

20MCA136 – NETWORKING & SYSTEM ADMINISTRATION LAB

Lab Report Submitted By

JENNY JOHNSON

Reg. No.: AJC22MCA-2053

In Partial Fulfilment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS (2 Year) (MCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC with 'A' grade. Koovapally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2023

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING KANJIRAPPALLY



CERTIFICATE

This is to certify that the lab report, “**20MCA136 NETWORKING & SYSTEM ADMINISTRATION LAB**” is the bonafide work of **JENNY JOHNSON (AJC22MCA-2053)** in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year **2022-23**.

Mrs. Ankitha Philip

Lab In- Charge

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

Internal Examiner

External Examiner

Course Code	Course Name	Syllabus Year	L-T-P-C
20MCA136	Networking & System Administration Lab	2020	0-1-3-2

VISION

To promote an academic and research environment conducive for innovation centric technical education.

MISSION

- MS1 - Provide foundations and advanced technical education in both theoretical and applied Computer Applications in-line with Industry demands.
- MS2 - Create highly skilled computer professionals capable of designing and innovating real life solutions.
- MS3 - Sustain an academic environment conducive to research and teaching focused to generate up-skilled professionals with ethical values.
- MS4 - Promote entrepreneurial initiatives and innovations capable of bridging and contributing with sustainable, socially relevant technology solutions.

COURSE OUTCOME

CO	Outcome	Target
CO1	Install and configure common operating systems in virtual environment.	60
CO2	Perform system administration tasks including network configurations, user creations and trouble shooting.	65
CO3	Install and manage servers for web applications.	60
CO4	Write shell scripts required for system administration.	60
CO5	Acquire skill sets required for a DevOps.	60

COURSE END SURVEY

CO	Survey Question	Answer Format
CO1	At what extend you are able to install and configure common operating systems.	Excellent/Very Good/Good Satisfactory/Needs improvement
CO2	At what extend you are able to perform system administration tasks.	Excellent/Very Good/Good Satisfactory/Needs improvement
CO3	At what extend you are able to install and manage servers for web applications.	Excellent/Very Good/Good Satisfactory/Needs improvement
CO4	At what extend you are able to write shell scripts required for system administration.	Excellent/Very Good/Good Satisfactory/Needs improvement
CO5	At what extend you are able to acquire skill sets required for a DevOps.	Excellent/Very Good/Good Satisfactory/Needs improvement

CONTENT

Sl. No.	Experiment	Date	CO	Page No.
1	Introduction to Computer hardware: Physical identification of major components of a computer system such as mother board, RAM modules, daughter cards, bus slots, SMPS, internal storage devices, interfacing ports. Specifications of desktop and server class computers. Installation of common operating systems for desktop and server use.	27-03-2023	CO1	1
2	Study of a terminal based text editor such as Vim or Emacs. Basic Linux commands, familiarity with following commands/operations expected 1. man 2. ls, echo, read 3. more, less, cat, 4. cd, mkdir, pwd, find 5. mv, cp, rm 6. wc, cut, paste 7. head, tail, grep, expr 8 chmod, chown 9. useradd, usermod, userdel, passwd 10. df,top, ps 12 ssh, ssh-keygen	06-03-2023, 07-03-2023, 13-03-2023, 14-03-2023, 20-03-2023, 21-03-2023	CO2	5
3	File system hierarchy in a common Linux distribution, file and device permissions, study of system configuration files in /etc, familiarizing log files for system events, user activity, network events.	10-07-2023	CO2	20
4	Shell scripting: study bash syntax, environment variables, variables, control constructs such as if, for and while, aliases and functions, accessing command line arguments passed to shell scripts. 1. Write a shell script to count lines and words in a file 2. Shell Script to check a number is even or odd 3. Shell script to check whether a number is positive or negative 4. Shell script to find the greatest of three numbers 5. Shell Script to demonstrate String Operators 6. Shell Script to analyze people of certain age groups who are eligible for getting a suitable job if their condition and norms get satisfied using nested if statement. 7. Write a shell script to display the capital of a state using case...esac statement. 8. Write a shell script to count the number in reverse direction.	28-03-2023, 03-04-2023, 04-04-2023, 11-04-2023, 18-04-2023, 27-06-2023	CO4	23

Sl. No.	Experiment	Date	CO	Page No.
4	9. Write a shell script to check whether the number is palindrome or not 10. Write a shell script to check whether a given number is Armstrong or not 11. Write a shell script to check whether a number is prime or not 12. Write a shell script for factorial of a number 13. Write a shell Script to print Fibonacci series 14. Write a shell script to check if the current year is a leap year or not	28-03-2023, 03-04-2023, 04-04-2023, 11-04-2023, 18-04-2023, 27-06-2023	CO4	
5	Install latest version of Ubuntu on a virtual box	05-06-2023	CO1	31
6	Installation and configuration of LAMP stack. Deploy an open source application such as phpmyadmin	26-06-2023	CO3	35
7	Build and install software from source code, familiarity with make and cmake utilities expected. 1. Write a program to find factorial of a number using make utility 2. Write a program to add two numbers using cmake utility	04-07-2023, 11-07-2023	CO4	44
8	Introduction to command line tools for networking IPv4 networking, network commands: ping route traceroute, nslookup, ip.	20-06-2023, 24-07-2023	CO5	47
9	Analyzing network packet stream using tcpdump and wireshark. Perform basic network service tests using nc.	20-06-2023	CO5	49
10	Insatllation of KVM and perform basic KVM Commands	12-06-2023, 13-06-2023 19-06-2023	CO3	51
11	Docker, installation and deployment.	04-07-2023	CO3	56

Experiment No.: 1

Aim: Introduction to Computer hardware: Physical identification of major components of a computer system such as mother board, RAM modules, daughter cards, bus slots, SMPS, internal storage devices, Interfacing ports. Specifications of desktop and server class computers. Installation of common operating systems for desktop and server use.

CO1: Install and configure common operating systems in virtual environment.

Procedure

1. Motherboard:

A motherboard is the main circuit board inside a computer that connects all of the computer's hardware components together, such as the central processing unit (CPU), memory, storage drives, and expansion cards. It serves as a communication hub between these components,



allowing them to work together and perform various functions. The motherboard typically includes several important components, including the chipset and the BIOS (Basic Input/Output System). Modern motherboards come in different form factors, such as ATX, microATX, and mini-ITX, which determine their size and layout.

2. RAM:

RAM (Random Access Memory) modules are computer components that store data and instructions temporarily while the computer is running. RAM is a type of volatile memory, which means that its contents are erased when the computer is turned off or restarted. RAM modules come in various types, speeds, and capacities. They are typically installed in slots on the motherboard and can be easily upgraded or replaced. RAM modules are also used to support multitasking, where multiple programs can run simultaneously. RAM modules are an essential component of a computer that provides temporary storage for data and instructions. They help to improve the computer's performance and support multitasking.



3. Daughter Card:

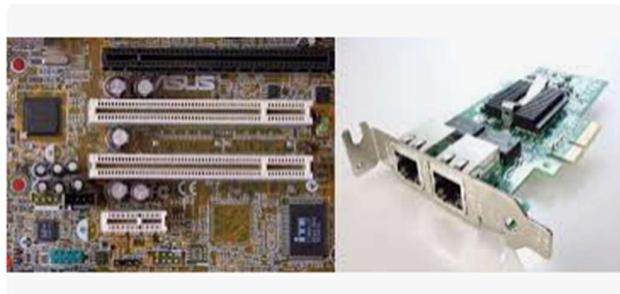
A daughter card, also known as a daughterboard or expansion card, is a circuit board that connects to the main motherboard of a computer to add new functionality or enhance existing features. Daughter cards are commonly used to expand the capabilities of a computer, such as adding additional ports, memory, or processing power. Examples of daughter cards include graphics cards, sound cards, network interface cards (NICs), and storage expansion cards.



4. Bus Slot:

A bus slot, also known as an expansion slot, is a socket on the motherboard of a computer that allows expansion cards to be inserted and connected to the computer's bus system. There are several types of bus slots commonly used in computers, including Peripheral Component

Interconnect (PCI), PCI Express (PCIE), and Accelerated Graphics Port (AGP) slots. These slots vary in their bandwidth, power, and physical size, and are designed to accommodate different types of expansion cards. Expansion cards, such as graphics cards, sound cards, and network interface cards, are connected to the computer's bus system through the bus



slot, allowing them to communicate with other components and exchange data.

5. SMPS:

It stands for Switched-Mode Power Supply, a type of power supply used in computers and other electronic devices. The SMPS is responsible for converting AC power from a wall outlet into DC power that the computer can use to operate. SMPS



uses high-frequency switching and regulation to convert AC power to DC power more efficiently. SMPS units are widely used in modern computers, as they are more reliable, energy-efficient, and generate less heat than linear power supplies. SMPS is an essential component of modern computers that plays a vital role in powering the system and ensuring its proper operation.

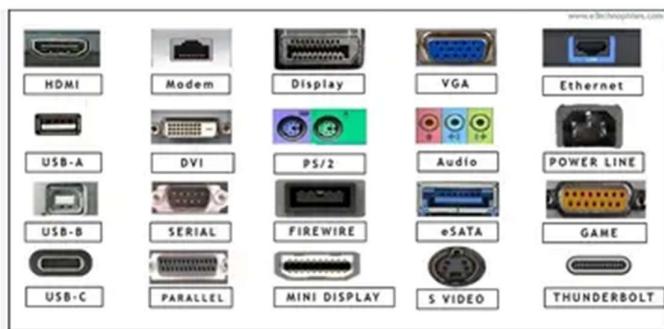
6. Internal Storage Devices:

Internal storage devices are electronic components used to store data within a computer or other electronic devices. There are two main types of internal storage devices: Hard Disk Drives (HDDs) and Solid State Drives (SSDs). HDDs are the traditional type of internal storage device and are typically larger in capacity than SSDs. They consist of spinning disks that store data magnetically and read/write heads that move over the disks to access the data. SSDs, on the other hand, use flash memory to store data and have no moving parts, which makes them faster, more durable, and more energy-efficient than HDDs. They are typically more expensive than HDDs.



7. Interfacing Ports:

Interfacing ports refers to the process of connecting different devices or components together in order to exchange information or perform certain tasks. Ports are used to facilitate communication between devices, and they can come in different forms, such as USB ports, Ethernet ports, HDMI ports, and so on. Interfacing ports is an important aspect of modern computing, as it allows us to connect a wide range of devices and components together in order to create complex systems and networks. Whether it's connecting a keyboard and mouse to a computer, or setting up a network of servers and workstations in a large enterprise, interfacing ports is a crucial part of the process.



Desktop Computer Specifications:

- Processor (CPU):** Desktop computers can come with various types of processors, ranging from basic to high-performance CPUs. Common brands include Intel and AMD. Higher clock speeds and more cores generally translate to better performance.
- Memory (RAM):** Desktops typically have anywhere from 4GB to 32GB or more of RAM. The amount you need depends on your intended use (e.g., basic tasks vs. gaming or content creation).
- Storage:** Desktops can have a variety of storage options, including Hard Disk Drives (HDDs) and Solid State Drives (SSDs). SSDs offer faster performance compared to HDDs.
- Graphics Card (GPU):** This is important for gaming and graphic-intensive tasks. Integrated graphics are suitable for basic tasks, while dedicated GPUs from NVIDIA or AMD are necessary for gaming and professional applications.
- Ports:** Desktops typically come with a range of USB ports, audio jacks, video outputs (HDMI, DisplayPort), and sometimes older ports like VGA or DVI.
- Expansion Slots:** Desktops may have PCIe slots for adding dedicated graphics cards, sound cards, network cards, and more.
- Form Factor:** Desktops come in various sizes, including tower, mini-tower, small form factor (SFF), and all-in-one (AIO) designs.

Server-Class Computer Specifications:

1. **Processor (CPU):** Servers often feature multi-core CPUs with high clock speeds and advanced features for improved reliability and performance. Brands like Intel Xeon and AMD EPYC are common in server environments.
2. **Memory (RAM):** Servers usually have more RAM than desktops, ranging from 16GB to hundreds of gigabytes, depending on the server's role (web hosting, database, virtualization, etc.).
3. **Storage:** Servers utilize various storage configurations, including RAID arrays, hot-swappable drives, and SSDs for faster data access.
4. **Redundancy:** Servers often have redundant power supplies, cooling systems, and sometimes even redundant components for high availability.
5. **Remote Management:** Servers usually include features like remote management interfaces (IPMI, iDRAC, iLO) to allow administrators to manage and monitor the server remotely.
6. **Network Connectivity:** Servers may have multiple Ethernet ports to provide redundancy and accommodate high network traffic.
7. **Form Factor:** Server form factors vary widely, including rack-mounted servers for data centers and tower servers for smaller environments.
8. **Virtualization Support:** Many server-class CPUs have features specifically designed to enhance virtualization performance, allowing multiple virtual machines to run efficiently on a single physical server.

Result

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

Experiment No.: 2

Aim: Study of a terminal based text editor such as Vim or Emacs. Basic Linux commands, familiarity with following commands/operations expected 1. man 2. ls, echo, read 3. more, less, cat, 4. cd, mkdir, pwd, find 5. mv, cp, rm 6. wc, cut, paste 7. head, tail, grep, expr 8 chmod, chown 9. useradd, usermod, userdel, passwd 10. df,top, ps 12 ssh, ssh-keygen

CO2: Perform system administration tasks including network configurations, user creations and troubleshooting.

Procedure

1.man

we can learn and understand different commands right from the shell.

Syntax: \$man ls

```
LS(1)                               User Commands                               LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILEs (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..
```

2. ls, echo, read

The ls command is used to list files in Linux.

a. **ls -R** : prints subdirectory contents

Syntax: \$ls -R

```
./jensample:
jenny

./jensample/jenny:
connect months names.txt
```

b. ls -l : long listing

Syntax: \$ls -l

```
-rw-rw-r-- 1 student student 18 Mar 6 12:26 week2.txt
-rw-rw-r-- 1 student student 33 Mar 6 12:27 weeks.txt
-rw-rw-r-- 1 student student 15 Mar 6 12:26 week.txt
```

c. ls -a : to list all hidden files

Syntax: \$ls -a

```
bubblesort.c          .gnupg           .profile
bubblesort.o         'insertion sort.c'
.cache                .java             Public
                           PycharmProjects
```

d. ls -al : list the files and directory with detailed information

Syntax: \$ls -al

```
student@t2:~$ ls -al
total 524
drwxr-xr-x 23 student student 4096 Mar 7 15:19 .
drwxr-xr-x  6 root    root    4096 Jun 17 2022 ..
```

e. ls -t : list the file sorted in the order of the last modified file.

