Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
Institute of Computer Technology

Name: Tushar Panchal

En.No: 21162101014

Sub: ITIM (IT Infrastructure & Management)

Branch: CBA

Batch:61

----------PRACTICAL 03----------

1. **You need to create a partition on a new storage device and format it with an ext4 file system, configure it to be mounted at boot, and mount it for use. (The mount point should be a directory named after you) :**

» **Check the empty space and disk partitions and initiate creation of partition using fdisk tool :**

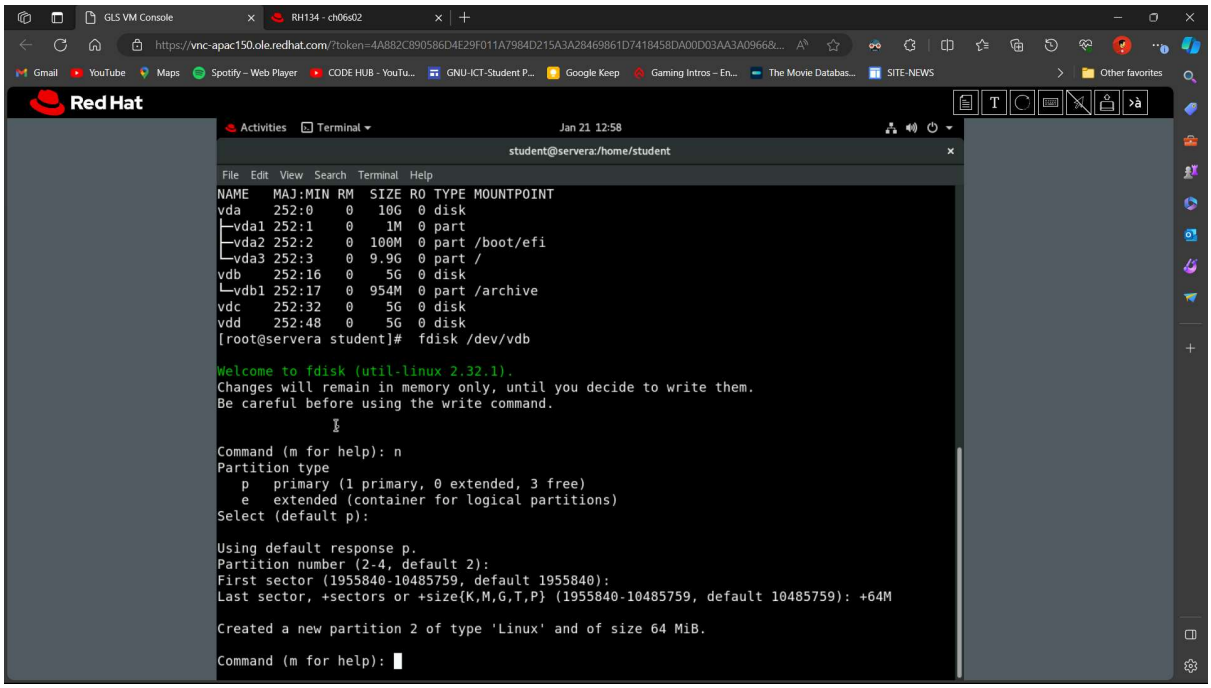
```
[student@servera ~]$ su root
Password:
[root@servera student]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    252:0    0   10G  0 disk
├─vda1 252:1    0    1M  0 part
├─vda2 252:2    0  100M  0 part /boot/efi
└─vda3 252:3    0  9.9G  0 part /
vdb    252:16   0    5G  0 disk
└─vdb1 252:17   0  954M  0 part /archive
vdc    252:32   0    5G  0 disk
vdd    252:48   0    5G  0 disk
[root@servera student]#  fdisk /dev/vdb

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help):
```

Let's break down the commands we've executed:

1. **lsblk**
    - This command is used to list information about all available block devices (storage devices) on your system. In the output you provided, it shows information about several devices:
    - **/dev/vda**: A 10GB disk with partitions (**vda1**, **vda2**, **vda3**). The third partition (**Lvda3**) is mounted as the root filesystem (**/**).
    - **/dev/vdb**: A 5GB disk with a partition (**Lvdb1**) that is mounted at **/archive**.
    - **/dev/vdc** and **/dev/vdd**: Two 5GB disks.
2. **fdisk /dev/vdb**
    - This command is used to interactively manage disk partitions on the specified block device (**/dev/vdb**). The **fdisk** command opens an interactive menu where you can perform various operations on the disk's partition table.

