**Lab Project 02 – Partitions, Filesystems, and Package Managers**

**Introduction.** This lab introduces the critical skillsets of creating, managing, mounting, and unmounting Linux partitions and filesystems from the command line interface (CLI). Additionally, this lab will introduce you to package managers, uses of a package manager, and the minimum package manager commands you should build familiarity with as you continue to work with Linux.

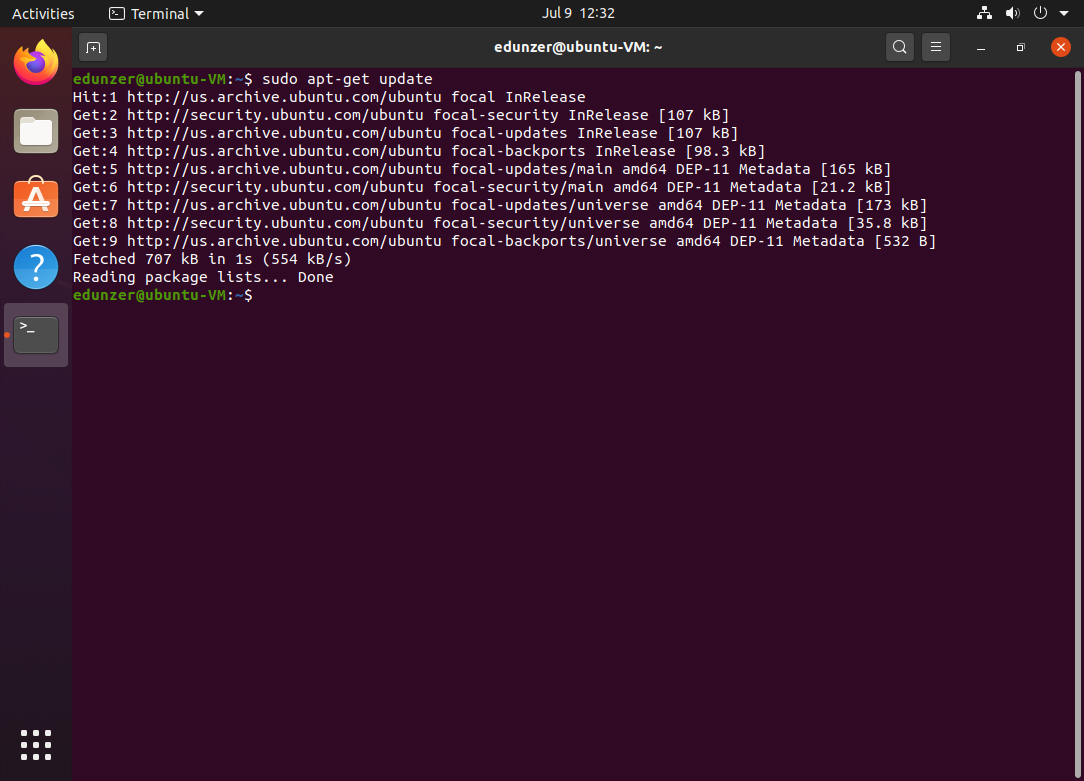
**Important Note.** As we progress through this course, I will start reducing the amount of detail provided in the assignment instructions. My expectation is you will research and develop your own way for completing the required task. There is always more than one way to accomplish a single task. This does not mean you are completely alone on this venture. Please review all material for each week and please utilize the weekly discussion forum for help from your peers.