Syntax: \$ls -t

```
student@t2:~$ ls -t
jensample               linklist.c
jenny                   linklis.c
new.txt                 merge.c
weeks.txt               'insertion sort.c'
week2.txt               bubblesort
week.txt                bubblesort.o
file2.txt               bubblesort.c
file1.txt               merge
Downloads              merge.o
```

f. ls -r : to reverse the natural sorting order

Syntax: \$ls -r

```
student@t2:~$ ls -r
week.txt            mergesort.c           'doubley link list creation.c'
weeks.txt          mergesort.c           Documents
week2.txt          merge.o              Desktop
```

The read command is used to read a value/data from the user and echo is used to print the value in that variable.

a. \$read x y z: Declare variables to store data

To print : echo “[\\$x][\\$y][\\$z]”

```
student@t2:~$ read x y z
jenny johnson mca
student@t2:~$ echo "[\$x][\$y][\$z]"
[jenny][johnson][mca]
```

- b. To read contents through multiple lines we use “\” at the end of each line.

```
student@t2:~$ read
j\
> e\
> n\
> n\
> y
student@t2:~$ echo $REPLY
jenny
```

- c. **\$read -p [prompt message]**: Prompt user to enter data

```
student@t2:~$ read -p"Enter your name"
Enter your name Jenny
student@t2:~$ echo "My name is $REPLY"
My name is Jenny
```

- d. **\$read -n [limit]**: Specifies the limit.

```
Enter 6 characters: jenny
student@t2:~$ read -n 6 -p "Enter 6 characters: "
Enter 6 characters: jennyjstudent@t2:~$
```

- e. **\$read -s** : it gives the security (hides the data)

```
student@t2:~$ read -s -p "Enter password: "
Enter password: student@t2:~$ echo "Password is $REPLY"
Password is 123456
```

3. more, cat

\$more: The more command is similar to cat command to display content. The only difference is that, in case of large files cat command will scroll off your screen while more command display output one screen full at a time.

```
student@t2:~$ more corona.txt
```

```
The IBV-like novel cold viruses were soon shown to be also morphologically related to the mouse hepatitis virus.[19] This new group of viruses were named coronaviruses after their distinctive morphological appearance.[7] Human coronavirus 229E and human coronavirus OC43 continued to be studied in subsequent decades
--More-- (99%)
```

- a. **\$more +20 [filename]**

```
student@t2:~$ more +20 corona.txt

Transmission electron micrograph of organ cultured coronavirus OC43
Scottish virologist June Almeida at St Thomas' Hospital in London, collaborating with Tyrrell, compared the structures of IBV, B814 and 229E in 1967.[32][33]
```

b. more +/[pattern] [filename]

Used to search the string inside your text document you can view all the instances navigating through the result.

```
student@t2:~$ more +/Human corona.txt

...skipping
as not realized at the time that these three different viruses were related.[20]
[12]

Human coronaviruses were discovered in the 1960s[21][22] using two different methods in the United Kingdom and the United States.[23] E.C. Kendall, Malcolm By
```

c. \$more -d[filename]

It helps the user to navigate according to instructions.

```
student@t2:~$ more -d corona.txt

d successfully cultivated rhinoviruses, adenoviruses and other known common cold viruses. In 1965, Tyrrell and Bynoe successfully cultivated the novel virus b
--More--(4%)[Press space to continue, 'q' to quit.]
```

\$cat

Syntax: cat > [filename] : create a new file and open it to add content.

```
student@t2:~$ cat > colors
red
blue
yellow
black
^Z
[3]+  Stopped                  cat > colors
```

a. cat >> [filename] : to append new contents to existing file contents

```
student@t2:~$ cat >> colors
green
white
```

b. cat [filename] : to display file contents.

```
student@t2:~$ cat colors
red
blue
yellow
black
green
white
```

c. cat -n [filename] : to display content with line numbers

```
student@t2:~$ cat -n colors
 1 red
 2 blue
 3 yellow
 4 black
 5 green
 6 white
```

d. cat -b [filename] : No line numbering for blank spacing.

```
student@t2:~$ cat -b colors
 1 red
 2 blue
 3 yellow
 4 black
 5 green
 6 white

 7 magenta
```

e. cat -e [filename] : to display \$ character at the end of each line.

```
student@t2:~$ cat -e colors
red$
blue$
yellow$
black$
green$
white$
$
$
$
magenta$
```

4. mkdir, cd, pwd, find

mkdir : creates new directory

Syntax: \$mkdir jenny

cd: change to the specified directory

Syntax: \$cd jenny

```
student@t2:~$ man ls
student@t2:~$ mkdir Jenny
student@t2:~$ cd Jenny
student@t2:~/Jenny$ ls
student@t2:~/Jenny$ 
```

pwd : print the working directory

Syntax: \$pwd

```
student@t2:~$ pwd
/home/student
student@t2:~$ 
```

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 . -name “*.txt”;

5. mv, cp, rm

mv: To move one file content to another file(contents will be overwritten).

Syntax: \$mv [filename][filename]

\$mv -b mark Bio : Backups file

```
student@t2:~/Jenny$ cat >Bio
Jenny John
RMCA
^Z
[4]+  Stopped                  cat > Bio
student@t2:~/Jenny$ mv -b mark Bio
student@t2:~/Jenny$ ls
Bio Bio~ marks
student@t2:~/Jenny$ cat Bio~
Jenny John
RMCA
```

mv -i : prompt user for confirmation

Syntax: \$mv -i [filename][filename]

\$mv -i Bio Bio1

```
student@t2:~/Jenny$ cat >Bio1
AmalJyothi
^Z
[5]+ Stopped                  cat > Bio1
student@t2:~/Jenny$ mv -i Bio Bio1
mv: overwrite 'Bio1'? n
student@t2:~/Jenny$ mv -i Bio Bio1
mv: overwrite 'Bio1'? y
student@t2:~/Jenny$ cat Bio1
1
2
3
4
5
```

cp - To copy the content to a new file

a) cp file1 file2 - To copy file1 contents in file2

```
jenny@jenny-VirtualBox:~$ cp file1 file2
jenny@jenny-VirtualBox:~$ cat file2
jenny
riya
tinu
alfiya
```

6. wc, cut, paste

\$wc [filename]

Prints the no.of lines,words,bytes etc.

```
student@t2:~$ cat > Details
Jenny Johnson
RMCA
AmalJyothi College of Engineering
^Z
[2]+ Stopped                  cat > Details
student@t2:~$ wc Details
3 7 53 Details
```

a. \$wc -l [filename]

Prints no.of lines

```
student@t2:~$ wc -l Details
3 Details
```

b. \$wc -m [filename]

Prints no.of characters

```
student@t2:~$ wc -m Details
53 Details
```

c. \$wc -w [filename]

Prints no.of words

```
student@t2:~$ wc -w Details
7 Details
```

d. \$wc -L [filename]

Prints the length of the longest line.

```
student@t2:~$ wc -L Details
33 Details
```

cut : used to cut file contents

a. cut -d - -f1 file2 : cut command to just print the first field of the file using the delimiter “-”

```
jenny@jenny-VirtualBox:~$ cut -d - -f1 file2
jenny
riya
tinu
```

2.**paste**:used to join files horizontally(Each file consisting of different lines)

a) paste file1 file2-To paste file1 contents in file2

```
jenny@jenny-VirtualBox:~$ paste file1 file2
jenny    jenny-3
riya     riya-29
tinu     tinu-45
alfiya
```

b) paste file1 file2 > file3-To paste file1 and file2 contents in a new file

```
jenny@jenny-VirtualBox:~$ paste file1 file2 > file3
jenny@jenny-VirtualBox:~$ cat file3
jenny    jenny-3
riya     riya-29
tinu     tinu-45
alfiya
```

c) paste -d ‘%’ file1 file2- By specifying the delimiter, we can also split the lines into columns with specified delimiter.

```
jenny@jenny-VirtualBox:~$ paste -d '%d' file1 file2
jenny%jenny-3
riya%riya-29
tinu%tinu-45
alfiya%
```

7. grep, head, tail, expr

`$grep -i [word][filename]`

Used to filter the content which makes our search easier(Case insensitive)

`$ grep -i Hindi mark`

```
student@t2:~/Jenny$ cat > mark
English 99
Maths 56
Hindi 78
Malayalam 89
```

```
student@t2:~/Jenny$ grep -i Hindi mark
Hindi 78
```

`$grep -v[word][filename]`

To view all the contents expect the searched one

`$grep -v Hindi mark`

```
student@t2:~/Jenny$ grep -v Hindi mark
English 99
Maths 56
Malayalam 89
```

`$grep -A1[word][filename]`

To view the content along with one line before that

`$grep -A1 Hindi mark`

```
student@t2:~/Jenny$ grep -A1 Hindi mark
Hindi 78
Malayalam 89
```

`$grep -B1[word][filename]`

To view the content along with one line before that.

`$grep -B1 Hindi mark`

```
student@t2:~/Jenny$ grep -B1 Hindi mark
Maths 56
Hindi 78
```

```
$grep -C1[word][filename]
```

To view the content along with one line after and before.

```
$grep -C1 Hindi mark
```

```
student@t2:~/Jenny$ grep -C1 Hindi mark
Maths 56
Hindi 78
Malayalam 89
```

```
$ head [filename]
```

To display the top lines of the file. By default it will display top 10 lines

```
$ head txt
```

```
$ head -5[filename]
```

Display top 5 lines

```
$head -5 txt
```

```
student@t2:~/Jenny$ head txt
1
2
3
4
5
6
7
8
9
11
student@t2:~/Jenny$ head -5 txt
1
2
3
4
5
```

```
$tail [filename]
```

To display the last contents of the file, by default it will display the last 10 lines.

```
$tail txt
```

```
$tail -5[filename]
```

To display the last five contents of the file.

```
$tail -5 txt
```

```
student@t2:~/Jenny$ tail txt
9
11
22
33
44
55
66
77
88
99
student@t2:~/Jenny$ tail -5 txt
55
66
77
88
99
```

\$expr [expressions]: Calculate the expressions and return the output.

```
student@t2:~/jenny$ expr 45 + 9
54
student@t2:~/jenny$ expr 45 - 9
36
student@t2:~/jenny$ expr 45 \* 9
405
student@t2:~/jenny$ expr 45 / 9
5
```

8 chmod, chown

\$chmod

Used to change the access permission of files and directories. It stands for change mode.

a) chmod -wx [filename] :- deny permission to write and execute for file

\$chmod -wx names

```
mca@t2:~$ chmod -wx names
mca@t2:~$ cat >> names
bash: names: Permission denied
mca@t2:~$ chmod +rws names
mca@t2:~$ cat >> names
jomol
^Z
[2]+  Stopped                  cat >> names
mca@t2:~$
```

7. \$chown

It is used to change a file ownership or directory ownership for a user or a group. It stands for change owner.

\$sudo chown [directory][filename]

\$chown jenny names

```
mca@t2:~$ sudo chown jenny names
[sudo] password for mca:
mca@t2:~$ chmod +rwx names
chmod: changing permissions of 'names': Operation not permitted
mca@t2:~$ ls -l names
-rw-rwSr-- 1 jenny mca 25 Mar 20 11:58 names
mca@t2:~$
```

9. useradd, usermod, userdel, passwd

\$ sudo userdel user :- Delete user

\$ sudo userdel jenny

9. \$ sudo groupdel: deletes group

\$ sudo group mcastudent

```
mca@t2:~$ sudo userdel jenny
[sudo] password for mca:
mca@t2:~$ sudo userdel jenny
userdel: user 'jenny' does not exist
mca@t2:~$ sudo groupdel mcastudent
mca@t2:~$ sudo groupdel mcastudent
groupdel: group 'mcastudent' does not exist
mca@t2:~$
```

10. df, ps, top

\$df

used to find report on disc utilisation

```
student@t2:~/jenny$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev              3953504      0   3953504   0% /dev
tmpfs             797752    1736   796016   1% /run
/dev/sda6        143074460 28142036 107591832  21% /
tmpfs             3988756   32404   3956352   1% /dev/shm
tmpfs                 5120       4     5116   1% /run/lock
tmpfs             3988756      0   3988756   0% /sys/fs/cgroup
```

ps - Stands for Process. Currently running programs and running instances.

\$ps

a)ps -u [user] :- Display all running processes of a particular user

\$ps -u mca

```
*** END OPENSSH PRIVATE KEY ***
mca@t2:~$ ps
  PID TTY      TIME CMD
  7220 pts/1    00:00:00 bash
  9000 pts/1    00:00:00 ps
mca@t2:~$ ps -u mca
  PID TTY      TIME CMD
 1364 ?        00:00:00 systemd
 1365 ?        00:00:00 (sd-pam)
 1371 ?        00:00:00 pulseaudio
 1373 ?        00:00:00 tracker-miner-f
 1376 ?        00:00:00 dbus-daemon
 1381 ?        00:00:00 gnome-keyring-d
 1384 ?        00:00:00 gvfsd
```

b)ps -C :- Specific process

\$ps -C firefox

```
mca@t2:~$ ps -C firefox
  PID TTY      TIME CMD
  9047 ?        00:00:01 firefox
mca@t2:~$ █
```

c)ps -f -p PID :- List the process by id

\$ps -f -p 9047

```
mca@t2:~$ ps -f -p 9047
UID      PID  PPID  C STIME TTY          TIME CMD
mca      9047   1364  10 15:58 ?        00:00:10 /usr/lib/firefox/firefox -ne
mca@t2:~$ █
```

top command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.

Syntax; \$ top

11 ssh, ssh-keygen

ssh user@ip address- Stands for Secure Shell Protocol used to securely connect to a remote server or system. ssh is secure in the sense that it transfers data in encrypted form between host and client.

\$ssh mca@192.168.6.29

```
mca@t2:~$ ssh mca@192.168.6.29
ssh: connect to host 192.168.6.29 port 22: Connection refused
█
```

a. sudo apt-get install openssh -server :- Update port

```
mca@t2:~$ sudo apt-get update
[sudo] password for mca:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:3 https://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Get:4 http://ppa.launchpad.net/maarten-fonville/android-studio/ubuntu focal InRe
lease [17.6 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,079 B
]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2,04
6 kB]
Get:8 http://ppa.launchpad.net/maarten-fonville/android-studio/ubuntu focal/main
amd64 Packages [2,052 B]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:10 http://ppa.launchpad.net/maarten-fonville/android-studio/ubuntu focal/mai
n Translation-en [324 B]
```

```
mca@t2:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 682 not upgraded.
Need to get 1,359 kB of archives.
After this operation, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssh-clien
```

b. sudo ufw allow 22

```
mca@t2:~$ sudo ufw allow 22
Rules updated
Rules updated (v6)
```

c. \$ssh mca@192.168.6.29

```
ssh: connect to host 192.168.6.29 port 22: Connection refused
mca@t2:~$ ssh mca@192.168.6.29
The authenticity of host '192.168.6.29 (192.168.6.29)' can't be established.
ECDSA key fingerprint is SHA256:kKk7s0MYRkq6/H06Go97XKajNqNDSThuvCv+GSJz40U.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.6.29' (ECDSA) to the list of known hosts.
mca@192.168.6.29's password:
Permission denied, please try again.
mca@192.168.6.29's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-26-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

698 updates can be installed immediately.
459 of these updates are security updates.
To see these additional updates run: apt list --upgradable
```

d. ssh-keygen :- Generating a key for secure shell

\$ssh-keygen

```
mca@t2:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mca/.ssh/id_rsa): jen.txt
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in jen.txt
Your public key has been saved in jen.txt.pub
The key fingerprint is:
SHA256:VY++5295UY4kWP5ceeeQ2JqhgKjQxsSiN/0q8CCK1vo mca@t2
The key's randomart image is:
+---[RSA 3072]----+
| . . . . |
| . o ..o |
| .= . . . +.o... |
| o * o . ....= =.= |
| + o . S. .0 *+|
| + . . . . o.+.o |
| ++. . . . o |
| o.o.. o .o |
| ..oE .oo |
+---[SHA256]-----+
```

Result

The program was executed and the result was successfully obtained. Thus, CO₂ was obtained.

