NETWORKING & SYSTEM ADMINISTRATION LAB

(20MCA136)

LAB RECORD

Submitted in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications of A P J Abdul Kalam Technological University, Kerala.

Submitted by:

SANU K JOSEPH (SJC23MCA-2049)

MASTER OF COMPUTER APPLICATIONS

ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

CHOONDACHERRY P.O, KOTTAYAM

KERALA

May 2024

ST. JOSEPH' S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

(An ISO 9001: 2015 Certified College)

CHOONDACHERRY P.O, KOTTAYAM, KERALA

CERTIFICATE

This is to certify that the Networking & System Administration Lab Record (20MCA136) submitted by **Sanu K Joseph**, student of **Second** semester **MCA** at **ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI** in partial fulfilment for the award of Master of Computer Applications is a bonafide record of the lab work carried out by him under our guidance and supervision. This record in any form has not been submitted to any other University or Institute for any purpose.

Dr. Rahul Shajan	**Mr. Anish Augustine K**
Associate Professor	Assistant Professor ,
(Head, Department of	Dept. of Computer Application
Computer Application)	(Faculty In-Charge)

Submitted for the End Semester Examination held on ______________________

Examiner 1 **Examiner 2**

DECLARATION

I Sanu K Joseph, do hereby declare that the Networking & System Administration Lab Record (20MCA136) is a record of work carried out under the guidance of Mr. Anish Augustine K, Asst. Professor, Department of Computer Applications, SJCET, Palai as per the requirement of the curriculum of Master of Computer Applications Programme of A P J Abdul Kalam Technology University, Thiruvananthapuram. Further, I also declare that this record has not been submitted, full or part thereof, in any University / Institution for the award of any Degree / Diploma.

Place: Choondacherry SANU K JOSEPH

Date : (SJC23MCA-2049)

DEPARTMENT OF COMPUTER APPLICATIONS

VISION

To emerge as a center of excellence in the field of computer education with distinct identity and quality in all areas of its activities and develop a new generation of computer professionals with proper leadership, commitment and moral values.

MISSION

- Provide quality education in Computer Applications and bridge the gap between the academia and industry.
- Promoting innovation research and leadership in areas relevant to the socio economic progress of the country.
- Develop intellectual curiosity and a commitment to lifelong learning in students, with societal and environmental concerns.

COURSE OUTCOMES

After the completion of the course 20MCA136 Networking & System Administration Lab the student will be able to :

CO 1	Install and configure common operating systems.	K3 (Apply)
CO 2	Perform system administration tasks.	K3 (Apply)
CO 3	Install and manage servers for web applications.	K2 (Understand)
CO 4	Write shell scripts required for system administration.	K3 (Apply)
CO 5	Acquire skill sets required for a DevOps.	K3 (Apply)

CONTENT

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.

Procedure:

What is Computer Hardware?

Computer hardware is a hardware part of a computer system. In simple words, only those parts of the computer system which we can see or touch are called computer hardware.

Hardware is an important part of our computer system without which the computer is incomplete. You cannot use a computer without hardware and without hardware, there cannot be a computer system or construction.

1. Mouse

A mouse is a hardware input device that is used to move the cursor or pointer on computer screens. It can also be used to run computer programs, select items in a graphical user interface, and manipulate objects in the computer world. Some common examples of how it can be used are clicking on buttons, scrolling up and down the screen, selecting files, opening folders, and so on.

2. Keyboard

A keyboard is an input device that you use to enter data into a computer. It's also called the input device for your computer. Keyboards are used with PCs, laptops, tablets, and other devices. There are many different types of keyboards, but the most common one is the QWERTY keyboard. A QWERTY keyboard has all the letters in alphabetical order on it. This is different from some other types of keyboards, like Dvorak or Colemak keyboards. For example, these keyboards have keys arranged differently than what you're used to seeing on a QWERTY keyboard. And that means that typing on these keyboards will feel like typing in another language at first! But don't worry - once you get accustomed to it, it feels natural!

3. Monitor

Personal computers use a monitor to display data, run the software, and interact with the user. A monitor is an electronic visual display that connects to your computer or laptop. It is used for displaying images, text, videos, games, web pages, and more. Monitors are available in different sizes depending on the needs of the person using them. The most common types of monitors are CRT (cathode ray tube), LCD (liquid crystal display), and LED (light-emitting diode).

4. Motherboard

The motherboard is the backbone of our computer system. It's the central processing unit or CPU. It connects all the other components, like memory and graphics card, to the power supply. The motherboard is where all the wires are plugged in and it's also where you place your RAM, which is your computer's working memory. The motherboard is what makes one machine different from another.

Motherboards are made up of tiny transistors that control the flow of electricity through copper tracks on their surface. These transistors are called Integrated Circuits or ICs for short.

5. CPU (Central Processing Unit)

A CPU, or central processing unit, is the brain of a computer. The CPU processes information and runs programs. It functions as a control unit that executes programs according to instructions in its program memory. The CPU contains elements such as registers, an arithmetic logic unit (ALU), and control logic for sequencing instructions.

6. RAM Memory

A computer's RAM is a type of computer memory that stores information so the CPU can access it directly. Computer systems use main memory to store both data and programs. The more RAM you have, the more data your system can process at one time. This will lead to more efficient operations on your computer, which translates into better performance for the user.

7. 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 on-board computer system and a calculator's data tables.

8. Hard Disk Drive

A hard disk drive is a piece of hardware inside a computer that stores information. It’s used to store software and data in a safe place, which can be accessed when needed. With magnetic storage, there are no moving parts - unlike a CD or DVD player in which you need to move a disk in order to access data. You can think of it as "a closet" where all your stuff is stored safely. As long as you have power, you can get to your things when you need them.

9. 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. They are often used for installing software on your computer, moving data between computers, or writing new programs.

10.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.

2. Install latest version of Ubuntu on a VirtualBox

Procedure:

1. Download and VirtualBox Windows 10 Installation
2. Ubuntu ISO download
3. Install VirtualBox
4. Create an Ubuntu VM
5. Install Ubuntu on VirtualBox Windows 10 6. Install VirtualBox Guest Additions

Download and VirtualBox Windows 10 Installation

1. Install Ubuntu on VirtualBox

2. How-to Install Ubuntu on VirtualBox?

2.1. Open VirtualBox

2.2. Click on “New” to create a virtual machine

2.3. Enter Name for your Virtual Machine

2.4. Select “Linux” Operating System from “Type”

2.5. Click “Next”

2.6. Enter amount of memory (RAM) =1024 MBand click “Next”

2.7. Click “Create” to create hard drive

2.8. Click “Next”

2.9. Click “Next”

2.10. Enter Size of Virtual Hard Drive= 20 GBand Click “Create”

2.11. Select Virtual Machine

2.12. Click on “Start” to start the virtual machine

2.13. Select disk file source
2.14. After selecting the OS file to be installed click “Open”

2.15. Click “Start”

2.16. Click “Ok”

2.17. Click “Install Ubuntu”

2.18. Click “Continue”

2.19. Click “Install Now”

2.20. Click “Continue”

2.21. Select location and click “Continue”

2.22. Select keyboard layout & click “Continue”

2.23. Fill all the details and Click “Continue”

2.24. Now the installation process will start and installation window will appear

2.25. Click “Restart Now”

2.26. When the system will get restarted the following message will appear. Press “Enter”

2.27. Close the pop-up messages by clicking on the Close (×) button

3. Steps To Maximize the Size of Ubuntu Desktop

3.1. Go to “Devices”

3.2. Click “Insert Guest Additions CD Image…”

3.3. Click “Run”

3.4. Click “Authenticate”

3.5. Press “Enter”

3.6. Now “Restart” your system for the changes to be applied.

3.7. After the system gets restarted. Go to “View”

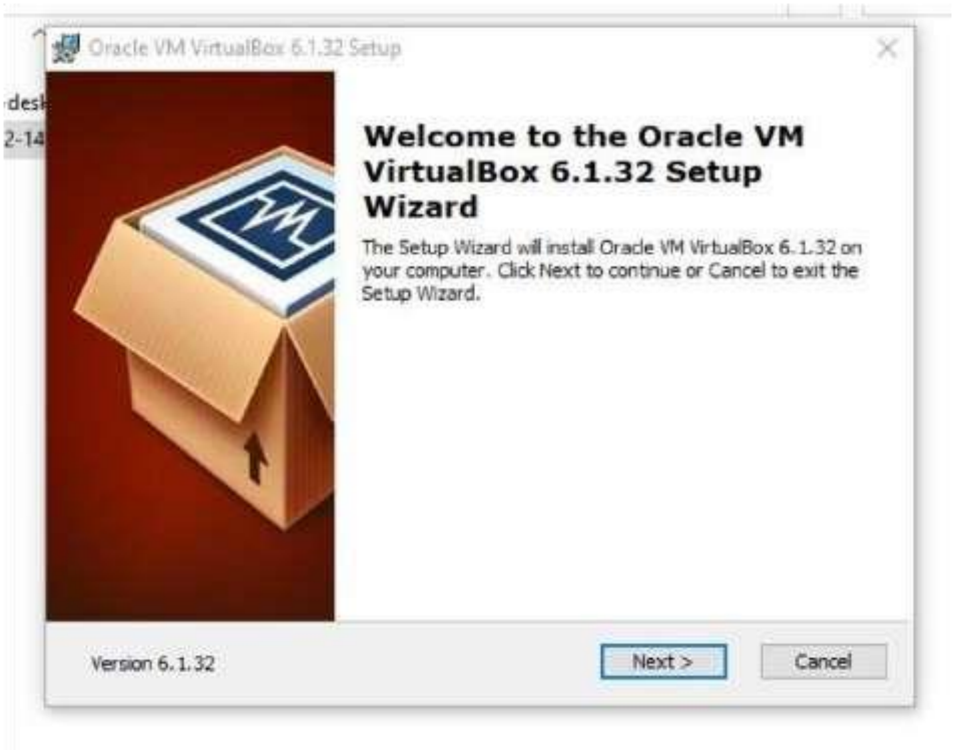
3.8. Click “Switch to Full screen”

3.9. Click “Switch”

Output Screenshot

STEP 1: Installing Virtual Box.

a. Starting pop-up window for the installation.



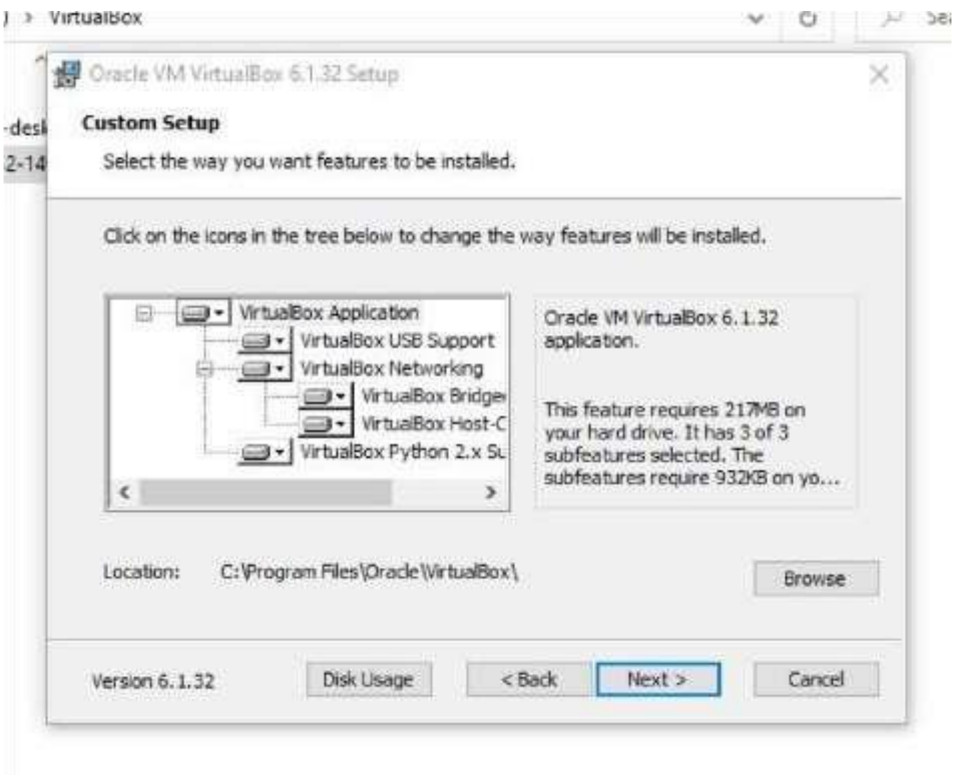
Oracle VM VirtualBox 6.1.32 Setup

Welcome to the Oracle VM VirtualBox 6.1.32 Setup Wizard

The Setup Wizard will install Oracle VM VirtualBox 6.1.32 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Version 6.1.32

b. Custom setup window to select the features you want and select installation location



Oracle VM VirtualBox 6.1.32 Setup

Custom Setup

Select the way you want features to be installed.

Click on the icons in the tree below to change the way features will be installed.

VirtualBox Application
VirtualBox USB Support
VirtualBox Networking
VirtualBox Bridge
VirtualBox Host-C
VirtualBox Python 2.x Su

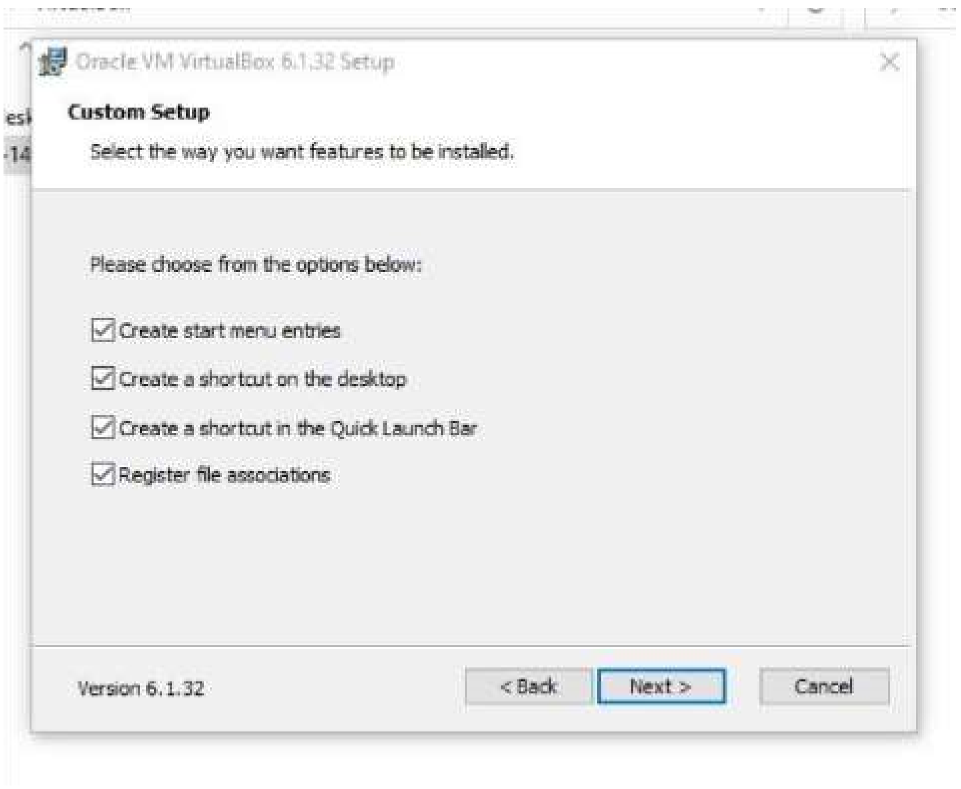
Oracle VM VirtualBox 6.1.32 application.

This feature requires 217MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 932KB on yo...

Location: C:\Program Files\Oracle\VirtualBox\

Version 6.1.32

c. Custom setup window to choose from the option below.



Oracle VM VirtualBox 6.1.32 Setup

Custom Setup

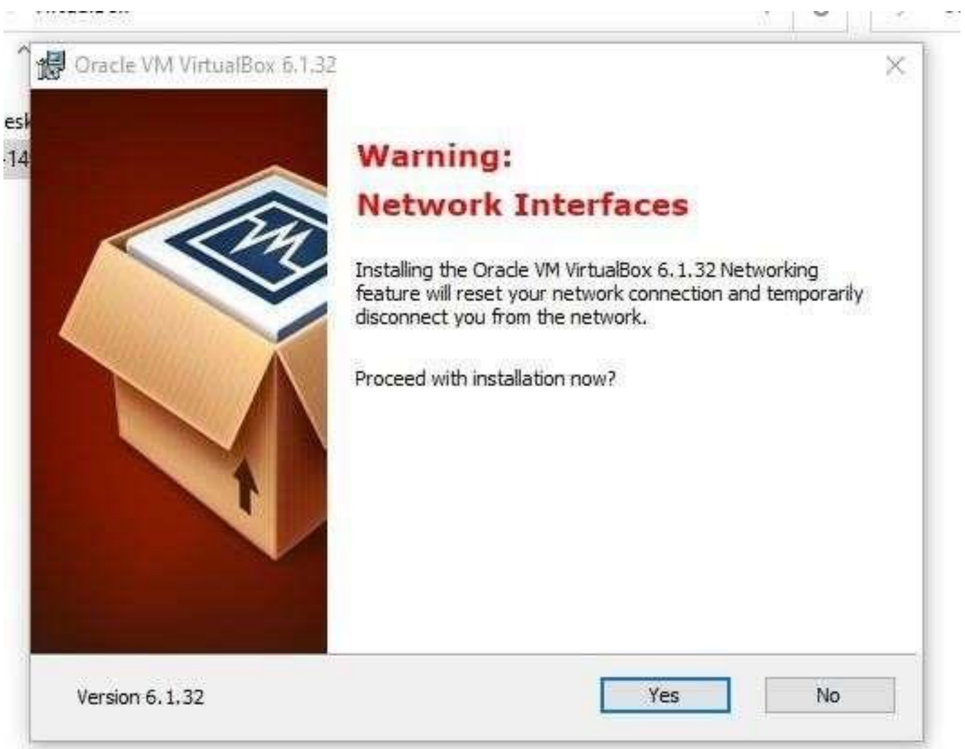
Select the way you want features to be installed.

Please choose from the options below:

- [✓] Create start menu entries
- [✓] Create a shortcut on the desktop
- [✓] Create a shortcut in the Quick Launch Bar
- [✓] Register file associations

Version 6.1.32

d. Custom setup window to confirm the installation.



Oracle VM VirtualBox 6.1.32

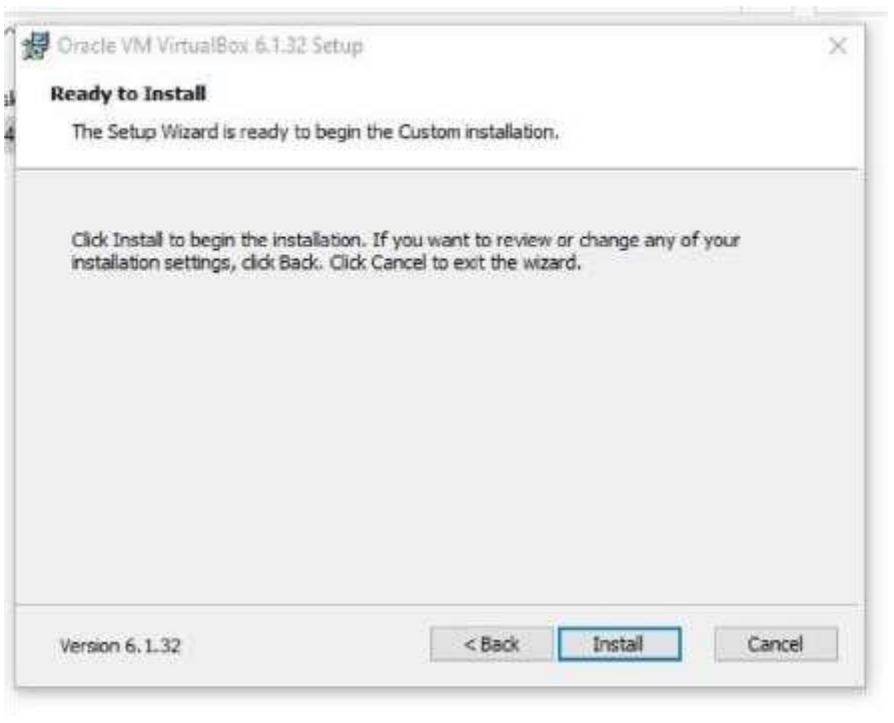
Warning:
Network Interfaces

Installing the Oracle VM VirtualBox 6.1.32 Networking feature will reset your network connection and temporarily disconnect you from the network.