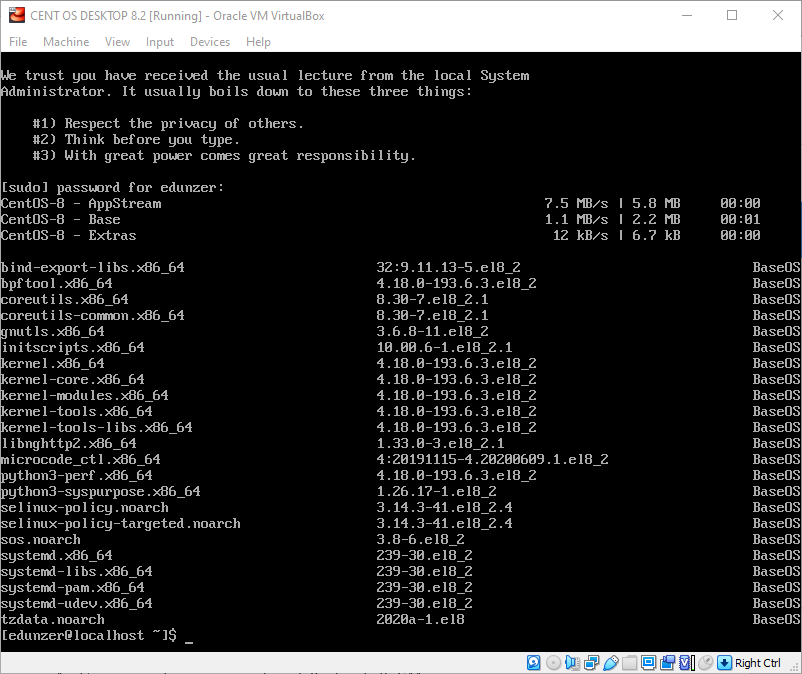
**Instructions.** Provide screenshots of your work as proof of your progress with a brief explanation of each screenshot. This lab assignment includes written answers, which must meet college-level writing requirements regarding formatting, citations, spelling, and grammar.

**Package Managers**

1. Update your Linux distribution with its respective package manager.
   1. yum – the native package manager in CentOS
   2. apt-get – the native package manager tool for apt in Ubuntu

**\*\*\*Screenshot #1, show completed update commands for BOTH VMs**

****

****

1. Explain the steps needed to completely remove a package in both VMs.
   1. Hint: yum and apt-get are tools used to interface with the package manager, meaning, you may not use yum or apt-get to perform these steps.

In CENT OS all you need to do is locate the package name, and then in the command line type in “yum remove [package name]”