Experiment No.:3

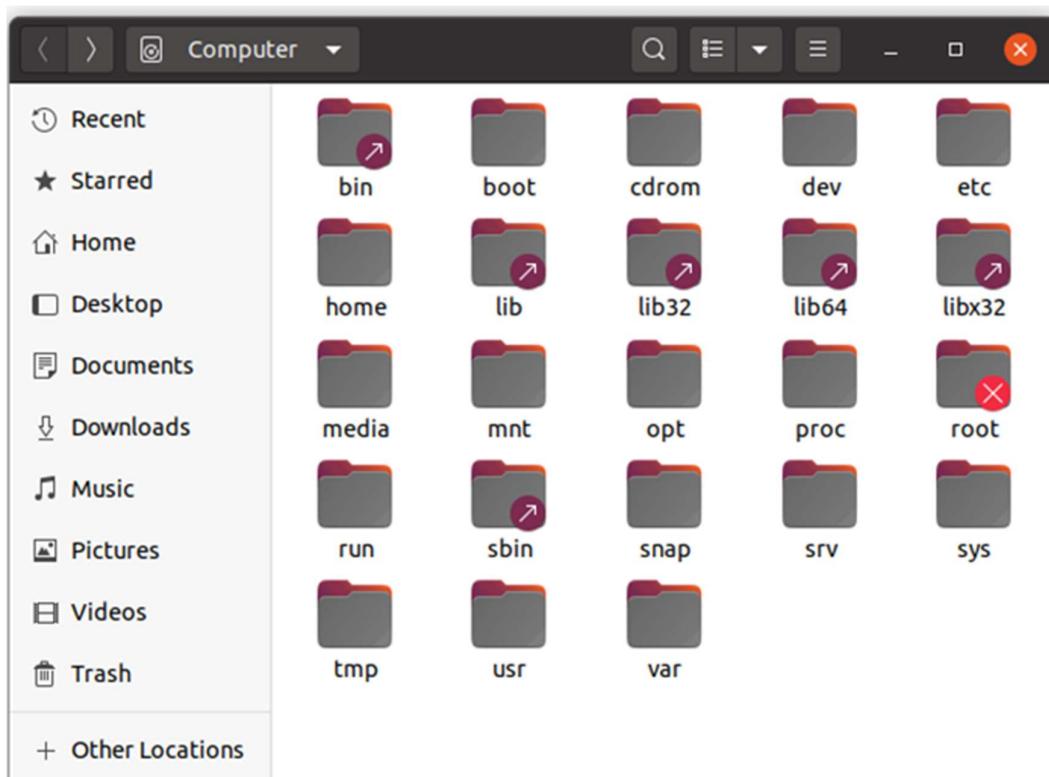
Aim: File system hierarchy in a common Linux distribution, file and device permissions, study of system configuration files in /etc, familiarizing log files for system events, user activity, network events

CO2: Perform system administration tasks including network configurations, user creations and trouble shooting.

Procedure

1. **/bin** : stores executable files and programs. Essential binary files which are accessible to all users. These files are required for basic functions and for various system operations and system interactions.
2. **/boot** : Files required for boot process including kernel , boot loaders and initial ram disk. We want to load the os and prepare the system for use. All the files supporting the booting process are stored in the boot directory.
3. **/dev** : Device file representing the physical and virtual devices such as hazard drive ,printers, cpu etc.
4. **/etc** : The system configuration files or various applications and services. Configuration files determine the behaviours, the functionality and the appearance of the software.
5. **/home** : Directory for regular users
6. **/lib** : Libraries required for the binary in /bin
7. **/mnt** : Temporary attaching file systems. eg. external drives
8. **/media** : it is a temporary file system for removable media. eg. usb
9. **/opt** : For sharing optional software packages.
10. **/sys** : Exposes information about system hardware and devices
11. **/tmp** : Temporary file created by the user or applications
12. **/usr** : It is basically user program and libraries
13. **/var** : Variable data that changes frequently such as log files and cache files

Output Screenshot



Installing tree

\$ sudo apt install tree

```
mca@u15:~/Desktop$ sudo apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libllvm9 linux-headers-5.4.0-26 linux-headers-5.4.0-26-generic
  linux-image-5.4.0-26-generic linux-modules-5.4.0-26-generic
  linux-modules-extra-5.4.0-26-generic shim
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 318 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 1s (42.4 kB/s)
Selecting previously unselected package tree.
(Reading database ... 231483 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
mca@u15:~/Desktop$
```

Run tree**\$tree**

```
mca@u15:~$ tree
.
├── Desktop
│   └── root_directory.png
├── Documents
├── Downloads
├── folder1
│   ├── file1
│   ├── file2
│   └── folder1
│       ├── filex
│       └── filey
└── folder2
    ├── filez
    └── folder2
        ├── filed
        └── fileh
.
└── Marvel
    ├── Endgame.txt
    ├── infinity
    └── new
        └── war.txt
.
└── Music
└── Pictures
└── Public
└── snap
    └── snap-store
        ├── 959
        └── common
            └── current -> 959
.
└── Templates
└── ubuntu-20.04-desktop-amd64.iso
└── Videos

18 directories, 13 files
mca@u15:~$
```

Result

The program was executed and the result was successfully obtained. Thus, CO₂ was obtained.

Experiment No.: 4

Aim: Shell scripting: study bash syntax, environment variables, variables, control constructs such as if, for and while, aliases and functions, accessing command line arguments passed to shell scripts.

CO4: Write shell scripts required for system administration.

Procedure

1. Write a shell script to count lines and words in a file

```
#!/bin/bash
file_path=/home/jenny/Jenny/prg3_var.sh
countlines=`wc --lines < $file_path`
countwords=`wc --word < $file_path`
echo "Number of lines: $countlines"
echo "Number of words: $countwords"
```

```
jenny@jenny-VirtualBox:~/Jenny$ vi prg4_countl.sh
jenny@jenny-VirtualBox:~/Jenny$ chmod +x prg4_countl.sh
jenny@jenny-VirtualBox:~/Jenny$ ./prg4_countl.sh
Number of lines: 8
Number of words: 20
```

2. Shell Script to check a number is even or odd

```
#!/bin/bash
read -p "Enter the number:" n
if(( $n % 2 == 0 ))
then
echo Number is even
else
echo Number is odd
fi
```

```
jenny@jenny-VirtualBox:~/Jenny$ vi even_odd.sh
jenny@jenny-VirtualBox:~/Jenny$ chmod +x even_odd.sh
jenny@jenny-VirtualBox:~/Jenny$ ./even_odd.sh
Enter the number:7
Number is odd
jenny@jenny-VirtualBox:~/Jenny$ ./even_odd.sh
Enter the number:2
Number is even
```

3. Shell script to check whether a number is positive or negative

```
#!/bin/bash
read -p "Enter the number: " n
if(( $n < 0))
then
echo Number is negative
elif((n > 0))
then
echo Number is positive
elif((n == 0))
then
echo Number is zero
else
echo Invalid
fi
```

```
jenny@jenny-VirtualBox:~/Jenny$ vi num_check.sh
jenny@jenny-VirtualBox:~/Jenny$ chmod +x num_check.sh
jenny@jenny-VirtualBox:~/Jenny$ ./num_check.sh
Enter the number: 7
Number is positive
jenny@jenny-VirtualBox:~/Jenny$ ./num_check.sh
Enter the number: -1
Number is negative
jenny@jenny-VirtualBox:~/Jenny$ ./num_check.sh
Enter the number: 0
Number is zero
```

4. Shell script to find the greatest of three numbers

```
#!/bin/bash
echo "Enter the first number"
read n1
echo "Enter second number"
read n2
echo "Enter third number"
read n3
if [[ $n1 > $n2 && $n1 > $n3 ]]
then
echo "$n1 is largest"
elif [[ $n2 > $n1 && $n2 > $n3 ]]
then
echo "$n2 is largest"
else
echo "$n3 is largest"
fi
```

```
student@t2:~/Jenny$ ./largest3.sh
Enter the first number
2
Enter second number
3
Enter third number
1
3 is largest
```

5. Shell Script to demonstrate String Operators

```
#!/bin/bash
read -p "Enter first string: " s1
read -p "Enter second string: " s2

if [ $s1 = $s2 ]
then
    echo "$s1 is equal to $s2"
else
    echo "$s1 is not equal to $s2"
fi

if [ $s1 != $s2 ]
then
    echo "$s1 is not equal to $s2"
else
    echo "$s1 is equal to $s2"
fi

if [ -z $s1 ]
then
    echo "Size not zero"
else
    echo "size zero"
fi

if [ -n $s2 ]
then
    echo "size zero"
else
    echo "size not zero"
fi

if [ $s1 ]
then
    echo "String is not empty"
else
    echo "String is empty"
fi
```

```
jenny@jenny-VirtualBox:~/Jenny$ vi string_o1.sh
jenny@jenny-VirtualBox:~/Jenny$ chmod +x string_o1.sh
jenny@jenny-VirtualBox:~/Jenny$ ./string_o1.sh
Enter first string: jenn
Enter second string: jenn
jenn is equal to jenn
jenn is equal to jenn
size zero
size zero
String is not empty
```

6. Shell Script to analyse people of certain age groups who are eligible for getting a suitable job if their condition and norms get satisfied using nested if statement.

```
#!/bin/bash
echo -n "Enter your age: "
read age
if [ "$age" -ge 18 ] && [ "$age" -le 60 ]; then
    echo "You are eligible for the job."
else
    echo "You are not eligible for the job."
fi
```

```
jenny@jenny-VirtualBox:~$ vi job_age.sh
jenny@jenny-VirtualBox:~$ chmod +x job_age.sh
jenny@jenny-VirtualBox:~$ ./job_age.sh
Enter your age: 16
You are not eligible for the job.
jenny@jenny-VirtualBox:~$ ./job_age.sh
Enter your age: 18
You are eligible for the job.
jenny@jenny-VirtualBox:~$ ./job_age.sh
Enter your age: 80
You are not eligible for the job.
```

7. Write a shell script to display the capital of a state using case...esac statement.

```
#!/bin/bash
read -p "Enter the State: " s
case "$s" in
"karnataka") echo "Capital of karnataka is Bangalore";;
"maharashtra") echo "Capital of maharashtra is Mumbai";;
"kerala") echo "Capital of kerala is Thiruvananthapuram";;
"tamilnadu") echo "Capital of Tamilnadu is Chennai";;
esac
```

```
mca@u3:~/jenny$ vi capital.sh
mca@u3:~/jenny$ chmod +x capital.sh
mca@u3:~/jenny$ ./capital.sh
Enter the Statekerala
Capital of kerala is Thiruvananthapuram
```

8. Write a shell script to count the number in reverse direction.

```
#!/bin/bash
for (( i=10; i>=0; i-- ))
do
echo $i
done
```

```
mca@u3:~/jenny$ ./forloop2.sh
10
9
8
7
6
5
4
3
2
1
0
```

9. Write a shell script to check whether the number is palindrome or not.

```
#!/bin/bash

read -p "Enter a number: " n

dup=$n

while [ $n -gt 0 ]

do

r=$(( n%10 ))

s=$(( r+$(( s*10 )) ))

n=$(( n/10 ))

done

if [ $dup -eq $s ]

then

echo "$dup is palindrome"

else

echo "$dup is not palindrome"

fi
```

```
jenny@jenny-VirtualBox:~/jenny$ ./palindrome.sh
Enter a number: 123
123 is not palindrome
jenny@jenny-VirtualBox:~/jenny$ ./palindrome.sh
Enter a number: 123321
123321 is palindrome
```

10. Write a shell script to check whether a given number is Armstrong or not

```
#!/bin/bash

read -p "Enter a number: " n

dup=$n

s=0

while [ $n -gt 0 ]
do
r=$(( $n%10 ))
s=$(( $s+$(($r*$r*$r)) ))
n=$(( n/10 ))
done

if [ $dup -eq $s ]
then
echo "$dup is an Armstrong number"
else
echo "$dup is not an armstrong number"
fi
```

```
jenny@jenny-VirtualBox:~$ ./armstrong.sh
Enter a number: 153
153 is an Armstrong number
jenny@jenny-VirtualBox:~$ ./armstrong.sh
Enter a number: 123
123 is not an armstrong number
```

11. Write a shell script to check whether a number is prime or not

```
#!/bin/bash

read -p "Enter a number: " n

i=2

while [ $i -lt $n ]
do
if [ `expr $n % $i` -eq 0 ]
then
echo "$n is not a prime number"
exit
fi
```

```
i=$(( i+1 ))
done

echo "$n is a prime number"
```

```
jenny@jenny-VirtualBox:~$ ./prime.sh
Enter a number: 4
4 is not a prime number
jenny@jenny-VirtualBox:~$ ./prime.sh
Enter a number: 7
7 is a prime number
```

12. Write a shell script for factorial of a number

```
#!/bin/bash

read -p "Enter the number: " n

f=1

for (( i=1; i<=n; i++ ))
do
    f=$(( f*i ))
done

echo "Factorial of $n is $f"
```

```
jenny@jenny-VirtualBox:~$ ./factorial.sh
Enter the number: 5
Factorial of 5 is 120
```

13. Write a shell Script to print Fibonacci series

```
#!/bin/bash

read -p "Enter the limit: " n

echo "Fibonacci Series: "

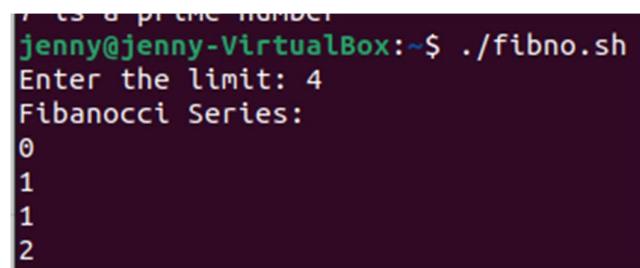
a=0
b=1
c=0

echo "$a"
echo "$b"
for(( i=0; i<=$n; i++ ))
```

```

do
c=$(( $a + $b ))
a=$b
b=$c
echo "$c"
done

```



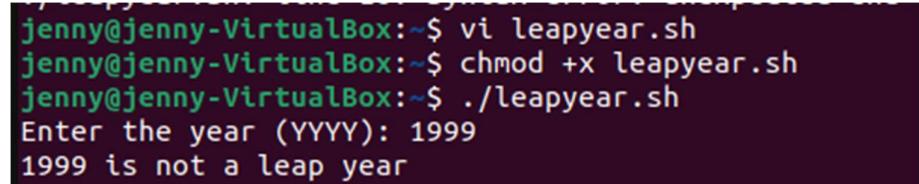
```
jenny@jenny-VirtualBox:~/Documents$ ./fibno.sh
Enter the limit: 4
Fibonacci Series:
0
1
1
2
```

14. Write a shell script to check if the current year is a leap year or not

```

#!/bin/bash
read -p "Enter the year (YYYY): " y
c1=$((y % 4))
c2=$((y % 100))
c3=$((y % 400))
if [ $c1 -eq 0 ]
then
if [ $c2 -ne 0 -o $c3 -eq 0 ]
then
echo "$y is a leap year" else
echo "$y is not a leap year" fi
else
echo "$y is not a leap year"
fi

```



```
jenny@jenny-VirtualBox:~/Documents$ vi leapyear.sh
jenny@jenny-VirtualBox:~/Documents$ chmod +x leapyear.sh
jenny@jenny-VirtualBox:~/Documents$ ./leapyear.sh
Enter the year (YYYY): 1999
1999 is not a leap year
```

Result

The program was executed and the result was successfully obtained. Thus, CO4 was obtained.

Experiment No.: 5

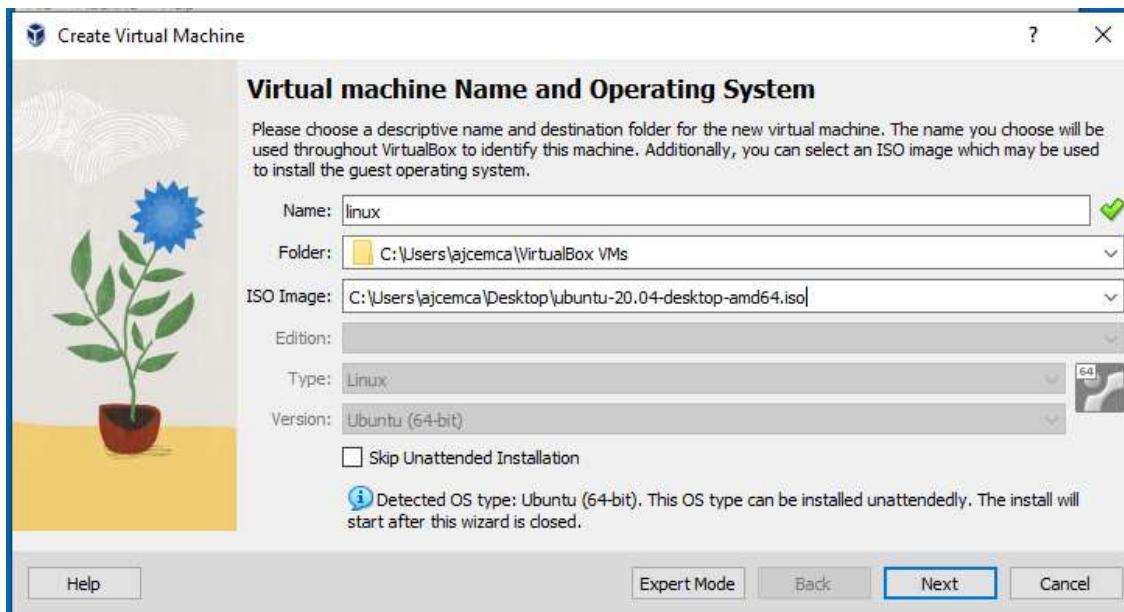
Aim: Install latest version of Ubuntu on a virtual box.

