**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 1**

**Aim**

Install the latest version of Ubuntu on an Oracle VM VirtualBox.

**Procedure**

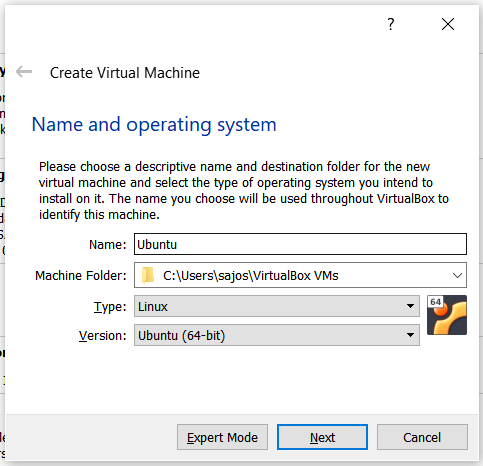
## Step 1: [Download VirtualBox](https://www.virtualbox.org/wiki/Downloads) for Windows and install it on the computer.



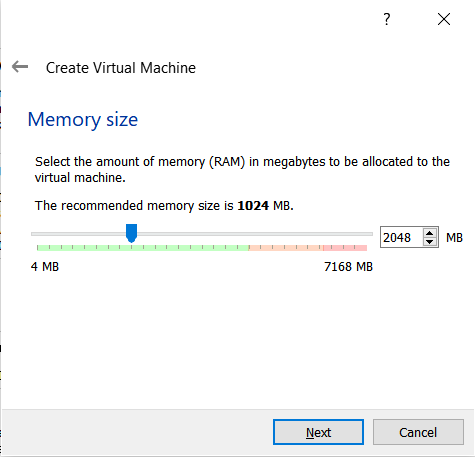
## Step 2: Download the Ubuntu [ISO file](https://www.lifewire.com/iso-file-2625923) you want to install from the [Ubuntu download page](https://ubuntu.com/download/desktop).

## Step 3: Open VirtualBox and select **New** in the top taskbar.

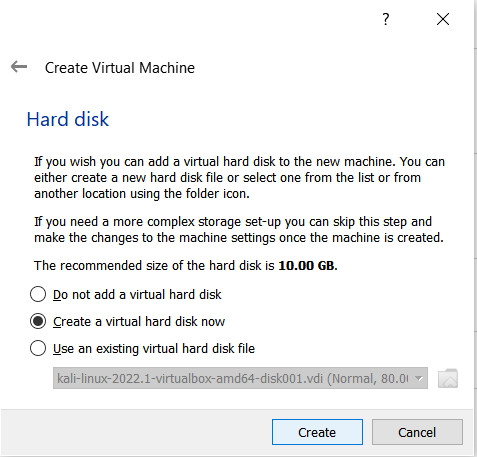
## Step 4: Give VM a name, choose **Linux** and then choose **Ubuntu** as the **Version**and select **Next**.



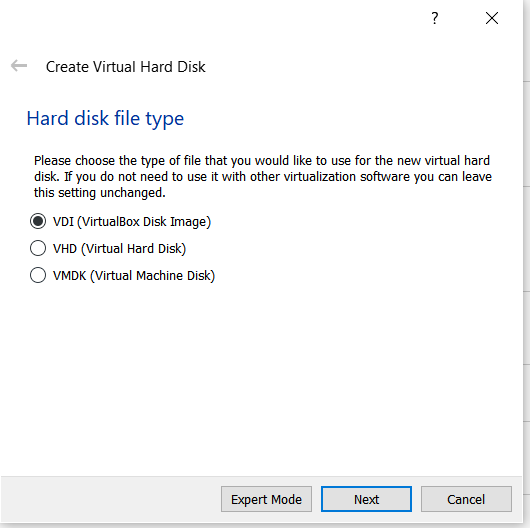
**Step 5:** We want to specify Memory size.

****

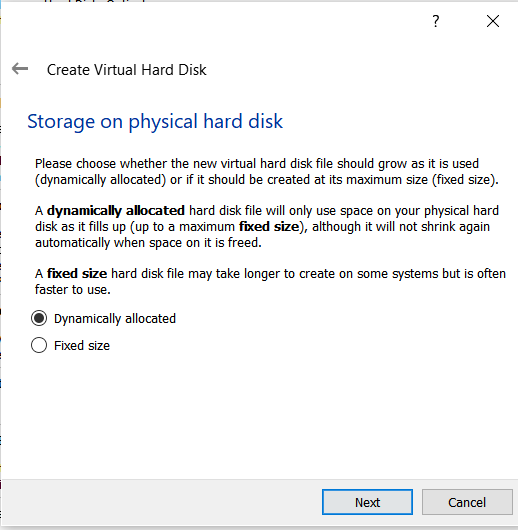
**Step 6:** Check the "Create a virtual hard disk now" option so we can later define our Ubuntu OS virtual hard disk size.



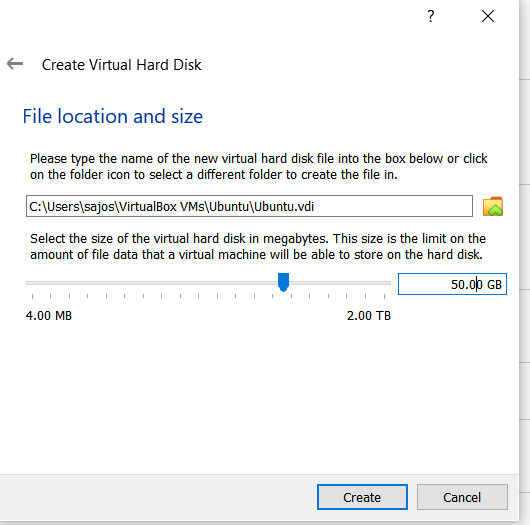
**Step 7:** Select "VHD (Virtual Hard Disk)".

****

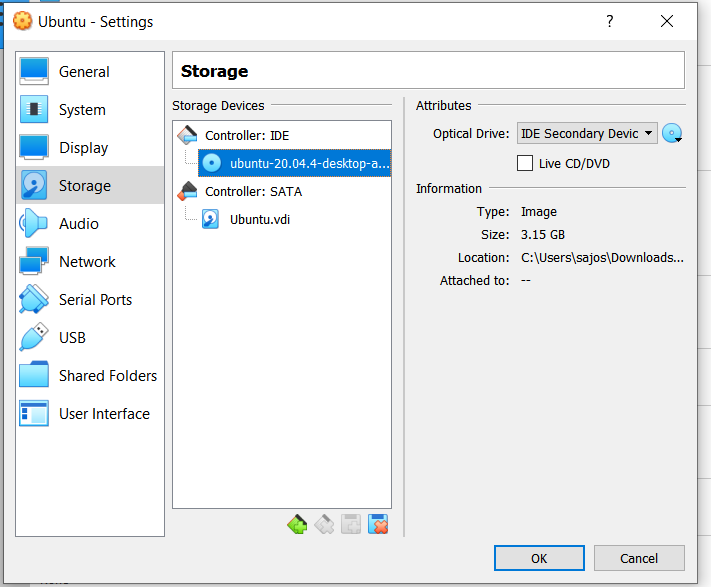
**Step 8:** Next, we'll dynamically allocate storage on our physical hard disk.



**Step 9:** We want to specify our Ubuntu OS's size. The recommended size is 10 GB, but you can increase the size if you wish.

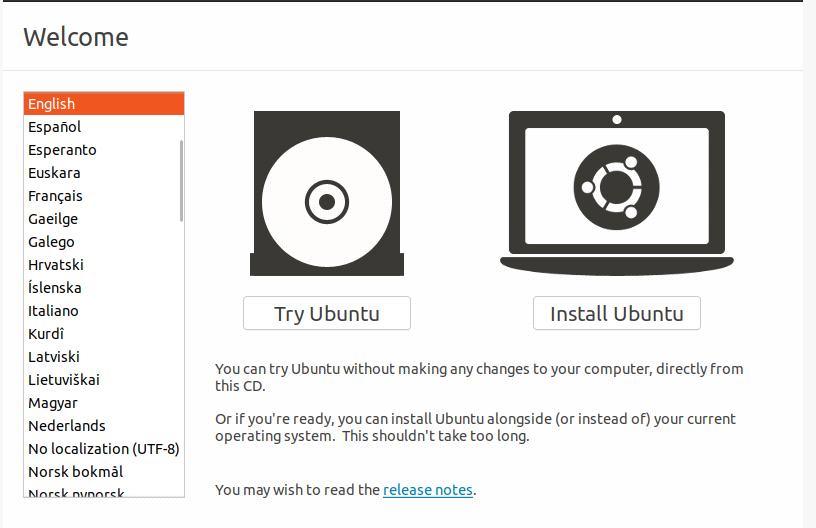
****

**Step 10:** Select virtual disk image.

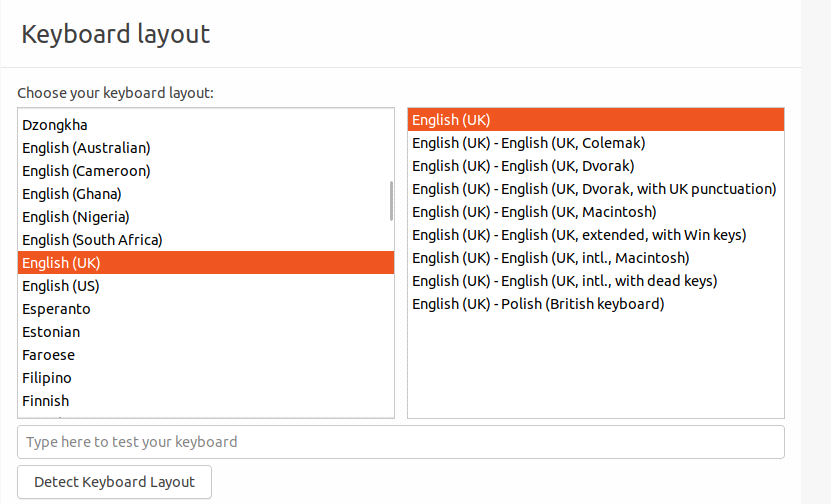


**Step 11**: The name of your virtual machine will now appear on the left side of the VirtualBox manager. Select Start in the toolbar to launch your VM.