» Creating the partition :

```
[root@servera student]#  fdisk /dev/vdb

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (1955840-10485759, default 1955840):
Last sector, +sectors or +size{K,M,G,T,P} (1955840-10485759, default 10485759): +64M

Created a new partition 2 of type 'Linux' and of size 64 MiB.

Command (m for help):
```

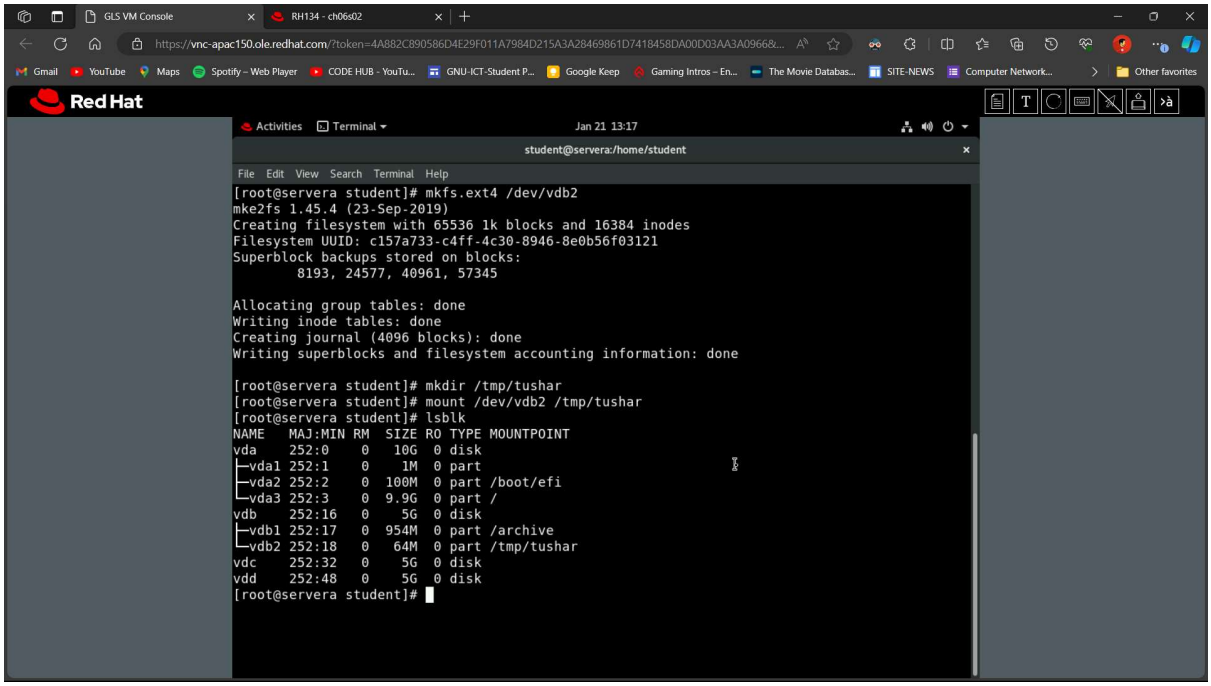


Let's break down the commands we've executed:

1. **fdisk /dev/vdb**:
    - This command launches the **fdisk** utility and specifies the block device **/dev/vdb** as the target for partitioning. You are essentially telling **fdisk** that you want to make changes to the partition table of the **/dev/vdb** disk.

2. **Command (n for create a new partition):**:
    - You entered **n** to create a new partition. This initiates the process of creating a new partition on the specified disk.
3. **Partition type (default p):**:
    - **fdisk** is asking you to choose the partition type. The default is a primary partition (**p**). You accepted the default by pressing Enter.
4. **Last sector, +sectors or +size (default 10485759):**:
    - **fdisk** is asking you to specify the size of the new partition. The default is 10485759 sectors. You chose to allocate 64 MiB for the new partition by entering **+64M**.
5. **Created a new partition 2 of type 'Linux' and of size 64 MiB:**:
    - **fdisk** confirms that it has created a new partition (partition 2) of type 'Linux' and with a size of 64 MiB.

» Assigning filesystem to the newly created partition :



Let's break down the commands we've executed:

1. **mkfs.ext4 /dev/vdb2**:
    - This command is used to create an ext4 filesystem on the specified block device, in this case, **/dev/vdb2**. The output of this command indicates the creation of the filesystem with a specific UUID, the allocation of blocks and inodes, the creation of journal blocks, and the writing of superblocks and filesystem accounting information.

output provides details about the filesystem creation process.