Proceed with installation now?

Version 6.1.32

e. Custom setup window to install the virtual box with the install button.



Oracle VM VirtualBox 6.1.32 Setup

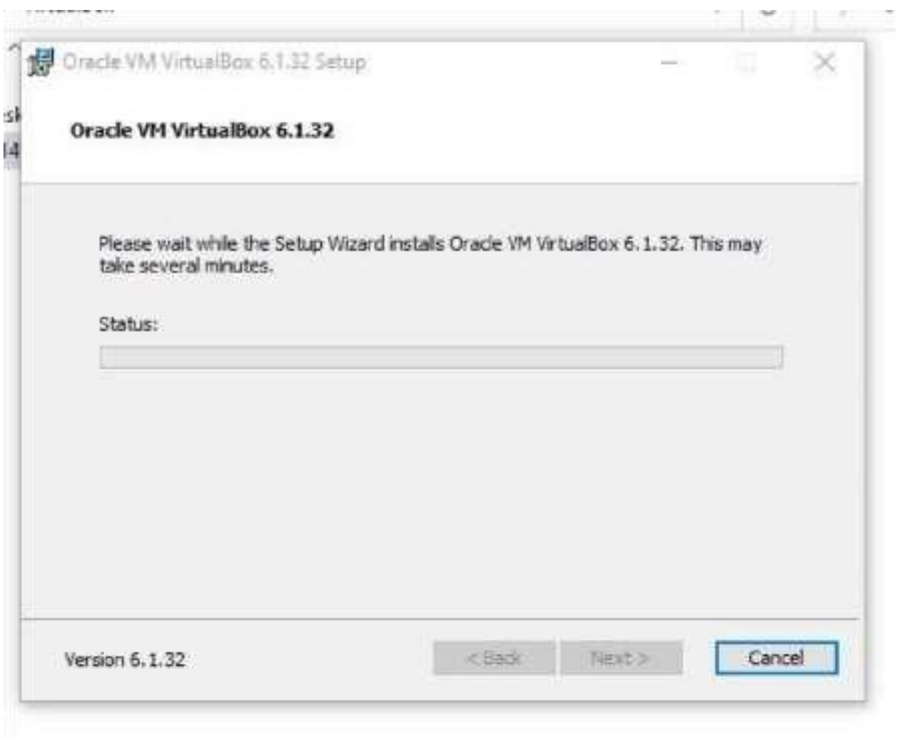
Ready to Install

The Setup Wizard is ready to begin the Custom installation.

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Version 6.1.32

f. Installation box showing the installation status.



Oracle VM VirtualBox 6.1.32 Setup

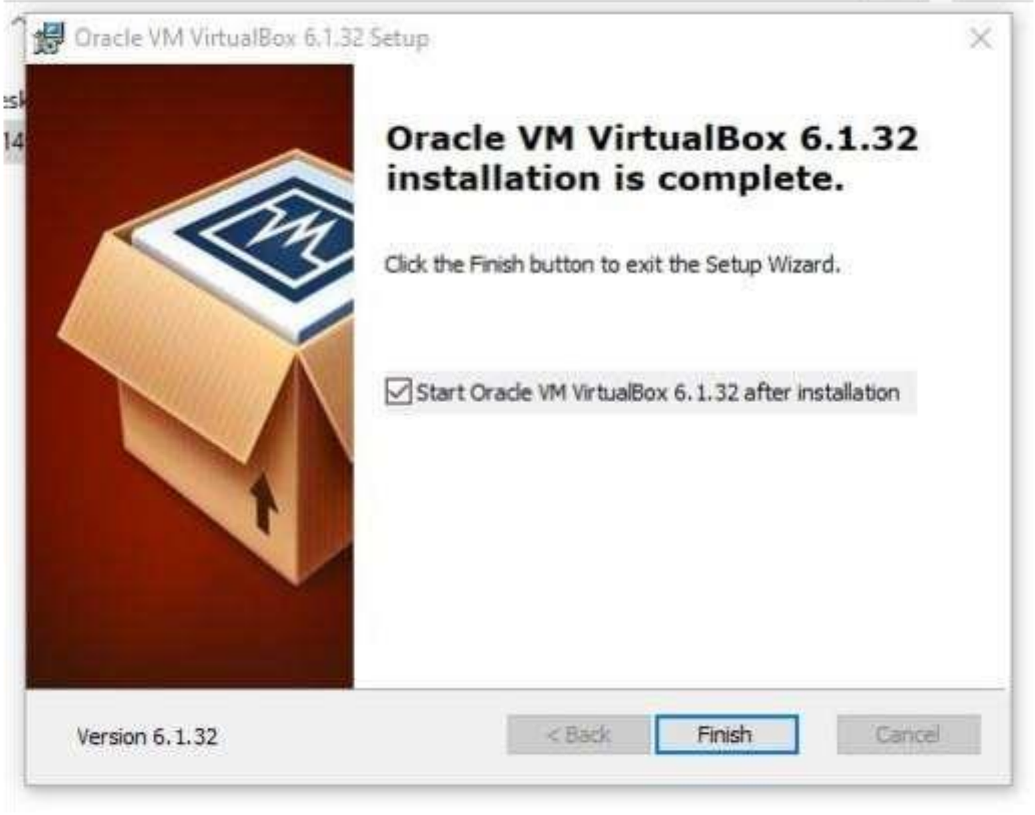
Oracle VM VirtualBox 6.1.32

Please wait while the Setup Wizard installs Oracle VM VirtualBox 6.1.32. This may take several minutes.

Status:

Version 6.1.32

g. Installation complete pop-up windows with the finish button.


Oracle VM VirtualBox 6.1.32 Setup

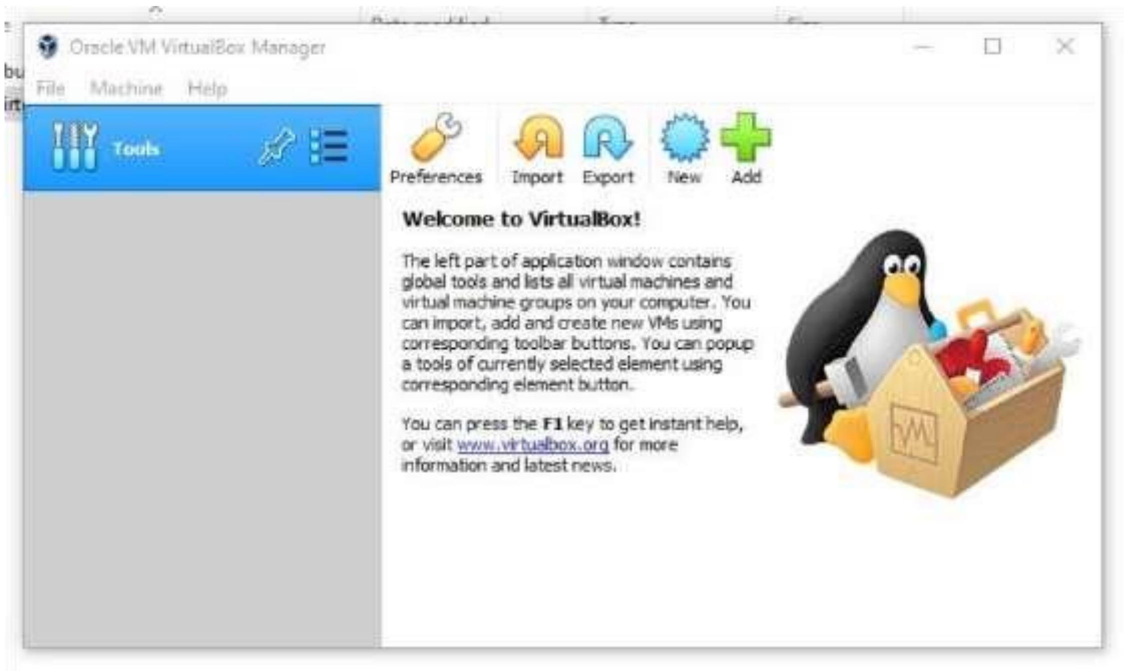
Oracle VM VirtualBox 6.1.32 installation is complete.

Click the Finish button to exit the Setup Wizard.

☑ Start Oracle VM VirtualBox 6.1.32 after installation

Version 6.1.32

STEP 2: Setup the Ubuntu Instance in the VM Virtual Box.

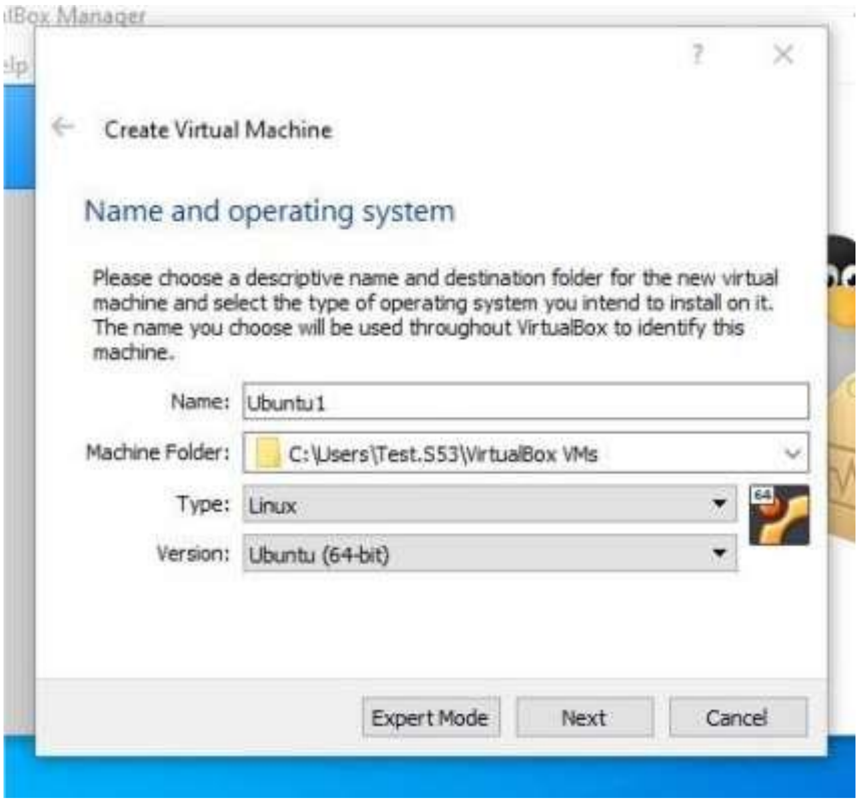

Oracle VM VirtualBox Manager

Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

a. After selecting the NEW button to create the Ubuntu instance, the pop-up window to enter & select the name, type, and version of the OS.



Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

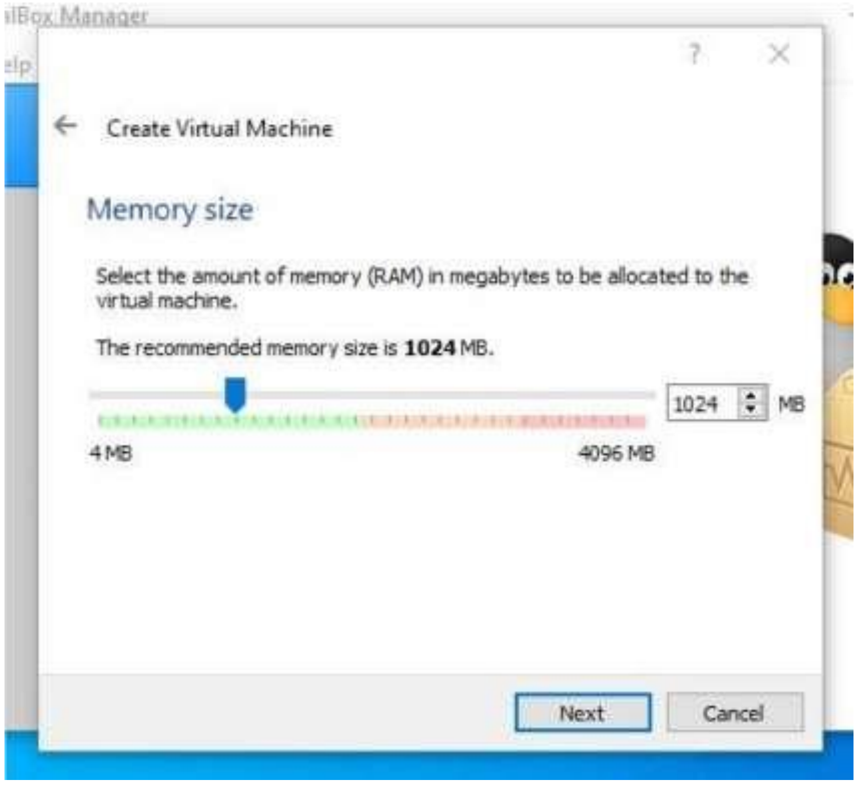
Name: Ubuntu1

Machine Folder: C:\Users\Test.S53\VirtualBox VMs

Type: Linux

Version: Ubuntu (64-bit)

b. Choose the main memory size for the OS.



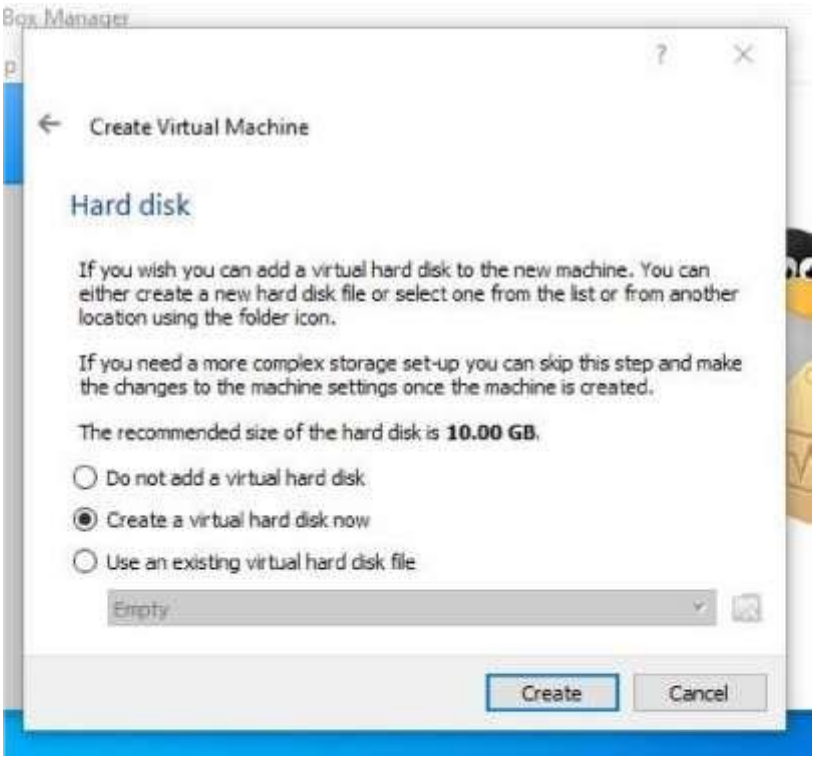
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

1024 MB

c. Option to add a virtual hard disk to the new machine instance.



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

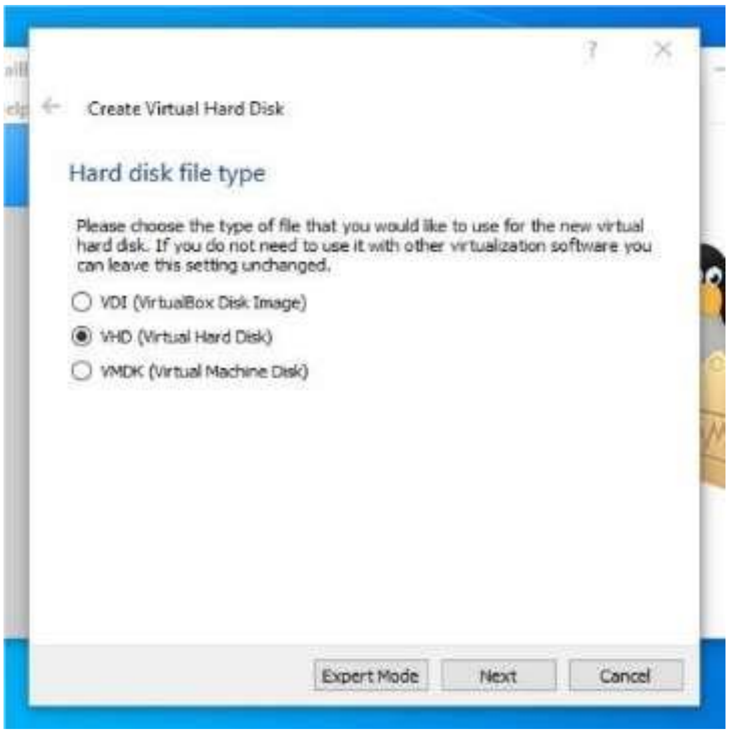
The recommended size of the hard disk is **10.00 GB**.

- ○ Do not add a virtual hard disk
- ● Create a virtual hard disk now
- ○ Use an existing virtual hard disk file

Empty

Create Cancel

d. Option to choose the type of new virtual hard disk for new instance of OS.