CO1: Install and configure common operating systems in virtual environment.

Procedure

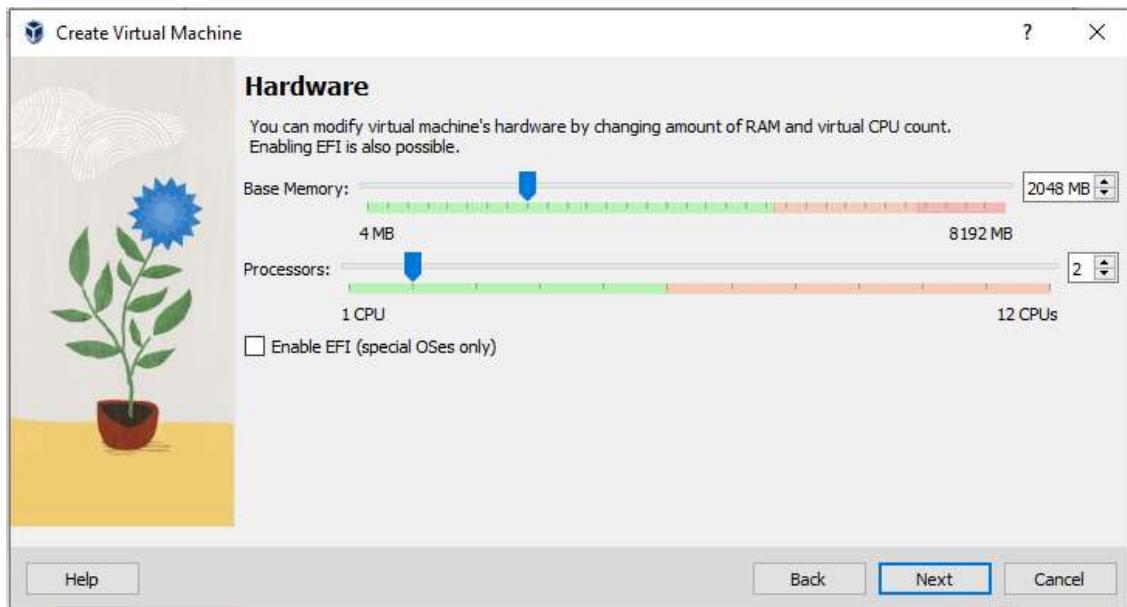


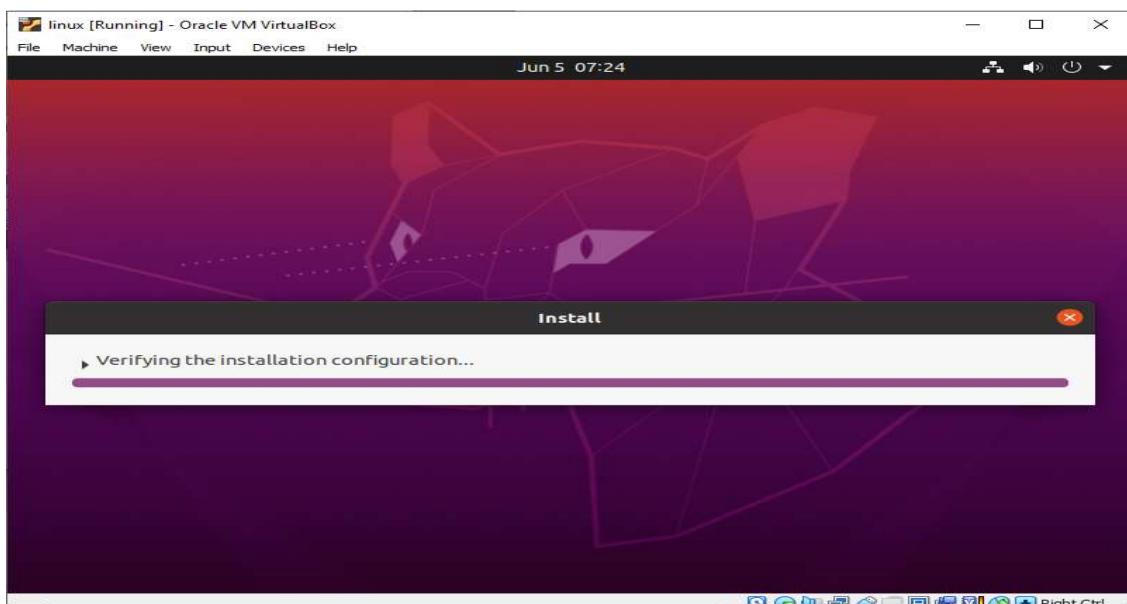
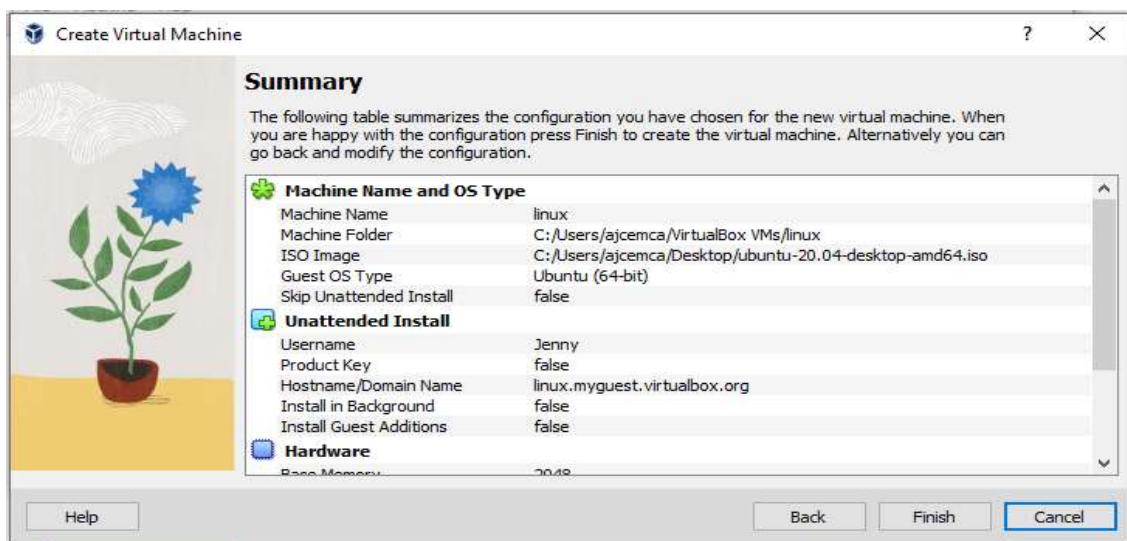
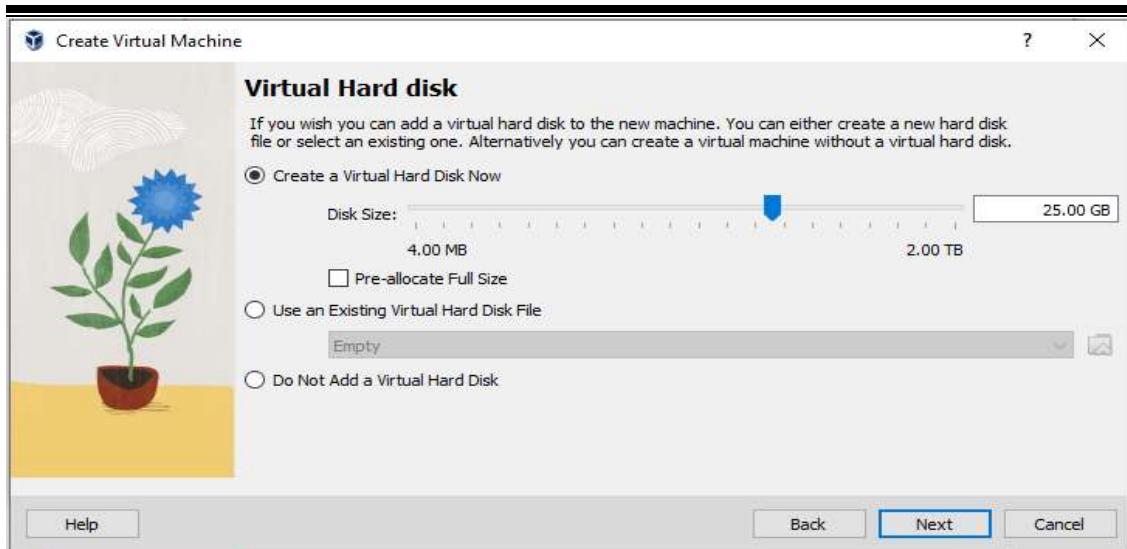
Installation of Linux

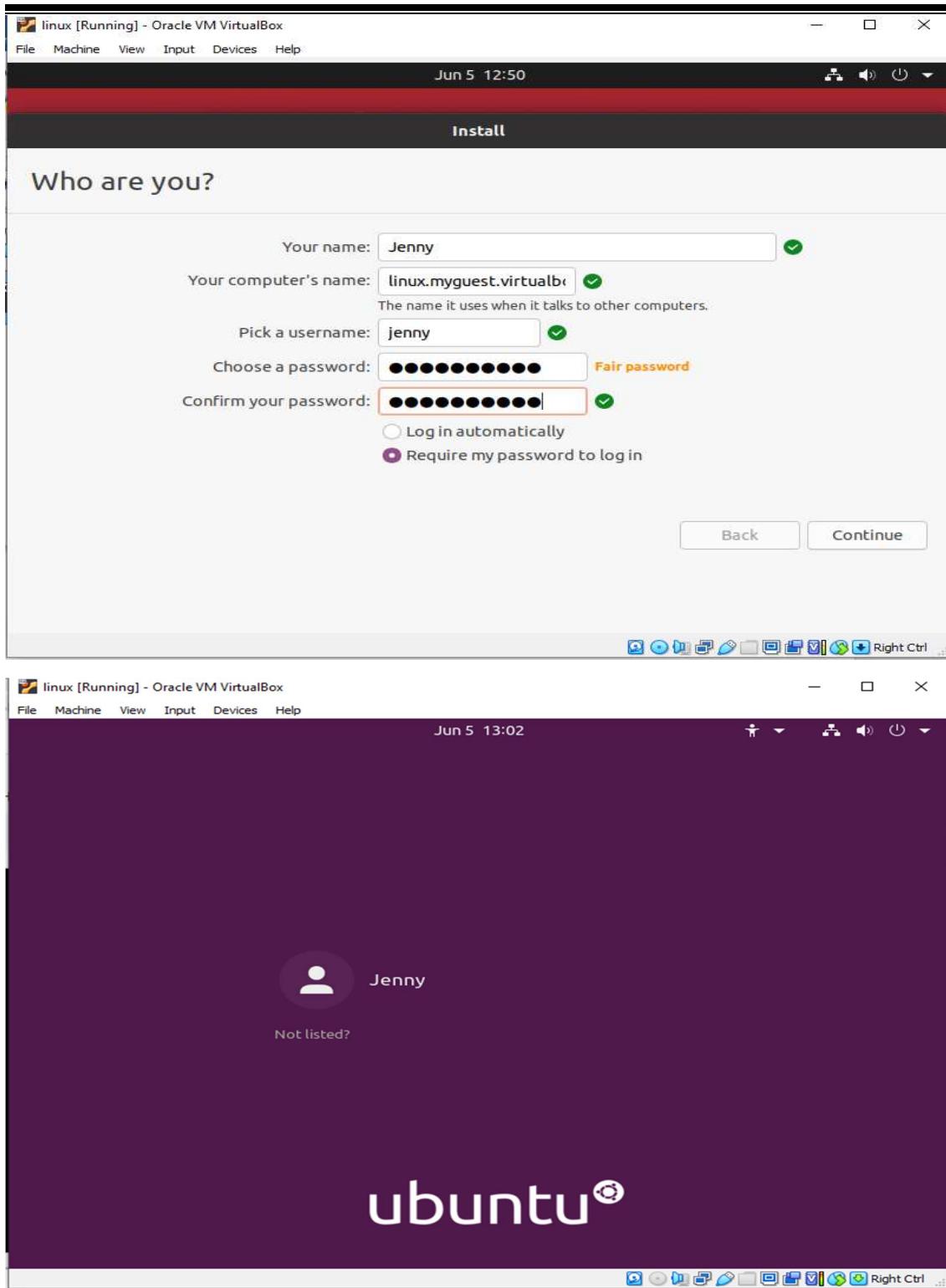




Setting up resources







Result

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

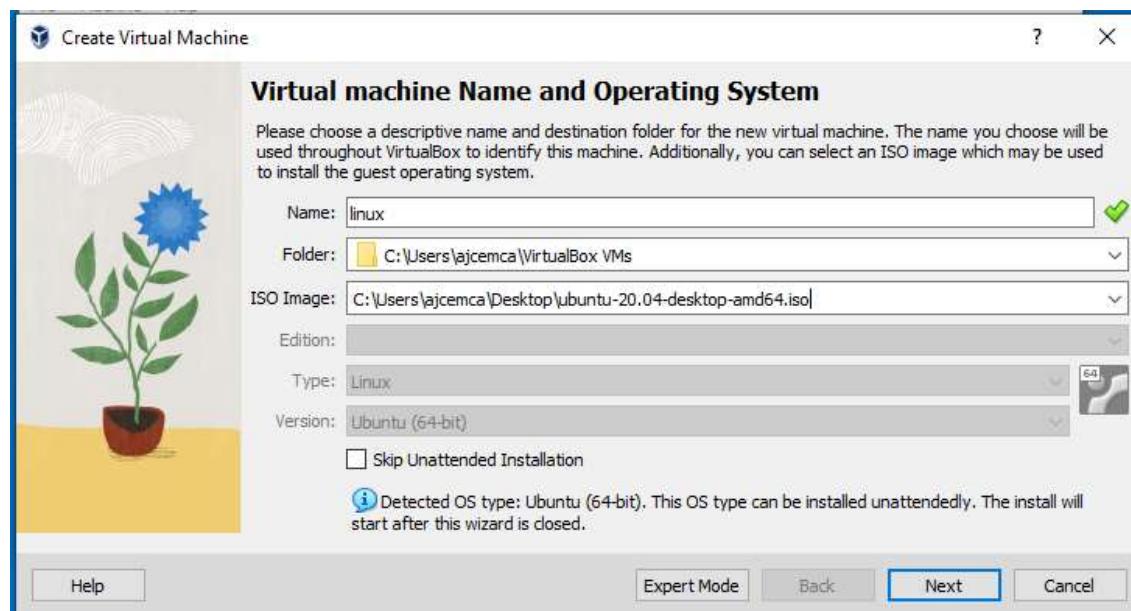
Experiment No.:6

Aim: Installation and configuration of LAMP stack. Deploy an open-source application such as phpmyadmin