2. **mkdir /tmp/tushar**:
    - This command creates a new directory named **tushar** inside the **/tmp** directory. The **-p** option is not used, so the command assumes that the parent directories (**/tmp** in this case) already exist.
3. **mount /dev/vdb2 /tmp/tushar**:

- This command mounts the ext4 filesystem created on **/dev/vdb2** to the **/tmp/tushar** directory. This means that the contents of the filesystem on **/dev/vdb2** will be accessible under the **/tmp/tushar** directory.

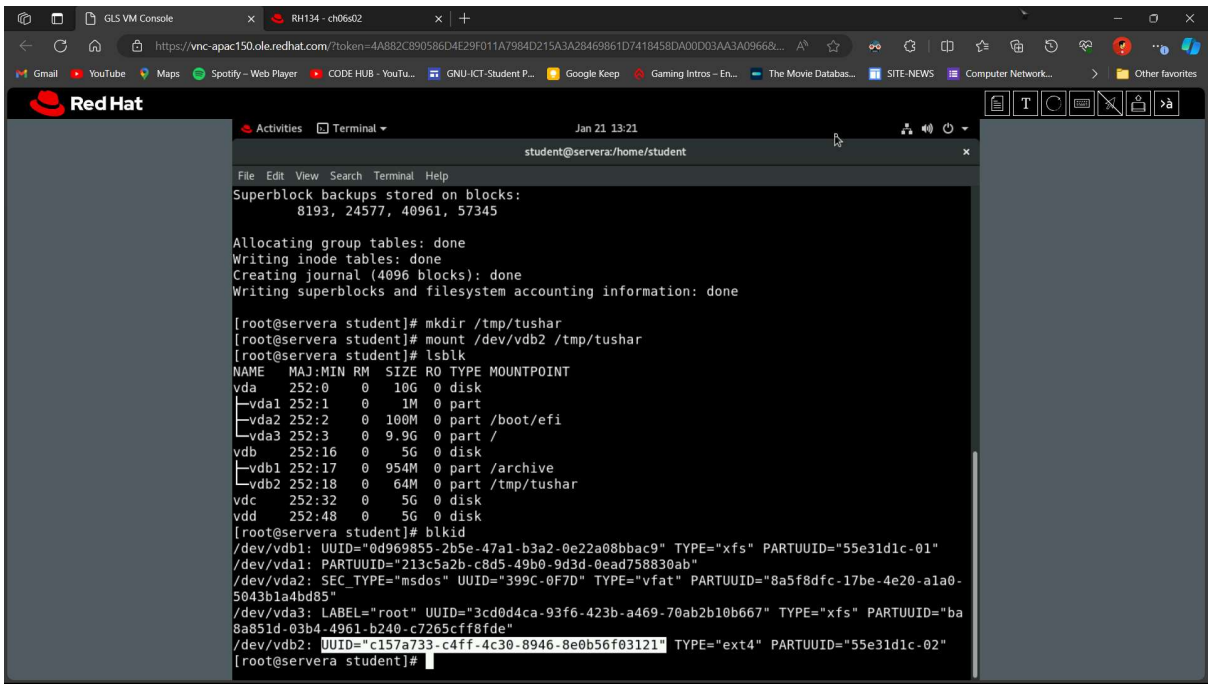
4. **lsblk**:

   - This command is used to list information about all available block devices on your system. In the output, you can see the current state of your block devices, including the newly created ext4 partition **/dev/vdb2** that has been mounted on **/tmp/tushar**.

The output shows the devices, their sizes, types, and mount points.

Your newly created ext4 partition (**/dev/vdb2**) has been successfully formatted, mounted to **/tmp/tushar**, and you can now use this directory to store files and data on that partition.

» Getting the UUID of the partition to create its manual entry in fstab :



Let's break down the commands we've executed:

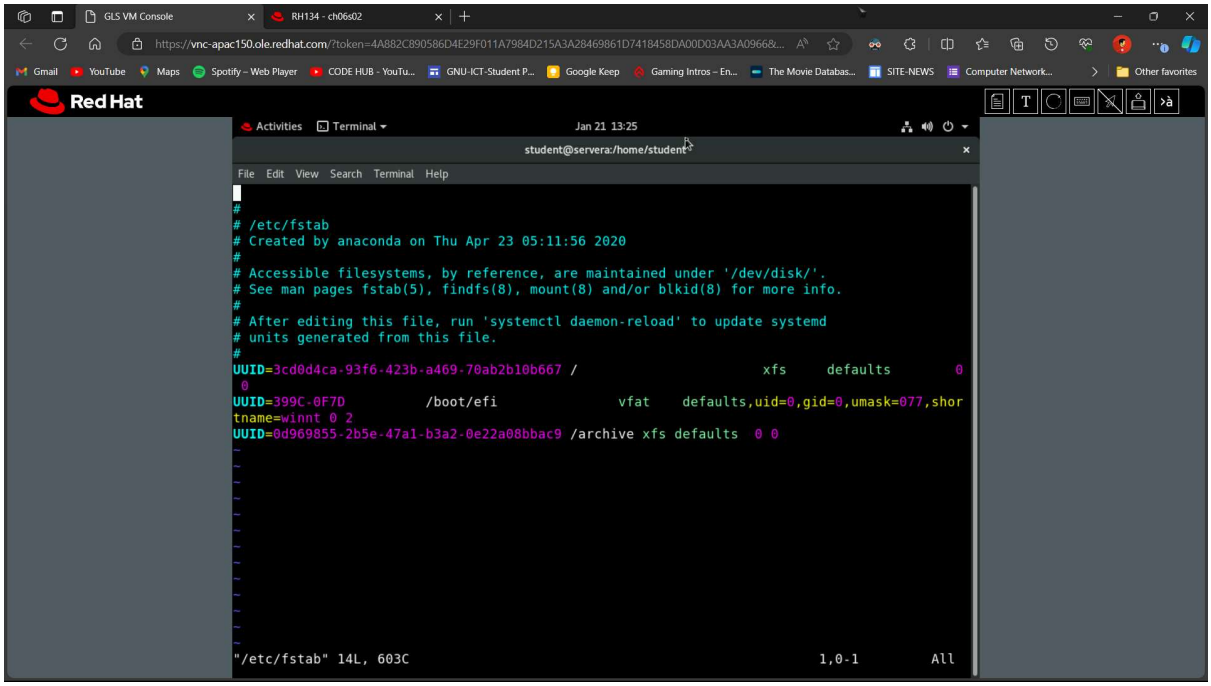
The **blkid** command is used to display information about block devices, specifically information about the attributes (UUID, filesystem type, labels, etc.) of each device. Let's break down the output :

Here's an explanation :

1. **/dev/vdb2: UUID="c157a733-c4ff-4c30-8946-8e0b56f03121" TYPE="ext4" PARTUUID="55e31d1c-02"**
    - This line provides information about the **/dev/vdb2** partition.
    - UUID: Universally Unique Identifier for the ext4 filesystem.
    - TYPE: The filesystem type is ext4.
    - PARTUUID: Partition UUID.

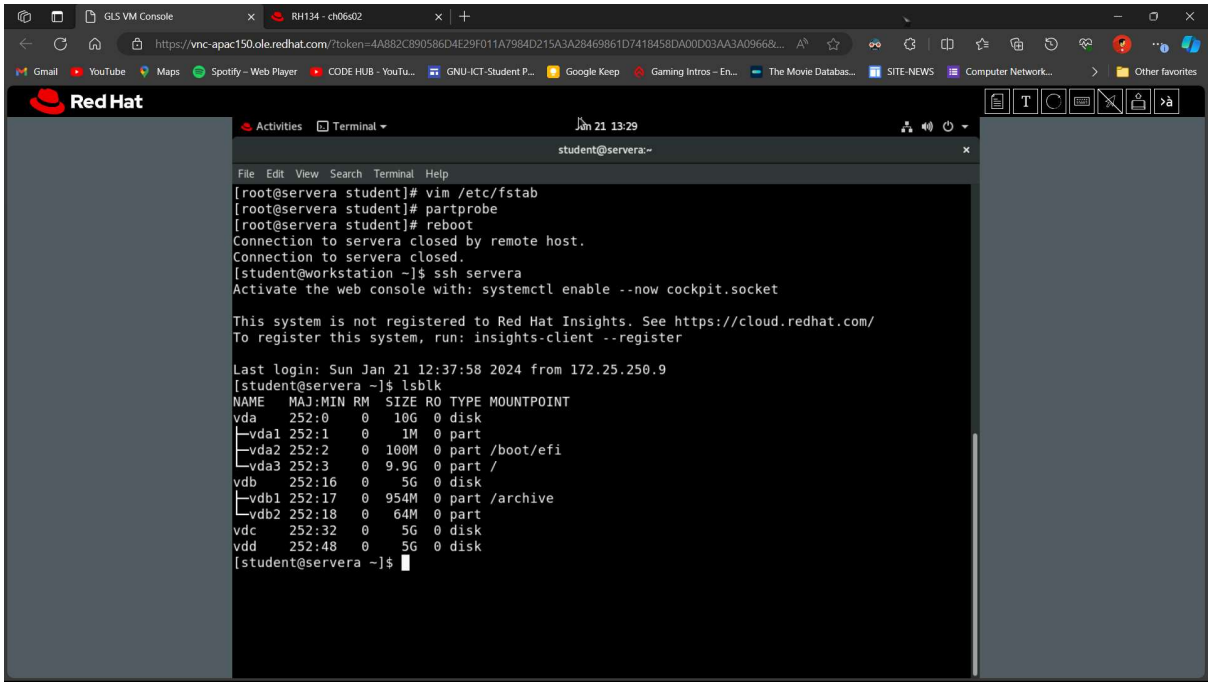
These details are useful for identifying and managing your block devices, especially when working with multiple disks and partitions on a system. The UUID is a unique identifier for each filesystem, and it's often used in configuration files like **/etc/fstab** to reference specific partitions.