Create Virtual Hard Disk

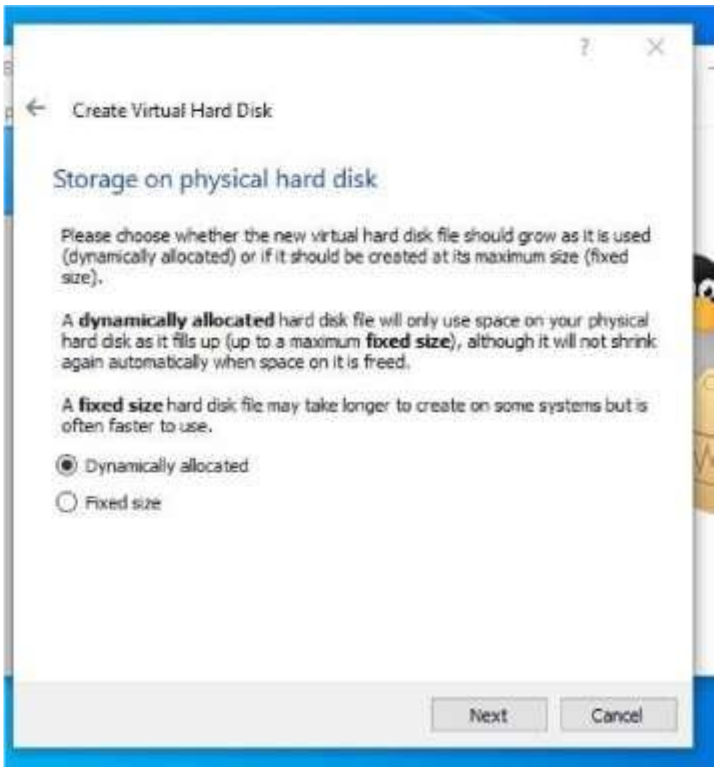
Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

- ○ VDI (VirtualBox Disk Image)
- ● VHD (Virtual Hard Disk)
- ○ VMDK (Virtual Machine Disk)

Expert Mode Next Cancel

e. Options to choose the methods of accessing the physical hard disk space for the new instance from the existing hard disk



Create Virtual Hard Disk

Storage on physical hard disk

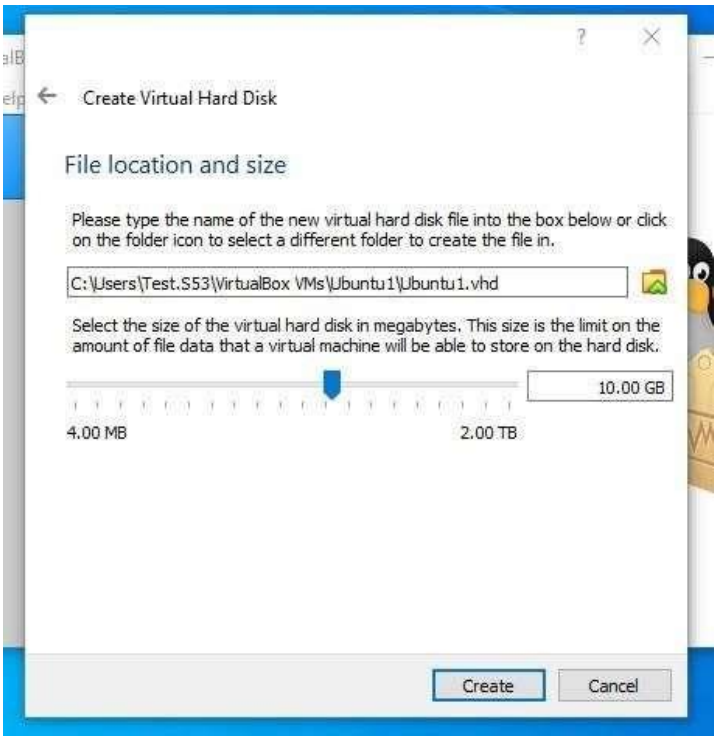
Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

[✓ Dynamically allocated]
Fixed size

f. Panel to select the size of the virtual disk in megabytes and location and name of the instance and final submit to create the instance of OS.



Create Virtual Hard Disk

File location and size

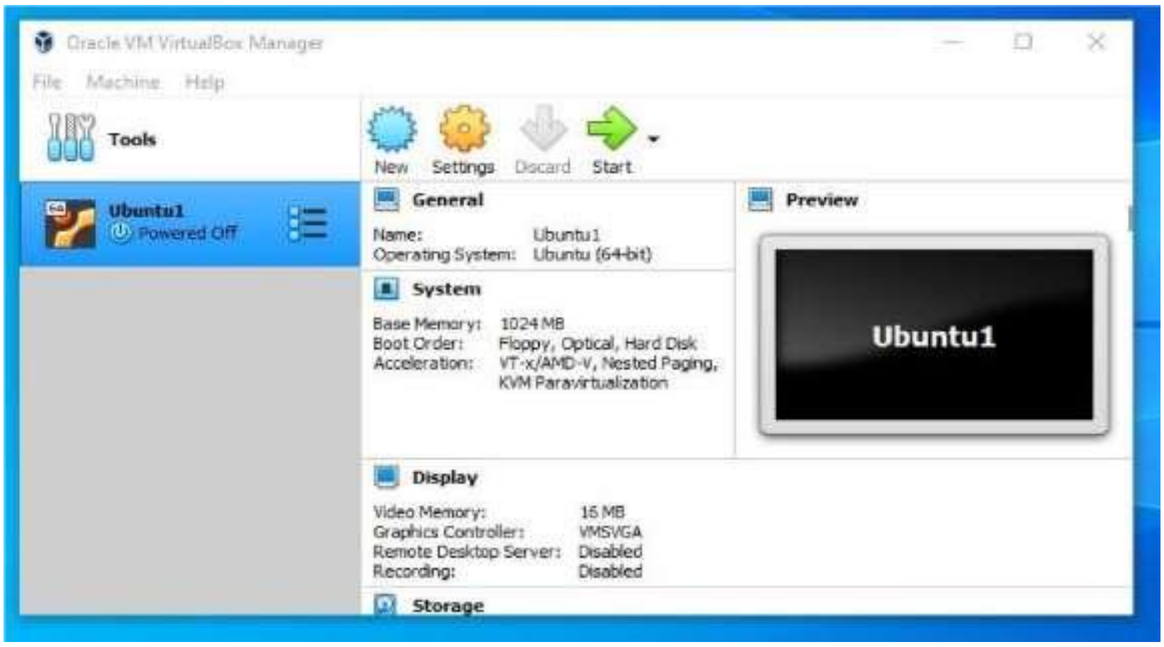
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

C:\Users\Test.S53\VirtualBox VMs\Ubuntu1\Ubuntu1.vhd

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

10.00 GB

g. The newly created OS instance and at the left side of the application.

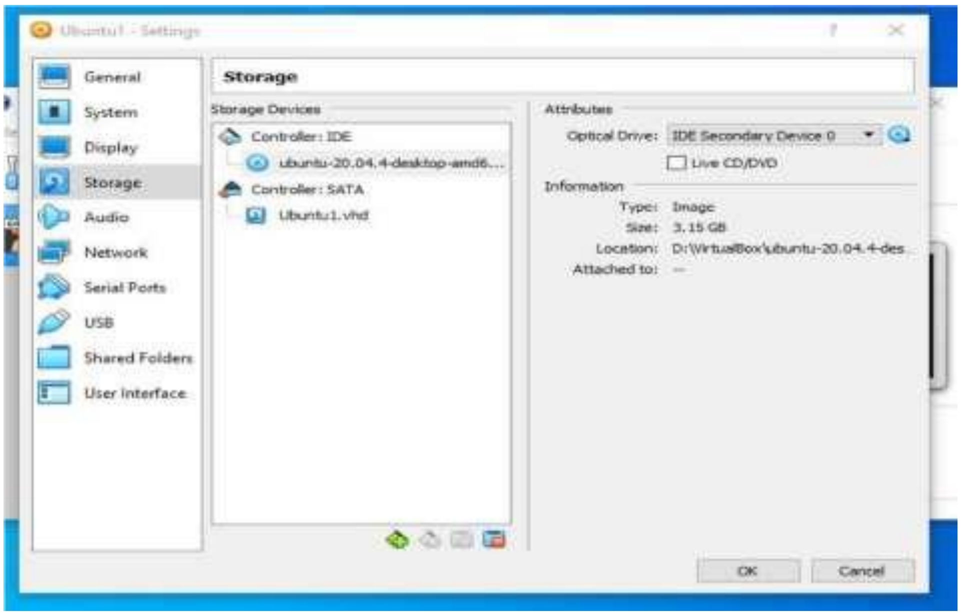


General
Name: Ubuntu1
Operating System: Ubuntu (64-bit)

System
Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

h. Settings the configurations of the instance created and adding the ISO image file of the OS correspondingly and Selecting the ISO image file from the local device.



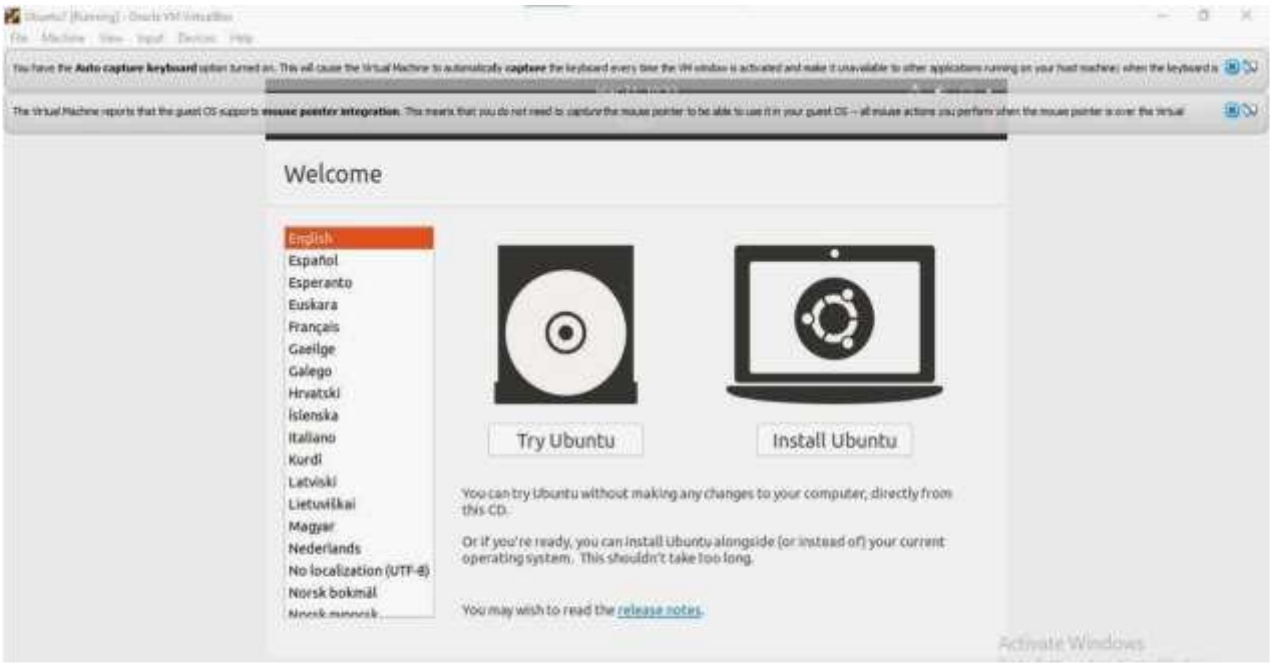
Storage
Storage Devices
Controller: IDE
ubuntu-20.04.4-desktop-amd6...
Controller: SATA
Ubuntu1.vhd

Attributes
Optical Drive: IDE Secondary Device 0
Live CD/DVD

Information
Type: Image
Size: 3.15 GB
Location: D:\VirtualBox\ubuntu-20.04.4-des
Attached to: --

STEP 3: Installation of the Ubuntu OS within the newly created instance.

a. Running the new OS instance and selecting the "Install Ubuntu" to install the loaded ISO file.


You have the **Auto capture keyboard** option turned on. This will cause the Virtual Machine to automatically **capture** the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is

The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to capture the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual

Welcome

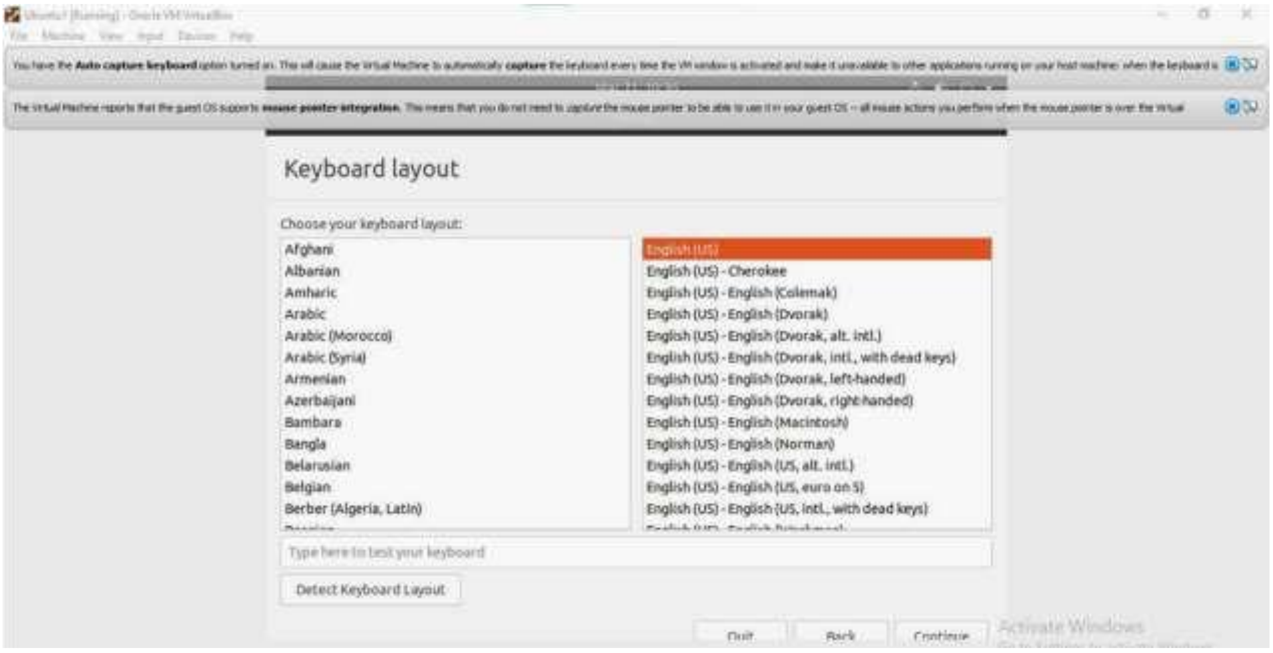
English
Español
Esperanto
Euskara
Français
Gaeilge
Galego
Hrvatski
Íslenska
Italiano
Kurdî
Latviski
Lietuviškai
Magyar
Nederlands
No localization (UTF-8)
Norsk bokmål
Norsk nynorsk

You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

b. Selecting the language for install the ubuntu OS.


You have the **Auto capture keyboard** option turned on. This will cause the Virtual Machine to automatically **capture** the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is

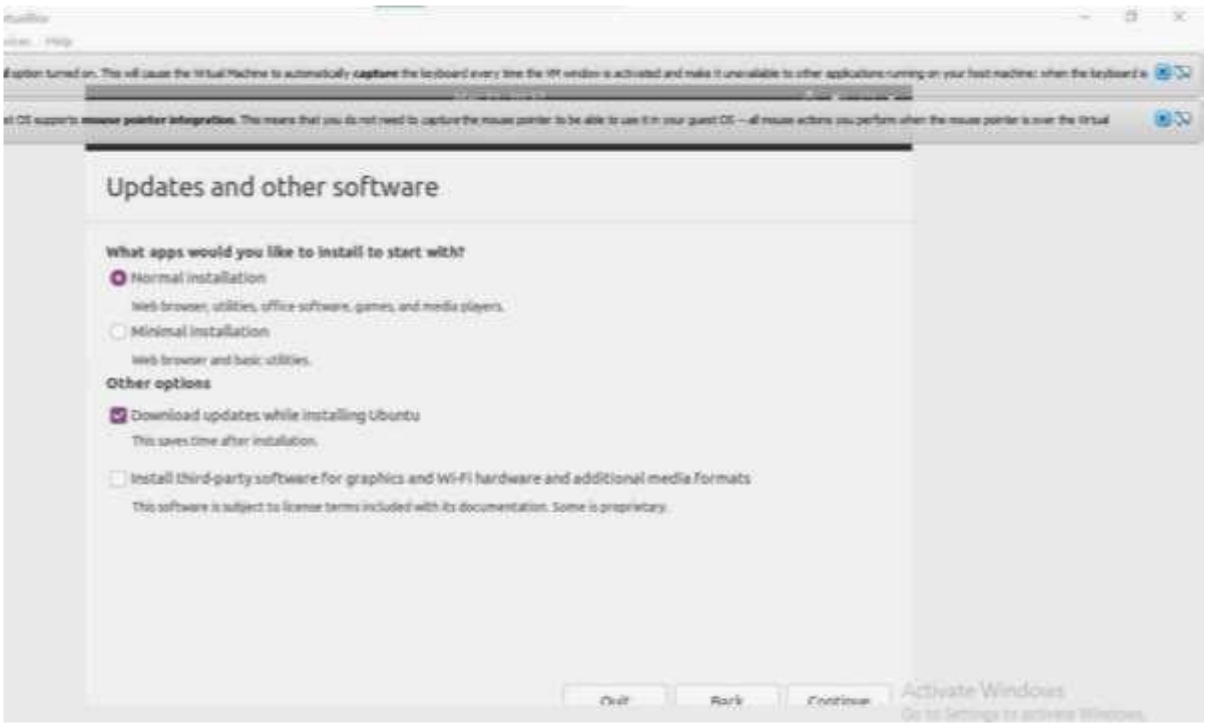
The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to capture the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual

Keyboard layout

Choose your keyboard layout:

Afghani	English (US)
Albanian	English (US) - Cherokee
Amharic	English (US) - English (Colemak)
Arabic	English (US) - English (Dvorak)
Arabic (Morocco)	English (US) - English (Dvorak, alt. intl.)
Arabic (Syria)	English (US) - English (Dvorak, intl., with dead keys)
Armenian	English (US) - English (Dvorak, left-handed)
Azerbaijani	English (US) - English (Dvorak, right-handed)
Bambara	English (US) - English (Macintosh)
Bangla	English (US) - English (Norman)
Belarusian	English (US) - English (US, alt. intl.)
Belgian	English (US) - English (US, euro on 5)
Berber (Algeria, Latin)	English (US) - English (US, intl., with dead keys)

c. Selection of other installation along & within with the installation of ubuntu.



option turned on. This will cause the Virtual Machine to automatically **capture** the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is

OS supports **mouse pointer integration**. This means that you do not need to capture the mouse pointer to be able to use it in your guest OS – all mouse actions you perform when the mouse pointer is over the Virtual

Updates and other software

What apps would you like to install to start with?

[✓ Normal installation]
Web browser, utilities, office software, games, and media players.

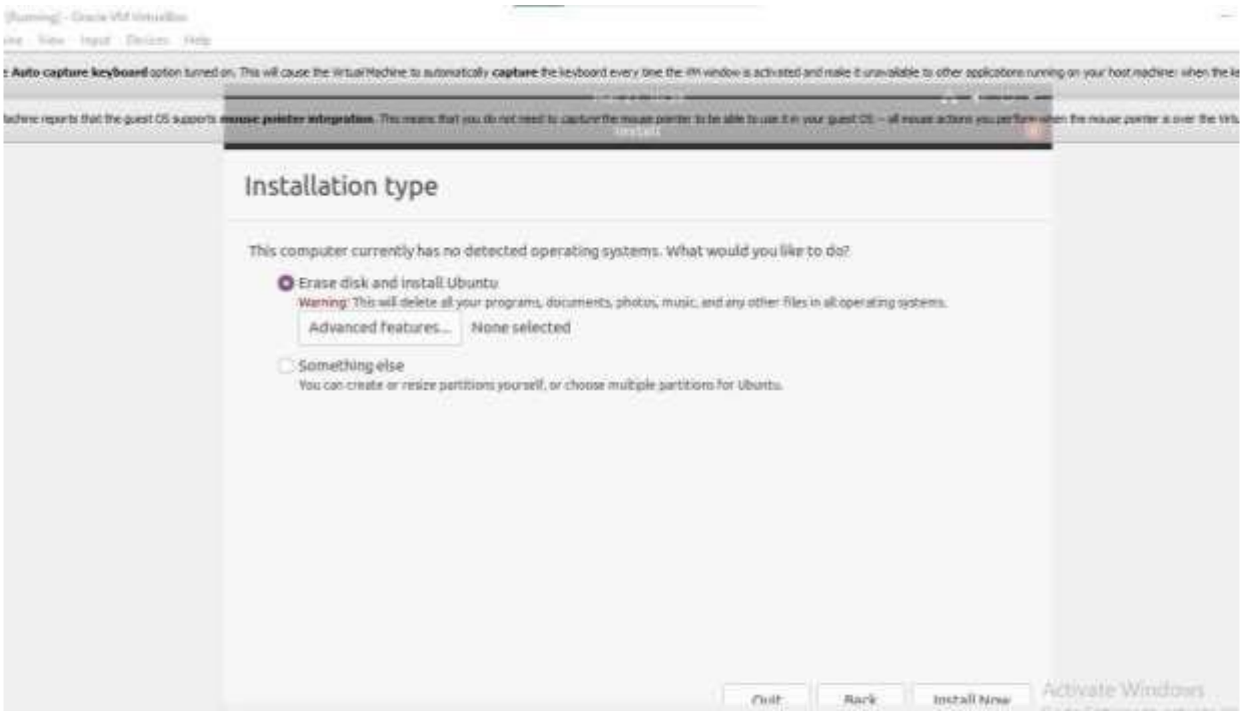
Minimal installation
Web browser and basic utilities.