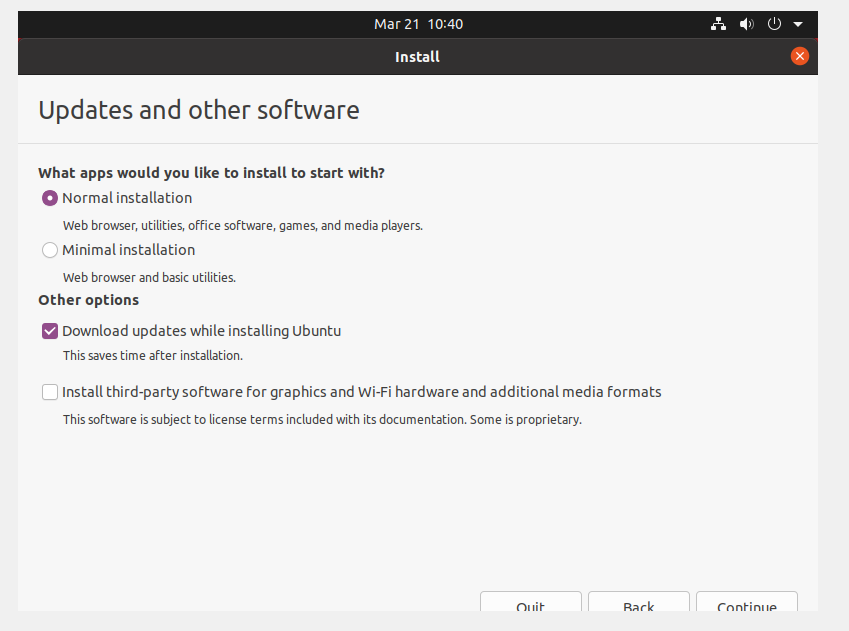
**Step 12:** Click Install Ubuntu.



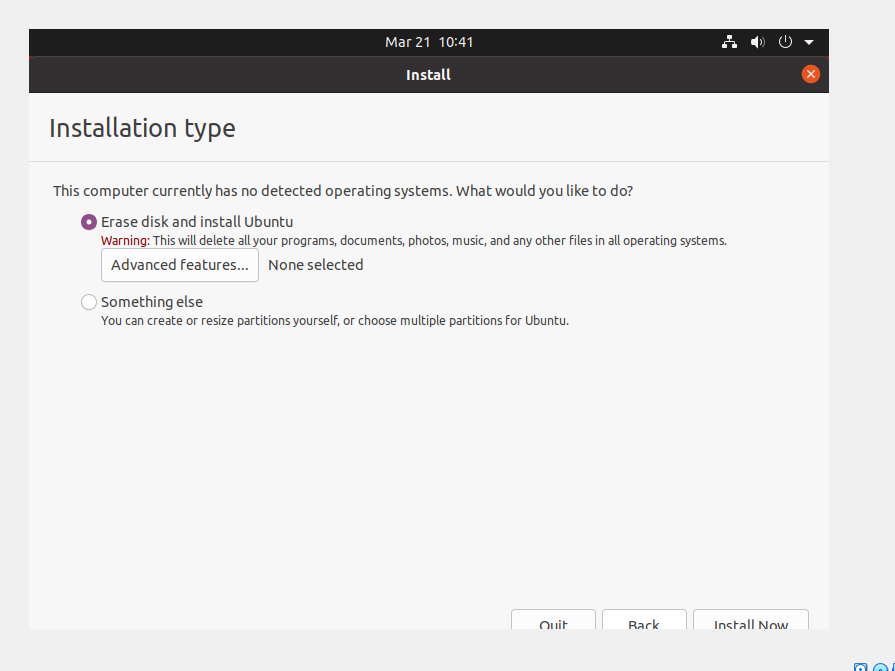
**Step 13:** Select your keyboard layout.



**Step 14:** In the "Updates and other software" section, check "Normal installation" and continue.

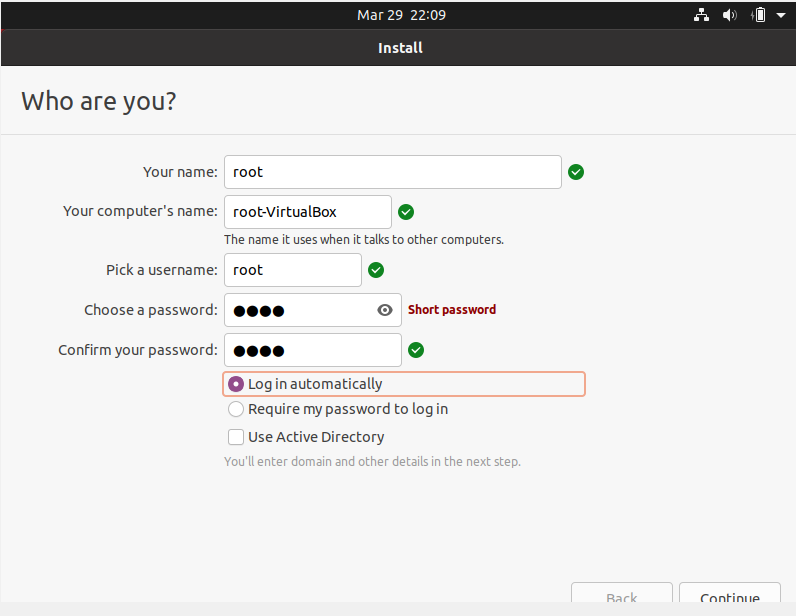


**Step 15:** Choose Erase disk and install Ubuntu and select Install Now, then select Continue to ignore the warning..



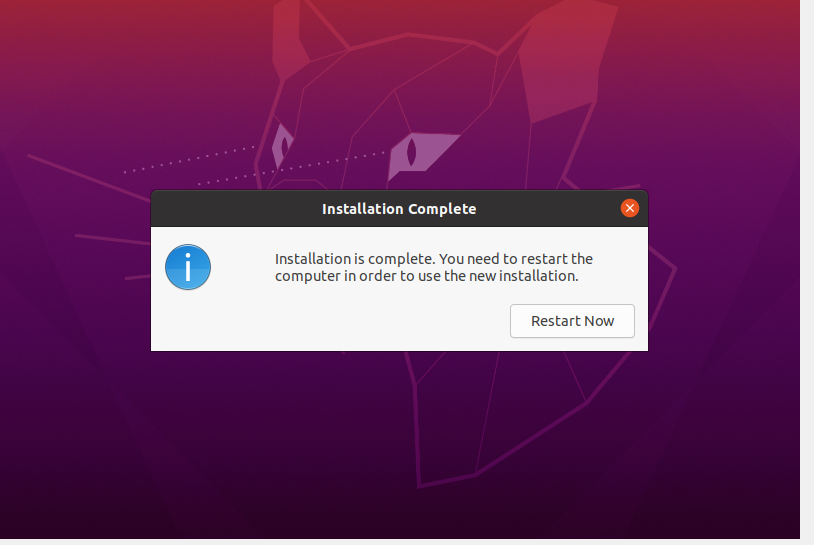
**Step 16:** Choose your time zone on the map, then select Continue.

**Step 17:** Set up your user account and select Continue.

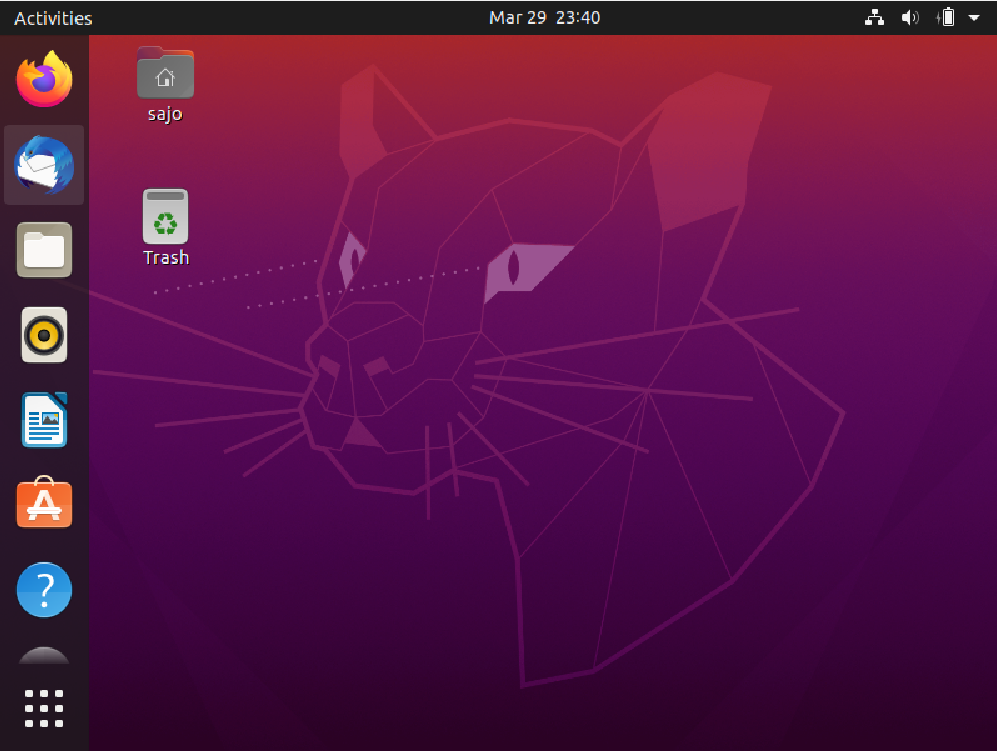


**Step 18:** Installation completed

**Step 19:** Restart your System

****

**Output Screenshort**

****

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 2**

**Aim**

Familiarization of Hardware Components in a Computer.