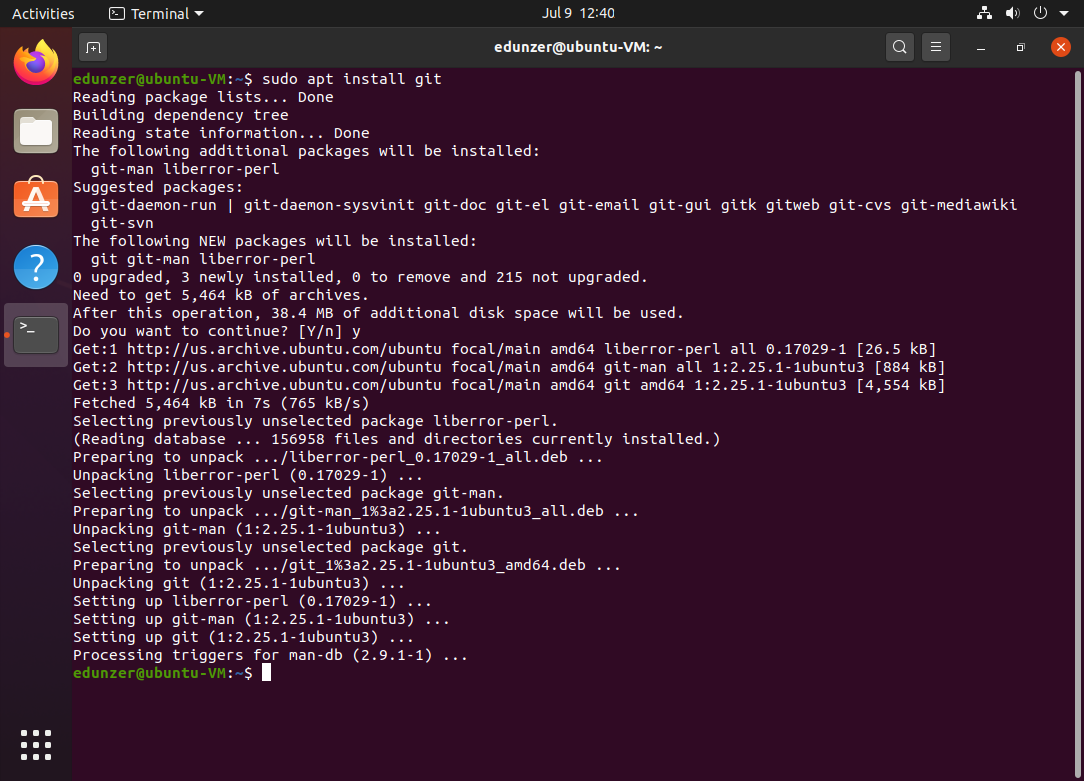
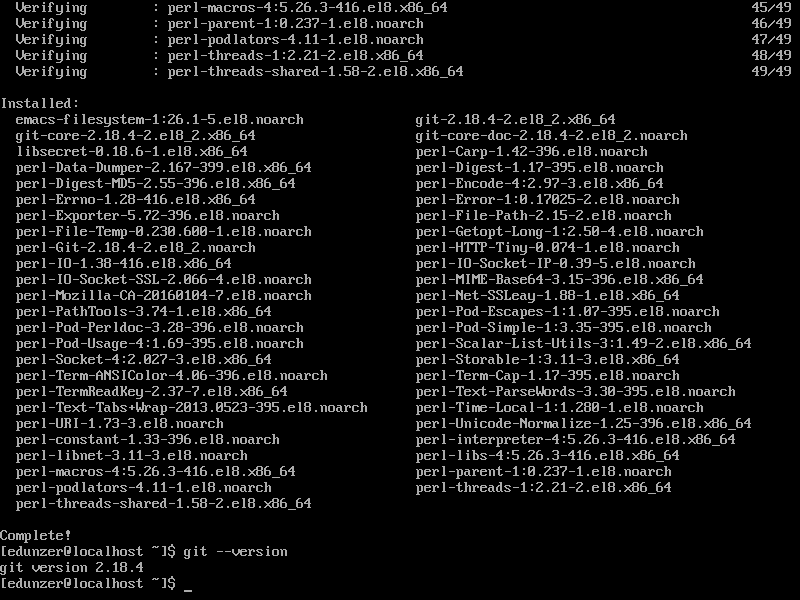
In Ubuntu you do a similar command in terminal but instead of yum you use “sudo apt remove [package name]”

1. Explain the steps and purpose of using the package manger to reconfigure an installed package.

The package manager allows for all dependencies of files to be installed, along with repositories, etc. and it also allows for upgrades and updating of existing packages and the Linux system.

1. Use your package managers to install git on both VMs.
   1. When installing one package, you noticed your Linux distro installed many packages. What is the purpose of these additional packages? Explain.

**\*\*\*Screenshot #2, show completed installation of git for BOTH VMs**

****

**Partitions**

The remaining portion of this lab is only for your Ubuntu VM. Before you start, shutdown your Ubuntu VM and attach a new virtual hard disk (VHD). To do so:

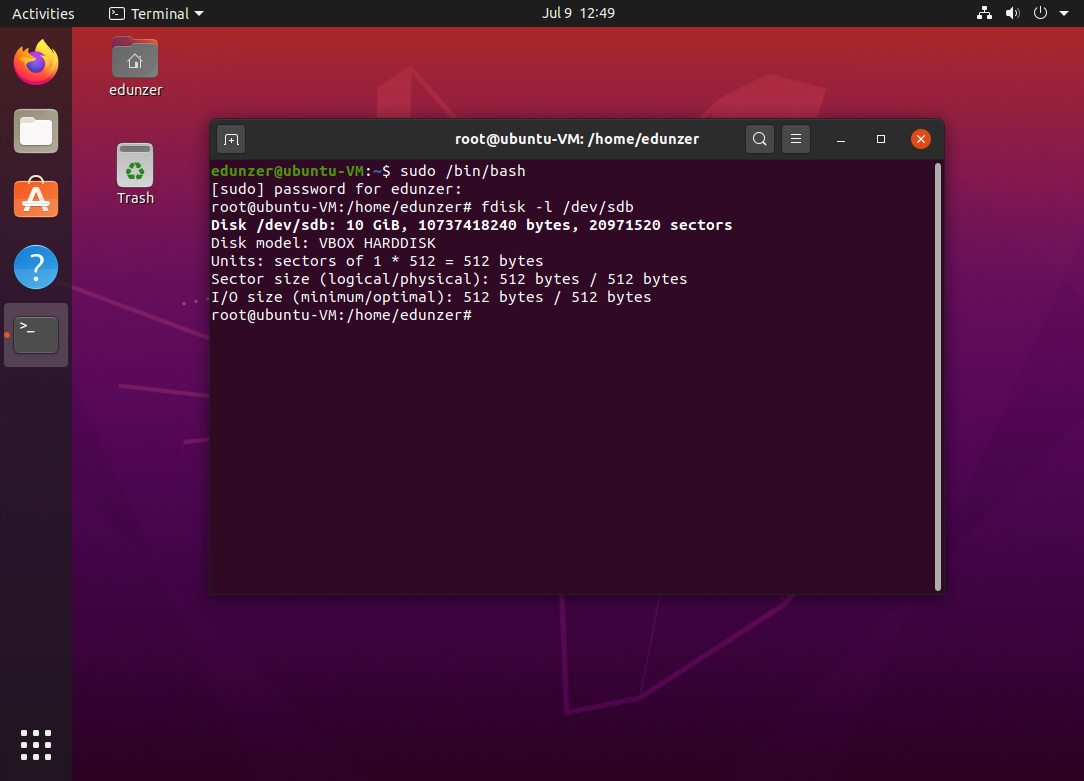
1. Enter Settings
2. Click on Storage
3. Click on the SATA Controller and click the icon to add a new hard disk (not optical)
4. Follow the disk creation wizard to create a new VHD with the size of 10GB.
5. Click OK to save your settings and start the Ubuntu VM

**Note:** It is important to understand that when you create virtual hard disks, or virtual disk images in VirtualBox, we always make them dynamic disks. This means we only use the amount of physical disk space on your host computer as needed. In other words, we did not create a 10GB file in the previous steps. It can, however, grow to 10GB if you fill the new virtual disk with data.

After your Ubuntu VM is up and running, login and open a terminal. In the terminal, switch users to the root user. A quick way to do what in Ubuntu is enter the command: **sudo /bin/bash**. This will ask for your password then drop you down into a root shell.

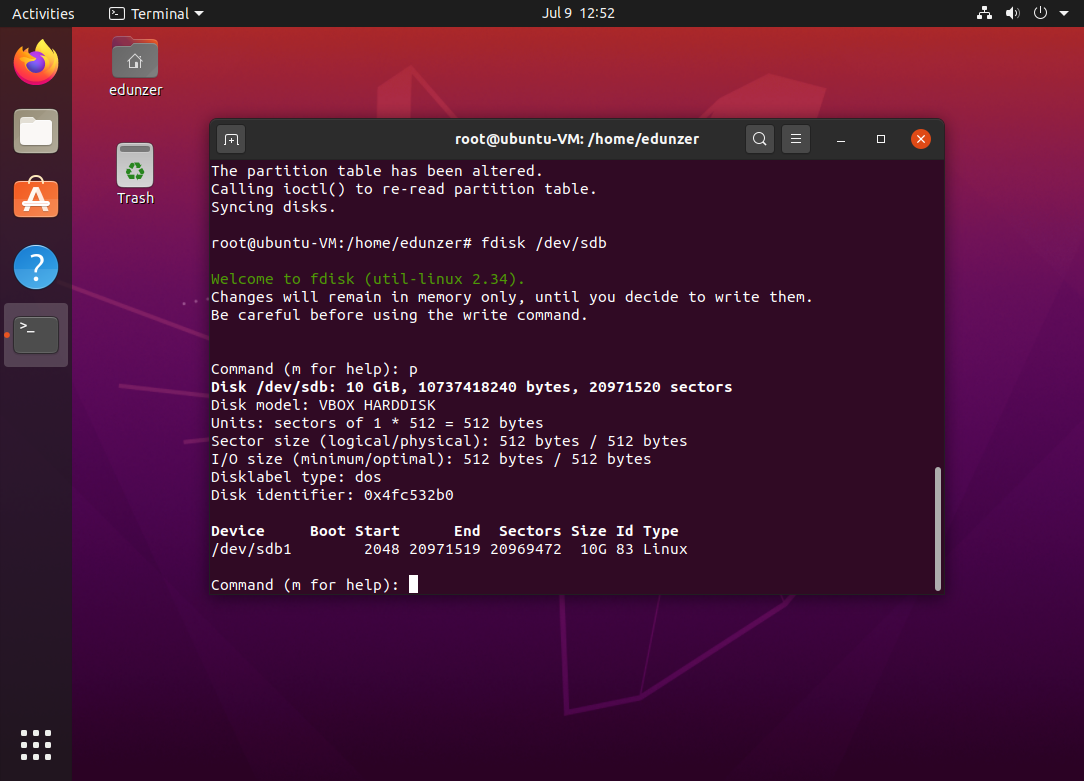
1. Use the command: **fdisk –l /dev/sdb**. The ‘l’ is a lowercase L, not a one (1).