Other options

[✓ Download updates while installing Ubuntu]
This saves time after installation.

Install third-party software for graphics and Wi-Fi hardware and additional media formats
This software is subject to license terms included with its documentation. Some is proprietary.

d. Selecting the disk partitioning allocation options from the given below.



[Running] - Oracle VM VirtualBox

Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically **capture** the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the ke

machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to capture the mouse pointer to be able to use it in your guest OS – all mouse actions you perform when the mouse pointer is over the Virtu

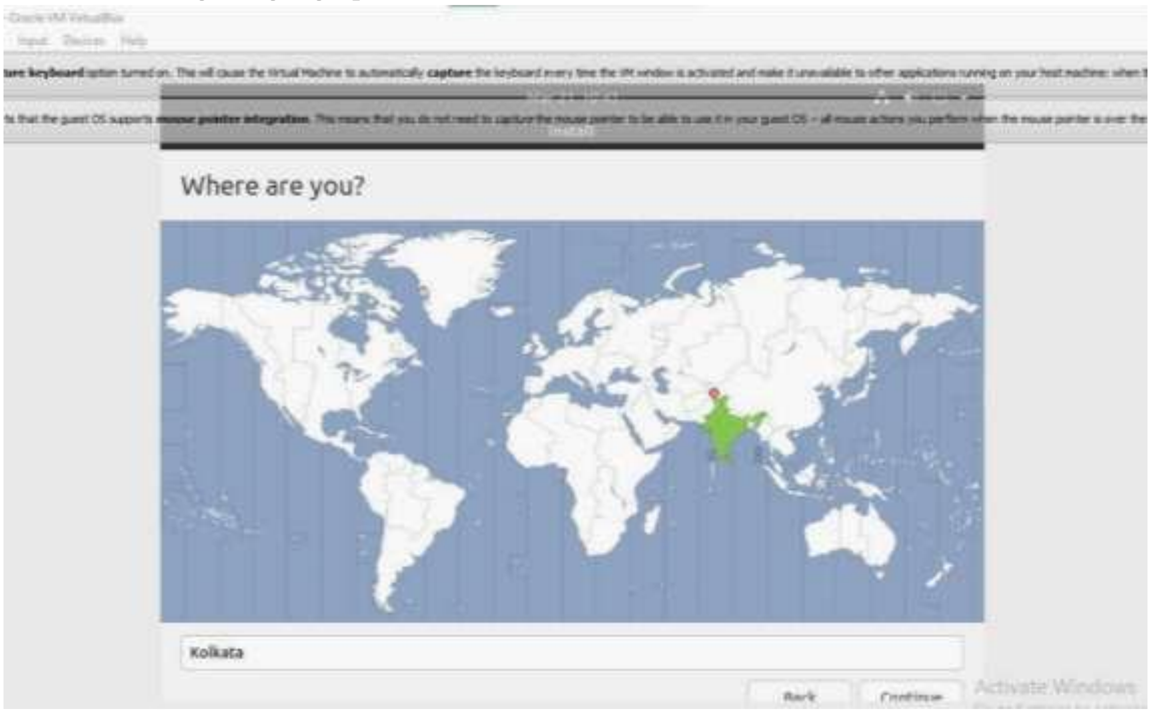
Installation type

This computer currently has no detected operating systems. What would you like to do?

[✓ Erase disk and install Ubuntu]
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
Advanced features... None selected

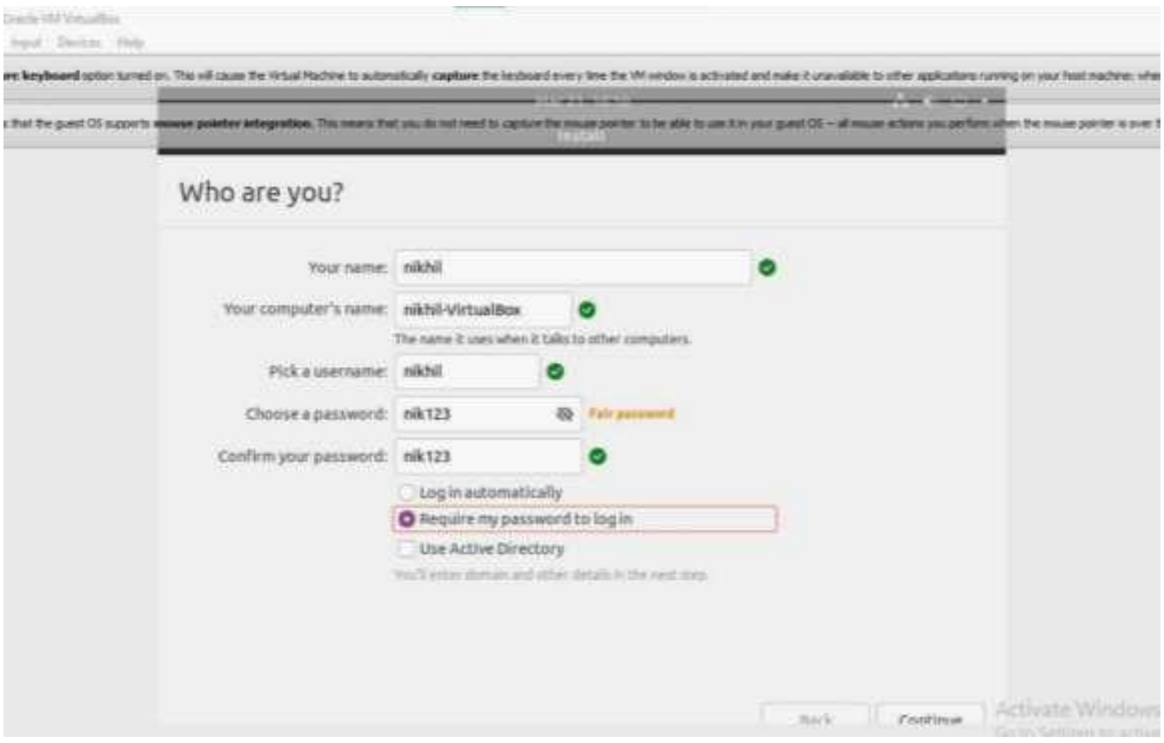
Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

e. Selecting the geographical location for the time/location.


Where are you?

Kolkata

f. Entering the name, username & password for the account to sign in to the ubuntu OS after installation.


Who are you?

Your name: nikhil

Your computer's name: nikhil-VirtualBox

The name it uses when it talks to other computers.

Pick a username: nikhil

Choose a password: nik123 Fair password

Confirm your password: nik123

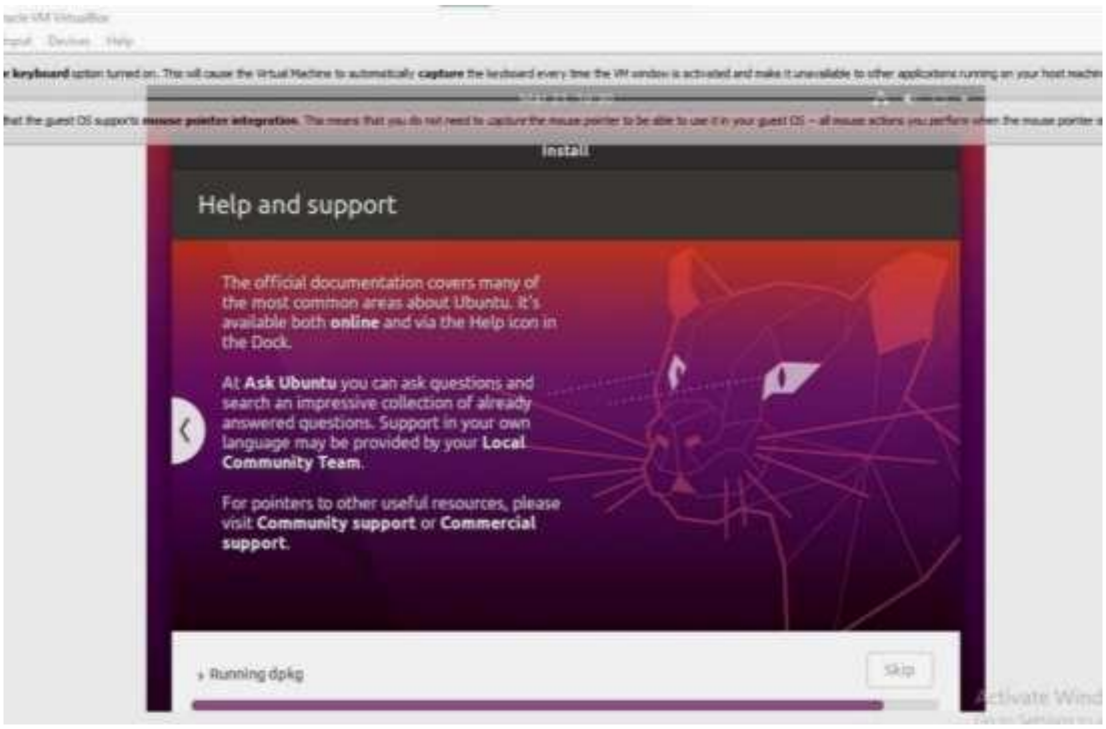
Log in automatically

[✓ Require my password to log in]

Use Active Directory

You'll enter domain and other details in the next step.

g. Installing the ubuntu OS in the instance, extracting the ubuntu ISO file, setting configurations, setting the various software within, etc.



Help and support

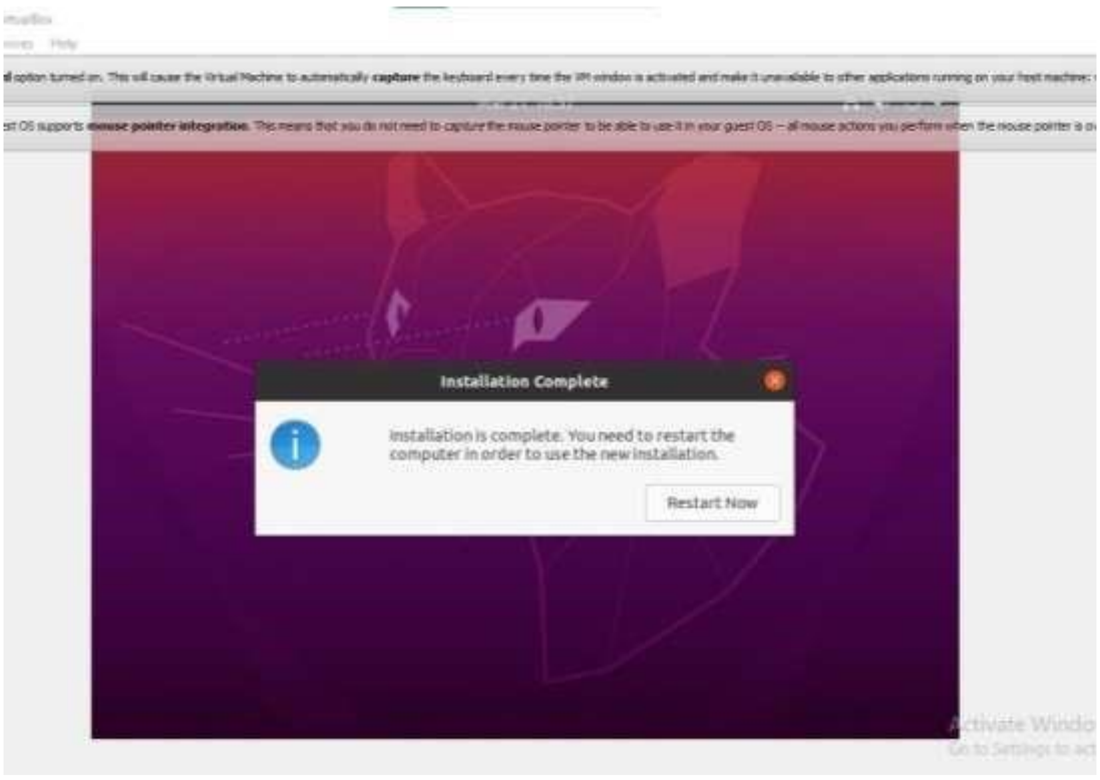
The official documentation covers many of the most common areas about Ubuntu. It's available both **online** and via the Help icon in the Dock.

At **Ask Ubuntu** you can ask questions and search an impressive collection of already answered questions. Support in your own language may be provided by your **Local Community Team**.

For pointers to other useful resources, please visit **Community support** or **Commercial support**.

Running dpkg

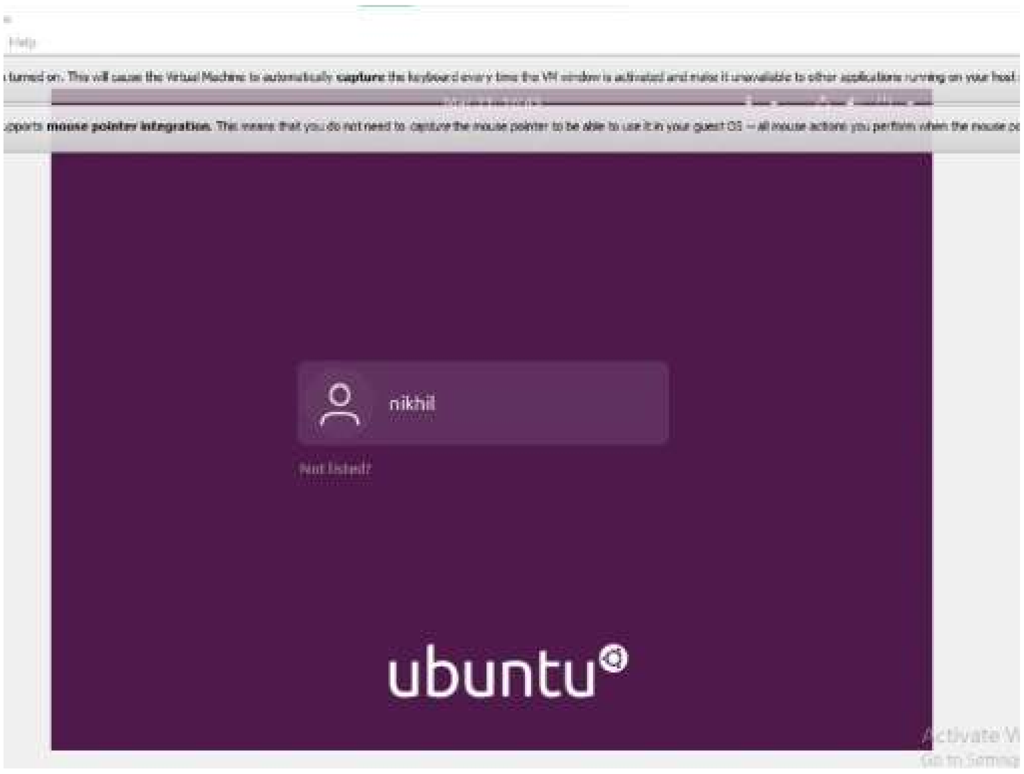
h. Restarting the OS instance to finalize the installation.



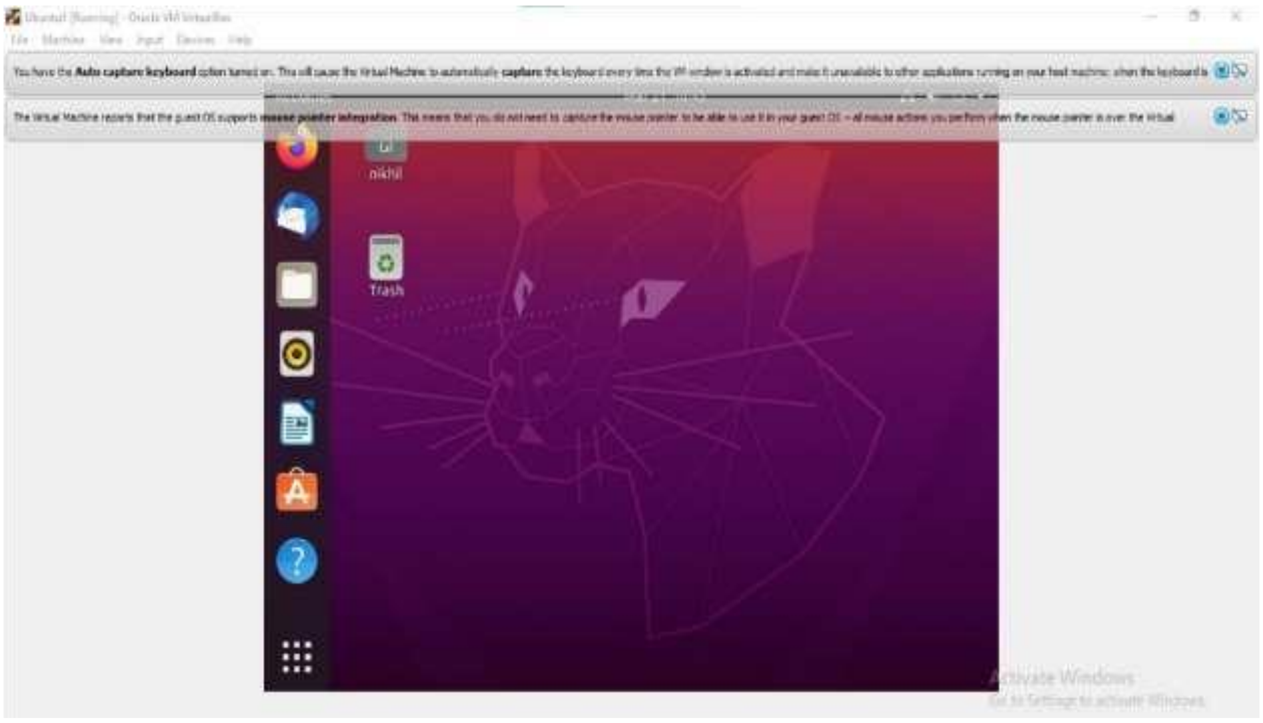
Installation Complete

Installation is complete. You need to restart the computer in order to use the new installation.

i. Signing in and visiting the home screen of the ubuntu OS using the previously registered username & password.



Output

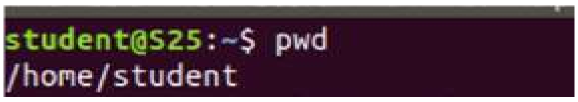


3. Study of a terminal based text editor such as Vim or Gedit, Basic Linux commands: - familiarity with following commands/operations expected

Procedure

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

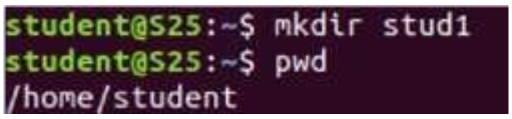
Syntax: -$ pwd



```
student@S25:~$ pwd
/home/student
```

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

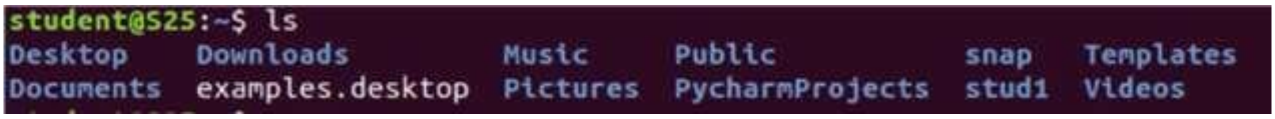
Syntax: -$ mkdir<directory name>



```
student@S25:~$ mkdir stud1
student@S25:~$ pwd
/home/student
```

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

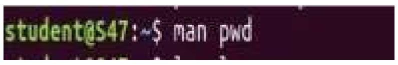
Syntax: -$ ls



```
student@S25:~$ ls
Desktop     Downloads          Music     Public            snap    Templates
Documents   examples.desktop   Pictures  PycharmProjects   stud1   Videos
```

Man: This command is used to display the user manual of any command that we can run on the terminal.