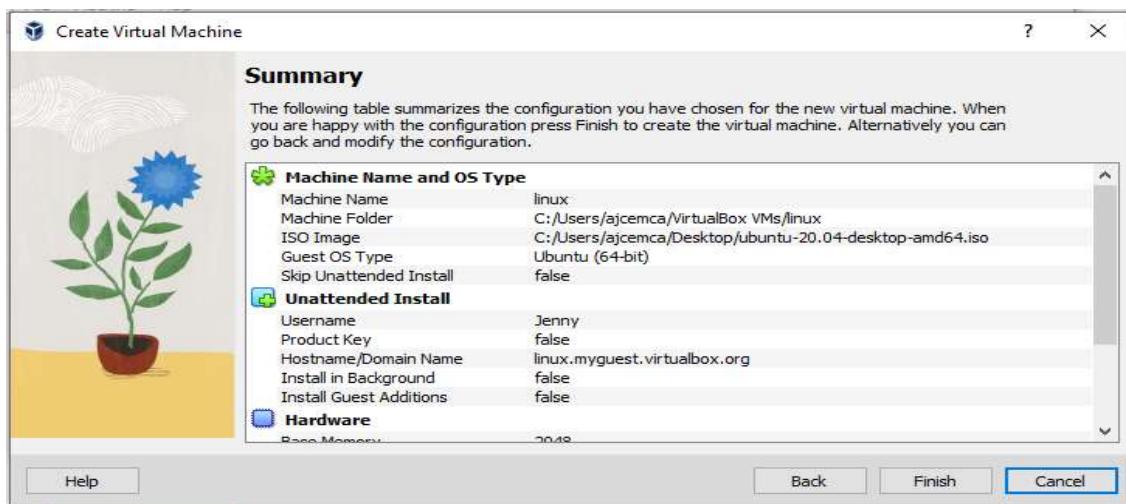
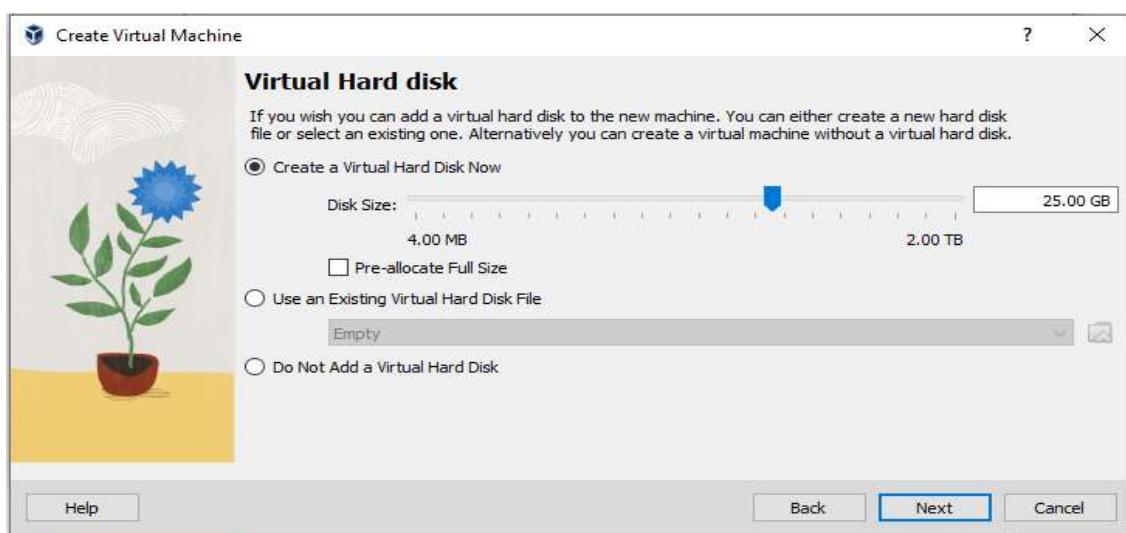
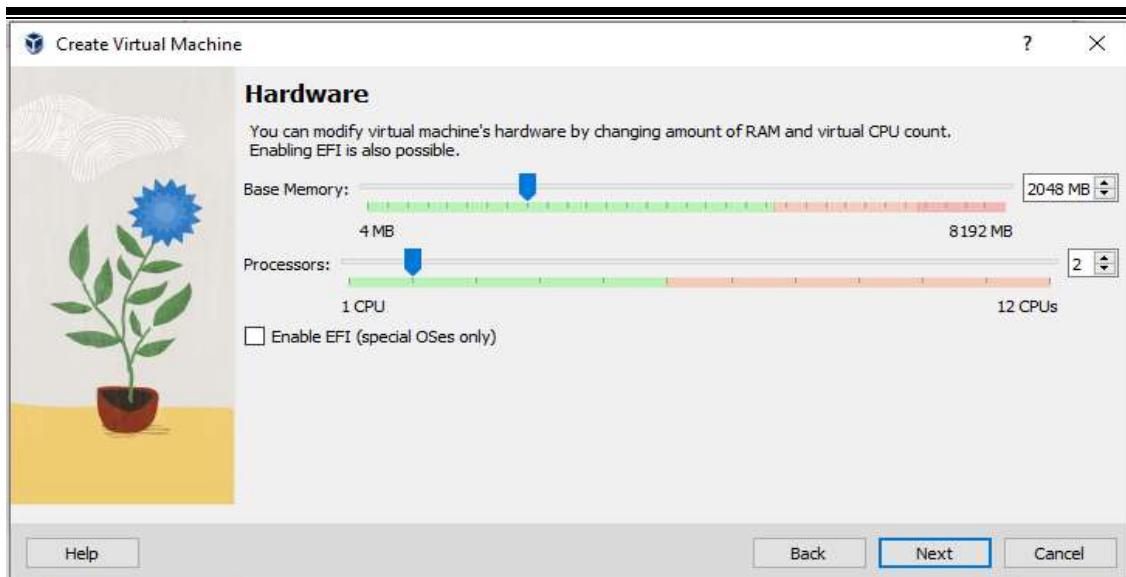
CO3: Install and manage servers for web applications.

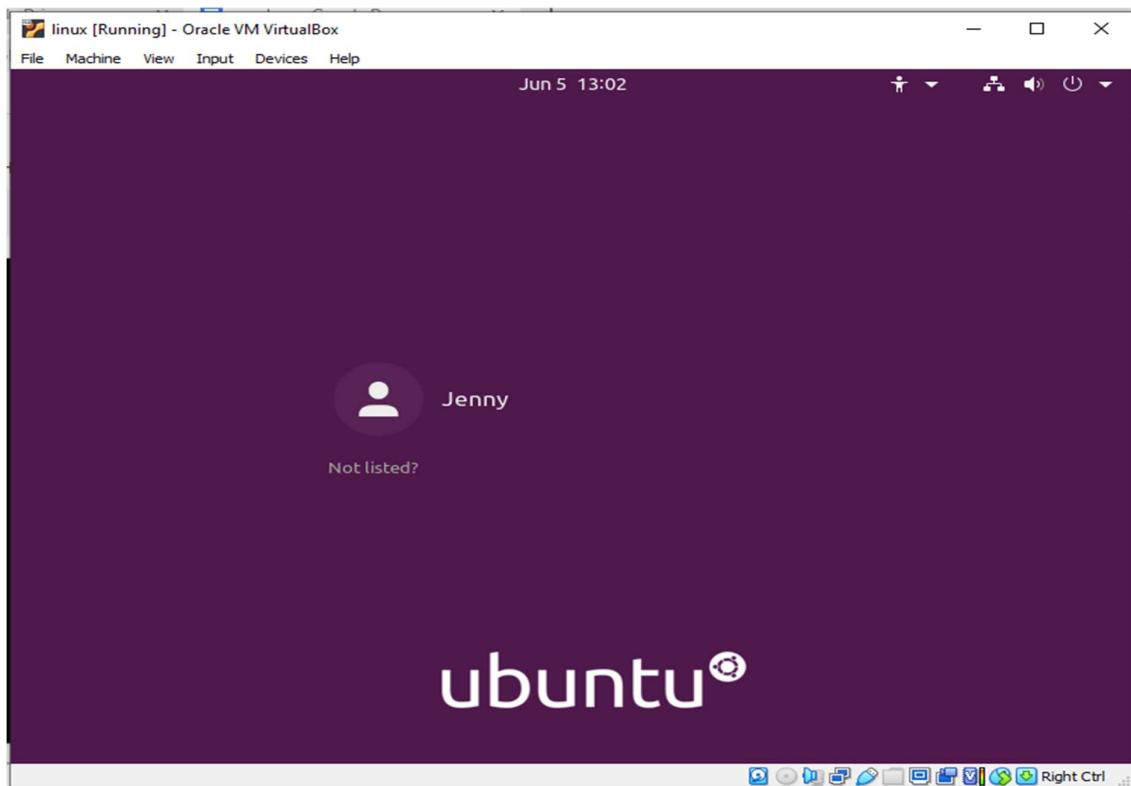
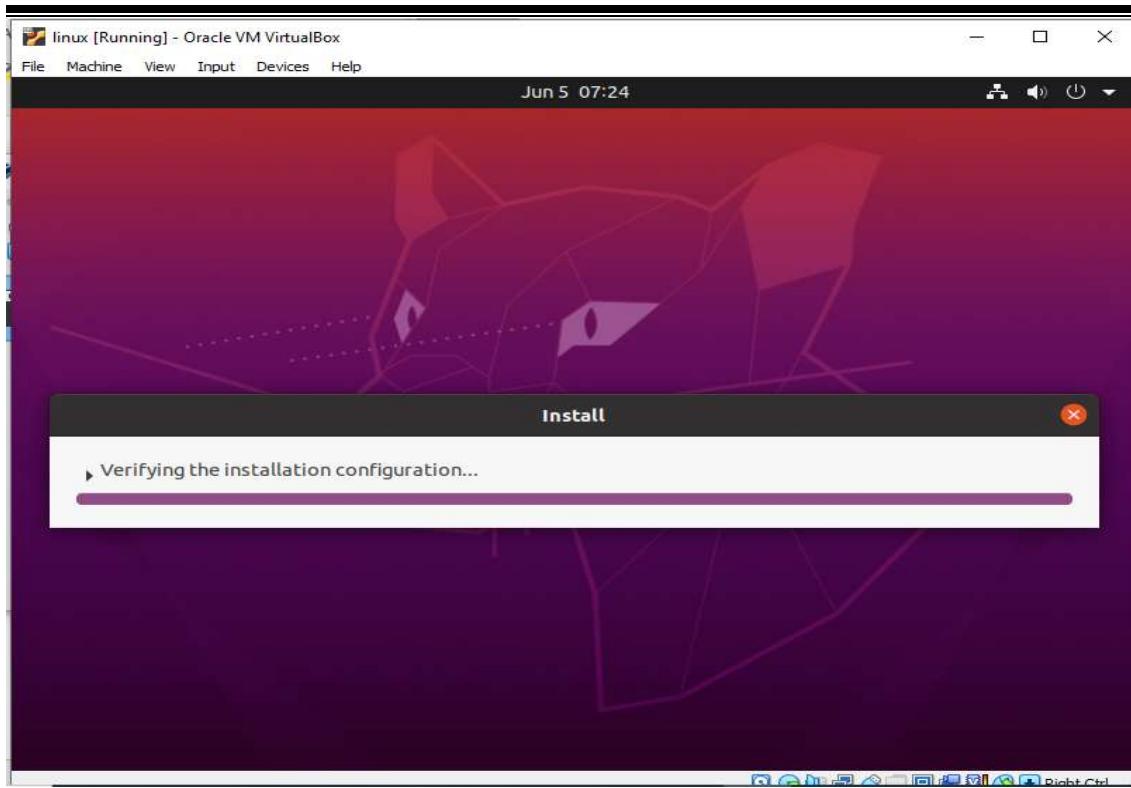
Procedure

Installation of Linux



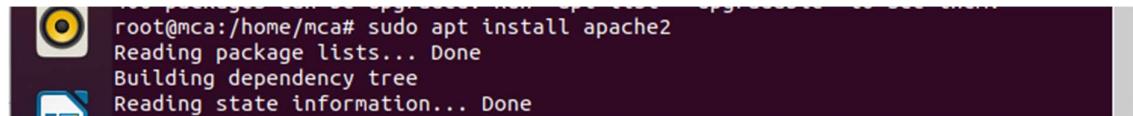
Setting up resources





Installation of Apache

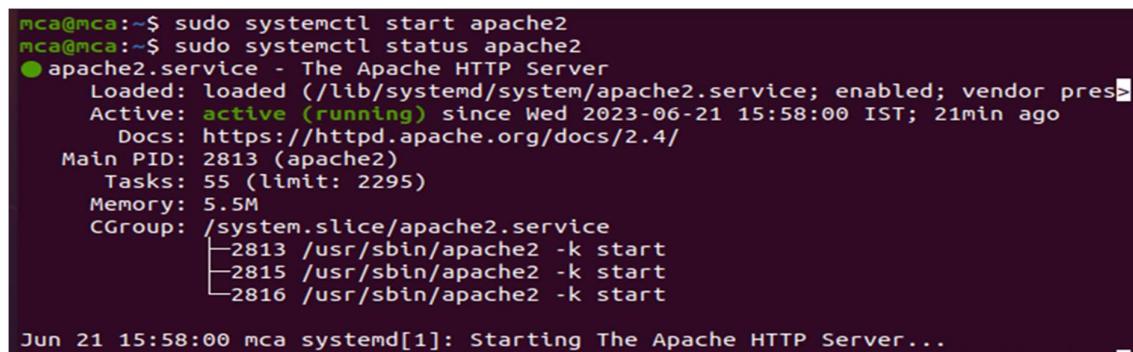
\$ apt install apache2



```
root@mca:/home/mca# sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Start Apache: \$ sudo systemctl start apache2

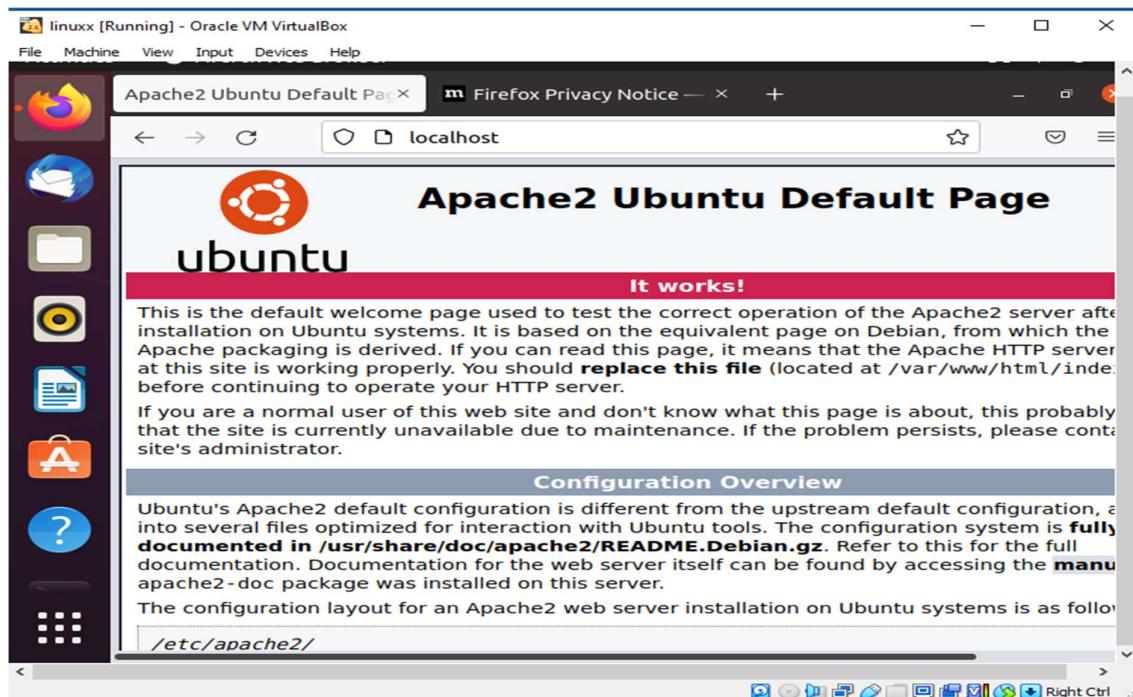
Check status of Apache: \$ sudo systemctl status apache2



```
mca@mca:~$ sudo systemctl start apache2
mca@mca:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pres>
   Active: active (running) since Wed 2023-06-21 15:58:00 IST; 21min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2813 (apache2)
      Tasks: 55 (limit: 2295)
     Memory: 5.5M
      CGroup: /system.slice/apache2.service
              └─2813 /usr/sbin/apache2 -k start
                  ├─2815 /usr/sbin/apache2 -k start
                  ├─2816 /usr/sbin/apache2 -k start

Jun 21 15:58:00 mca systemd[1]: Starting The Apache HTTP Server...
```

Check localhost



Installation of MySQL Server

\$ sudo apt install mysql-server -y