**\*\*\*Screenshot #3, show your output from the above command showing your new disk**

****

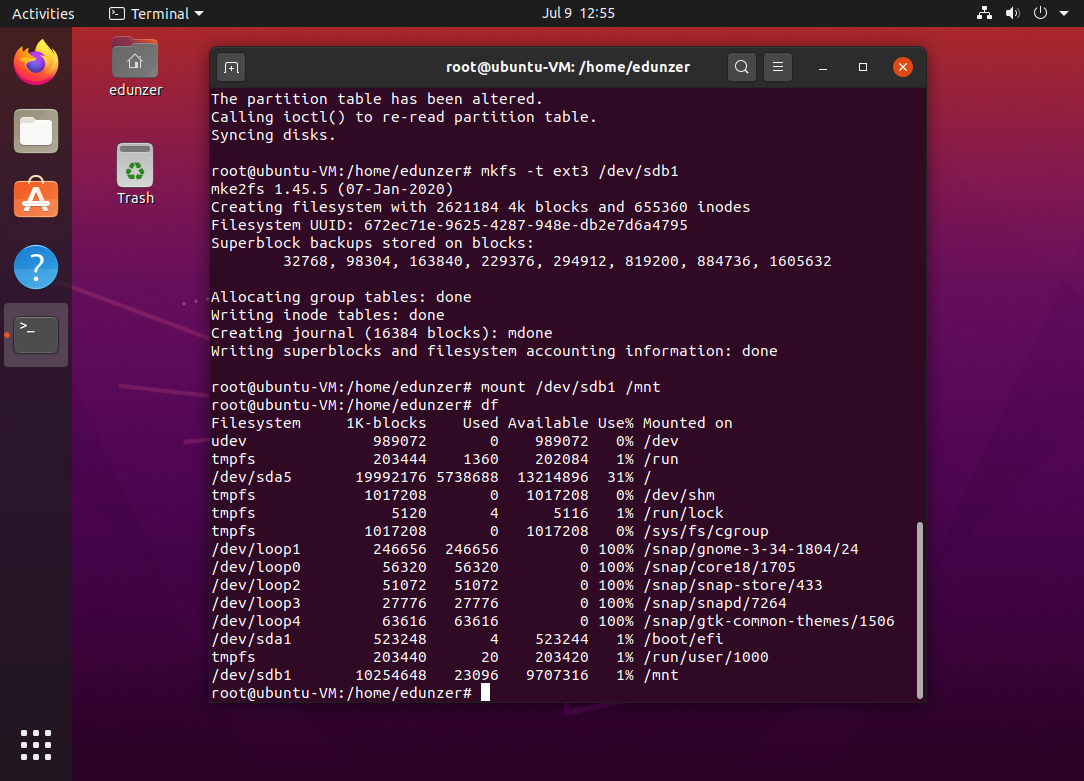
1. Now, setup a partition on the new disk to prepare it for a filesystem
   1. **fdisk /dev/sdb**
      1. Enter: **p**, this lists current partitions on your disk. At this point, you should not have any partitions listed here.
      2. Enter: **n**, this will start the process to create a partition. Use the default values shown on your screen (you can press **enter** without entering the values to automatically use the default values).
      3. Once fdisk shows an output that it has created the partition, save the changes and exit fdisk by entering: **w**.
   2. Enter fdisk again by entering: **fdisk /dev/sdb**
      1. Use the **p** option to view partitions