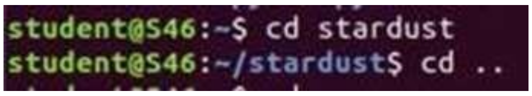
Syntax: -$ man <command name>



```
student@S47:~$ man pwd
```

Cd: This command is used to change the current directory.

Syntax: -$ cd <directory name>



```
student@S46:~$ cd stardust
student@S46:~/stardust$ cd ..
```

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

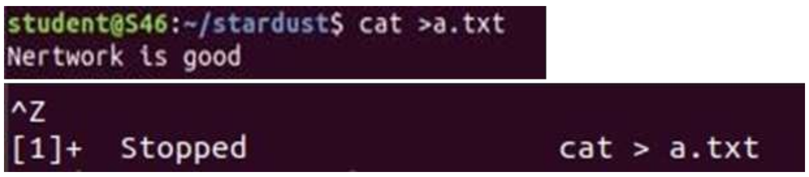
- **cd –:**This command is used to switch back to previous directory we were working earlier.

cat > filename: This command is used to create a file and add contents to that file.

Syntax: -$ cat > filename.txt **cat filename:**This command is used to

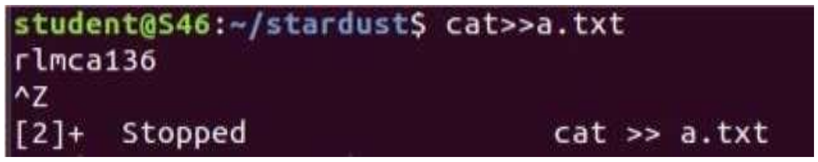
view the contents in the file.

Syntax: -$ cat filename.txt



```
student@S46:~/stardust$ cat >a.txt
Nertwork is good
^Z
[1]+  Stopped                 cat > a.txt
```

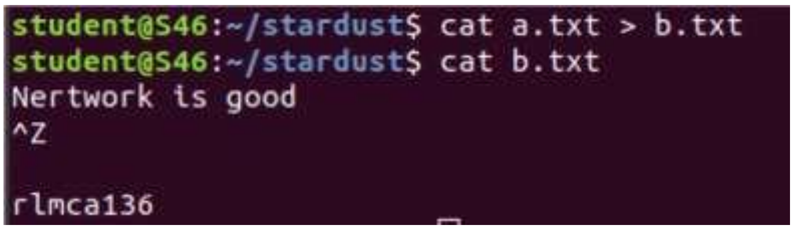
cat>>filename:This command is used to add contents to an existing file. Syntax: -$ cat >> filename.txt



```
student@S46:~/stardust$ cat>>a.txt
rlmca136
^Z
[2]+  Stopped                 cat >> a.txt
```

cat filename1 > filename2: This command is used to copy the content from one file to another file.

Syntax: -$ cat filename1 > filename2



```
student@S46:~/stardust$ cat a.txt > b.txt
student@S46:~/stardust$ cat b.txt
Nertwork is good
^Z

rlmca136
```

read :This command is used to read the content of a line to a variable.

Syntax: -$ read variablename

Find: This command is used to display contents of particular directory.

Syntax: -$ find filename.txt **grep** :This command will let you search through all the text in a given file.

Syntax: -$ grep word filename.txt

Output: -

- **grep -i: command** used for a case insensitive search

  Syntax: $ grep -i filename.txt
- **grep -v:** command used for inverted search.

  Syntax: $ grep -v filename.txt
- **grep -A1:** command used to display line after the result.

  Syntax: $ grep -A1 filename.txt
- **grep -B1:** command used to display line before the result.

  Syntax: $ grep -B1 filename.txt
- **grep -C1:** command used to display line before and after the result.

  Syntax: $ grep -C1 filename.txt
- **wc -word count:** This command is used for counting purpose which is used to find the number of lines, the number of words, the number of characters and the number of bytes.

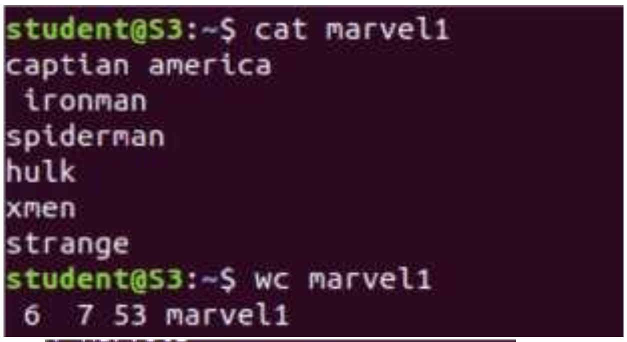
- **wc -l** (count number of lines)
- **wc -w** (count number of words)
- **wc -c** (count number of characters)
- **wc -m** (count number of bytes)

Syntax: - $ wc -l filename.txt

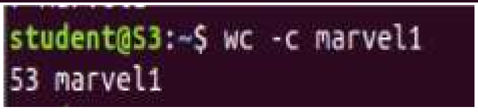
$ wc -w filename.txt

$ wc -c filename.txt

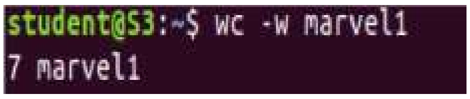
$ wc -m filename.txt



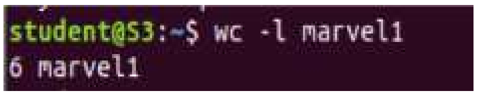
```
student@S3:~$ cat marvel1
captian america
 ironman
spiderman
hulk
xmen
strange
student@S3:~$ wc marvel1
 6  7 53 marvel1
```



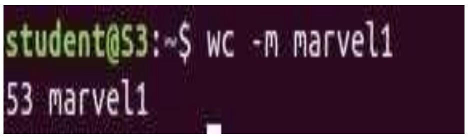
```
student@S3:~$ wc -c marvel1
53 marvel1
```



```
student@S3:~$ wc -w marvel1
7 marvel1
```



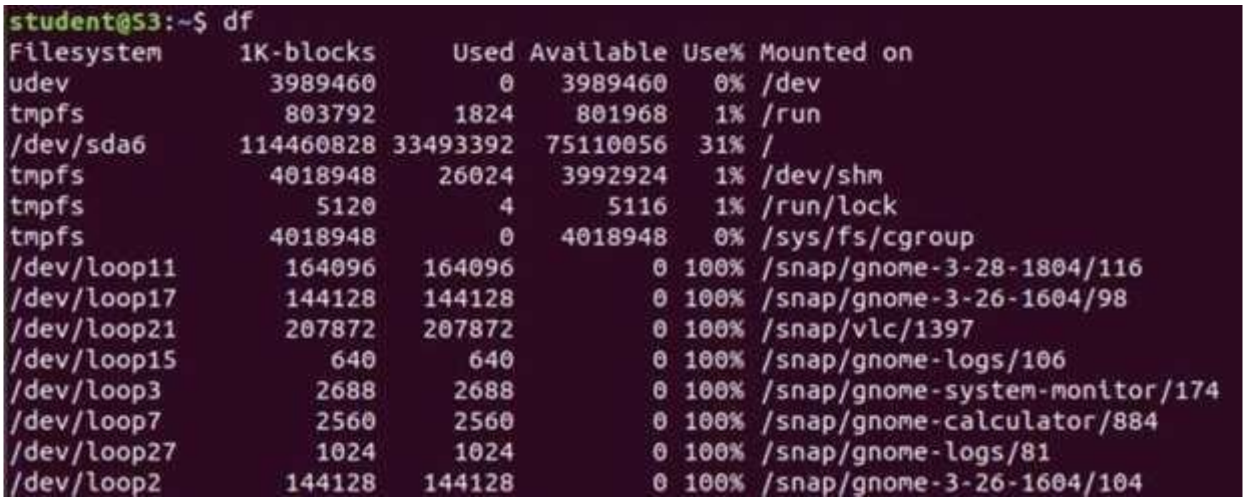
```
student@S3:~$ wc -l marvel1
6 marvel1
```



```
student@S3:~$ wc -m marvel1
53 marvel1
```

df: This command is used to get a report on system disc space usage.

Syntax: -$ df filename.txt

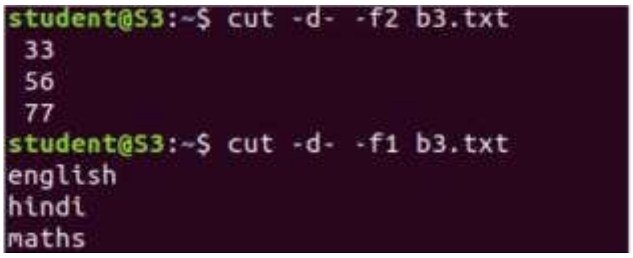


```
student@S3:~$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             3989460        0   3989460   0% /dev
tmpfs             803792     1824    801968   1% /run
/dev/sda6      114460828 33493392  75110056  31% /
tmpfs            4018948    26024   3992924   1% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
tmpfs            4018948        0   4018948   0% /sys/fs/cgroup
/dev/loop11       164096   164096         0 100% /snap/gnome-3-28-1804/116
/dev/loop17       144128   144128         0 100% /snap/gnome-3-26-1604/98
/dev/loop21       207872   207872         0 100% /snap/vlc/1397
/dev/loop15          640      640         0 100% /snap/gnome-logs/106
/dev/loop3          2688     2688         0 100% /snap/gnome-system-monitor/174
/dev/loop7          2560     2560         0 100% /snap/gnome-calculator/884
/dev/loop27         1024     1024         0 100% /snap/gnome-logs/81
/dev/loop2        144128   144128         0 100% /snap/gnome-3-26-1604/104
```

- **df -m** :This command is used to see the report in mega bytes.

Syntax: $ def -m filename.txt **cut -d**:This command is used to cut and display the content based on the delimiter given.

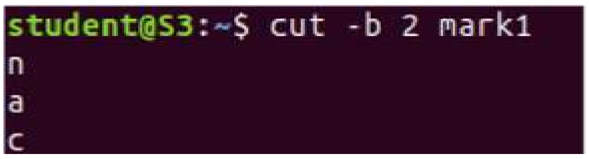
Syntax: -$ cut –d delimiter –fieldnumber filename



```
student@S3:~$ cut -d- -f2 b3.txt
 33
 56
 77
student@S3:~$ cut -d- -f1 b3.txt
english
hindi
maths
```

cut -b: This command is used tocut and display the content based on the specified byte number.

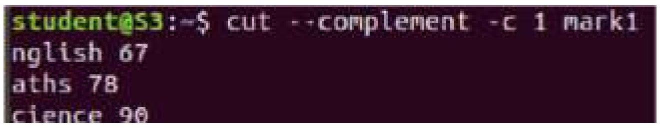
Syntax: -$ cut –b bytenumber filename



```
student@S3:~$ cut -b 2 mark1
n
a
c
```

cut --complement -c: This command is used to erase the specified character and display the remaining content of the file.

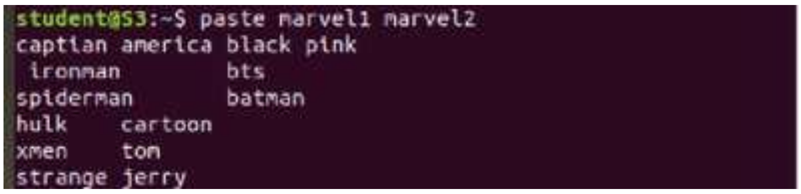
Syntax: -$ cut --complement –c characternumber filename.txt Output



```
student@S3:~$ cut --complement -c 1 mark1
nglish 67
aths 78
cience 90
```

Paste<u>:</u> This command is used to paste the contents from the specified file.

Syntax: -$ paste filename



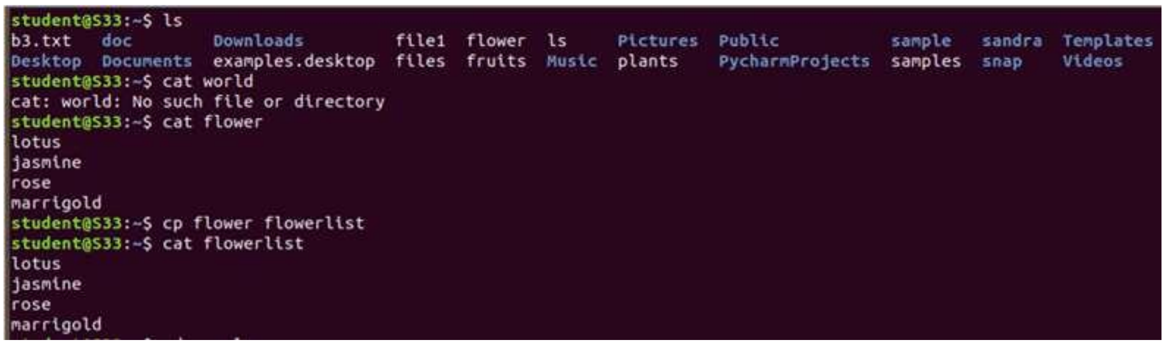
```
student@S3:~$ paste marvel1 marvel2
captian america black pink
 ironman        bts
spiderman       batman
hulk    cartoon
xmen    tom
strange jerry
```

More: This 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

Cp:This command is used to copy the contents from an existing file to a new file.

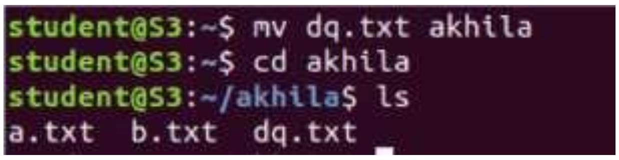
Syntax: -$ cpexisting_filenamenew_filename



```
student@S33:~$ ls
b3.txt   doc        Downloads         file1  flower  ls     Pictures  Public          sample   sandra  Templates
Desktop  Documents  examples.desktop  files  fruits  Music  plants    PycharmProjects samples  snap    Videos
student@S33:~$ cat world
cat: world: No such file or directory
student@S33:~$ cat flower
lotus
jasmine
rose
marrigold
student@S33:~$ cp flower flowerlist
student@S33:~$ cat flowerlist
lotus
jasmine
rose
marrigold
```

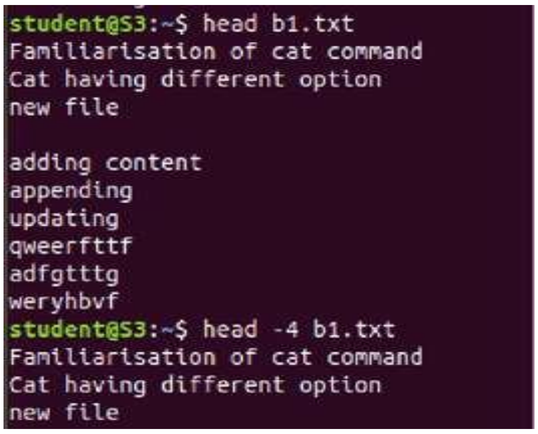
Mv:This command is used to move an existing file or directory from one location to another.

Syntax: -$ mv filename directory_name



```
student@S3:~$ mv dq.txt akhila
student@S3:~$ cd akhila
student@S3:~/akhila$ ls
a.txt  b.txt  dq.txt
```

Head:This command is used to display the first 10 lines of the file by default. Syntax:
-$ head filename

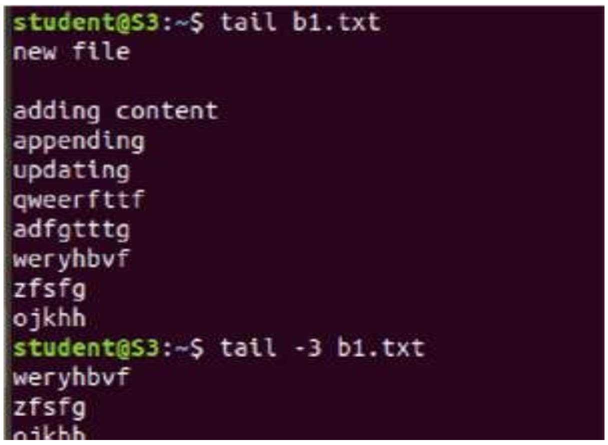


```
student@S3:~$ head b1.txt
Familiarisation of cat command
Cat having different option
new file

adding content
appending
updating
qweerfttf
adfgtttg
weryhbvf
student@S3:~$ head -4 b1.txt
Familiarisation of cat command
Cat having different option
new file
```

- **head -number**:This command is used to display the lines of the file to the specified number from head.

Tail: This command is used to display the last 10 lines of the file by default.

Syntax: -$ tail filename

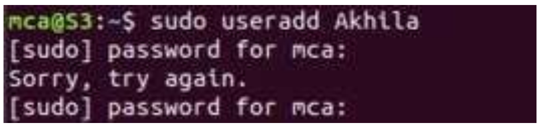


```
student@S3:~$ tail b1.txt
new file

adding content
appending
updating
qweerfttf
adfgtttg
weryhbvf
zfsfg
ojkhh
student@S3:~$ tail -3 b1.txt
weryhbvf
zfsfg
ojkhh
```

- **tail -number**:This command is used to display the lines of the file to the specified number from tail.

sudo useradd: This command is used to add new user.

Syntax: -$ sudo useradd username



```
mca@S3:~$ sudo useradd Akhila
[sudo] password for mca:
Sorry, try again.
[sudo] password for mca:
```

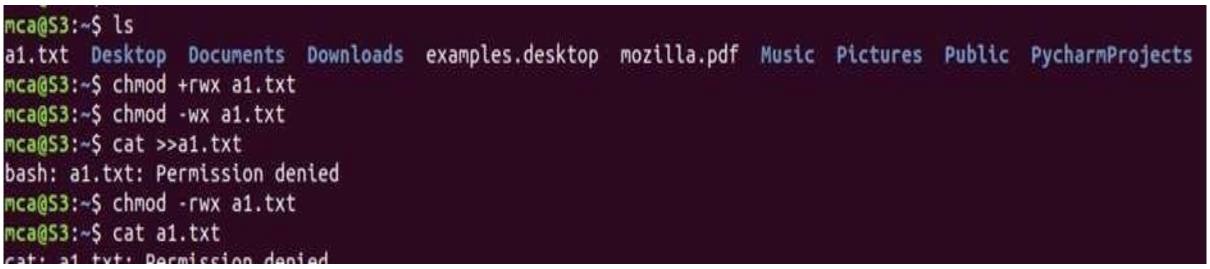
- **sudo passwd** :This command is used to add password to the user.
  - Syntax: -$ sudo passwd username
- **sudo usermod** :This command is used to add members.
  - Syntax: -$sudo usermod -G groupname username **delete**
- **sudo userdel username -** used to delete user.
- **sudo groupdel groupname -** used to delete group name.
  - Syntax: -$ sudo userdel username

- **sudo groupdel groupname chmod**: This command is used change directory permission of files.
- **chmod +rwx**
- **chmod -wx**
- **chmod -rwx**

Syntax: - $ chmod +wx filename

$ chmod -wx filename

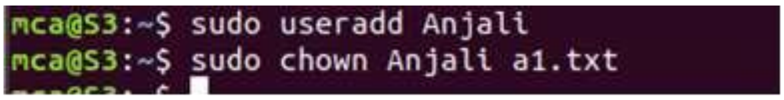
$ chmod -rwx filename



```
mca@S3:~$ ls
a1.txt  Desktop  Documents  Downloads  examples.desktop  mozilla.pdf  Music  Pictures  Public  PycharmProjects
mca@S3:~$ chmod +rwx a1.txt
mca@S3:~$ chmod -wx a1.txt
mca@S3:~$ cat >>a1.txt
bash: a1.txt: Permission denied
mca@S3:~$ chmod -rwx a1.txt
mca@S3:~$ cat a1.txt
cat: a1.txt: Permission denied
```

chown: This command is used to give ownership to user .