```
mca@mca:~$ sudo apt install mysql-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgifast-perl libcgipm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgifast-perl libcgipm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 16 newly installed, 0 to remove and 362 not upgraded.
Need to get 36.5 MB of archives.
```

Start MySQL

\$ sudo systemctl start mysql

Check status of MySQL

\$ sudo systemctl status mysql

```
mca@mca:~$ sudo systemctl start mysql
mca@mca:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2023-06-21 16:26:15 IST; 1min 24s ago
    Main PID: 29486 (mysqld)
      Status: "Server is operational"
        Tasks: 37 (limit: 2295)
       Memory: 354.2M
      CGroup: /system.slice/mysql.service
              └─29486 /usr/sbin/mysqld

Jun 21 16:26:14 mca systemd[1]: Starting MySQL Community Server...
Jun 21 16:26:15 mca_systemd[1]: Started MySQL Community Server.
```

Enter into MySQL from CLI

\$ sudo muysql

```
mca@mca:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.33-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

show databases;

```
mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.01 sec)
```

```
mysql> create database Jenny;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> show databases;
+-----+
| Database      |
+-----+
| Jenny         |
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.01 sec)
```

Installation of PHP

```
$apt install libapache2-mod-php php-mysql
```

```
root@jenny-VirtualBox:/home/jenny# apt install libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.1 php-common php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline

```

```
$systemctl restart apache2
```

```
root@jenny-VirtualBox:/home/jenny# systemctl restart apache2
root@jenny-VirtualBox:/home/jenny# systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2023-07-23 20:16:26 IST; 10s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 16976 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 16980 (apache2)
   Tasks: 6 (limit: 5510)
  Memory: 10.3M
     CPU: 38ms
```

Installation of phpMyAdmin

```
$apt install phpmyadmin
```

```
root@jenny-VirtualBox:/home/jenny# apt install phpmyadmin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

```

```
root@jenny-VirtualBox:/home/jenny# /var
bash: /var: Is a directory
root@jenny-VirtualBox:/home/jenny# cd /var
root@jenny-VirtualBox:/var# cd www
bash: cd: /www: No such file or directory
root@jenny-VirtualBox:/var# ls
backups cache crash lib local lock log mail metrics opt run snap spool tmp www
root@jenny-VirtualBox:/var# cd www
root@jenny-VirtualBox:/var/www# ls
html
root@jenny-VirtualBox:/var/www# cd html
root@jenny-VirtualBox:/var/www/html# cat >> hello.php
<?php
echo "Hello There...!";
phpinfo();
?>^Z
[8]+  Stopped                  cat >> hello.php
```

PHP Version 8.1.2-1ubuntu2.13

System	Linux jenny-VirtualBox 5.19.0-46-generic #47~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Jun 21 15:35:31 UTC 2 x86_64
Build Date	Jun 28 2023 14:01:49
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-mysqlind.ini, /etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/15-xml.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-curl.ini, /etc/php/8.1/apache2/conf.d/20-dom.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-fil.ini, /etc/php/8.1/apache2/conf.d/20-gd.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-icu.ini, /etc/php/8.1/apache2/conf.d/20-mbstring.ini, /etc/php/8.1/apache2/conf.d/20-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-simplexml.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshini.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini, /etc/php/8.1/apache2/conf.d/20-xmfreader.ini, /etc/php/8.1/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.1/apache2/conf.d/20-xsl.ini, /etc/php/8.1/apache2/conf.d/20-zip.ini

To enable phpMyAdmin open the /etc/apache2/apache2.conf file using a text editor.

```
$ vi /etc/apache2/apache2.conf
```

```
root@mca:/home/mca# vi /etc/apache2/apache2.conf
```

Include /etc/phpmyadmin/apache.conf

```
Include /etc/phpmyadmin/apache.conf
```

Restart apache.

go to localhost/phpmyadmin/

Welcome to phpMyAdmin

Language: English

Log in

Username:

Password:

Go

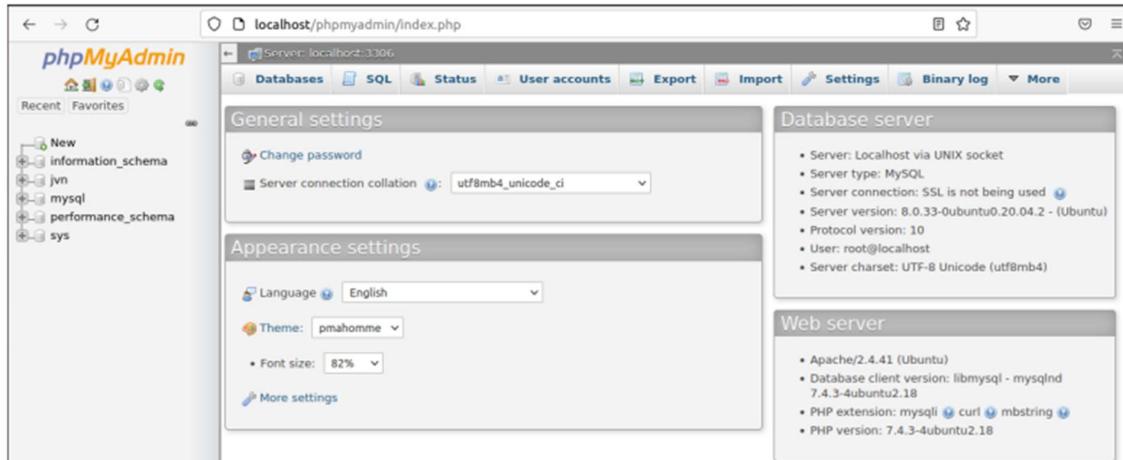
change password

```
mysql> alter user root@localhost identified with caching_sha2_password by "root";
Query OK, 0 rows affected (0.35 sec)
```

fetching some contents

```
mysql> select user,authentication_string,plugin, host from mysql.user;
+-----+-----+-----+-----+
| user | authentication_string | plugin | host |
+-----+-----+-----+-----+
| debian-sys-maint | $A$005$IsquoQ$xmYD\%MStmosXmws3pvInou5o4Hob\vrvc1641lRTpRfnFpp0 | caching_sha2_password | localhost |
| mysql.infoschema | $A$005$THISISACOMBINATIONOFLNVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED | caching_sha2_password | localhost |
| mysql.session | $A$005$THISISACOMBINATIONOFLNVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED | caching_sha2_password | localhost |
| mysql.sys | $A$005$THISISACOMBINATIONOFLNVALIDSALTANDPASSWORDTHATMUSTNEVERBEUSED | caching_sha2_password | localhost |
| root | | auth_socket | localhost |
+-----+-----+-----+-----+
5 rows in set (0.02 sec)
```

localhost/phpmyadmin/index.php



Result

The program was executed and the result was successfully obtained. Thus, CO3 was obtained.

Experiment No.:7

Aim: Build and install software from source code, familiarity with make and cmake utilities expected.

CO4: Write shell scripts required for system administration.

Procedure

1. Write a program to find factorial of a number using make utility

function1.cpp

```
#include "functions.h"
int factorial(int n){
    if(n!=1){
        return(n*factorial(n-1));
    }
    else return 1;
}
```

function2.cpp

```
#include <iostream>
#include "functions.h"
void print_hello(){
    std::cout << "Hello World";
}
```

functions.h

```
void print_hello();
int factorial(int n);
```

Main.cpp

```
#include <iostream>
#include "functions.h"
int main(){
    print_hello();
    std::cout << std::endl;
    std::cout << "the factorial of 5 is :" << factorial(5) << std::endl;
    return 0;
}
```

1. Open terminal, move to current directory.
 2. Make
 3. Ls
 4. ./hello
-

```

root@u3:/home/mca/Documents/sample_make# make
g++ main.cpp function1.cpp function2.cpp -o hello
root@u3:/home/mca/Documents/sample_make# ls
function1.cpp  function2.cpp  functions.h  hello  main.cpp  Makefile
root@u3:/home/mca/Documents/sample_make# ./hello
Hello World
the factorial of 5 is :120
root@u3:/home/mca# make --v
GNU Make 4.2.1

```

2. Write a program to add two numbers using cmake utility

add.cpp

```

#include "add.h"
int add(int a, int b)
{
    return a + b;
}

```

add.h

```

#pragma once
int add(int a, int b);

```

main.cpp

```

#include<iostream>
#include "add.h"
int main()
{
    std::cout << "Sum of 28 and 24 is " << add(28,24)<<"\n";
    return 0;
}

```

CMakeLists.txt

```

cmake_minimum_required(VERSION 3.16.3)
project("Hello World")
add_executable(a.out main.cpp add.cpp)

```

//add_executable: adds an executable target called < name> to be built from the source files listed in the command invocation

1. open terminal
2. move to the current folder

```
root@u3:/home/mca# cd Documents
root@u3:/home/mca/Documents# ls
sample_cmake  sample_make
root@u3:/home/mca/Documents# cd sample_cmake
```

3. Type the following commands : cmake .

```
root@u3:/home/mca/Documents/sample_cmake# cmake .
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
```

Make

```
root@u3:/home/mca/Documents/sample_cmake# make
Scanning dependencies of target a.out
[ 33%] Building CXX object CMakeFiles/a.out.dir/main.cpp.o
[ 66%] Building CXX object CMakeFiles/a.out.dir/add.cpp.o
[100%] Linking CXX executable a.out
[100%] Built target a.out
```

/a.out

```
root@u3:/home/mca/Documents/sample_cmake# ./a.out
Sum of 28 and 24 is 52
```

Result

The program was executed and the result was successfully obtained. Thus, CO4 was obtained.

Experiment No.:8

Aim: Introduction to command line tools for networking IPv4 networking, network commands: ping route traceroute, nslookup, ip.

CO5: Acquire skill sets required for DevOps.

Procedure

1. **ifconfig:** ifconfig command is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary

```
mca@u3:~$ ifconfig
enp5s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.6.173 netmask 255.255.255.0 broadcast 192.168.6.255
        inet6 fe80::1efd:d0a3:b384:c68 prefixlen 64 scopeid 0x20<link>
          ether 0c:9d:92:0f:6b:fd txqueuelen 1000 (Ethernet)
            RX packets 141067 bytes 96745845 (96.7 MB)
            RX errors 0 dropped 158 overruns 0 frame 0
            TX packets 38889 bytes 6864461 (6.8 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 4951 bytes 483464 (483.4 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 4951 bytes 483464 (483.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. **ifup and ifdown:** The ifup command is used to activate (up) a network interface, ifdown deactivates (down) it.

```
ifdown: unknown interface enp5s0
mca@u3:~$ sudo ifconfig enp5s0 down
mca@u3:~$ sudo ifconfig enp5s0 up
```

3. used to identify the connection between host and server. Used for detecting devices on a network and for troubleshooting problem

```
mca@u3:~$ ping google.com
PING google.com (142.250.182.78) 56(84) bytes of data.
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=1 ttl=248 time=16.0 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=2 ttl=248 time=16.0 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=3 ttl=248 time=15.9 ms
^Z
[8]+  Stopped                  ping google.com
mca@u3:~$ ping 192.168.6.174
PING 192.168.6.174 (192.168.6.174) 56(84) bytes of data.
64 bytes from 192.168.6.174: icmp_seq=1 ttl=64 time=0.399 ms
^Z
[9]+  Stopped                  ping 192.168.6.174
mca@u3:~$ ping -c 2 142.250.193.110
PING 142.250.193.110 (142.250.193.110) 56(84) bytes of data.
64 bytes from 142.250.193.110: icmp_seq=1 ttl=248 time=15.2 ms
64 bytes from 142.250.193.110: icmp_seq=2 ttl=248 time=15.2 ms
--- 142.250.193.110 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
```

4.traceroute: It is used to identify the route taken by the packet to reach the destination.

```
mca@u3:~$ traceroute google.com
traceroute to google.com (142.250.182.78), 30 hops max, 60 byte packets
 1 _gateway (192.168.6.100)  0.139 ms  0.125 ms  0.134 ms
 2 136.232.57.109 (136.232.57.109)  1.465 ms  1.436 ms  1.542 ms
 3 172.20.97.57 (172.20.97.57)  14.543 ms  14.491 ms  14.631 ms
 4 172.27.9.126 (172.27.9.126)  16.426 ms  16.373 ms  16.530 ms
 5 172.27.9.125 (172.27.9.125)  16.476 ms  17.274 ms  16.555 ms
 6 172.27.109.51 (172.27.109.51)  16.698 ms  16.086 ms  16.066 ms
```

5.whois: The whois system is a listing of records that contains details about both the ownership of domains and the owners.

```
mca@u3:~$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
```

6.nslookup: used to check DNS records propagation and resolution using different servers, and perform other troubleshooting steps.

```
mca@u3:~$ nslookup google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.193.110
Name:   google.com
Address: 2404:6800:4007:81f::200e
```

Result

The program was executed and the result was successfully obtained. Thus, CO5 was obtained.

Experiment No.:9

Aim: Analyzing network packet stream using tcpdump and wireshark. Perform basic network service tests using nc.

CO5: Acquire skill sets required for DevOps.

Procedure

tcpdump: used to capture the packets of the current network interface.

```
mca@u3:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp5s0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:49:07.138536 IP maa03s46-in-f14.1e100.net.https > u3.36990: Flags [P.], seq 513607617:513608289, ack 4281923877, win 5317, options [nop,nop
,TS val 221157450 ecr 1755977580], length 672
15:49:07.139289 IP maa03s46-in-f14.1e100.net.https > u3.36990: Flags [P.], seq 672:703, ack 1, win 5317, options [nop,nop,TS val 221157451 ecr
1755977580], length 31
```

a) **tcpdump -D:** used to display the current network interfaces.

```
mca@u3:~$ sudo tcpdump -D
1.enp5s0 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
```

b) **tcpdump -c 5 -i:** used to capture packets from one interface.

```
mca@u3:~$ sudo tcpdump -c 5 -i enp5s0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp5s0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:51:16.708365 ARP, Request who-has 192.168.6.146 tell _gateway, length 46
15:51:16.710108 IP u3.35210 > dns.google.domain: 34499+ [1au] PTR? 146.6.168.192.in-addr.arpa. (55)
15:51:16.725954 IP dns.google.domain > u3.35210: 34499 NXDomain 0/0/1 (55)
15:51:16.726189 IP u3.35210 > dns.google.domain: 34499+ PTR? 146.6.168.192.in-addr.arpa. (44)
15:51:16.735420 ARP, Request who-has 192.168.6.95 tell 192.168.6.56, length 46
5 packets captured
22 packets received by filter
0 packets dropped by kernel
```

c) **tcpdump -A -i:** to capture packets in ASCII format.

