-34da0fab9625
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

This image shows the `lsblk` command output, which lists block devices. It's used to confirm the presence of the newly attached 20GB disk (like `/dev/xvdf` or `/dev/sdb`). At this stage, the disk is **unpartitioned**, and this step is verifying the disk is recognized by the system before creating partitions.


```

ubuntu@ip-172-31-32-198:~$ sudo mkdir /Data1 /Data2 /Data3 /Data4
ubuntu@ip-172-31-32-198:~$ sudo mount /dev/nvme1nlp1 /Data1
ubuntu@ip-172-31-32-198:~$ sudo mount /dev/nvme1nlp2 /Data2
ubuntu@ip-172-31-32-198:~$ sudo mount /dev/nvme1nlp4 /Data3
mount: /Data3: special device /dev/nvme1nlp4 does not exist.
        dmesg(1) may have more information after failed mount system call
ubuntu@ip-172-31-32-198:~$ sudo mount /dev/nvme1nlp5 /Data3
ubuntu@ip-172-31-32-198:~$ sudo mount /dev/nvme1nlp6 /Data4
ubuntu@ip-172-31-32-198:~$ lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                 7:0      0  26.3M  1 loop /snap/amazon-ssm-agent/9881
loop1                 7:1      0  73.9M  1 loop /snap/core22/1748
loop2                 7:2      0  73.9M  1 loop /snap/core22/1802
loop3                 7:3      0  44.4M  1 loop /snap/snapd/23545
loop4                 7:4      0  44.4M  1 loop /snap/snapd/23771
nvme0n1               259:0     0    8G   0 disk
├─nvme0n1p1           259:2     0    7G   0 part /
├─nvme0n1p15          259:4     0   106M  0 part /boot/efi
└─nvme0n1p16          259:5     0   913M  0 part /boot
nvme1n1               259:1     0   20G   0 disk
├─nvme1n1p1           259:3     0    3G   0 part /Data1
├─nvme1n1p2           259:6     0    3G   0 part /Data2
├─nvme1n1p3           259:7     0    1K   0 part
├─nvme1n1p5           259:8     0    6G   0 part /Data3
└─nvme1n1p6           259:9     0    6G   0 part /Data4
ubuntu@ip-172-31-32-198:~$

```

Then, I again used 'lsblk' to check the result of partitioning. Now the output showed /dev/xvdf1, /dev/xvdf2 as primary partitions and /dev/xvdf5, /dev/xvdf6 as logical partitions inside the extended one. This confirmed that all partitions were created successfully.

Each command successfully formatted the partition and made it ready to be mounted.

Next, I created four directories in the root folder to use as mount points. I then mounted the newly formatted partitions onto these folders

```

Last login: Mon Apr  7 09:45:47 2025 from 110.224.90.98
ubuntu@ip-172-31-32-198:~$ ls -lh /Data4/file4
-rw-r--r-- 1 root root 0 Apr  7 09:44 /Data4/file4
ubuntu@ip-172-31-32-198:~$ ls -lh /Data3/file3
-rw-r--r-- 1 root root 0 Apr  7 09:44 /Data3/file3
ubuntu@ip-172-31-32-198:~$ ls -lh /Data2/file2
-rw-r--r-- 1 root root 0 Apr  7 09:43 /Data2/file2
ubuntu@ip-172-31-32-198:~$ ls -lh /Data1/file1
-rw-r--r-- 1 root root 0 Apr  7 09:43 /Data1/file1
ubuntu@ip-172-31-32-198:~$

```

At this point, each of the four partitions was attached to its respective folder under root.

To test the partitions, I created empty files of specific sizes inside each mounted folder

This created files of size 2 GB, 2 GB, 4 GB, and 4 GB respectively. These files helped me fill up space and verify storage utilization.

```
ubuntu@ip-172-31-32-198:/Data1$ while true; do sleep 5s; done
^Z
[1]+  Stopped                  sleep 5s
ubuntu@ip-172-31-32-198:/Data1$ df -hT /Data1 /Data2 /Data3 /Data4
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/nvme1n1p1  ext4  2.9G   24K  2.8G   1% /Data1
/dev/nvme1n1p2  ext4  2.9G   24K  2.8G   1% /Data2
/dev/nvme1n1p5  ext4  5.9G   24K  5.6G   1% /Data3
/dev/nvme1n1p6  ext4  5.9G   24K  5.6G   1% /Data4
ubuntu@ip-172-31-32-198:/Data1$
```

After that, I went into '/Data1' using 'cd /Data1' and ran a simple infinite loop, This kept the terminal busy. I paused the loop using 'Ctrl+Z' to simulate a background process running in the mount point.

```
Last login: Mon Apr  7 09:46:41 2025 from 110.224.90.98
ubuntu@ip-172-31-32-198:~$ sudo umount /Data1
ubuntu@ip-172-31-32-198:~$ sudo umount /Data2
umount: /Data2: not mounted.
ubuntu@ip-172-31-32-198:~$ sudo umount /Data3
umount: /Data3: not mounted.
ubuntu@ip-172-31-32-198:~$ sudo umount /Data4
umount: /Data4: not mounted.
ubuntu@ip-172-31-32-198:~$
```

This cleanly unmounted the devices from their respective folders.