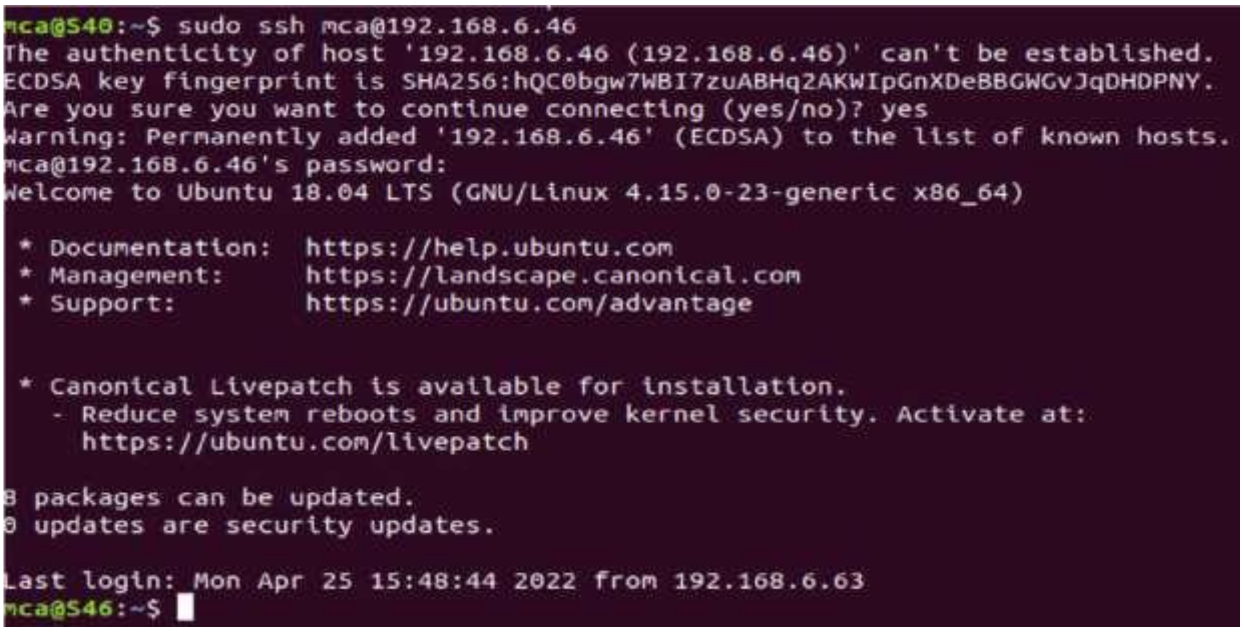
Syntax: - $ sudo chown username filename



```
mca@S3:~$ sudo useradd Anjali
mca@S3:~$ sudo chown Anjali a1.txt
```

Ssh:This command is used to provide a secure encrypted connection between two hosts over an insecure network.

Syntax: - $ ssh mca@ipaddress



```
mca@S40:~$ sudo ssh mca@192.168.6.46
The authenticity of host '192.168.6.46 (192.168.6.46)' can't be established.
ECDSA key fingerprint is SHA256:hQC0bgw7WBI7zuABHq2AKWIpGnXDeBBGWGvJqDHDPNY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.6.46' (ECDSA) to the list of known hosts.
mca@192.168.6.46's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-23-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

8 packages can be updated.
0 updates are security updates.

Last login: Mon Apr 25 15:48:44 2022 from 192.168.6.63
mca@S46:~$
```

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.

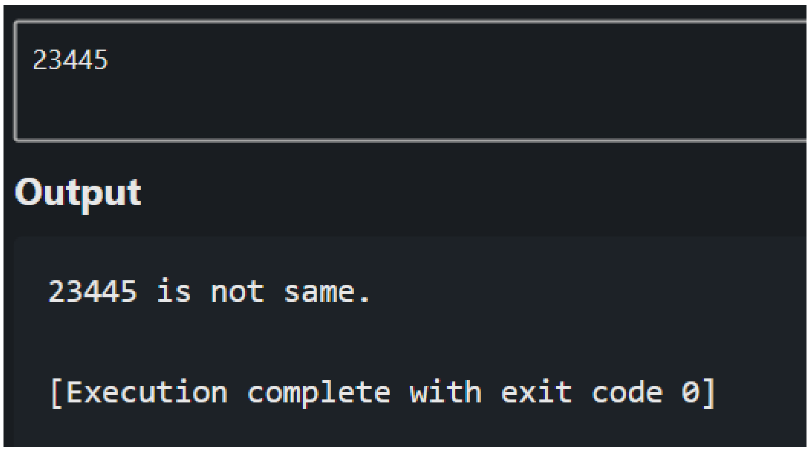
4.1AIM

Shell program to check the given number and its reverse are same.

Source Code

```
read -p "Enter a number: " num
echo "$num" | rev | grep -q "^$num$" && echo "$num is same." || echo "$num is not same."
```

Output



```
23445
```

Output

```
23445 is not same.

[Execution complete with exit code 0]
```

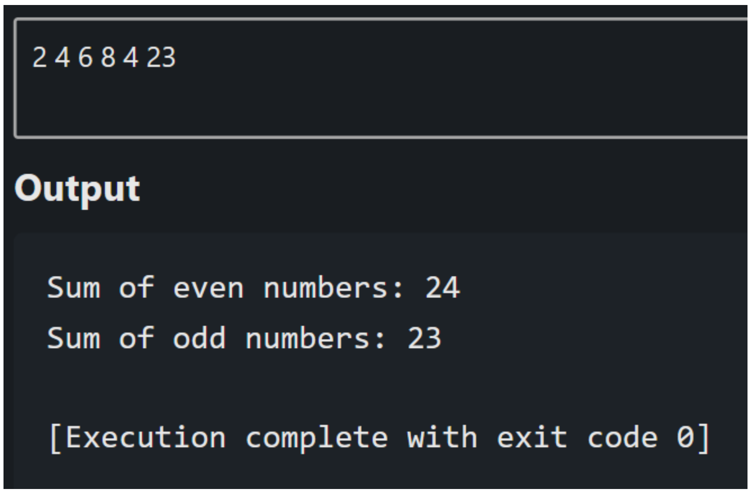
4.2AIM

Shell program to find the sum of odd and even numbers from a set of numbers.

Source Code

```
read -p "Enter numbers separated by spaces: " nums
sum_even=$(grep -oE '\b[0-9]*[02468]\b' <<< "${nums[@]}" | paste -sd+ | bc)
sum_odd=$(grep -oE '\b[0-9]*[13579]\b' <<< "${nums[@]}" | paste -sd+ | bc)
echo "Sum of even numbers: $sum_even"
echo "Sum of odd numbers: $sum_odd"
```

Output



```
2 4 6 8 4 23
```

Output

```
Sum of even numbers: 24
Sum of odd numbers: 23

[Execution complete with exit code 0]
```

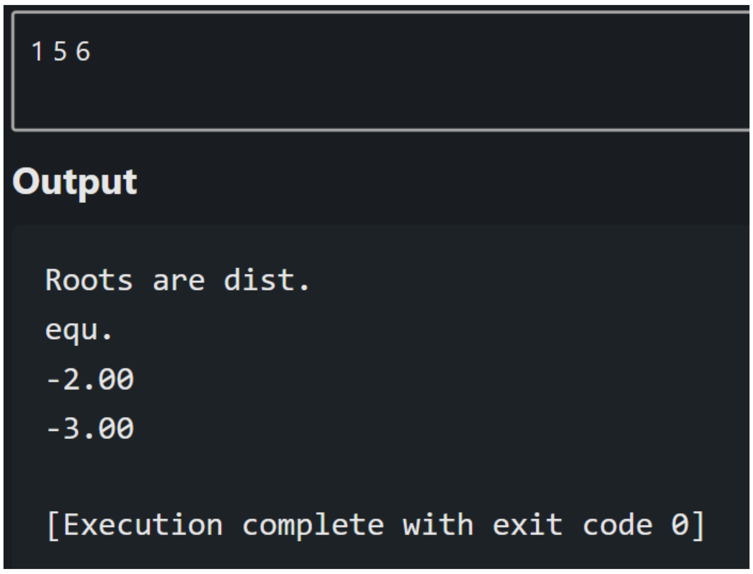
4.3AIM

Shell program to find the roots of a quadratic equation.

Source Code

```
read -p "Enter coefficients (a b c): " a b c
d=$((b*b-4*a*c))
echo "Roots are $( [ $d -gt 0 ] && echo "dist." || [ $d -eq 0 ] && echo "equ." || echo "complex." )"
[ $d -gt 0 ] && bc -l <<< "scale=2; (-$b + sqrt($d)) / (2 * $a)"
[ $d -gt 0 ] && bc -l <<< "scale=2; (-$b - sqrt($d)) / (2 * $a)"
```

Output



```
1 5 6
```

Output

```
Roots are dist.
equ.
-2.00
-3.00

[Execution complete with exit code 0]
```

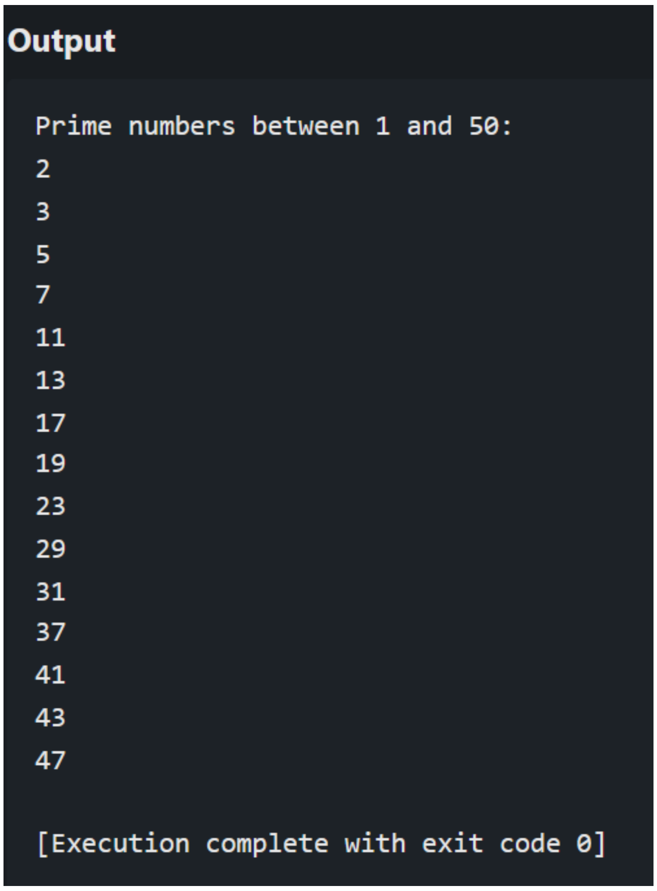
4.4AIM

Shell program to generate prime numbers between 1 and 50.

Source Code

```
echo "Prime numbers between 1 and 50:"
for num in $(seq 2 50); do
    if [ $(factor $num | wc -w) -eq 2 ]; then
        echo $num
    fi
done
```

Output



Output

```
Prime numbers between 1 and 50:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47

[Execution complete with exit code 0]
```

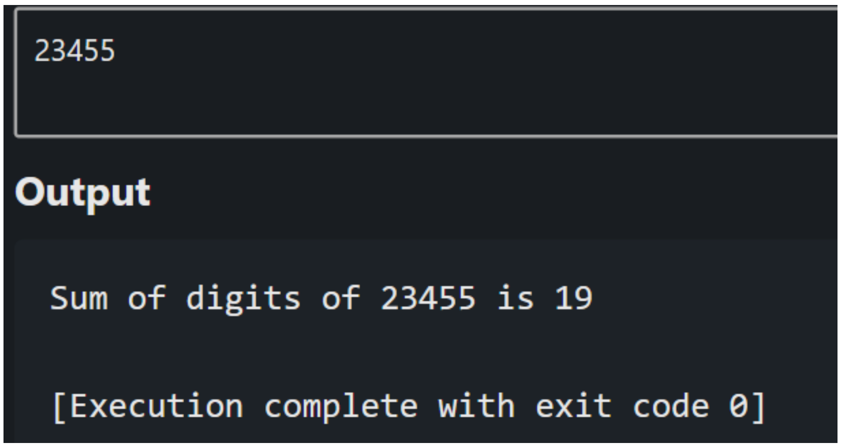
4.5AIM

Shell program to find the sum of digits of a number using function.

Source Code

```
sum_digits() {
   echo $1 | grep -o '.' | paste -sd+ | bc
}
read -p "Enter a number: " num
result=$(sum_digits $num)
echo "Sum of digits of $num is $result"
```

Output



```
23455
```

Output

```
Sum of digits of 23455 is 19

[Execution complete with exit code 0]
```

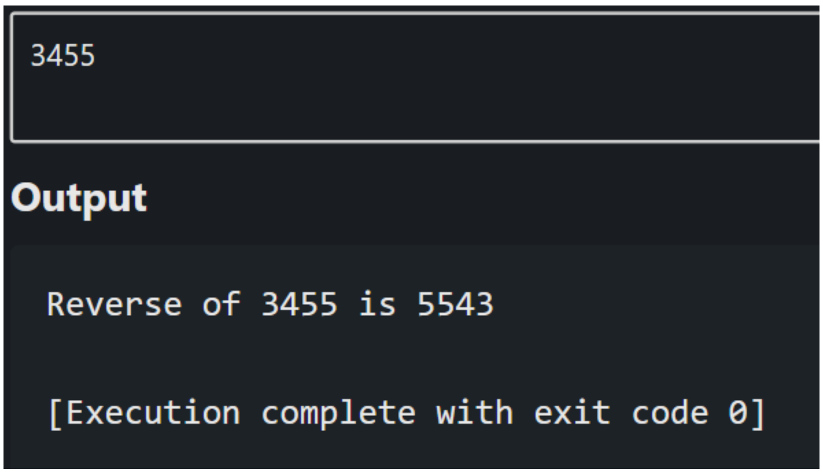
4.6AIM

Shell program to print the reverse of a number using function.

Source Code

```
reverse_number() {
   echo $1 | grep -o . | tac | tr -d '\n'
}
read -p "Enter a number: " num
result=$(reverse_number $num)
echo "Reverse of $num is $result"
```

Output



```
3455
```

Output

```
Reverse of 3455 is 5543

[Execution complete with exit code 0]
```

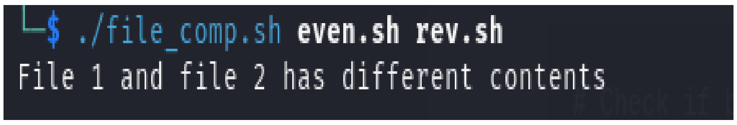
4.7AIM

Shell script, which receives two filenames as arguments. It checks whether the two files' contents are same or not. If they are same then second file is deleted.

Source Code

```
if [ "$(cat "$1")" = "$(cat "$2")" ]; then
    rm "$2"
    echo "Files have same contents. Second file deleted."
 else
    echo "Files have different contents."
 Fi
```

Output



```
└─$ ./file_comp.sh even.sh rev.sh
File 1 and file 2 has different contents
```

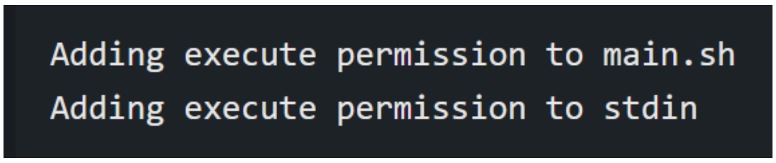
4.8AIM

Shell script to check executable rights for all files in the current directory, if a file does not have the execute permission, then make it executable.

Source Code

```
for file in *; do
     if [ ! -x "$file" ]; then
         chmod +x "$file"
         echo "Adding execute permission to $file"
     else
         echo "$file already has execute permissions."
     fi
 done
```

Output



```
Adding execute permission to main.sh
Adding execute permission to stdin
```

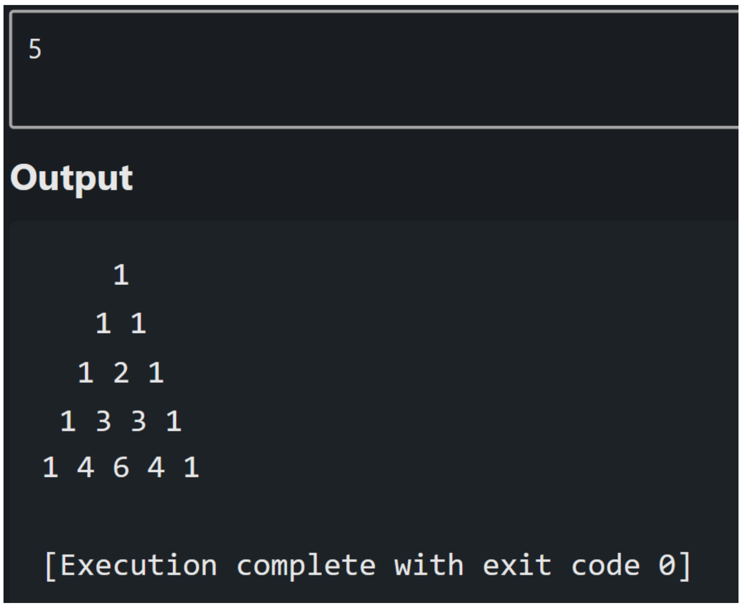
4.9AIM

Shell program to create Pascal’s triangle.

Source Code

```
read -p "Enter the number of rows for Pascal's Triangle: " rows
 for ((i = 0; i < rows; i++)); do
     printf "%*s" $((rows - i - 1)) ""
     coef=1
     for ((j = 0; j <= i; j++)); do
         printf "%d " $coef
         coef=$((coef * (i - j) / (j + 1)))
     done
     echo
 done
```

Output



```
5

Output

    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1

[Execution complete with exit code 0]
```

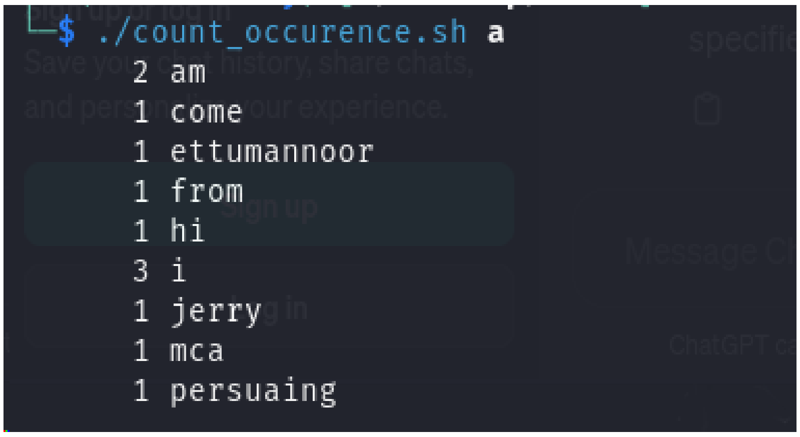
4.10 AIM

Shell script to find out the unique words in a file and count the occurrence of each of these words

Source Code

```
[ $# -ne 1 ] && echo "Usage: $0 <file>" && exit 1
 tr -s '[:space:]' '\n' < "$1" | tr '[:upper:]' '[:lower:]' | sort | uniq -c
```

Output



```
└─$ ./count_occurence.sh a
      2 am
      1 come
      1 ettumannoor
      1 from
      1 hi
      3 i
      1 jerry
      1 mca
      1 persuaing
```

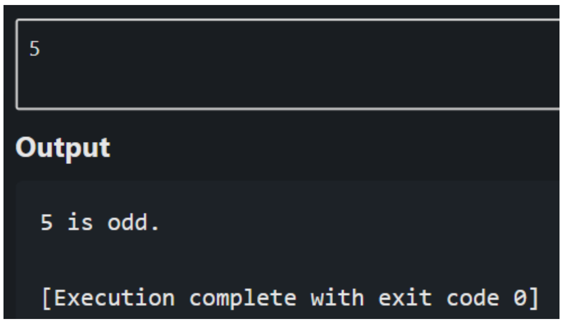
Program-1

Write a Shell program to check the given number is even or odd.