**Procedure**

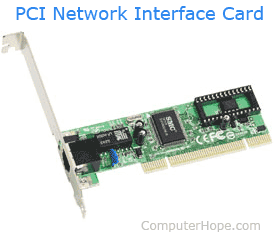
**Motherboard**

****

A motherboard provides connectivity between the hardware components of a computer, like the processor (CPU), memory (RAM), hard drive, and video card. There are multiple types of motherboards, designed to fit different types and sizes of computers.

Each type of motherboard is designed to work with specific types of processors and memory, so they don't work with every processor and type of memory. However, hard drives are mostly universal and work with the majority of motherboards, regardless of the type or brand.

**NIC (Network Interface Card)**

****

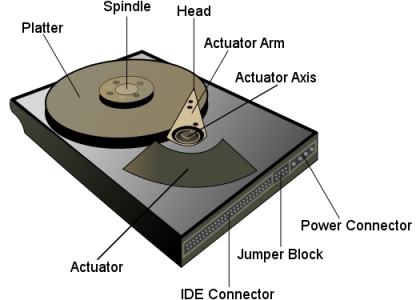
Short for network interface card, the NIC is also referred to as an Ethernet card and network adapter. A NIC is a computer expansion card for connecting to a network (e.g., home network or Internet) using an Ethernet cable with an RJ-45 connector.

**Random Access Memory**

****

Random access memory (RAM) is fast-access memory that is cleared when the computer is power-down. RAM attaches directly to the motherboard, and is used to store programs that are currently running. RAM is a set of integrated circuits that allow the stored data to be accessed in any order (why it is called random). There are many different types of RAM. Distinctions between these different types include: writable vs. read-only, static vs. dynamic, volatile vs. non-volatile, etc.

**Hard Disk Drive**

****

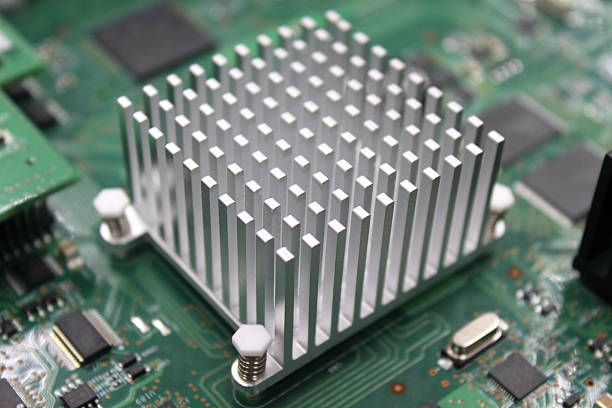
A hard disk drive (HDD) is a non-volatile storage device which stores digitally encoded data on rapidly rotating platters with magnetic surfaces. Just about every new computer comes with a hard disk these days unless it comes with a new solid-state drive. Typical desktop hard disk drives store between 120 and 400GB, rotate at 7,200 rpm, and have a media transfer rate of 1 Gbit/s or higher. Hard disk drives are accessed over one of a number of bus types, including parallel ATA(also called IDE), Serial ATA (SATA), SCSI, Serial Attached SCSI, and Fibre Channel.

**Processor**

The processor, also called the microprocessor or CPU (for Central Processing Unit), is the brain of the PC. It performs all general computing tasks and coordinates tasks done by memory, video, disk storage, and other system components. The CPU is a very complex chip that resides directly on the motherboard of most PCs, but may sometimes reside on a daughtercard that connects to the motherboard via a dedicated specialized slot.

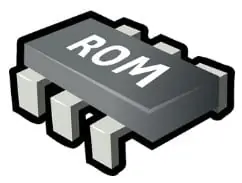
****

**Heat sink.**

****

This is a passive piece of hardware that draws heat away from components to regulate/reduce their temperature to help ensure they continue to function properly. Typically, a heat sink is installed directly atop the CPU, which produces the most heat among internal components.

**ROM Memory**

****

ROM stands for a type of memory chip that can be read from but not written to.

In other words, it's a form of data storage that can't be changed after being programmed.

It's sometimes called "non-volatile" memory because the stored information will remain even when not powered up or in use.

ROM is often used to store a computer's basic start-up instructions and certain types of data, such as your car's onboard computer system and a calculator's data tables.

**Optical Drive**

****

Optical Drives are used in PCs to read and write CDs and DVDs.

The optical drive reads the data from the disc, which can then be transformed into a digital file that is readable by the computer.

This makes it easy to backup files, play music or movies, or copy data from one disc to another.

The term "CD" refers to Compact Discs, which are the most common type of optical drive on modern computers.

**Power Supply**

****

A power supply is an electrical appliance that provides the necessary power to operate a computer.

Computers are powered by electricity, and the power supply converts the alternating current (AC) from the electric outlet into direct current (DC).

The power supply in a computer can be an internal or external component.

It’s important to make sure your power supply is functioning properly.

**Graphics Processing Unit (GPU)**

****

The graphics processing unit, or GPU, has become one of the most important types of computing technology, both for personal and business computing. Designed for parallel processing, the GPU is used in a wide range of applications, including graphics and video rendering. Although they’re best known for their capabilities in gaming, GPUs are becoming more popular for use in creative production and artificial intelligence (AI).

**Solid State Drive (SSD)**

****

A solid-state drive (SSD) is a new generation of storage device used in computers. SSDs use flash-based memory, which is much faster than a traditional mechanical hard disk. Upgrading to an SSD is one of the best ways to speed up your computer. Learn how SSDs work and how to keep them optimized with a specialized performance-boosting tool.

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 3**

**Aim**

Familiarization of the Linux commands.

**Procedure**

**1. pwd**

This command is used to display the location of the current working directory.

**Syntax :-** $ pwd

**Output :-**



**2. mkdir**

This command is used to create a new directory under any directory.

**Syntax :-** $ mkdir <directory name>

**Output :-**



**3. ls**

This command is used to display a list of content of directory.

**Syntax :-** $ ls

**Output :-**

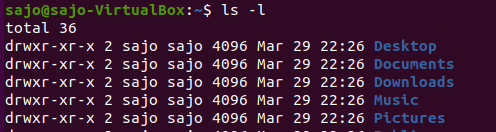


**5. ls –l**

This command is used to shows file or directory, size, modified date and time, file or folder name and owner of the file, and its permission.

**Syntax :-** $ ls –l

**Output:-**

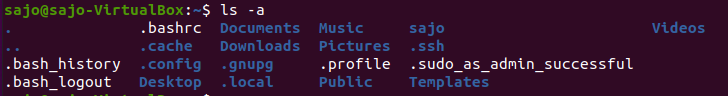


**6. ls –a**

This command is used to list all files including hidden files.

**Syntax :-** $ls –a

**Output :-**

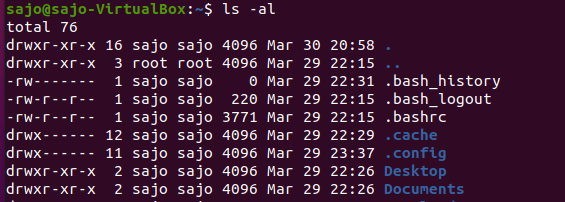


**7. ls –al**

This command is used to

**Syntax :-** $ ls -al

**Output :-**



**8. ls –t**

This command is used to display files in the last modified order.

**Syntax :-** $ ls –t

**Output :-**



**9. cd**

This command is used to change the current directory.

**Syntax :-** $ cd <directory name>

**Output :-**



**10. cd ..**

This command is used to move to the parent directory of current directory, or the directory one level up from the current directory.

**Syntax :-** $ cd ..

**Output :-**



**11. cd –**

This command is used to switch back to previous directory we were working earlier.

**Syntax :-** $ cd –

**Output :-**

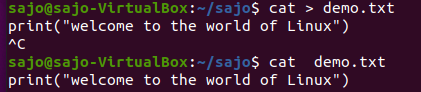


**12. cat > filename**

This command is used to create a file and add contents to that file.

**Syntax :-** $ cat > filename.txt

**Output :-**

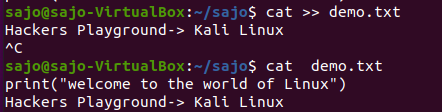


**13. cat>>filename**

This command is used to add contents to an existing file.

**Syntax :-** $ cat >> filename.txt

**Output :-**

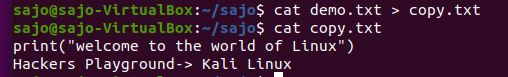


**14. cat filename1 > filename2**

This command is used to copy the content from one file to another file.

**Syntax :-** $ cat filename1 > filename2

**Output :-**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 4**

**Aim**

Familiarization of the Linux commands.

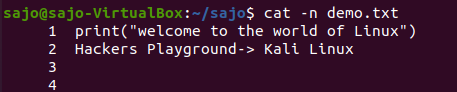
**Procedure**

**1. cat -n**

This command return contents with line numbers.