» To ensure block device boot, creating manual entry in /etc/fstab :



Let's break down the commands we've executed:
The **vim /etc/fstab** command is used to edit the **/etc/fstab** file using the Vim text editor. The **/etc/fstab** file is a system configuration file on Unix-like operating systems (including Linux) that contains information about disk drives and partitions, as well as the file systems and mount points associated with them. The file is read by the system during boot to determine how to mount these filesystems.

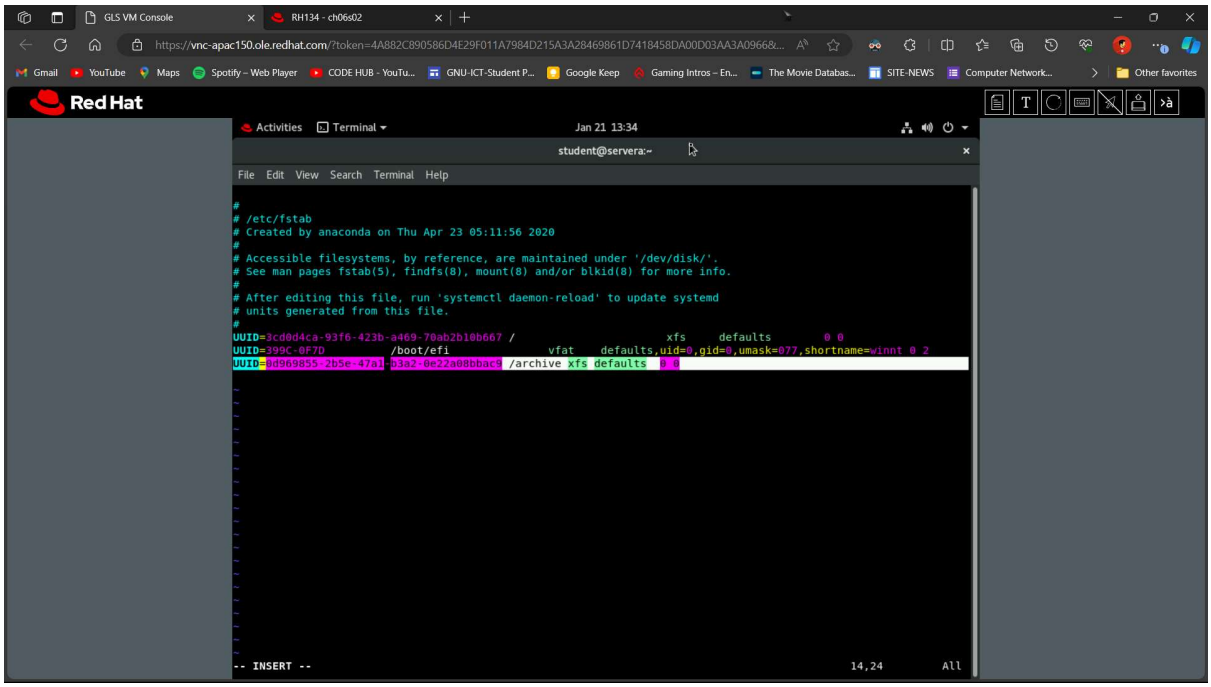
» Refreshing and rebooting the device to check the successful setup :