Source Code

```
read -p "Enter a number: " num
((num % 2 == 0)) && echo "$num is even." || echo "$num is odd."
```

Output



```
5
```

Output

```
5 is odd.

[Execution complete with exit code 0]
```

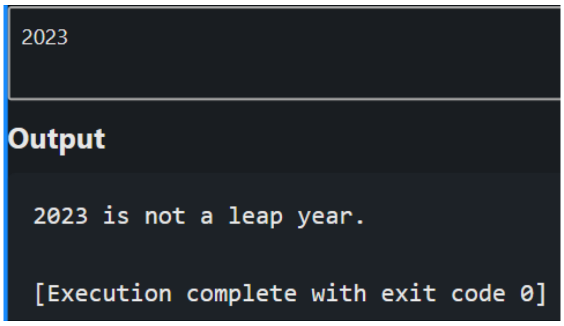
Program-2

Write a Shell program to check a leap year.

Source Code

```
read -p "Enter a year: " year
if (( (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) )); then
    echo "$year is a leap year."
else
    echo "$year is not a leap year."
fi
```

Output



```
2023
```

Output

```
2023 is not a leap year.

[Execution complete with exit code 0]
```

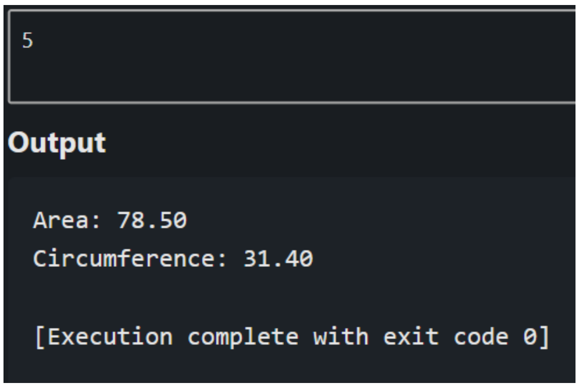
Program-3

Write a Shell program to find the area and circumference of a circle.

Source Code

```
read -p "Enter radius: " radius
area=$(echo "scale=2; 3.14 * $radius * $radius" | bc)
circumference=$(echo "scale=2; 2 * 3.14 * $radius" | bc)
echo "Area: $area"
echo "Circumference: $circumference"
```

Output



```
5

Output

Area: 78.50
Circumference: 31.40

[Execution complete with exit code 0]
```

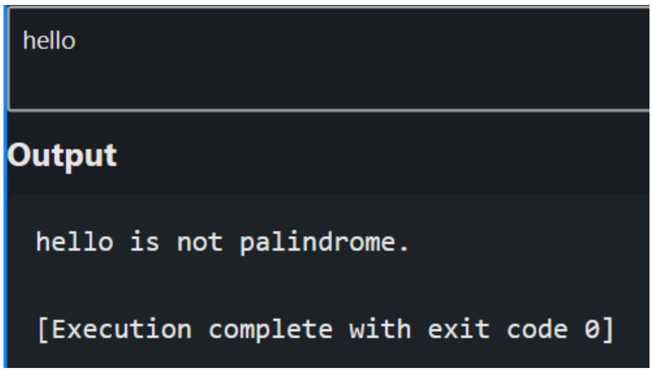
Program-4

Write a Shell program to check the given string is palindrome or not.

Source Code

```
read -p "Enter a string: " str
echo "$str" | rev | grep -q "^$str\$" && echo "$str is palindrome." || echo "$str is not palindrome."
```

Output



```
hello

Output

hello is not palindrome.

[Execution complete with exit code 0]
```

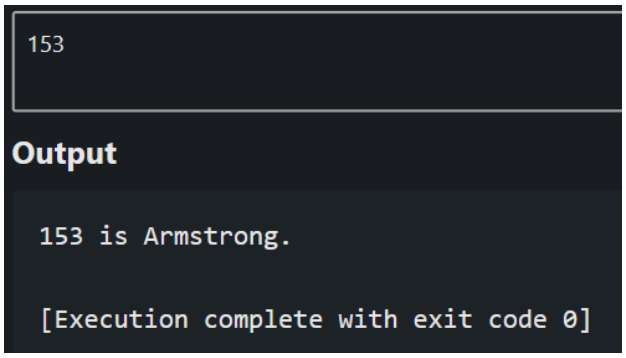
Program-5

Write a Shell program to check the given integer is Armstrong number or not.

Source Code

```
read -p "Enter a number: " num
s=0; for ((i=0;i<${#num};i++)); do ((s+=(${num:i:1})**${#num})); done
echo $s | grep -q "$num" && echo "$num is Armstrong." || echo "$num is not Armstrong."
```

Output



```
153
```

Output

```
153 is Armstrong.

[Execution complete with exit code 0]
```

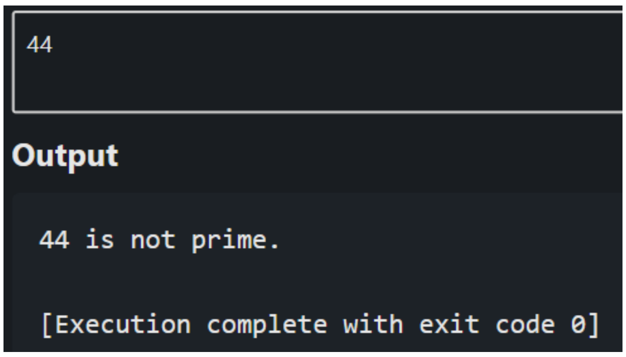
Program-6

Write a Shell program to check the given integer is prime or not.

Source Code

```
read -p "Enter a number: " num
is_prime=true
for (( i=2; i<num; i++ )); do
   [ $((num % i)) -eq 0 ] && is_prime=false && break
done
$is_prime && echo "$num is prime." || echo "$num is not prime."
```

Output



```
44
```

Output

```
44 is not prime.

[Execution complete with exit code 0]
```

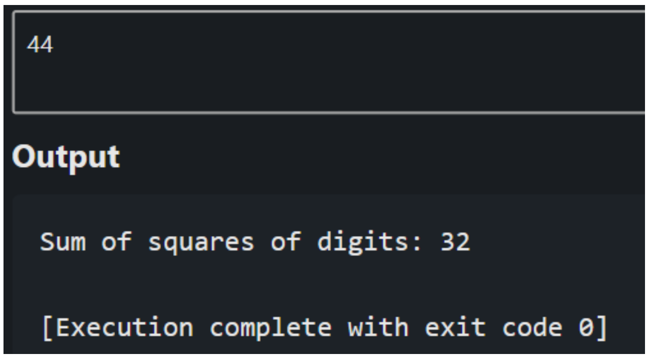
Program-7

Write a Shell program to find the sum of square of individual digits of a number.

Source Code

```
read -p "Enter a number: " num
sum=$(echo "$num" | grep -o . | awk '{s+=$1^2} END {print s}')
echo "Sum of squares of digits: $sum"
```

Output



```
44
```

```
Sum of squares of digits: 32

[Execution complete with exit code 0]
```

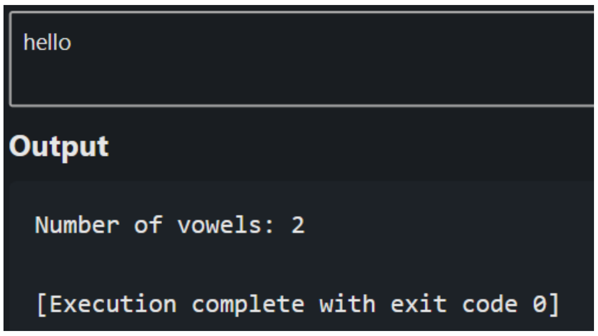
Program-8

Write a Shell program to count the number of vowels in a line of text.

Source Code

```
read -p "Enter a line of text: " text
vowel_count=$(echo "$text" | grep -io '[aeiou]' | wc -l)
echo "Number of vowels: $vowel_count"
```

Output



```
hello
```

```
Number of vowels: 2

[Execution complete with exit code 0]
```

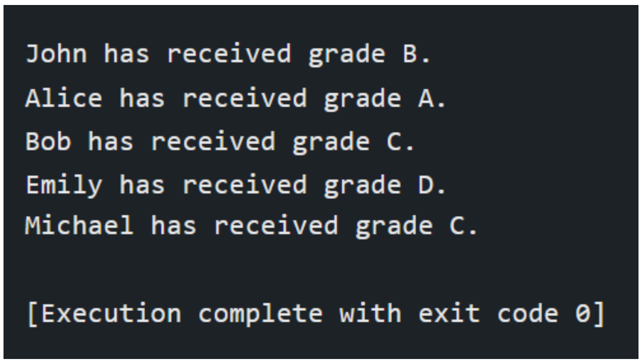
Program-9

Write a Shell program to display student grades.

Source Code

```
while read -r name marks; do
    grade=$(awk -v m="$marks" 'BEGIN{ grades="FDCBA"; print (m>=90) ? "A" : (m>=80) ? "B" : (m>=70) ? "C" : (m>=60) ? "D" : "F" }')
    echo "$name has received grade $grade."
done < input.txt
```

Output



```
John has received grade B.
Alice has received grade A.
Bob has received grade C.
Emily has received grade D.
Michael has received grade C.

[Execution complete with exit code 0]
```

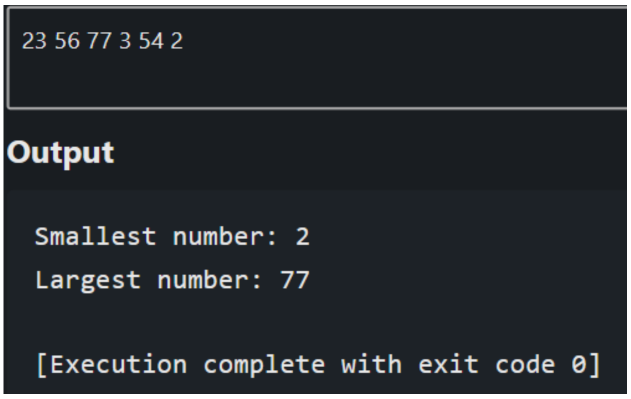
Program-10

Write a Shell program to find the smallest and largest numbers from a set of numbers.

Source Code

```
read -p "Enter numbers separated by spaces: " -a nums
echo "Smallest number: $(printf "%s\n" "${nums[@]}" | sort -n | head -n 1)"
echo "Largest number: $(printf "%s\n" "${nums[@]}" | sort -rn | head -n 1)"
```

Output



```
23 56 77 3 54 2
```

Output

```
Smallest number: 2
Largest number: 77

[Execution complete with exit code 0]
```

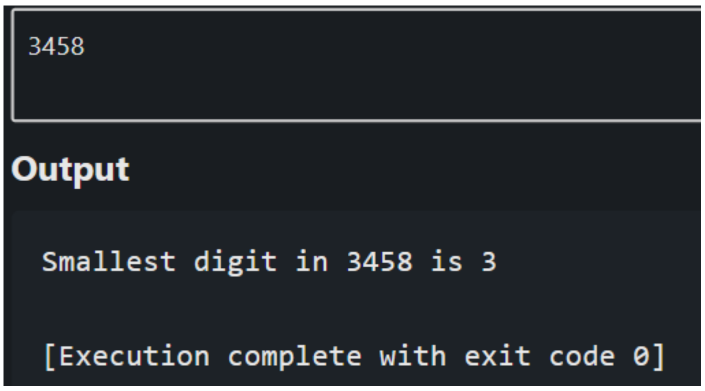
Program-11

Write a Shell program to find the smallest digit from a number.

Source Code

```
read -p "Enter a number: " num
smallest=$(echo "$num" | grep -o '[0-9]' | sort -n | head -n 1)
echo "Smallest digit in $num is $smallest"
```

Output



```
3458
```

Output

```
Smallest digit in 3458 is 3

[Execution complete with exit code 0]
```

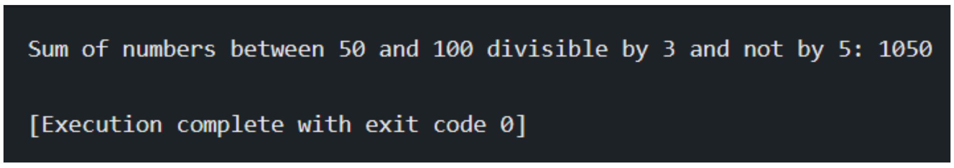
Program-12

Write a Shell program to find the sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5

Source Code

```
sum=$(seq 50 100 | awk '{if($1%3==0 && $1%5!=0) sum+=$1} END{print sum}')
echo "Sum of numbers between 50 and 100 divisible by 3 and not by 5: $sum"
```

Output



```
Sum of numbers between 50 and 100 divisible by 3 and not by 5: 1050

[Execution complete with exit code 0]
```

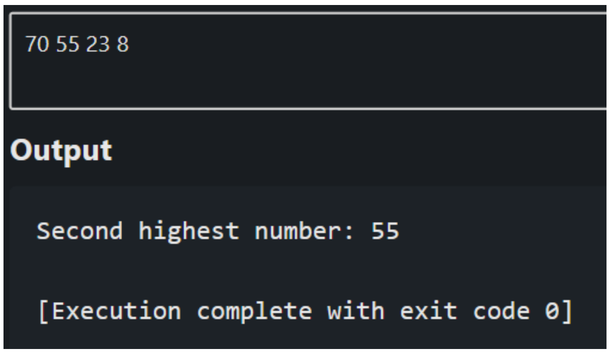
Program-13

Write a Shell program to find the second highest number from a set of numbers.

Source Code

```
read -p "Enter numbers separated by spaces: " -a nums
second_highest=$(printf '%s\n' "${nums[@]}" | sort -rn | awk 'NR==2{print $1}')
echo "Second highest number: $second_highest"
```

Output



```
70 55 23 8
```

Output

```
Second highest number: 55

[Execution complete with exit code 0]
```

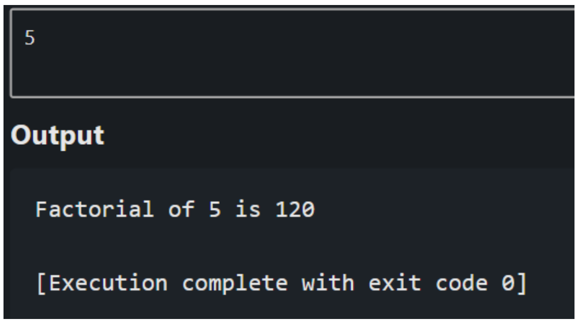
Program-14

Write a Shell program to find the factorial of a number using for loop.

Source Code

```
read -p "Enter a number: "  num
factorial=1
for (( i=1; i<=num; i++ )); do
   factorial=$((factorial * i))
done
echo "Factorial of $num is $factorial"
```

Output



```
5
```

Output

```
Factorial of 5 is 120

[Execution complete with exit code 0]
```

Program-15

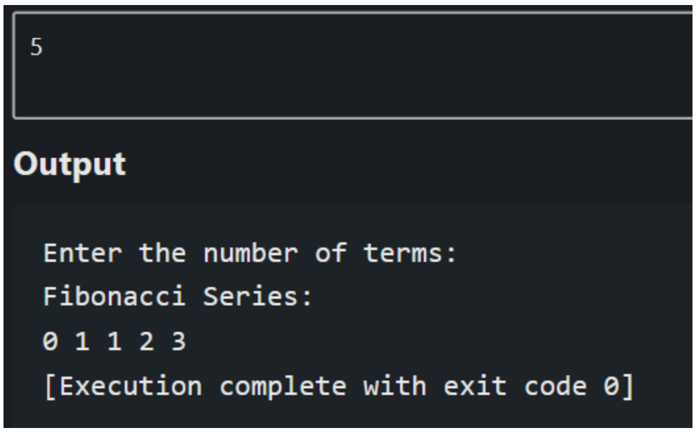
Write a Shell program to generate Fibonacci series.

Source Code

```
a=0
 b=1
 echo "Enter the number of terms: "
 read n
 echo "Fibonacci Series:"
```

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

Output



```
5
```

Output

```
Enter the number of terms:
Fibonacci Series:
0 1 1 2 3
[Execution complete with exit code 0]
```

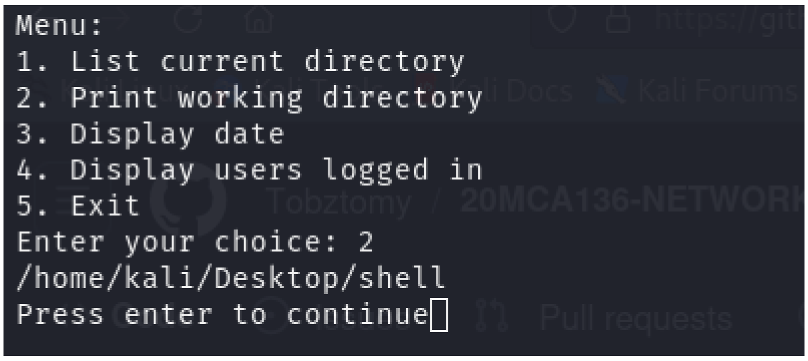
Program-16

Write a Menu driven Shell script that Lists current directory, Prints Working Directory, displays Date and displays Users logged in.

Source Code

```
while true; do
    echo "1. List current directory"
    echo "2. Print working directory"
    echo "3. Display date"
    echo "4. Display users logged in"
    echo "5. Exit"
    read -p "Enter your choice: " choice

    case $choice in
        1) ls ;;
        2) pwd ;;
        3) date ;;
        4) who ;;
        5) echo "Exiting..."; exit ;;
        *) echo "Invalid option. Please choose again." ;;
    esac
done
```

Output



```
Menu:
1. List current directory
2. Print working directory
3. Display date
4. Display users logged in
5. Exit
Enter your choice: 2
/home/kali/Desktop/shell
Press enter to continue
```

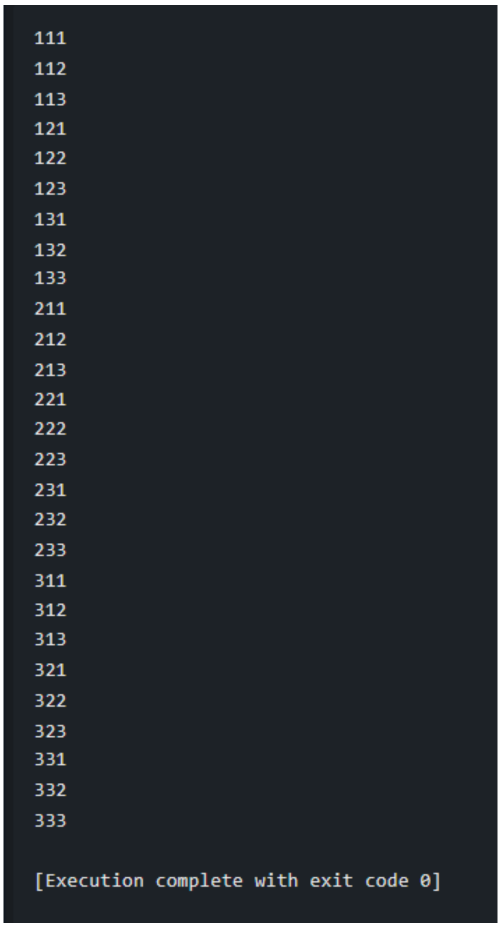
Program-17

Write a Shell program to generate all combinations of 1, 2, and 3 using loop.