```
mca@u3:~$ sudo tcpdump -A -i enp5s0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp5s0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:51:54.286494 ARP, Request who-has 192.168.1.1 tell 192.168.1.1, length 46
.....8.@@.....
15:51:54.287299 IP u3.57822 > dns.google.domain: 43032+ [1au] PTR? 1.1.168.192.in-addr.arpa. (53)
E..Q..@.@@.....5.=.....1.1.168.192.in-addr.arpa.....).....
15:51:54.288701 ARP, Reply 192.168.1.1 is-at 44:31:92:f1:0a:8c (oui Unknown), length 46
.....D1..
```

d) **tcpdump -XX -i:** to capture packets in hexadecimal format.

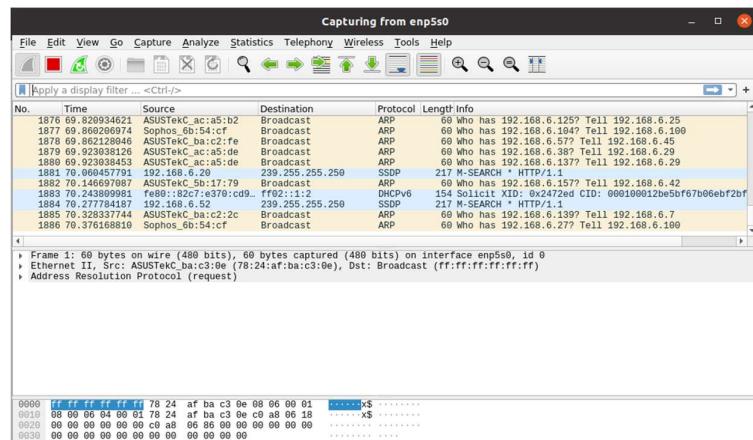
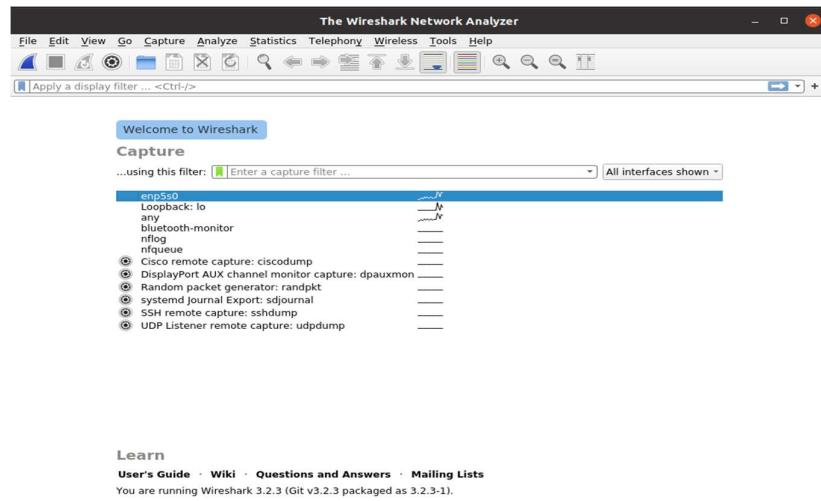
```
mca@u3:~$ sudo tcpdump -XX -i enp5s0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp5s0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:52:59.775355 ARP, Request who-has 192.168.6.107 tell _gateway, length 46
0x0000: ffff ffff ffff 001a 8c6b 54cf 0806 0001 .....kT....
0x0010: 0800 0604 0001 001a 8c6b 54cf c0a8 0664 .....kT....d
```

e) **tcpdump -w filename -i enp5s0:** used to write the data on a file.

```
mca@u3:~$ sudo tcpdump -w file.txt -i enp5s0
tcpdump: listening on enp5s0, link-type EN10MB (Ethernet), capture size 262144 bytes
^Z
[13]+ Stopped                  sudo tcpdump -w file.txt -i enp5s0
mca@u3:~$ ls
Desktop Documents Downloads file.txt Music Pictures Public Templates understanding-the-networking-commands
mca@u3:~$
```

wireshark: is a network protocol analyzer, or an application that captures packets from a network connection

```
mca@u3:~$ sudo wireshark
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
mca@u3:~$
```



Result

The program was executed and the result was successfully obtained. Thus, CO5 was obtained.

Experiment No.:10

Aim: Installation of KVM and perform basic KVM Commands

CO3: Install and manage servers for web applications.

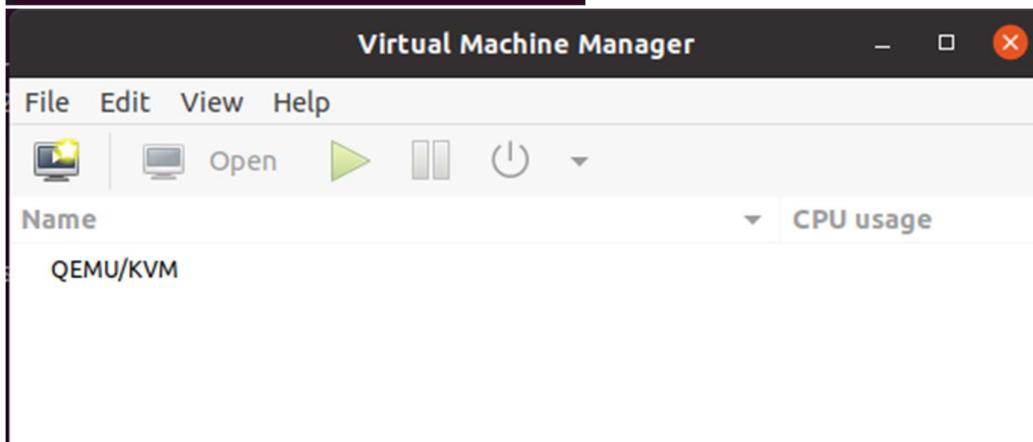
Procedure

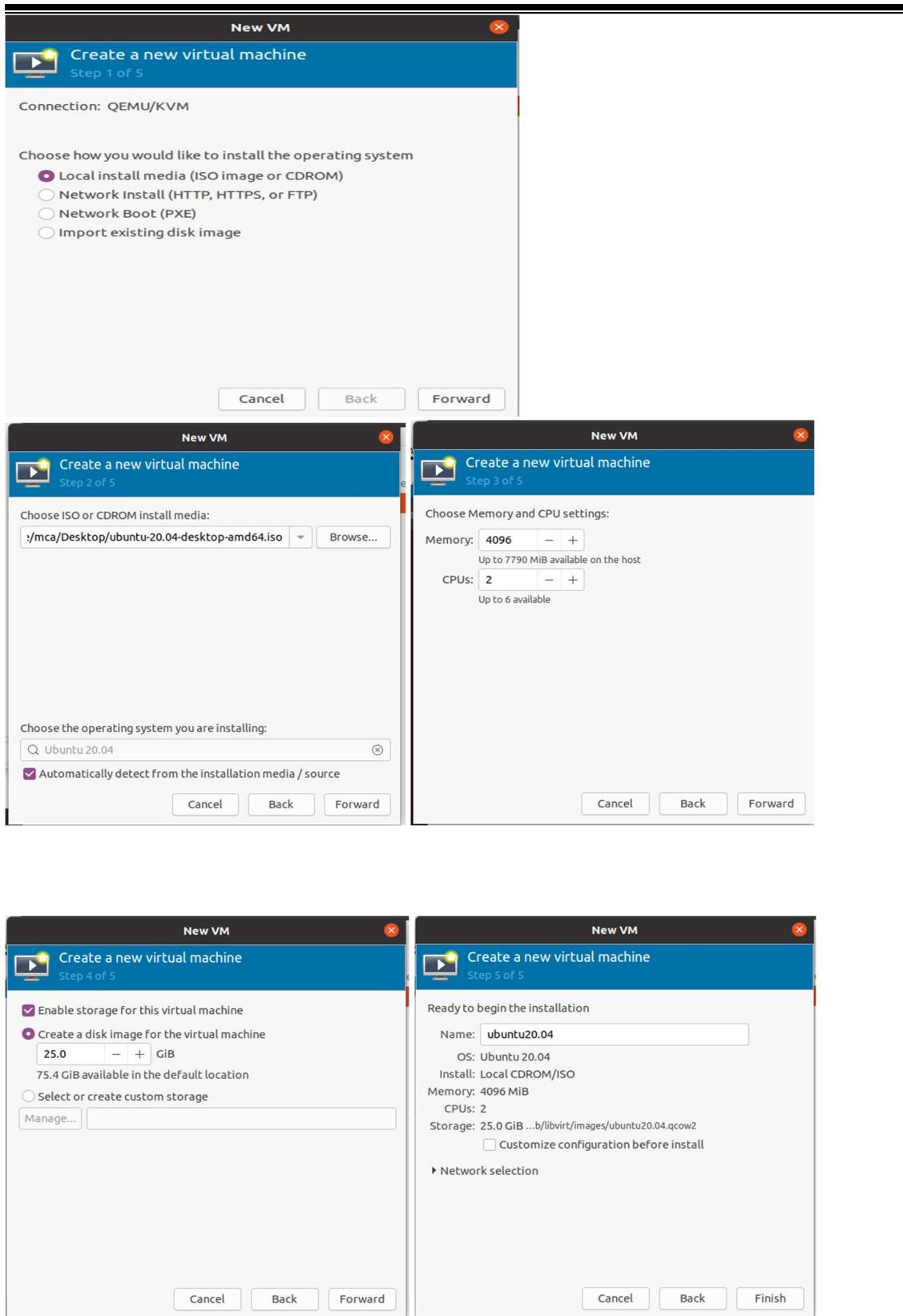
- **egrep -c '(vmx|svm)' /proc/cpuinfo:** it gives the cpu information. If the output is greater than 0, then your system is compactable.

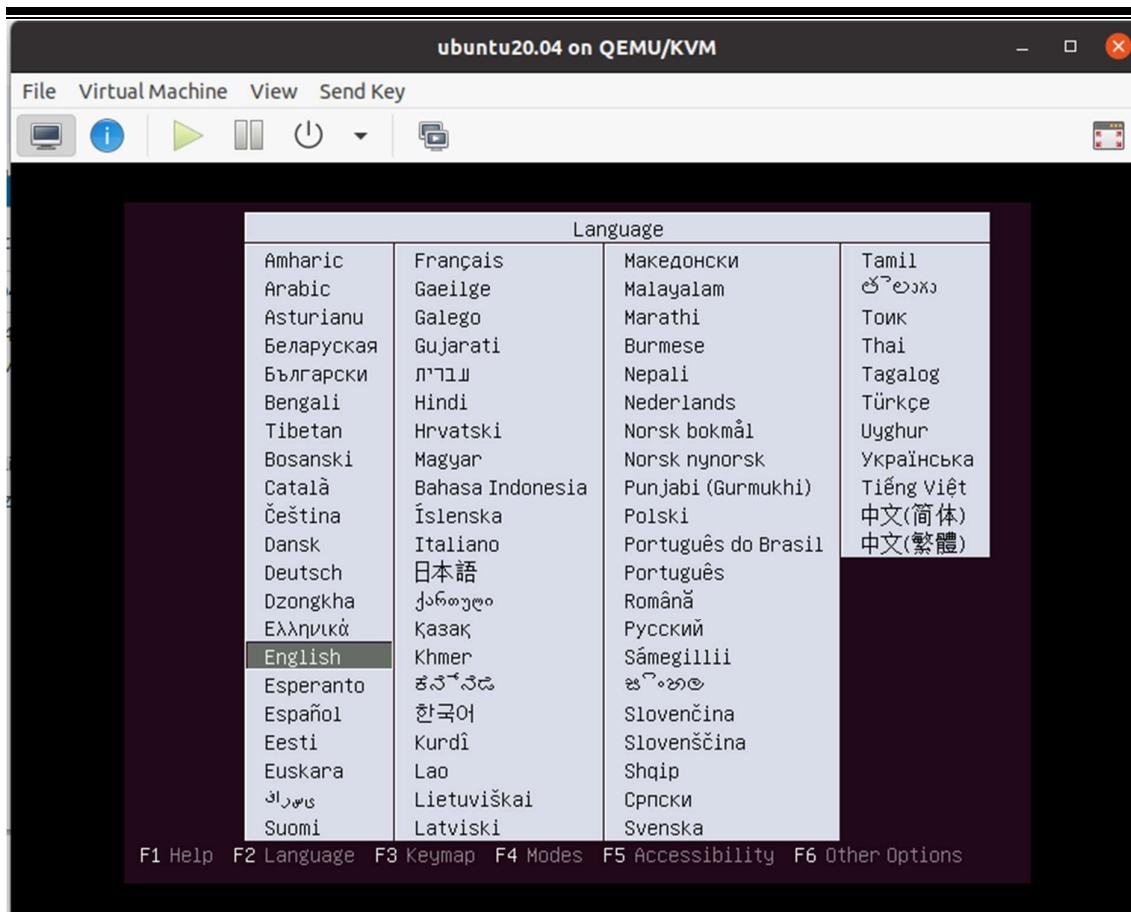
```
root@u3:/home/mca# egrep -c '(vmx|svm)' /proc/cpuinfo
12
```

- **kvm -ok:** check whether the CPU we're running on supports KVM acceleration.
- **apt -get install qemu qemu-kvm libvirt-daemon-system bridge-utils virt-manager virt-viewer -y:** installing kvm
- **libvirt:** library, used to interface with different virtualisation technologies.
- **virt-manager:** a graphical user front end for the libvirt library which provides vm management services.
- **virt-viewer:** displaying the graphical console of a guest user.

```
root@u3:/home/mca# kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
root@u3:/home/mca# apt-get install qemu qemu-kvm libvirt-daemon-system bridge-utils virt-manager virt-viewer -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
root@u3:/home/mca# apt install virt-manager
Reading package lists... Done
root@u3:/home/mca# virt-manager
root@u3:/home/mca#
```







virsh(virtual shell): flexible command line utility for managing vm controlled by libvirt.

1. sudo virsh list

```
mca@u3:~$ virsh list
 Id  Name      State
 -----
 1   ubuntu20.04  running
```

2. sudo virsh list --all

```
mca@u3:~$ sudo virsh list --all
 Id  Name      State
 -----
 1   ubuntu20.04  running
```

3. sudo virsh nodeinfo

```
mca@u3:~$ sudo virsh nodeinfo
CPU model:          x86_64
CPU(s):             6
CPU frequency:      2800 MHz
CPU socket(s):      1
Core(s) per socket: 6
Thread(s) per core: 1
NUMA cell(s):       1
Memory size:        7977500 KiB
```

3. sudo virsh help list

```
mca@u3:~$ sudo virsh help list
NAME
  list - list domains

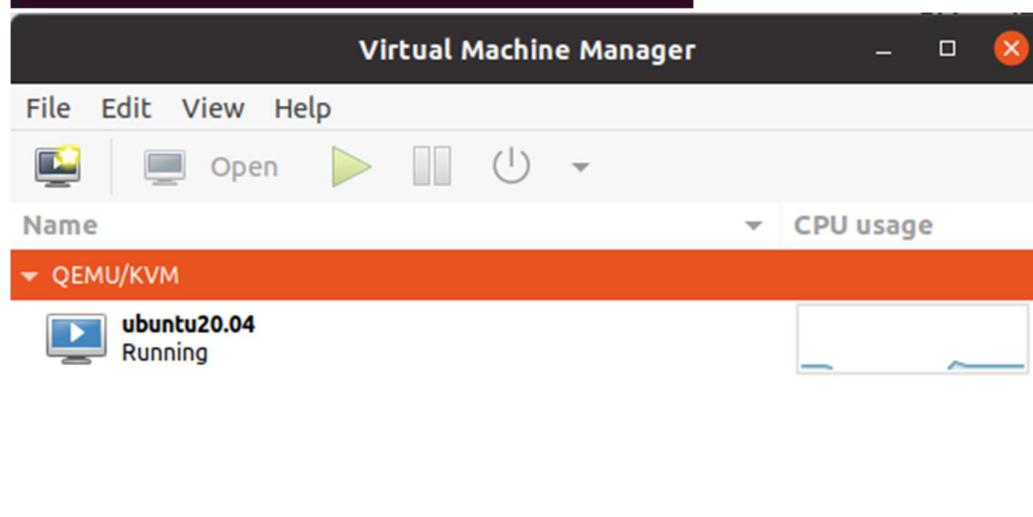
SYNOPSIS
  list [--inactive] [--all] [--transient] [--persistent] [--with-snapshot] [--without-snapshot]
  [--state-running] [--state-paused] [--state-shutoff] [--state-other] [--autostart] [--no-autostart]
  [--uuid] [--name] [--table] [--managed-save] [--title]

DESCRIPTION
  Returns list of domains.