**Syntax**:- $ cat -n filename

**Output**:-

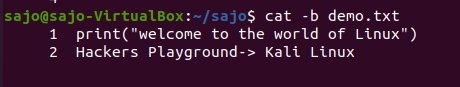


**2. cat -b**

This Returns contents with line numbers but excludes the empty lines

**Syntax**:- $ cat -b filename

**Output**:-



**3. touch**

This command is used to create a empty file

**Syntax**:- $ touch filename

**Output**:-



**4. echo command >> filename**

This command helps to add lines on the empty file

**Syntax**:- $ echo command>> filename

**Output**:-



**5. head**

This command returns first 10 lines in the file

**Syntax**:- $ head filename

**Output**:-

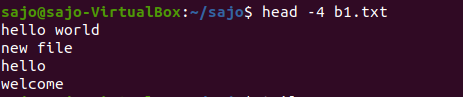


**6. head – (No. of lines)**

This will return beginning lines specified in the command

**Syntax**:- $ head -4 filename

**Output**:-

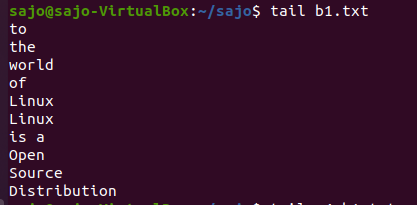


**7. tail**

This returns the last 10 lines in the file

**Syntax** $ tail filename

**Output**:-

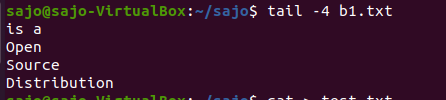


**8. tail – (No. of lines)**

This will return last no of lines specified in the command

**Syntax**:- $ tail -4 filename

**Output**:-

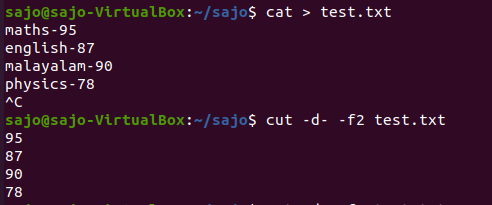


**9. cut -d -f2**

This commands in Linux allows you to select the second part of the content that split by ‘ - ‘ delimiter.

**Syntax**:- $ cut -d- f2 filename

**Output**:-

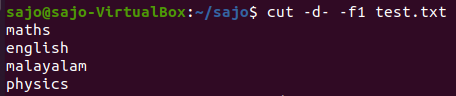


**10. cut -d -f1**

This commands in Linux allows you to select the First part of the content that split by ‘ - ‘ delimiter.

**Syntax**:- $ cut -d- f1 filename

**Output**:



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 5**

**Aim**

Familiarization of the Linux commands.

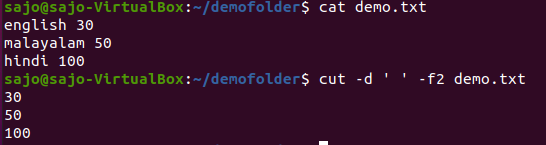
**Procedure**

**1. cut -d ‘ ‘ f2**

This commands in Linux allows you to select the second part of the content that split by ‘ ‘ delimiter.

**Syntax:-** $ cut -d ‘ ‘ f2 filename.txt

**Output:-**

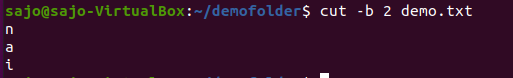


**2. cut -b 2**

This command in Linux allows to select only these bytes.

**Syntax:-** $ cut -b 2 filename

**Output:-**

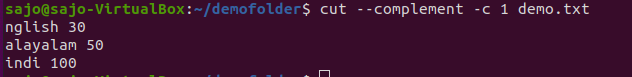


**3.** **cut –complement**

This command helps to exclude given bytes.

**Syntax:-** $ cut –complement -c 1 filename

**Output:-**

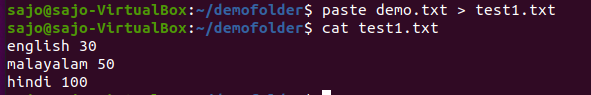


**4. paste**

This command helps to paste contents to other file

**Syntax:-** $ paste filename > filename2

**Output:-**

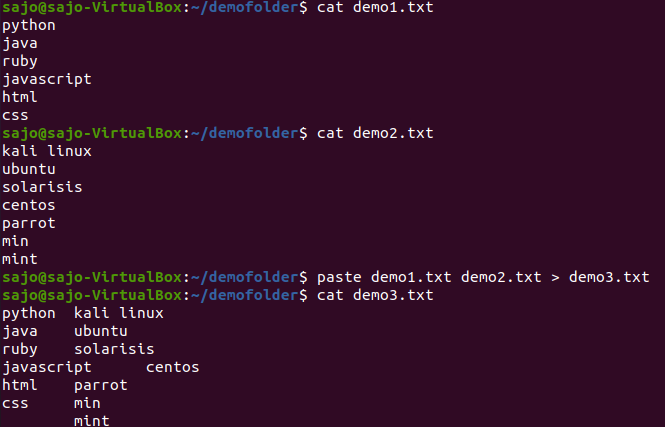


**5. paste a b > c**

This command helps to paste from multiple files.

**Syntax:-** $ paste file1 file2 > file3

**Output:-**

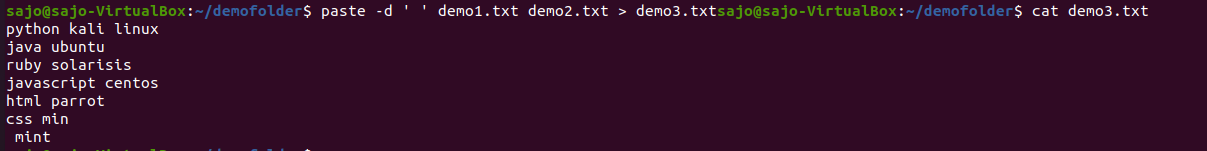


**6. paste -d ‘ ‘**

This command helps to paste contents with ‘ ‘ delimiter.

**Syntax:-** $ paste -d ‘ ‘ file1 file 2 > file3

**Output:-**

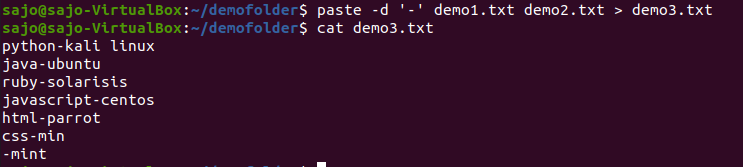


**7. paste -d ‘-‘**

This command helps to contents from multiple files with ‘-‘ delimiter.

**Syntax:-** $ paste -d ‘-‘ file1 file2 > file3

**Output:-**

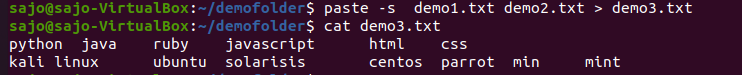
****

**8. paste -s**

This command helps to paste on file at a time instead of in parallel.

**Syntax:-** $ paste -s file1 file2 > file3.

**Output:-**



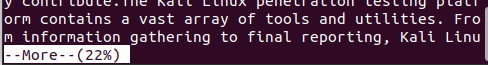
**9. more**

More command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large.

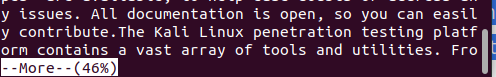
**Syntax:-** $ more filename.

**Output:-**

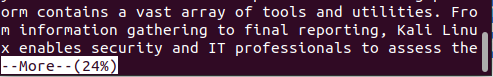
****

****

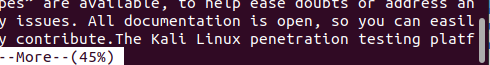
**SPACE Key** is act as a NEXT button.



**B Key** act as a PREVIOUS button.



**ENTER** **key** is used to view line by line.

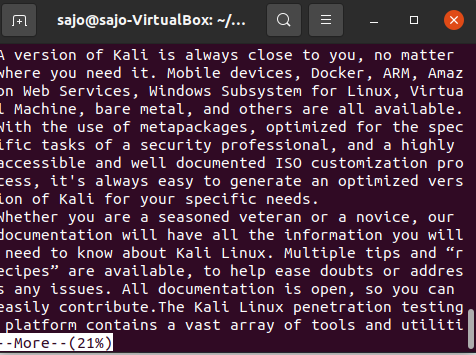


**10. more -s**

This command is used to squeeze multiple blank lines into one.

**Syntax:-** $ more -s filename.

**Output:-**

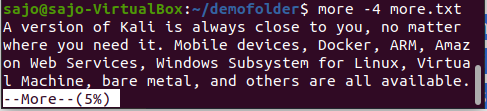


**11. more –(no.)**

This command helps to view content based on specified number of lines.

**Syntax:-** $ more -4 filename.

**Output:-**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 6**

**Aim**

Familiarization of the Linux commands.

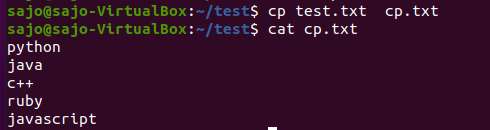
**Procedure**

1. **cp**

cp command used to copy .

**Syntax:-** .$cp filename1.txt filename2.txt

**output:**

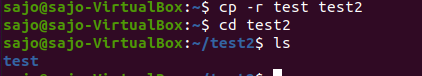
****

1. **cp -r**

cp -r command is used to copy entire directory.

**Syntax:-** $cp -r directory1 directory2

**output:**

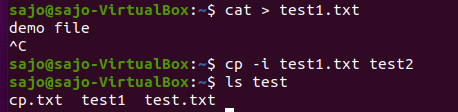
****

1. **$cp -i**

this command is used to ask permission to overwrite

**Syntax:-** $cp -i filename.txt directory

**output :**

****

1. **$mv**

mv command is used to move a file to another directory

**Syntax:-** .$mv filename.txt directory

**output:**

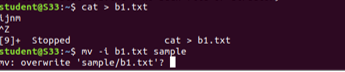
****

1. **$mv -i**

this command is used to ask permission to move one directory to anotherdirectory .

**Syntax:-** .$mv -i filename.txt directory

**output:**

****

**Name: Sajo sam**

**Roll No:32**

**Batch: MCA -B**

**Date:21-04-2022**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 7**

**Aim**

Familarization of linux commands

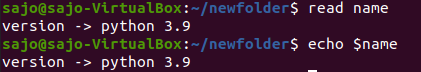
**Procedure**

1. **read**

The Linux **read** command is used to read the contents of a line into a variable.