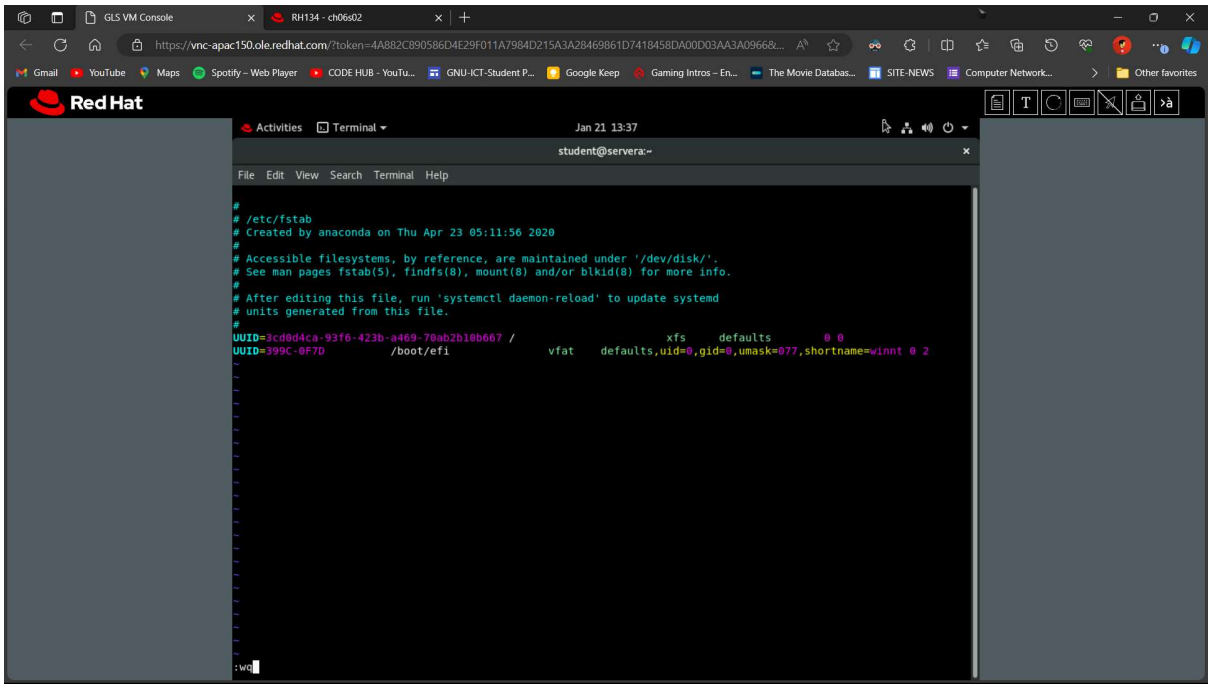


Let's break down the commands we've executed:
The **partprobe** command is used to inform the operating system kernel about changes in the partition table of a block device without requiring a system reboot. This is particularly useful when you've made changes to the partition table (e.g., created, deleted, or resized partitions) and you want the system to recognize these changes without having to restart.

2. **You need to delete the created partition and ensure that the changes are persistent, so that when the device is rebooted, the created partition is removed :**