OPTIONS
  --inactive      list inactive domains
  --all          list inactive & active domains
  --transient    list transient domains
  --persistent   list persistent domains
```

5. sudo virsh start <vmname>

```
mca@u3:~$ sudo virsh start ubuntu20.04
Domain ubuntu20.04 started
```

**6. sudo virsh reboot <vmname>****7. sudo virsh dominfo <vmname>**

```
mca@u3:~$ sudo virsh reboot ubuntu20.04
Domain ubuntu20.04 is being rebooted

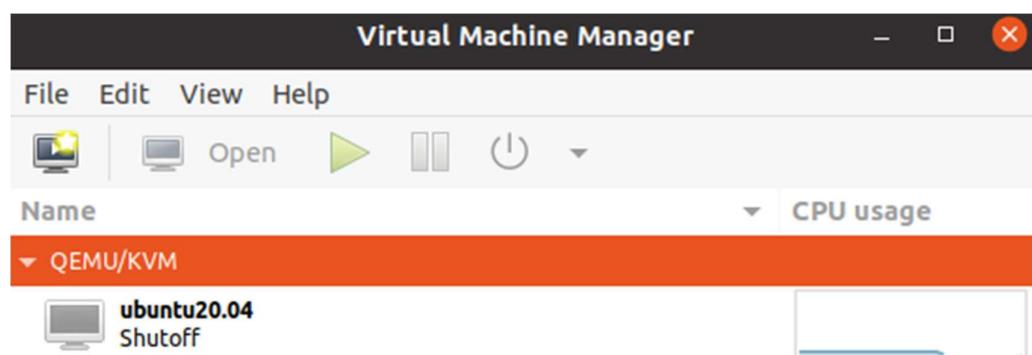
mca@u3:~$ sudo virsh dominfo ubuntu20.04
Id:                 1
Name:               ubuntu20.04
UUID:              fdd09262-69b4-4ad8-bfa4-f7a9816ee49b
OS Type:            hvm
State:              running
CPU(s):             2
CPU time:           163.9s
Max memory:         4194304 KiB
Used memory:        4194304 KiB
Persistent:         yes
Autostart:          disable
Managed save:       no
Security model:    apparmor
Security DOI:      0
Security label:    libvirt-fdd09262-69b4-4ad8-bfa4-f7a9816ee49b (enforcing)
```

8. sudo virsh suspend <vmname>**9. sudo virsh resume <vmname>****10.sudo virsh shutdown <vmname>**

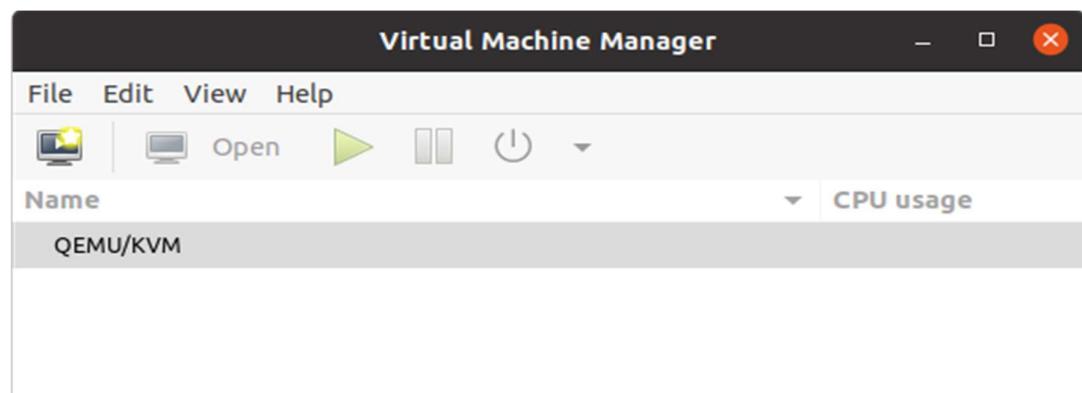
```
mca@u3:~$ sudo virsh suspend ubuntu20.04
Domain ubuntu20.04 suspended

mca@u3:~$ sudo virsh resume ubuntu20.04
Domain ubuntu20.04 resumed

mca@u3:~$ sudo virsh shutdown ubuntu20.04
Domain ubuntu20.04 is being shutdown
```



```
11.sudo virsh destroy <vmname>
12.sudo virsh undefine --domain <vmname> --remove-all-storage
mca@u3:~$ sudo virsh destroy ubuntu20.04
Domain ubuntu20.04 destroyed
mca@u3:~$ sudo virsh undefine --domain ubuntu20.04 --remove-all-storage
Domain ubuntu20.04 has been undefined
Volume 'vda'(/var/lib/libvirt/images/ubuntu20.04.qcow2) removed.
```



Result

The program was executed and the result was successfully obtained. Thus, CO3 was obtained.

Experiment No.:11

Aim: Docker, installation and deployment.

CO3: Install and manage servers for web applications.

Procedure

1.connect to instance

```
root@ip-172-31-33-80:~# ssh -i "dockerimg.pem" ec2-user@ec2-13-234-17-45.ap-south-1.compute.amazonaws.com
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ajcemca\Downloads>ssh -i "dockerimg.pem" ec2-user@ec2-13-234-17-45.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-234-17-45.ap-south-1.compute.amazonaws.com (13.234.17.45)' can't be established.
ECDSA key fingerprint is SHA256:SN6E4d2FOG8gIxNo+4o3v6+8P4epc0LkvkGVMH3Ra8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? Yes
Warning: Permanently added 'ec2-13-234-17-45.ap-south-1.compute.amazonaws.com,13.234.17.45' (ECDSA) to the list of known hosts.

              _\|_ / 
             _\|_|_|_ | Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-33-80 ~]$ pwd
/home/ec2-user
```

2.**which docker:** used to locate docker

3.**yum install docker -y:** Installing docker

4.**docker –version:** to check the version

```
[root@ip-172-31-33-80 ec2-user]# which docker
/usr/bin/which: no docker in (/sbin:/bin:/usr/sbin:/usr/bin)
[root@ip-172-31-33-80 ec2-user]# yum install docker -y

Complete!
[root@ip-172-31-33-80 ec2-user]# which docker
/bin/docker
[root@ip-172-31-33-80 ec2-user]# docker --version
Docker version 20.10.23, build 7155243
[root@ip-172-31-33-80 ec2-user]# docker info
```

5.**service docker start:** to start docker

```
[root@ip-172-31-33-80 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-33-80 ec2-user]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
     Active: active (running) since Fri 2023-06-23 09:09:25 UTC; 7s ago
       Docs: https://docs.docker.com
   Process: 3517 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Process: 3516 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 3520 (dockerd)
    CGroup: /system.slice/docker.service
```

6.**docker images:** This command is used to list all the Docker images that are currently available on your system.

```
[root@ip-172-31-33-80 ec2-user]# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
[root@ip-172-31-33-80 ec2-user]# docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
[root@ip-172-31-33-80 ec2-user]# docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
```

7.docker pull Ubuntu: This command is used to download a Docker image(Ubuntu) from a registry.

```
[root@ip-172-31-33-80 ec2-user]# docker pull ubuntu
Using default tag: latest
```

8.docker run -it –name jenny <image> /bin/bash: This command is used to download a Docker image from a registry if it is not available and then creates the container.

run: creates container

-it: iterative terminal

```
[root@ip-172-31-33-80 ec2-user]# docker run -it ubuntu /bin/bash
```

```
[root@ip-172-31-33-80 ec2-user]# docker run -it --name jenny ubuntu /bin/bash
```

```
[root@ip-172-31-33-80 ec2-user]# docker run -it --name jen hello-world
```

9.docker ps -a: This command is used to list all the Docker containers.

ps: process

-a: all

```
[root@ip-172-31-33-80 ec2-user]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
dcf27de1c786 mongo "docker-entrypoint.s..." 15 seconds ago Exited (0) 3 seconds ago je
713e1c0c81e4 hello-world "/hello" 3 minutes ago Exited (0) 3 minutes ago jen
c4a4473ca7a0 ubuntu "/bin/bash" 7 minutes ago Exited (0) 7 minutes ago Jenny
aab2a959efd3 ubuntu "/bin/bash" 10 minutes ago Exited (0) 9 minutes ago quizzical_hertz
```

10.docker start <container name>: starts the specified container

11.docker attach <container name>:attach local standard input, output, and error streams to a running container.

12.cat > dfile.txt: creating a file inside the container and adding contents.

13.ls: listing the contents in the container

```
[root@ip-172-31-33-80 ec2-user]# docker start jenny
jenny
[root@ip-172-31-33-80 ec2-user]# docker attach jenny
root@c4a4473ca7a0:/# cat > dfile.txt
jenny
^Z
[1]+  Stopped                  cat > dfile.txt
root@c4a4473ca7a0:/# cat dfile.txt
jenny
root@c4a4473ca7a0:/# ls
bin  boot  dev  dfile.txt  etc  home  lib  lib32  lib64
```

14.apt install apache2: installing apache in the container

```
root@c4a4473ca7a0:/# apt install apache2 -y
Reading package lists... Done
```

15.which apache2: file path

16.apache2 -v: version of apache

```
root@c4a4473ca7a0:/# which apache2
/usr/sbin/apache2
root@c4a4473ca7a0:/# apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2023-03-01T22:43:55
root@c4a4473ca7a0:/#
```

17.docker ps : This command is used to list all the running Docker containers.
ps: process

```
[root@ip-172-31-33-80 ec2-user]# docker start jenny
jenny
[root@ip-172-31-33-80 ec2-user]# docker ps
CONTAINER ID        IMAGE       COMMAND       CREATED          STATUS          PORTS     NAMES
c4a4473ca7a0        ubuntu      "/bin/bash"   22 minutes ago   Up 16 seconds
[root@ip-172-31-33-80 ec2-user]# docker start jenny
```

18.tree: displays contents in tree structure

```
root@41699f5f4796:/tmp# tree
.
|-- f1.txt
|-- f2.txt
`-- jenny
```

19.docker rename <current container name> <new container name>: renames the container name.

```
[root@ip-172-31-36-203 ec2-user]# docker ps -a
CONTAINER ID        IMAGE       COMMAND       CREATED          STATUS          PORTS     NAMES
41699f5f4796        image-jenny   "/bin/bash"   4 days ago     Exited (0) 4 days ago
[jroot@ip-172-31-36-203 ec2-user]# docker rename j-container container-j
[root@ip-172-31-36-203 ec2-user]# docker ps -a
CONTAINER ID        IMAGE       COMMAND       CREATED          STATUS          PORTS     NAMES
41699f5f4796        image-jenny   "/bin/bash"   4 days ago     Exited (0) 4 days ago
[jroot@ip-172-31-36-203 ec2-user]#
```

20.docker commit <container name> <imagename>: to convert a container as an image

21.docker run -it --name jen-container jen-img: creating a container using the image

```
[root@ip-172-31-36-203 ec2-user]# docker commit jenny image-jenny
sha256:516f5f0b346c16ea872759ac38038f82614b163f463028f6bcd427c773c1f176
[root@ip-172-31-36-203 ec2-user]# docker images
REPOSITORY          TAG        IMAGE ID      CREATED         SIZE
image-jenny        latest    516f5f0b346c  7 seconds ago  227MB
ubuntu              latest    99284ca6cea0  4 weeks ago   77.8MB
[root@ip-172-31-36-203 ec2-user]# docker run -it --name j-container image-jenny
root@41699f5f4796:/# ls
bin  boot  dev  etc  home  jen  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@41699f5f4796:/# cd tmp
root@41699f5f4796:/tmp# ls
f1.txt  f2.txt  jenny
root@41699f5f4796:/tmp# which apache2
/usr/sbin/apache2
root@41699f5f4796:/tmp# tree
.
|-- f1.txt
|-- f2.txt
`-- jenny
```

22.docker login: login to docker hub

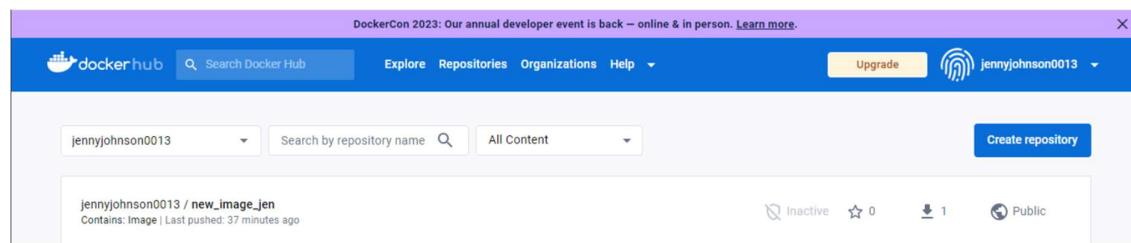
```
[root@ip-172-31-36-203 ec2-user]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: jennyjohnson0013
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

23.docker tag image-jenny jennyjohnson0013/new_image_jen: tagging image in docker hub

24.docker push jennyjohnson0013/new_image_jen: pushing image to dockerhub.

```
[root@ip-172-31-36-203 ec2-user]# docker tag image-jenny jennyjohnson0013/new_image_jen
[root@ip-172-31-36-203 ec2-user]# docker push jennyjohnson0013/new_image_jen
Using default tag: latest
The push refers to repository [docker.io/jennyjohnson0013/new_image_jen]
```



The pushed image is pulled using another instance

Install and start docker in new instance

```
aws Services Q Search [Alt+S]
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-32-4 ~]$ sudo su
[ec2-user@ip-172-31-32-4 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
Failed to start docker.service: Unit not found.
[ec2-user@ip-172-31-32-4 ec2-user]# which docker
/usr/bin/which: no docker in (/sbin:/bin:/usr/sbin:/usr/bin)
[ec2-user@ip-172-31-32-4 ec2-user]# yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-32-4 ec2-user]# yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.23-1.amzn2.0.1 will be installed
```

25.docker login:login to docker

```
[root@ip-172-31-36-203 ec2-user]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: jennyjohnson0013
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

26.docker pull jennyjohnson0013/new_image_jen: pulling image from docker hub

```
Login Succeeded
[root@ip-172-31-32-4 ec2-user]# docker pull jennyjohnson0013/new_image_jen
Using default tag: latest
latest: Pulling from jennyjohnson0013/new_image_jen
6b851dcae6ca: Pull complete
7c24a1e0b4b0: Pull complete
Digest: sha256:ee20ef6cda0d70715ab2130e17021b81408b4b01fa24e38c8d2fde237a4d0b74
Status: Downloaded newer image for jennyjohnson0013/new_image_jen:latest
docker.io/jennyjohnson0013/new_image_jen:latest
[root@ip-172-31-32-4 ec2-user]# docker images
REPOSITORY           TAG      IMAGE ID      CREATED       SIZE
jennyjohnson0013/new_image_jen  latest   516f5f0b346c  40 minutes ago  227MB
```

27:**docker run -it --name jenny jennyjohnson0013/new_image_jen:** creating new container using the pulled image from docker hub

```
[root@ip-172-31-32-4 ec2-user]# docker run -it --name jenny jennyjohnson0013/new_image_jen
root@3df4dc701296:/# ls
bin  boot  dev  etc  home  jen  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@3df4dc701296:/# cd tmp
root@3df4dc701296:/tmp# ls
f1.txt  f2.txt  jenny
root@3df4dc701296:/tmp# exit
exit
```

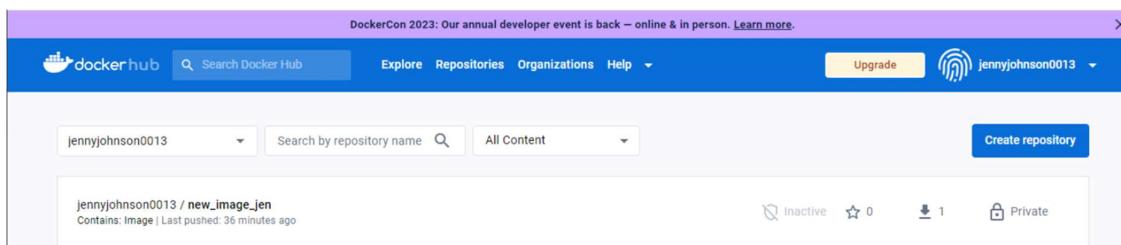
28.**docker rm jenny:** removing container

```
[root@ip-172-31-32-4 ec2-user]# docker rm jenny
jenny
[root@ip-172-31-32-4 ec2-user]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
[root@ip-172-31-32-4 ec2-user]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
```

29.**docker rmi jennyjohnson0013/new_image_jen:**removing image

```
[root@ip-172-31-32-4 ec2-user]# docker rmi jennyjohnson0013/new_image_jen
Untagged: jennyjohnson0013/new_image_jen:latest
Untagged: jennyjohnson0013/new_image_jen@sha256:ee20ef6cda0d70715ab2130e17021b81408b4b01fa24e38c8d2fde237a4d0b74
```

30.Making docker repository private



Result

The program was executed and the result was successfully obtained. Thus, CO3 was obtained.