**Syntax:-** $read variable\_name

**Output:**-

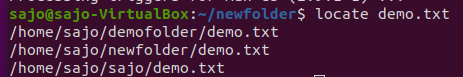


1. **locate**

The locate command and [find](https://www.javatpoint.com/linux-find)command is used to search a file by name. But, the difference between both commands is that locate command is a background process and searches the file in the database whereas, find command searches in the filesystem. The locate command is much faster than find command.

**Syntax:-** $locate filename.txt

**Output:-**

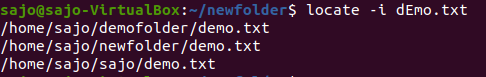
****

1. **locate -i**

It is used to ignore case sensitivity of the specified patterns.

**Syntax :-** $locate -i filename.txt

**Output:**-

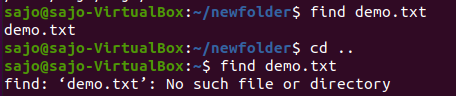
****

1. **find**

The find command helps us to find a particular file within a directory. It is used to find the list of files for the various conditions like permission, user ownership, modification, date/time, size, and more.

**Syntax :-** $find filename.txt

**Output:-**

****

1. **grep**

The 'grep' command stands for **"global regular expression print"**. grep command filters the content of a file which makes our search easy.

**Syntax:-** $grep word filename.txt

**Output:-**

****

1. **grep -i**

The 'grep -i' command filters output in a case-insensitive way.

**Syntax:-** $grep -i word filename.txt

**Output:-**

****

1. **grep -v**

The 'grep -v' command displays lines not matching to the specified word.

**Syntax:-** $grep -v word filename.txt

**Output:-**

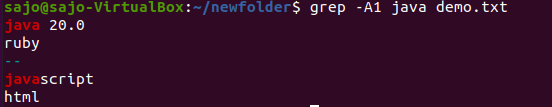
****

1. **grep -A**

grep -A command is used to display the **line after the result**.

**Syntax:-** $grep -A word file.txt

**Output:-**

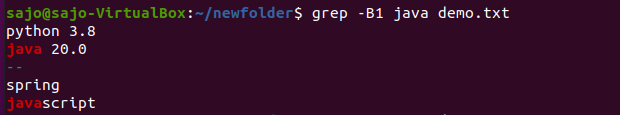


1. **grep -B**

grep -B command is used to display the **line before the result**.

**Syntax:-** $grep -B word file.txt

**Output:-**

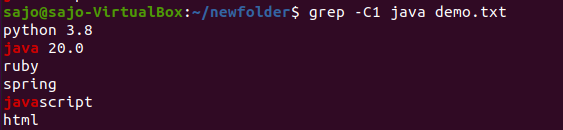


1. **grep -C**

grep -C command is used to display the **line after and line before** the result.

**Syntax:-** $grep -C word file.txt

**Output:-**



1. **df**

Linux df command is used to display the **disk space used in the file system**. The 'df' stands for **"disk filesystem**." It defines the number of blocks used, the number of blocks available, and the directory where the file system is mounted.

**Syntax:-** $df

**Output:-**

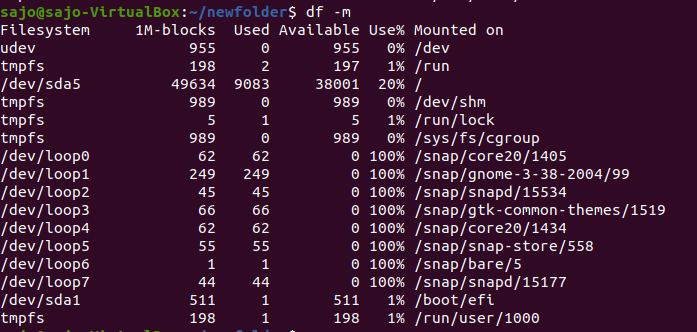
****

1. **df -m**

it is used to see the report in mega byte

**Syntax:-** $df -m

**Output:-**

****

1. **du**

To check how much space a file or directory take.

**Syntax:-** $du

**Output:-**

****

1. **wc**

Linux wc command helps in counting the lines, words, and characters in a file. It displays the number of lines, number of characters, and the number of words in a file. Mostly, it is used with pipes for counting operation.

**Syntax :-** $wc filename.txt

**Output:-**

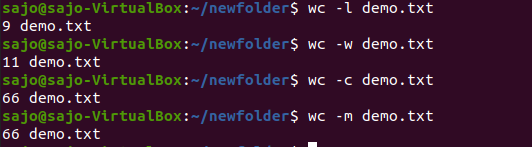
****

**-l, --lines:** It is used to print the newline counts.

**-w, --words:** It is used to print the word counts.

**-c, --bytes:** It is used to print the byte counts.

**-m, --chars:** It is used to print the character counts.

****

**Name: Sajo Sam**

**Roll No: 32**

**Batch: MCA -B**

**Date:25-04-2022**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 8**

**Aim**

Familarization of linux commands

**Procedure**

1. **useradd & passwd**

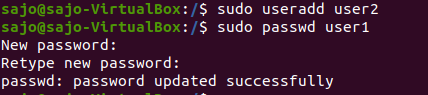
useradd is a command in Linux that is used to add user accounts to your system

passwd command in Linux is used to change the user account passwords. The root user reserves the privilege to change the password for any user on the system, while a normal user can only change the account password for his or her own account.

**Syntax:-** $ sudo useradd user1

$ sudo passwd user1

**Output:-**





1. **groupadd**

groupaddcommand is used to create a new user group.

**Syntax:-** groupadd [option] group\_name

**Output:-**





1. **usermod**

usermod command or modify user is a command in Linux that is used to change the properties of a user in Linux through the command line

**Syntax:-** $sudo usermod [option] groupname username

**Output:-**

****

****

1. **compgen**

compgen is a bash built-in command which is used to list all the commands that could be executed in the Linux system. This command could also be used to count the total number of commands present in the terminal or even to look for a command with the specific keyword.

**Syntax:-** $compgen -g

**Output:-**

****

1. **userdel**

userdel command in Linux system is used to delete a user account and related files. This command basically modifies the system account files, deleting all the entries which refer to the username LOGIN. It is a low-level utility for removing the users.

**Syntax:-** sudo userdel username

**Output:-**



1. **groupdel**

*groupdel*command is used to delete a existing group. It will delete all entry that refers to the group, modifies the system account files, and it is handled by superuser or root user.

**Syntax:-** $sudo groupdel groupname

**Output:-**

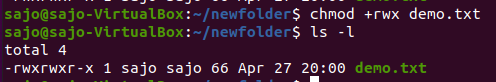


1. **chmod**

the chmod command is used to change the access mode of a file.  
The name is an abbreviation of change mode.

**Syntax:-** $chmod permission filename

**Output:-**

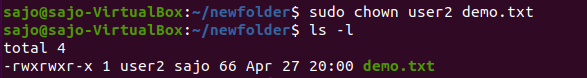


1. **chown**

chown command is used to change the file Owner or group. Whenever you want to change ownership you can use chown command.

**Syntax:-** sudo chown username filename

**Output:-**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 9**

**Aim**

Shell program to print a value

**Procedure**

#!/bin/bash

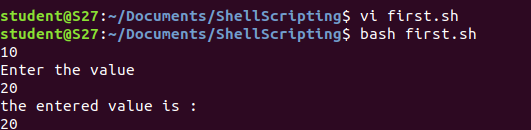
echo "10"

echo “enter the value”

read a

echo “the entered value is : “ $a

**Output**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 10**

**Aim**

Shell program to add 2 numbers

**Procedure**

#!/bin/bash

echo "enter the value 1"

read a

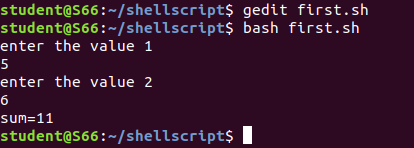
echo "enter the value 2"

read b

d=$(($a + $b))

echo "sum="$d

**Output**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 11**

**Aim**

Shell program to check largest of 2 numbers

**Procedure**

#!/bin/bash

echo "enter 2 numbers"

read a

read b

echo $a $b

if [ $a -gt $b ]

then

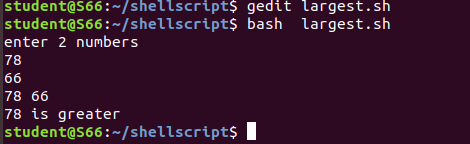
echo "$a is greater"

else

echo "$b is greater"

fi

**Output**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 12**

**Aim**

Shell program to check largest of 3 numbers

**Procedure**

#!/bin/bash

echo "enter 3 numbers"

read a

read b

read c

if [ $a -gt $b  -a  $a -gt $c ]

then

echo "$a is greater"

elif [ $b -gt $a  -a  $b -gt $c ]

then

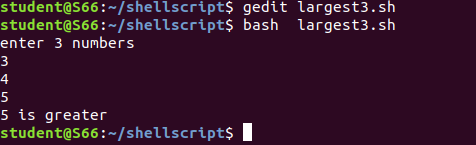
echo "$b is greater"

else

echo "$c is greater"

fi

**Output**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 13**

**Aim**

Shell program to check given number is odd or even

**Procedure**

#!/bin/bash

echo "enter the value"

read a

if [ $(($a % 2)) -eq 0 ]

then

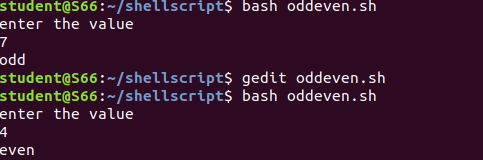
echo "even"

else

echo "odd"

fi

**Output**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 14**

**Aim**

Write a shellscript to display current date and calendar

**Procedure**

#!/bin/bash

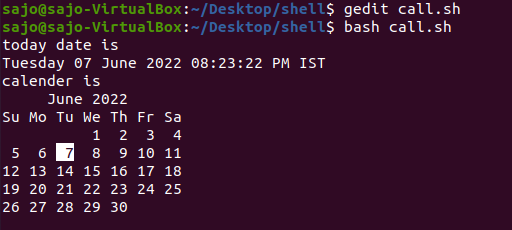
echo " Todays date="

echo $(date)

echo "Calender="

cal

**Output**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 15**

**Aim**

Write a shellscript to check a number is greater than or less than or equal to another number