» **Removing the entry from /etc/fstab :**



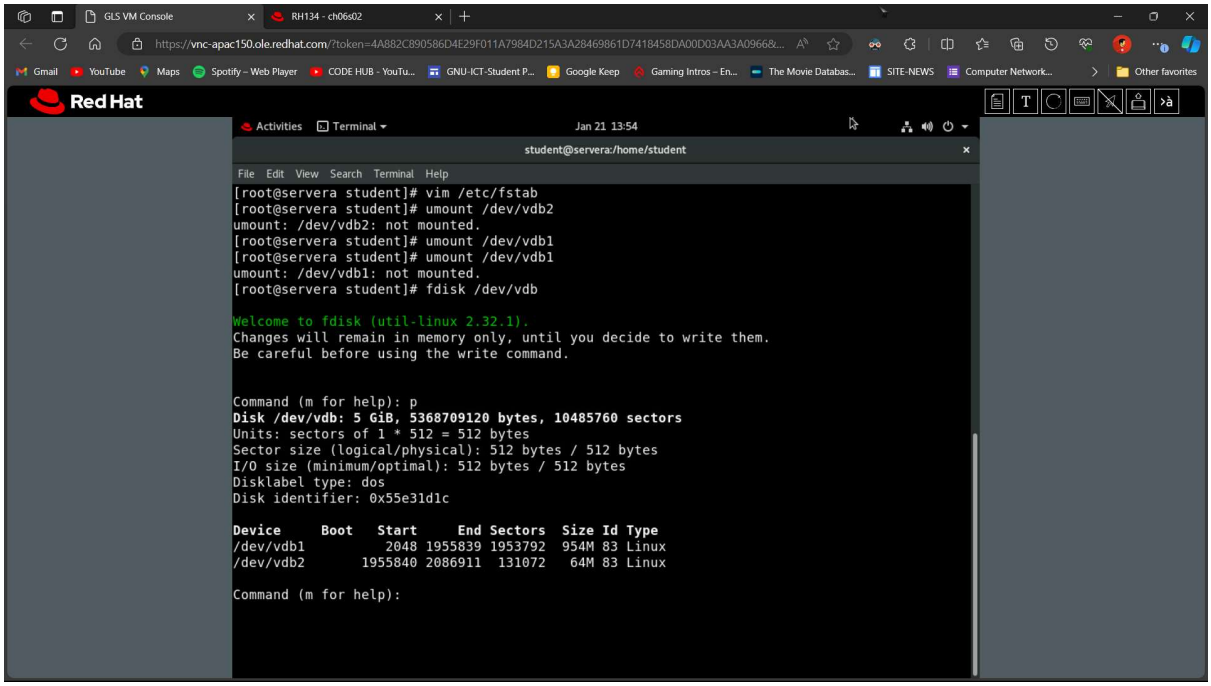


after removing it escape with (:wq) write and quit

Remove the highlighted line from above

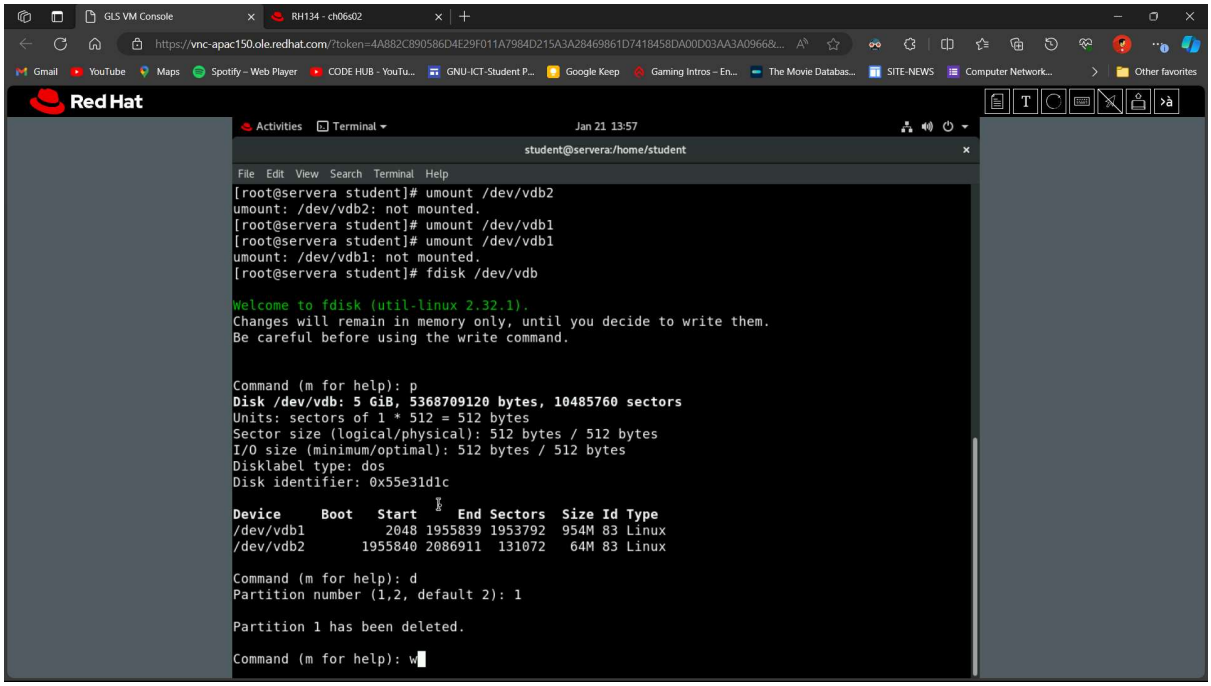
Command : `vim /etc/fstab`

» Unmounting the block and removing the partition :




```
[root@servera student]# umount /dev/vdb1
[root@servera student]# umount /dev/vdb1
umount: /dev/vdb1: not mounted.
```

```
[root@servera student]# fdisk /dev/vdb

Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/vdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x55e31d1c

Device     Boot   Start     End Sectors  Size Id Type
/dev/vdb1          2048 1955839 1953792  954M 83 Linux
/dev/vdb2       1955840 2086911  131072   64M 83 Linux

Command (m for help):
```



Let's break down the commands we've executed:

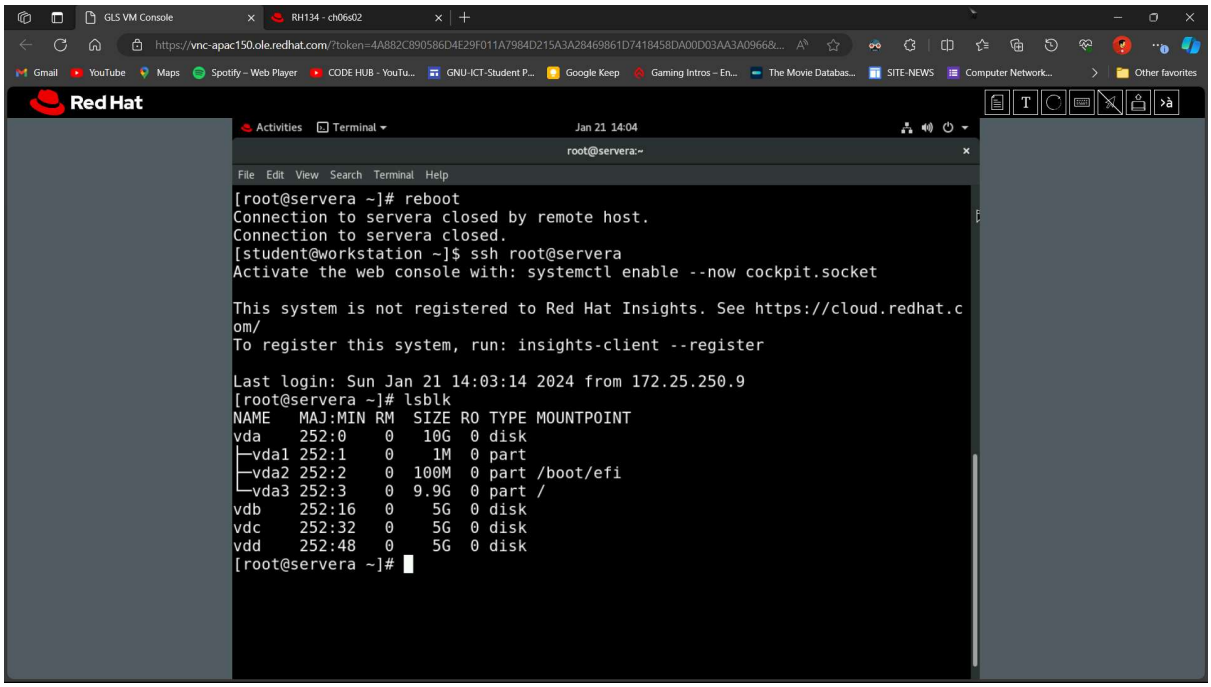
1. **umount /dev/vdb1**:
    - This command unmounts the filesystem on the **/dev/vdb1** partition. Before making changes to a partition using tools like **fdisk**, it's often necessary to unmount it to ensure that the filesystem is not in use. The unmount operation frees up the device so that it can be modified safely.
2. **fdisk /dev/vdb**:
    - This command launches the **fdisk** utility for the block device **/dev/vdb**. You are entering the interactive mode of **fdisk** to make changes to the partition table of the **/dev/vdb** disk.
3. **p (Command for printing the partition table)**:
    - You entered **p** to print the current partition table for **/dev/vdb**. This shows information about the existing partitions on the disk.

The output provides information about the disk's size, sector size, disk label type, and identifier.

4. **d (Command for deleting a partition)**:
    - You entered **d** to delete a partition. The utility prompts you to select the partition to delete.
5. **W (Command for writing changes to disk)**:
    - After selecting and deleting the partition, you entered **W** to write the changes to the disk.

The **W** command writes the changes to the disk, and the utility exits.

» Reboot and check if the partition is successfully deleted :



Let's break down the commands we've executed:

1. **reboot**:
    - This command is used to initiate a system reboot. It gracefully shuts down the system, terminates all

running processes, and restarts the machine. The **reboot** command effectively reboots the server.

The system has now initiated a reboot, and the SSH connection to the server has been closed.

2. **ssh root@servera**:
    - After the server has completed the reboot, you use the **ssh** command to log in again. The **[student@workstation ~]$** prompt indicates that you are logging in from a workstation as a user named "student." Then, you log in as the root user on the server.

The server prompts you with the root user's shell prompt.

3. **lsblk**:
    - This command is used to list information about all available block devices (storage devices) on the system.

The output provides information about the block devices on the server, including their names, sizes, and mount points.