**\*\*\*Screenshot #4, show your newly created partition on sdb (sdb1).**

****

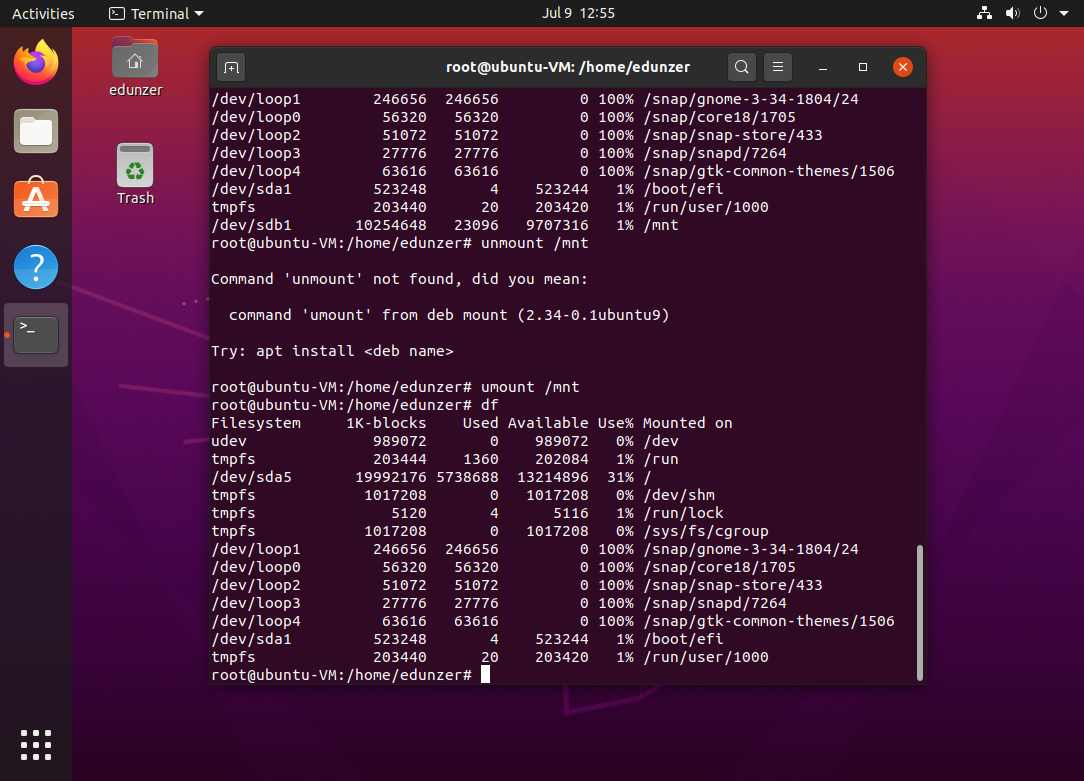
1. Next, create a filesystem to make the disk a partition usable
   1. Enter the command: **mkfs –t ext3 /dev/sdb1**
2. Mount the filesystem
   1. Enter: **mount /dev/sdb1 /mnt**