**Procedure**

#!/bin/bash

echo "enter the two numbers"

read a

read b

if [ $a -gt  $b ]

then

echo " $a is greater than $b"

elif [ $a -lt  $b ]

then

echo "$a is less than $b"

elif [ $a -eq  $b ]

then

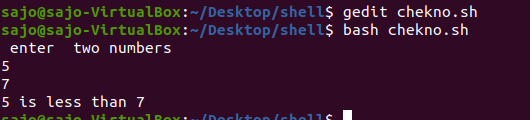
echo "$a is equal to $b "

else

echo "invalid"

fi

**Output**

****

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 16**

**Aim**

Find the sum of first 10  numbers

**Procedure**

#!/bin/bash

i=1

summ=0

while [ $i -le 10 ]

 do

   summ=$((summ+i))

    i=$((i+1))

 done

echo "sum of first 10 numbers = $summ"

**Output Screenshot**

****

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 17**

**Aim**

Write a shellscript to find the sum ,average and product of 4 numbers

**Procedure**

#!/bin/bash

echo "enter 4 numbers"

read a

read b

read c

read d

sum1=$((a+b+c+d))

p=`expr $a \\* $b \\* $c \\* $d`

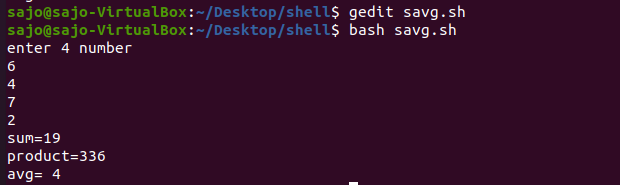
avg=$((sum1/4| bc -l))

echo "Sum=$sum1"

echo "product= $p"

echo "average=$avg"

**Output Screenshot**

****

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 18**

**Aim**

Find the factorial of the given number

**Procedure**

#!/bin/bash

fact=1

echo "enter the number"

read n

for (( i=2 ; i<=n ; i++ ))

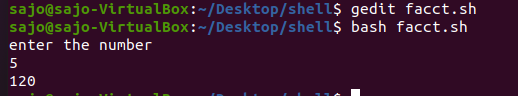
do

fact=`expr $fact \\* $i`

done

echo "$n != $fact"

**Output Screenshot**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 19**

**Aim**

Find the palindrome of a given number

**Procedure**

#!/bin/bash

echo "enter the number"

read n

a=$n

rev=0

while [ $n -gt 0 ]

do

r=$((n%10))

rev=`expr $rev \\* 10 + $r`

n=`expr $n / 10 `

done

if [ $a -eq  $rev ]

then

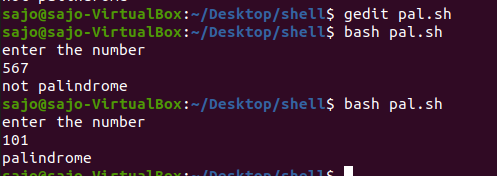
echo "palindrome"

else

echo "not palindrome"

fi

**Output Screenshot**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 20**

**Aim**

Check whether the given year is leap year or not

**Procedure**

#!/bin/bash

echo "enter the year"

read y

if [ $((y%4)) == 0 ]  &&  [  $((y%100)) != 0 ]  ||  [  $((y%400)) == 0  ]

 then

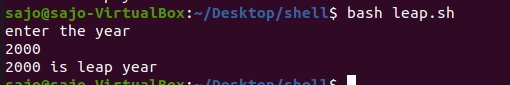
 echo "$y is a leap year"

 else

 echo "$y is not a leap year"

 fi

**Output Screenshot**

****

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 21**

**Aim**

Write a shellscript program to find the sum of all the digits in a number

**Procedure**

#!/bin/bash

echo "Enter the number"

read a

while [ $a -ne 0 ]

do

r=$((a%10))

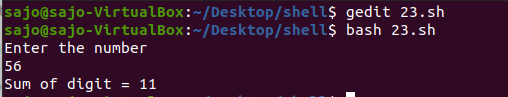
s=$((s+r))

a=$((a/10))

done

echo "Sum of digit = "$s

**Output Screenshot**



**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 22**

**Aim**

Write a shellscript program to find the sum  and average of numbers in command line

**Procedure**

#!/bin/bash

i=0

echo "enter the size"

read n

echo "enter the numbers"

while [ $i -lt $n ]

do

 read a

sum=$((sum+a))

avg=$((sum/n))

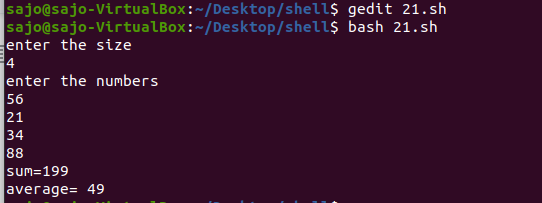
i=$((i+1))

done

echo "sum=$sum"

echo "average= $avg"

**Output Screenshot**

****

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 23**

**Aim**

Write a shellscript program to  do mathematical calculations using switch case

**Procedure**

#!/bin/bash

echo "enter two number"

read a

read b

echo " 1.Addition "

echo " 2.Substraction "

echo " 3.Multiplication  "

echo "4.Division"

echo "5.Exit"

s=1

while [ $s == 1 ]

do

echo "Enter the choice"

read ch

case $ch in

1)

echo "sum= "$((a+b))

;;

2)

echo "Substraction="$((a-b))

;;

3)

echo "Multiplication="$((a\*b))

;;

4)

echo "divison="$((a/b))

;;

5)exit

;;

\*)

echo "invalid"

;;

esac

done

**Output Screenshot**

****

**Name:**

**Roll No:**

**Batch:**

**Date:**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 24**

**Aim**

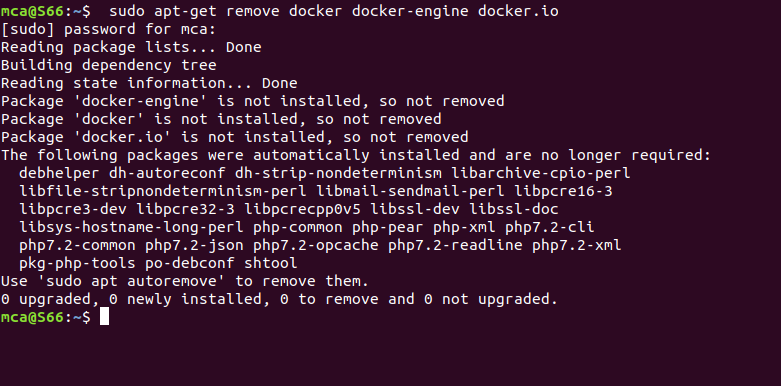
Installation of Docker on Ubuntu

**Procedure**

step 1: Open the terminal on Ubuntu.

step 2: Remove any [Docker files](https://www.simplilearn.com/tutorials/docker-tutorial/what-is-dockerfile) that are running in the system, using the following command:

**$ sudo apt-get remove docker docker-engine docker.io**



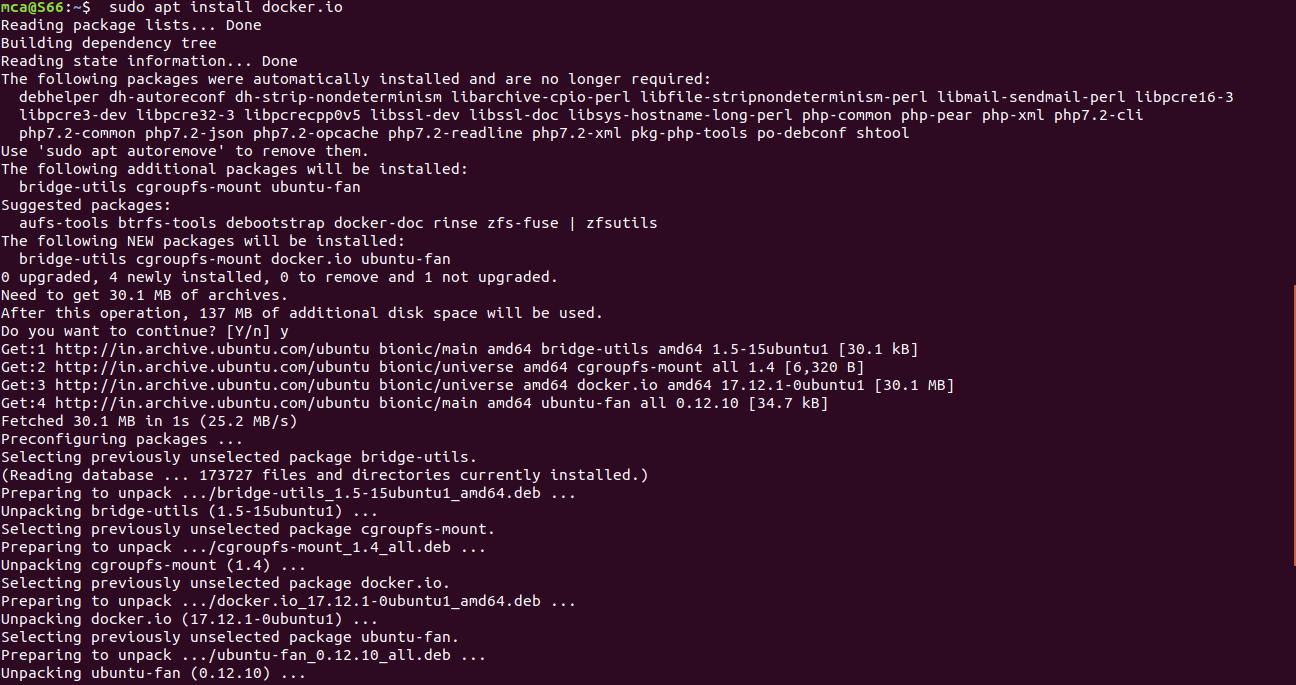
step 3: Check if the system is up-to-date using the following command:

**$ sudo apt-get update**



Step 4: Install Docker using the following command:

**$ sudo apt install docker.io**



Step 5: Install all the dependency packages using the following command:

**$ sudo snap install docker**



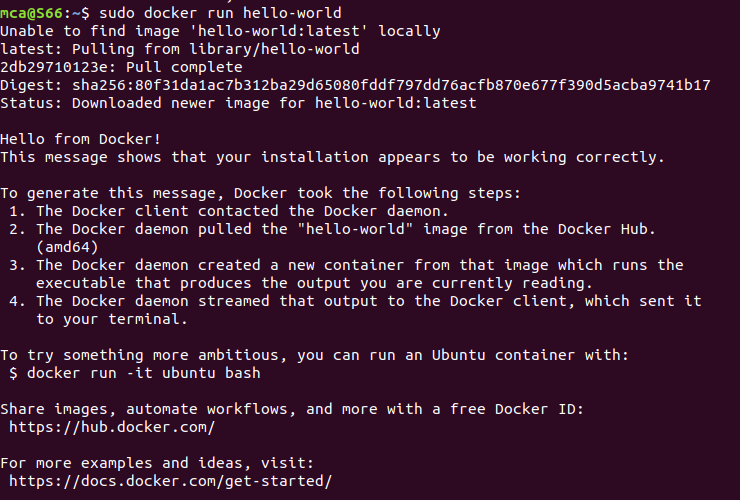
Step 6: Before testing Docker, check the version installed using the following command:

**$ docker --version**



Step 7: Pull an image from the Docker hub using the following command:

**$ sudo docker run hello-world**



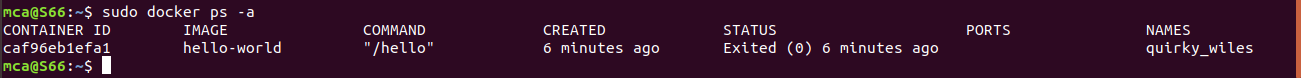
Step 8: Check if the docker image has been pulled and is present in your system using the following command:

**$ sudo docker images**



Step 9: To display all the containers pulled, use the following command:

**$ sudo docker ps -a**

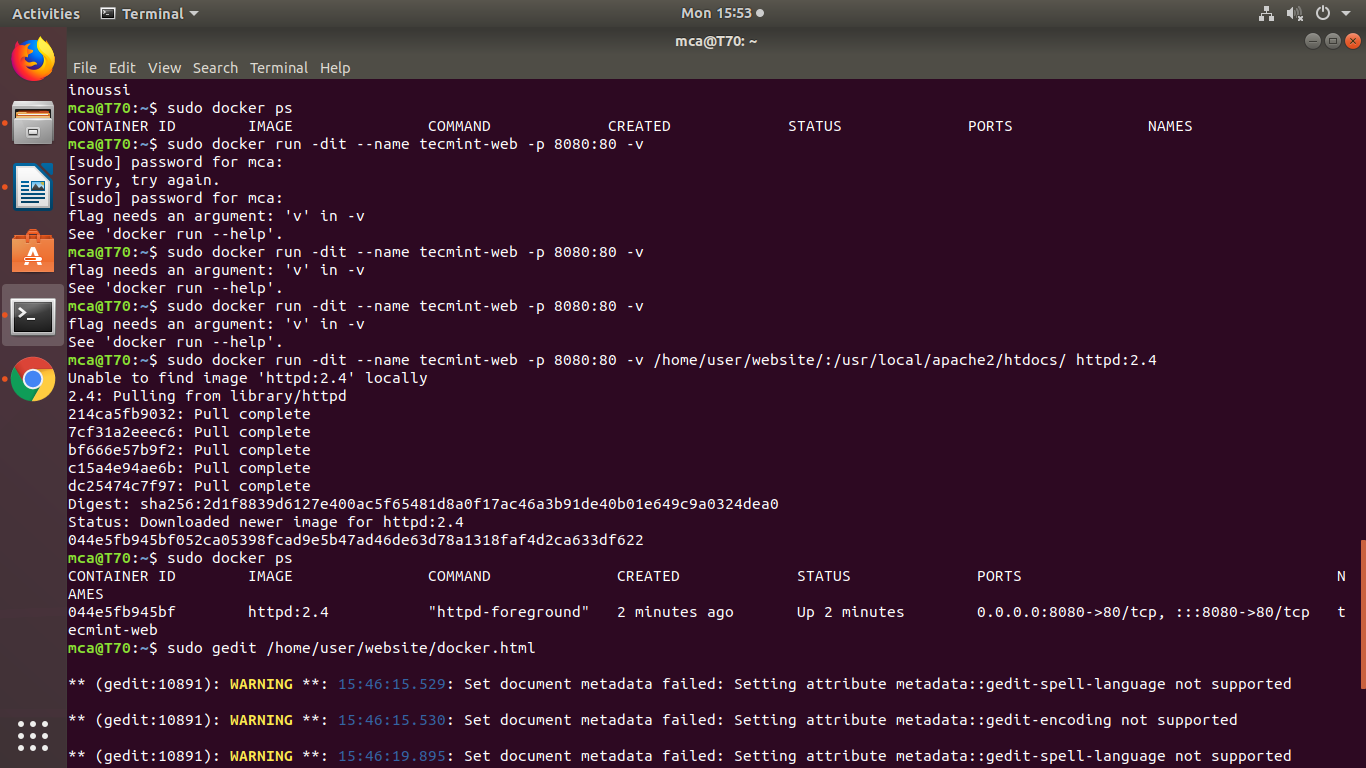


Step 10: To check for containers in a running state, use the following command:

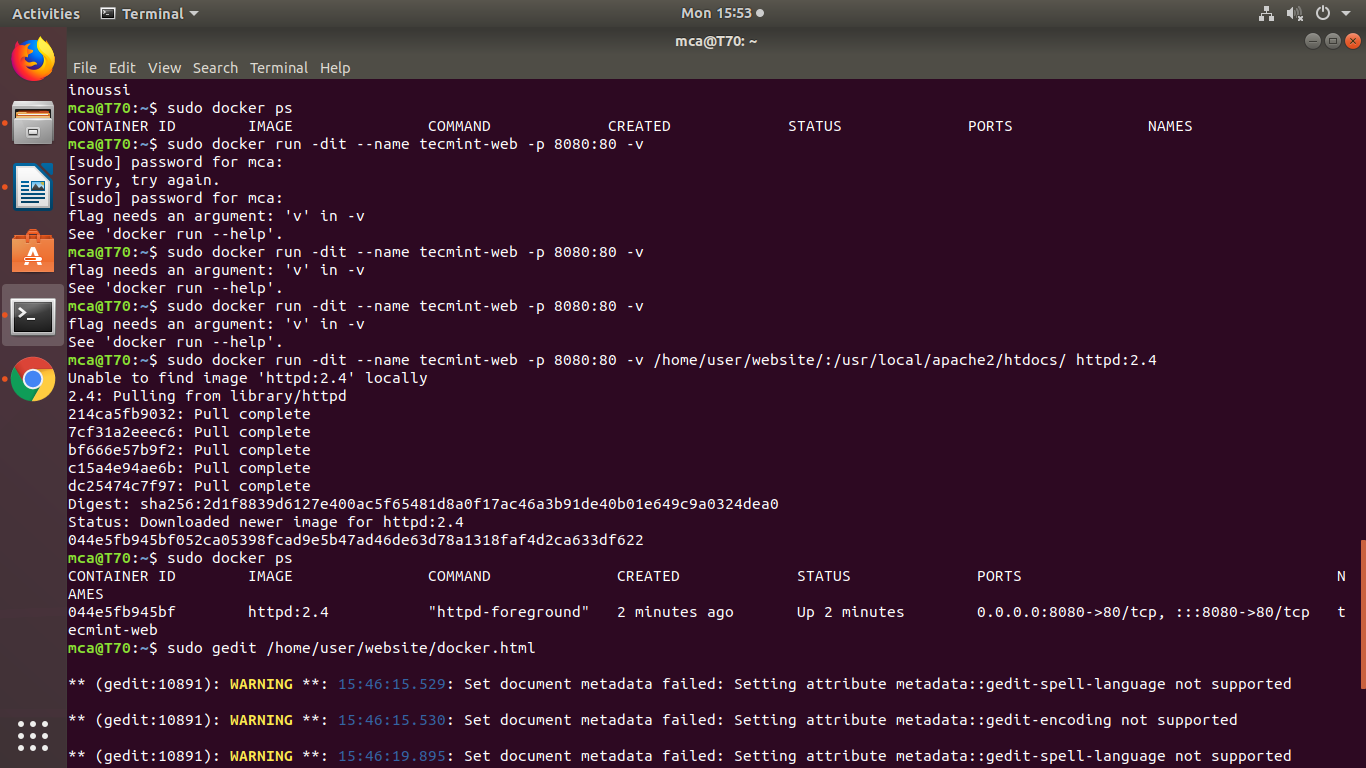
**$ sudo docker ps**



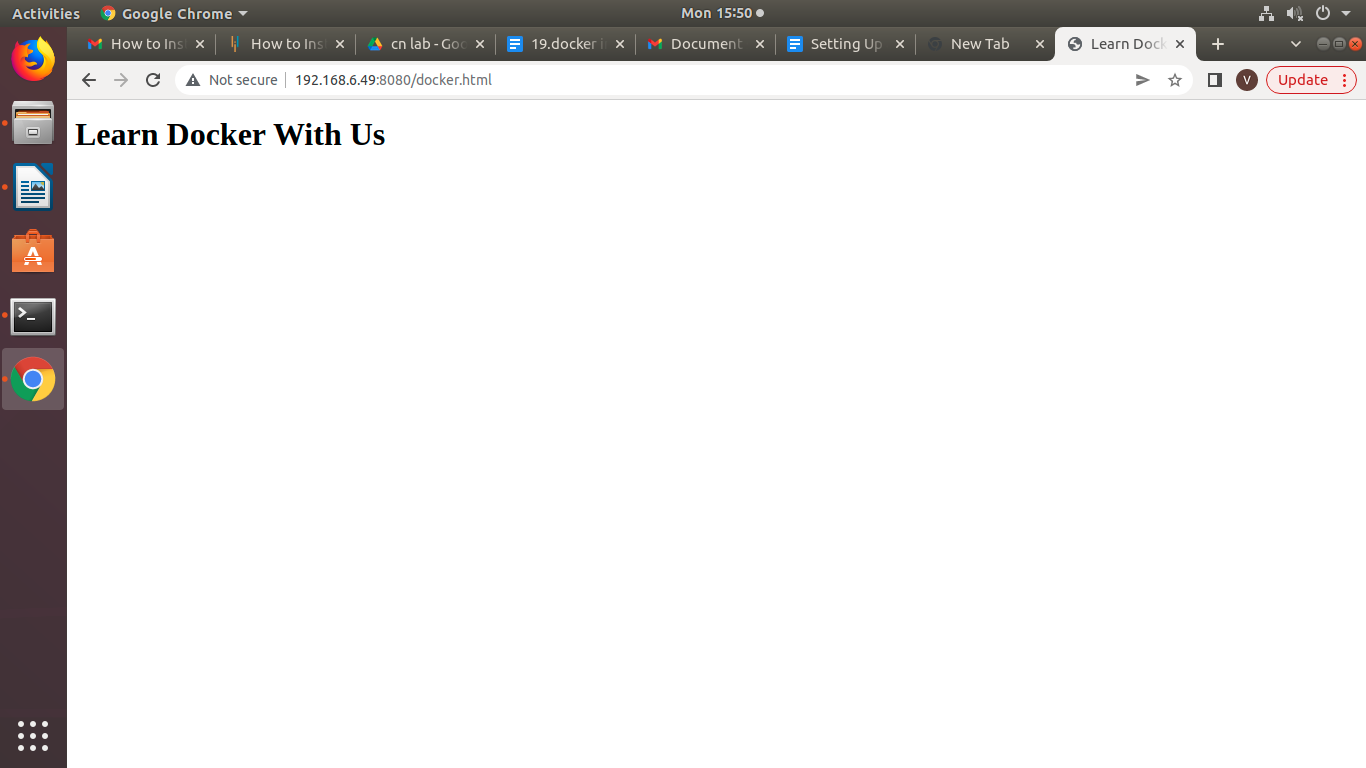
 step 11: **sudo docker run -dit --name tecmint-web -p 8080:80 -v** /home/user/website/:/usr/local/apache2/htdocs/ httpd:2.4



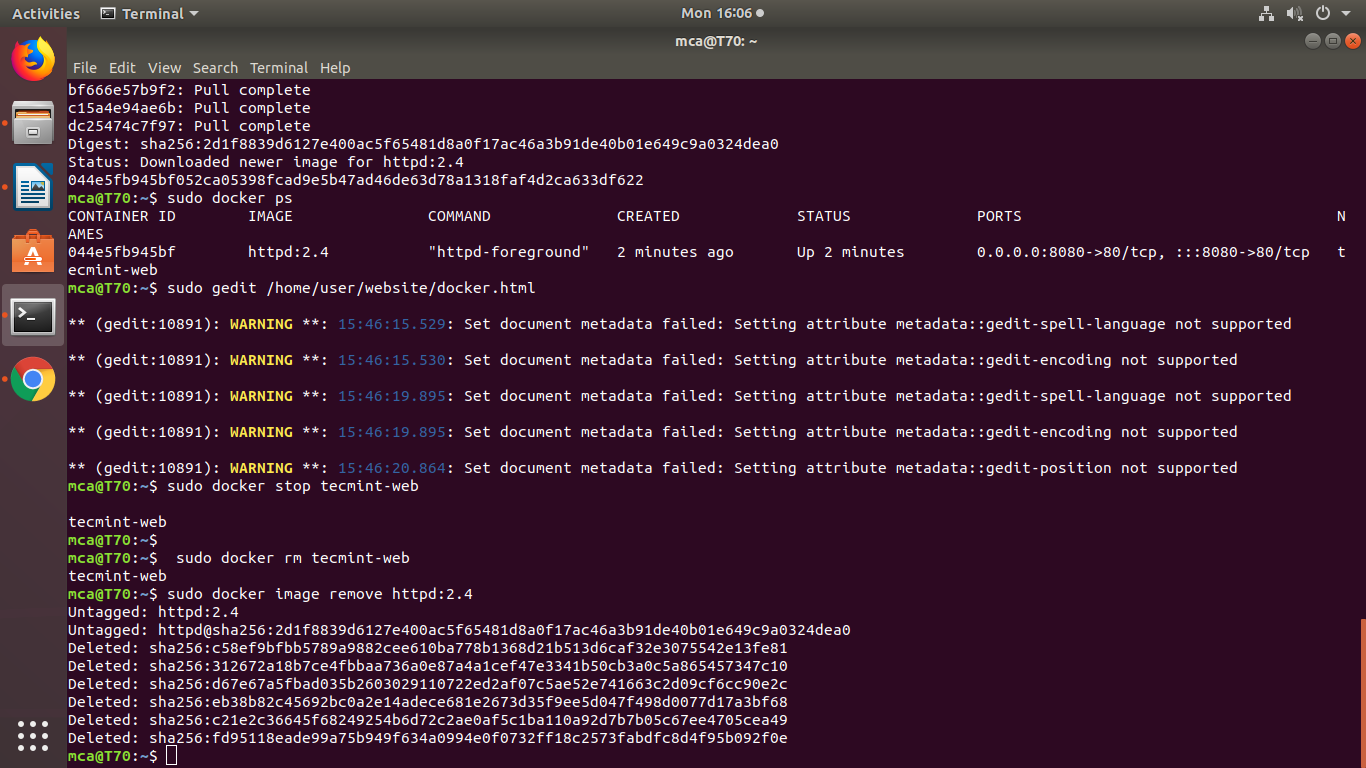
step 12:  **$ sudo docker ps**



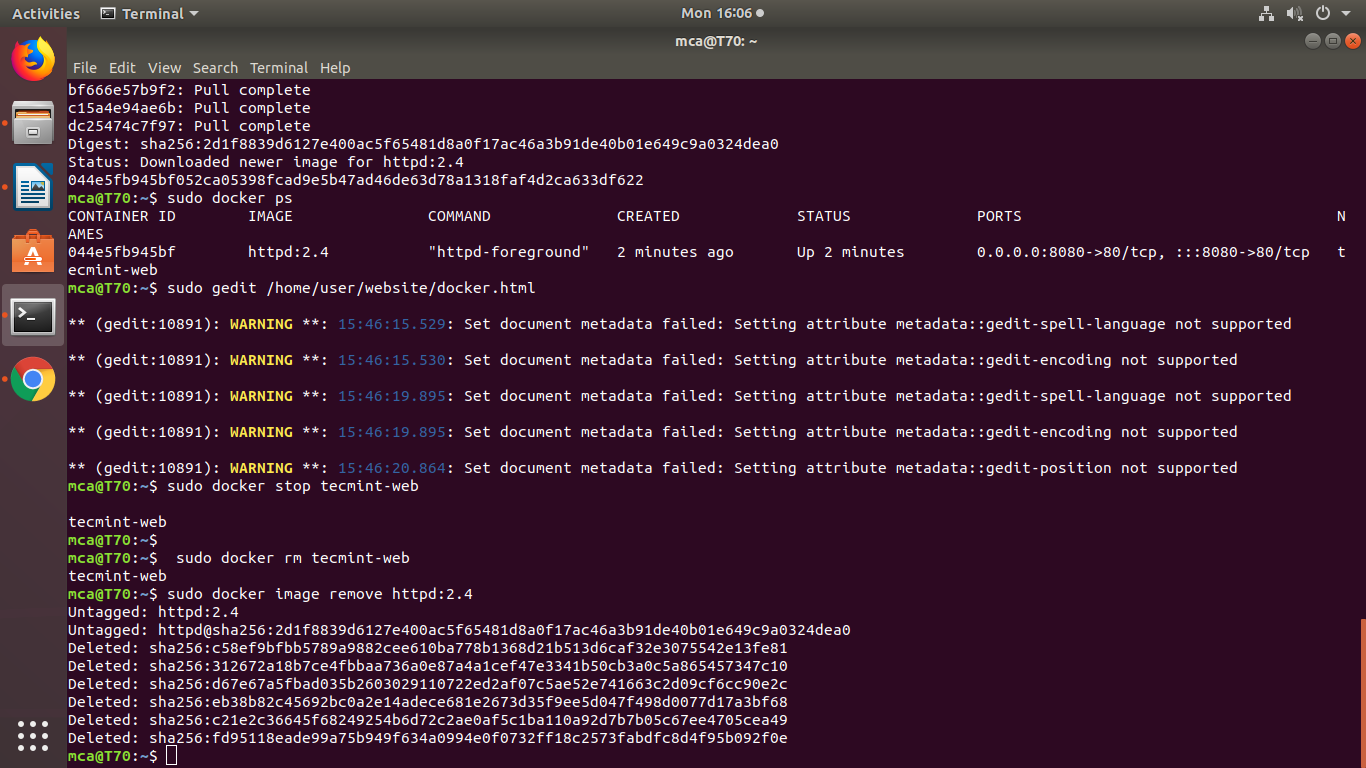
step 13 : **$ sudo gedit  /home/user/website/docker.html**



 step 14 : **$ sudo docker stop tecmint-web**



 step 15: **$ sudo docker rm tecmint-web**



step 16: **$ sudo docker image remove httpd:2.4**