**\*\*\*Screenshot #5, use the df command to show your mounted filesystem**

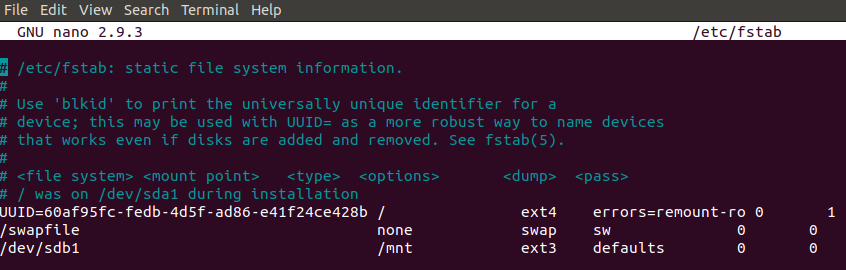
****

1. Unmount the filesystem
   1. Enter: **umount /mnt**

**\*\*\*Screenshot #6, use the df command to show the filesystem is no longer mounted**

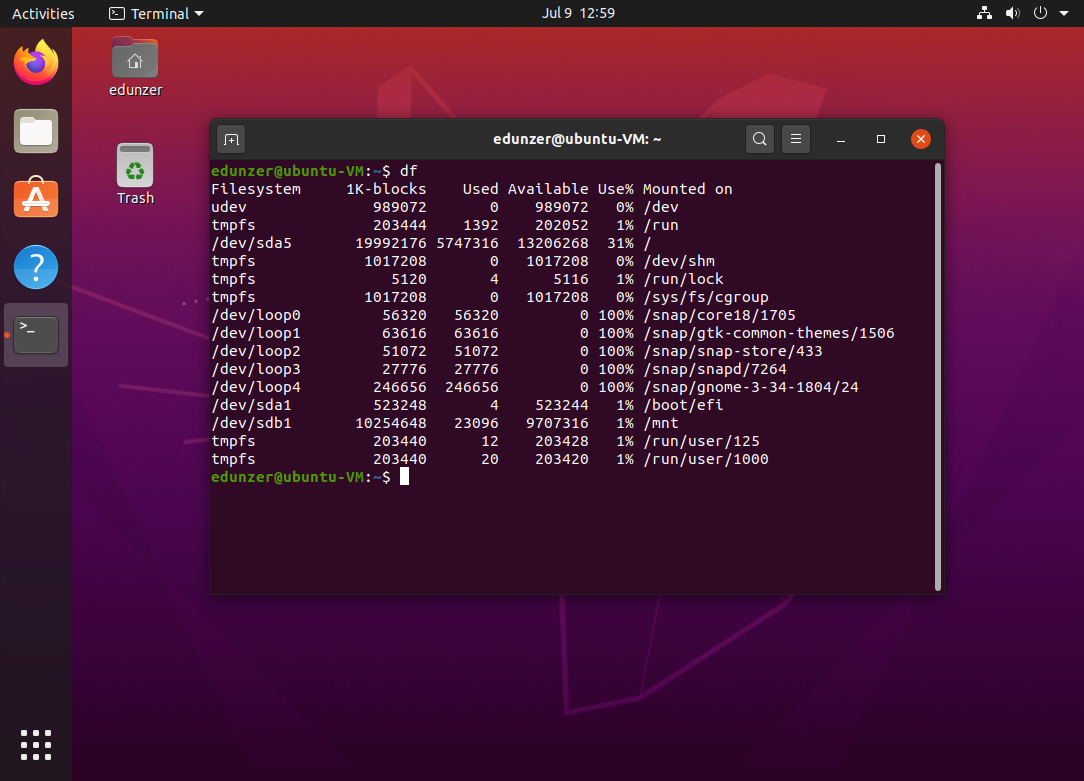
****

1. Edit the /etc/fstab file using nano to permanently mount the filesystem after every reboot.
   1. Enter: **nano /etc/fstab**
   2. At the bottom of the file enter the following, align it with the /swapfile entry above it. Your output should look like this:



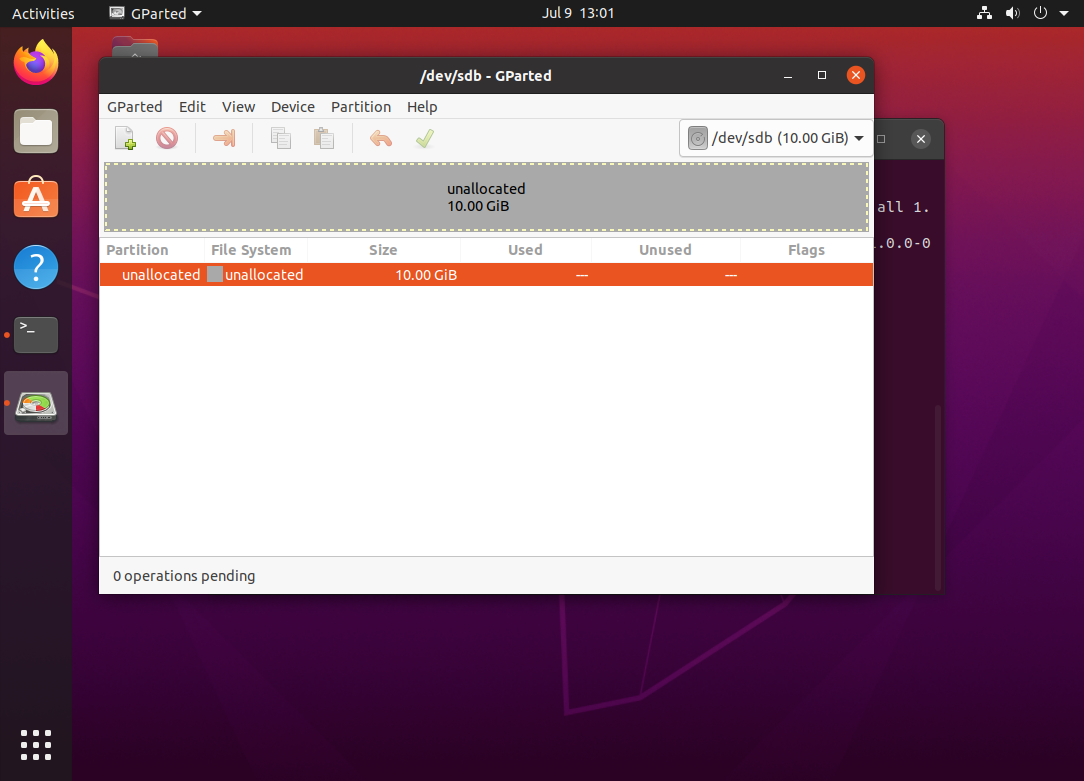
* 1. Save the file by pressing the **Ctrl** key and **W**, press **enter**, then press **Ctrl** and **X** to exit nano
  2. Reboot the system

**\*\*\*Screenshot #7, use the df command to show your mounted filesystem immediately following a fresh reboot**

****

1. Use your package manager to install the package gparted.
2. After installation is complete, open gparted (an application now available in the application browser in Ubuntu)
   1. From the desktop, click the 9 dots forming a square in the lower left-hand corner. Click gparted.
   2. In the top, right-hand corner of gparted, you can select /dev/sdb
   3. Unmount the filesystem and delete the partition. Make sure to click the green check mark to apply the changes.

**\*\*\*Screenshot #8, show your gparted successful completion screen**

****

1. Make sure to remove your permanent mount from /etc/fstab or you will receive system errors the next time you reboot.

**Filesystem Questions**

1. In addition to your work above, explain the main differences between ext2, ext3, and ext4. Notice, your Ubuntu is using ext4 and this lab had you use ext3.

The main difference is that ext4 supports much larger file sizes. And ext3 allows journaling until ext2.

1. What other filesystems are available in Linux?

XFS, JFS, btrfs are the main file systems besides the ext family.

1. Why would you use a different filesystem other than the default? In other words, when would you use something other than ext4 when install Ubuntu?

One of the biggest reasons is support, and speed. New file systems could come out with better speed and be more stable. But you can also switch to other file systems to support your needs. Some are better at pooling, snapshots, disk striping, etc. then the ext family.

**Save and Upload.** Save your in one document and upload to Canvas for grading.