Source Code

```
for i in {1..3}{1..3}{1..3}; do
   echo $i
 done
```

Output



```
111
112
113
121
122
123
131
132
133
211
212
213
221
222
223
231
232
233
311
312
313
321
322
323
331
332
333

[Execution complete with exit code 0]
```

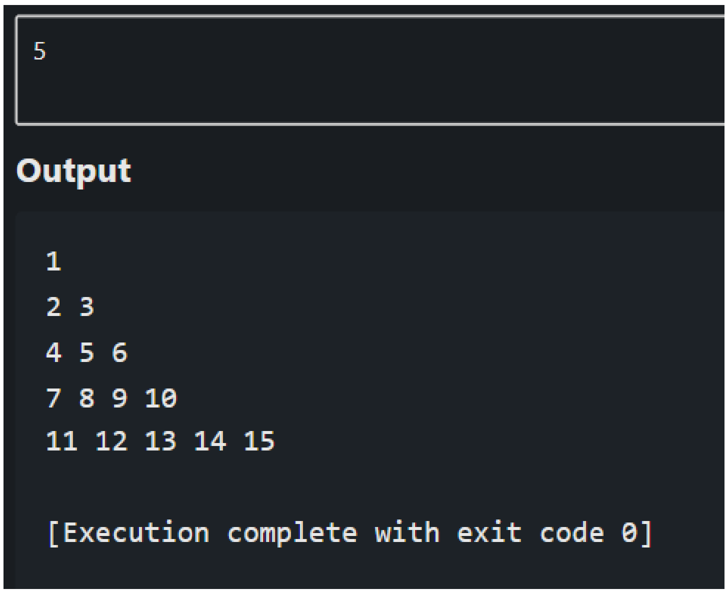
Program-18

Write a Shell program to create the number series.

Source Code

```
read -p "Enter the number of rows: " rows
 num=1
 for ((i=1; i<=rows; i++)); do
     for ((j=1; j<=i; j++)); do
         echo -n "$num "
         ((num++))
     done
     echo
 done
```

Output



```
5
```

Output

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

[Execution complete with exit code 0]
```

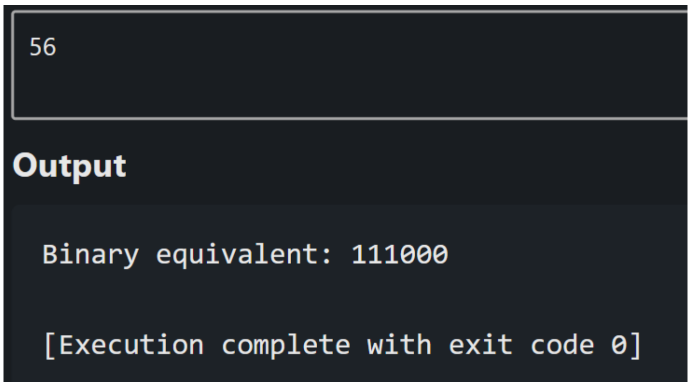
Program-19

Write a Decimal to Binary Conversion Shell Script.

Source Code

```
read -p "Enter a decimal number: " decimal
 echo "Binary equivalent: $(echo "obase=2; $decimal" | bc)"
```

Output



```
56
```

Output

```
Binary equivalent: 111000

[Execution complete with exit code 0]
```

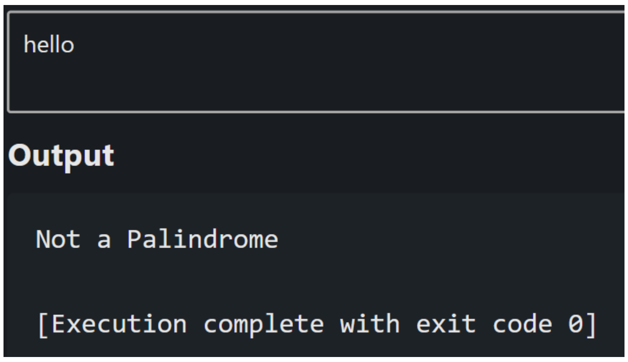
Program-20

Write a Shell Script to Check Whether a String is Palindrome or not.

Source Code

```
read -p "Enter a string: " str
 [ "$str" = "$(echo $str | rev)" ] && echo "Palindrome" || echo "Not a Palindrome"
```

Output



```
hello
```

Output

```
Not a Palindrome

[Execution complete with exit code 0]
```

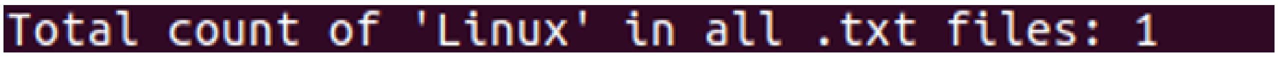
Program-21

Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across files present in subdirectories.

Source Code

```
count=$(grep -r -o -w "Linux" *.txt | wc -l)
 echo "Total count of 'Linux' in .txt files: $count"
```

Output



```
Total count of 'Linux' in all .txt files: 1
```

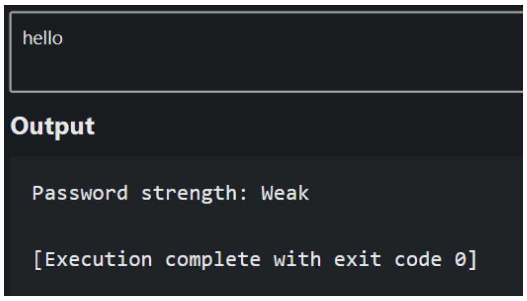
Program-22

Write a shell script to validate password strength. Here are a few assumptions for the password string. (Length – minimum of 8 characters. Contain both alphabet and number. Include both the small and capital case letters.)

Source Code

```
read -p "Enter your password: " password
 if [[ ${#password} -ge 8 && "$password" =~ [0-9] && "$password" =~ [a-z] && "$password" =~
[A-Z] ]]; then
    echo "Password strength: Strong"
 else
    echo "Password strength: Weak"
 fi
```

Output



```
hello

Output

  Password strength: Weak

  [Execution complete with exit code 0]
```

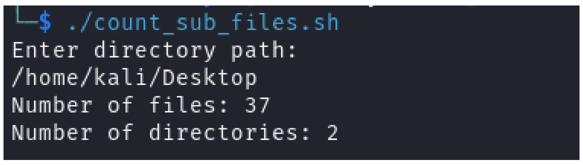
Program-23

Write a shell script to print the count of files and subdirectories in the specified directory.

Source Code

```
read -p "Enter directory path: " directory
 #error / exist check ( optional )
 [ -z "$directory" ] && { echo "Usage: $0 <directory>"; exit 1; }
 file_count=$(find "$directory" -type f | wc -l)
 dir_count=$(find "$directory" -type d | wc -l)
 echo "Files: $file_count"
 echo "Directories: $dir_count"
```

Output



```
└─$ ./count_sub_files.sh
Enter directory path:
/home/kali/Desktop
Number of files: 37
Number of directories: 2
```

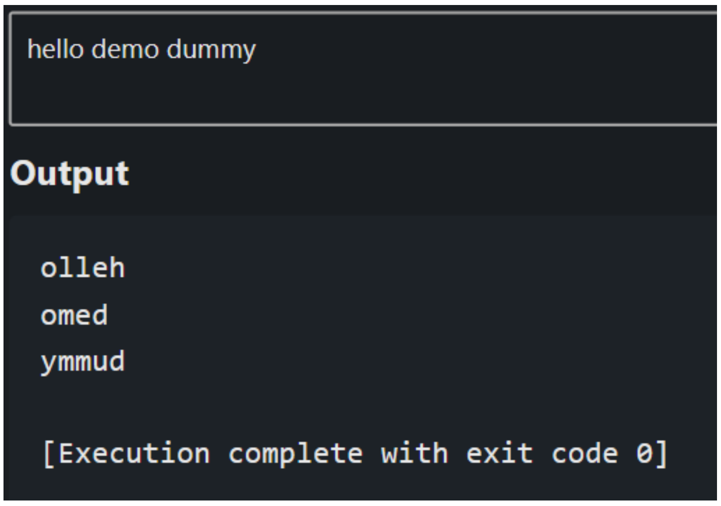
Program-24

Write a shell script to reverse the list of strings and reverse each string further in the list.

Source Code

```
read -p "Enter strings separated by spaces: " -a list
 for str in "${list[@]}"; do
    echo "$str" | rev
 done
```

Output



```
hello demo dummy
```

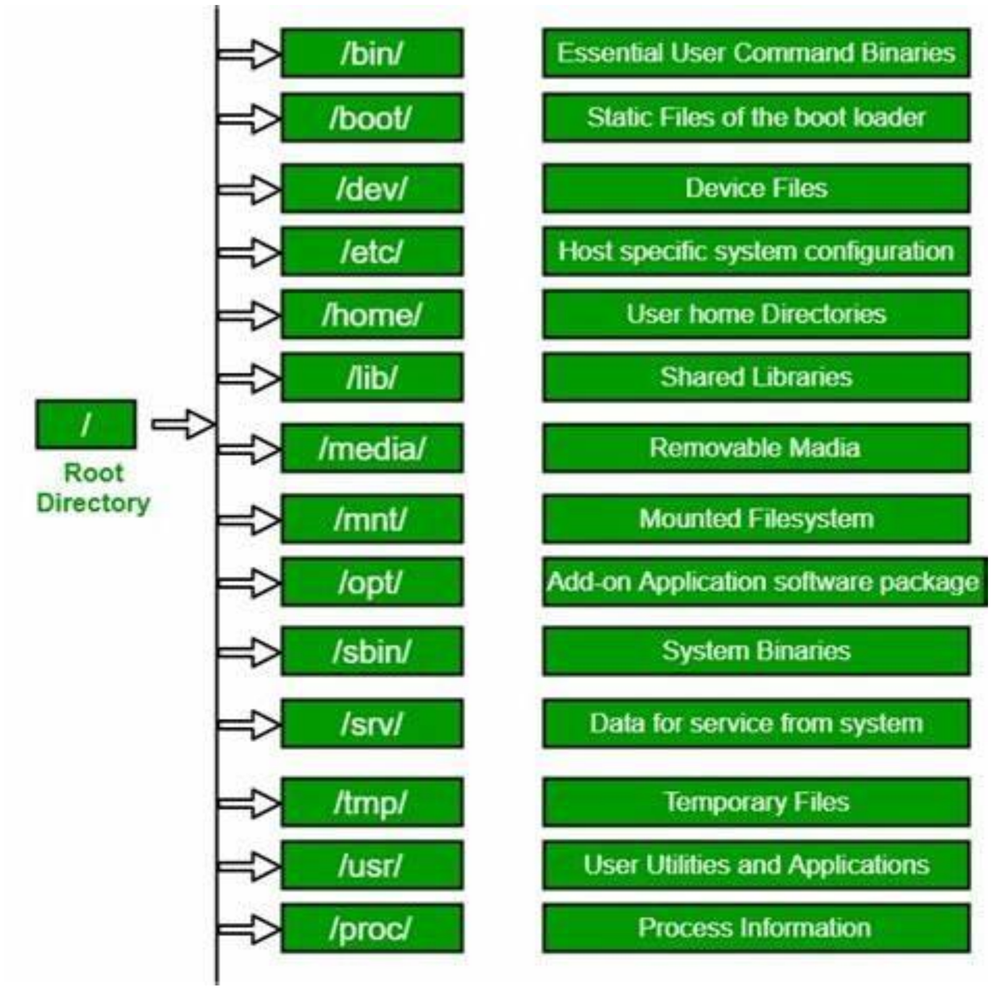
Output

```
olleh
omed
ymmud

[Execution complete with exit code 0]
```

5. File system hierarchy in a common Linux distribution

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating systems. It is maintained by the Linux Foundation.

- ❖ In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.
- ❖ Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.
- ❖ Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS, and are not considered authoritative for platforms other than Linux.



Root directory

The root directory, denoted by a forward slash (/), marks the starting point of the Linux File System. It is positioned at the topmost level, serving as the foundation for the entire file system. All other directories are classified as children or grandchildren of the root directory.

/home directory

The **/home** directory serves as a private directory for each user, containing individual user home directories such as Documents, Downloads, Desktop, Pictures, etc. Each registered user on the system has their own subdirectory within /home. This directory ensures user privacy and isolates user data from system-wide files.

/bin directory

The **/bin** directory holds essential binary files (executables) that are essential for the proper functioning of the system. For example, basic Linux commands like **cp, ls,** and **echo** exist as binary files within the **/bin** directory. These files are accessible to all users.

/boot

The /boot directory contains the files needed to boot the system – for example, the GRUB boot loader's files and your Linux kernels are stored here. The boot loader's configuration files are not located here, though – they are in /etc with the other configuration files.

/dev

Linux exposes devices as files, and the /dev directory contains several special files that represent devices. These are not actual files as we know them, but they appear as files – for example, /dev/sda represents the first SATA drive in the system. If you wanted to partition it, you could start a partition editor and tell it to edit /dev/sda. This directory also contains pseudo-devices, which are virtual devices that do not actually correspond to hardware. For example, /dev/random produces random numbers. /dev/null is a special device that produces no output and automatically discards all input – when you pipe the output of a command to /dev/null, you discard it.

/lib

The /lib directory contains libraries needed by the essential binaries in the /bin and /sbin folder. Libraries needed by the binaries in the /usr/bin folder are in /usr/lib.

/media

The /media directory contains subdirectories where removable media devices inserted into the computer are mounted. For example, when you insert a CD into your Linux system, a directory will automatically be created inside the /media directory. You can access the contents of the CD inside this directory.

/mnt

Historically speaking, the /mnt directory is where system administrators mounted temporary file systems while using them. For example, if you're mounting a Windows partition to perform some file recovery operations, you might mount it at /mnt/windows. However, you can mount other file systems anywhere on the system.

/opt

The /opt directory contains subdirectories for optional software packages. It is commonly used by proprietary software that does not obey the standard file system hierarchy – for example, a proprietary program might dump its files in /opt/application when you install it.

/proc

The /proc directory like the /dev directory because it does not contain standard files. It contains special files that represent system and process information.

/sbin

The /sbin directory is like the /bin directory. It contains essential binaries that are generally intended to be run by the root user for system administration

/srv

The /srv directory contains “data for services provided by the system.” If you were using the Apache HTTP server to serve a website, you’d likely store your website’s files in a directory inside the /srv directory.

/tmp

Applications store temporary files in the /tmp directory. These files are generally deleted whenever your system is restarted and may be deleted at any time by utilities such as tmpwatch**.**

/usr

The /usr directory contains applications and files used by users, as opposed to applications and files used by the system. For example, non-essential applications are located inside the /usr/bin directory instead of the /bin directory and non-essential system administration binaries are in the /usr/sbin directory instead of the /sbin directory. Libraries for each are located inside the /usr/lib directory. The /usr directory also contains other directories – for example, architecture-independent files like graphics are in /usr/share. The /usr/local directory is where locally compiled applications install to by default – this prevents them from mucking up the rest of the system.

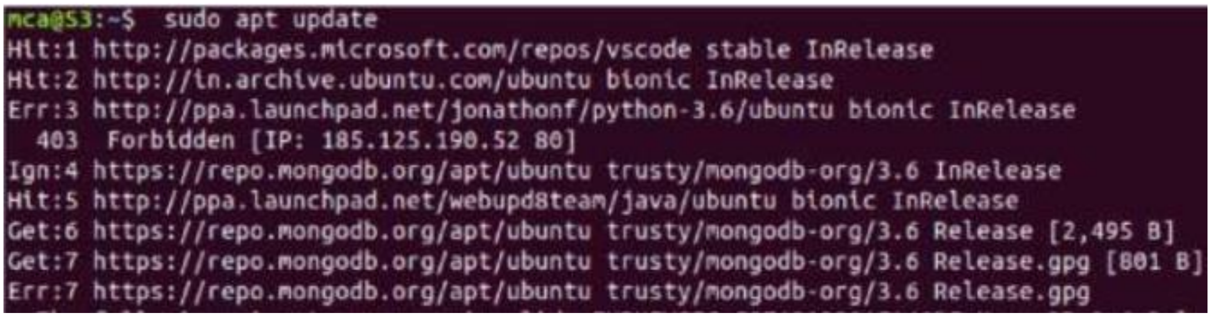
6. Installation and configuration of LAMP stack. Deploy in an open-source application such as phpMyAdmin and WordPress.

Procedure

Install Apache2

- **Update your system:**

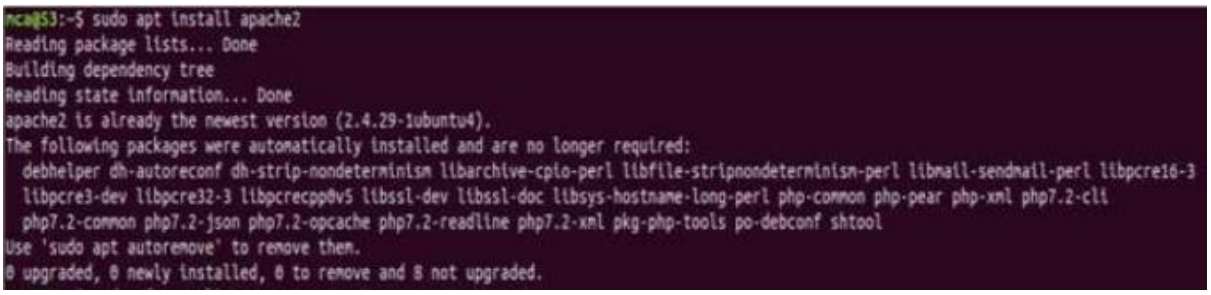
sudo apt update



```
mca@S3:~$  sudo apt update
Hit:1 http://packages.microsoft.com/repos/vscode stable InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Err:3 http://ppa.launchpad.net/jonathonf/python-3.6/ubuntu bionic InRelease
  403  Forbidden [IP: 185.125.190.52 80]
Ign:4 https://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.6 InRelease
Hit:5 http://ppa.launchpad.net/webupd8team/java/ubuntu bionic InRelease
Get:6 https://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.6 Release [2,495 B]
Get:7 https://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.6 Release.gpg [801 B]
Err:7 https://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.6 Release.gpg
```

- **Install Apache using apt:**

sudo apt install apache2



```
mca@S3:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.29-1ubuntu4).
The following packages were automatically installed and are no longer required:
  debhelper dh-autoreconf dh-strip-nondeterminism libarchive-cpio-perl libfile-stripnondeterminism-perl libmail-sendmail-perl libpcre16-3
  libpcre3-dev libpcre32-3 libpcrecpp0v5 libssl-dev libssl-doc libsys-hostname-long-perl php-common php-pear php-xml php7.2-cli
  php7.2-common php7.2-json php7.2-opcache php7.2-readline php7.2-xml pkg-php-tools po-debconf shtool
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
```

Confirm that Apache is now running with the following command:

sudo systemctl status apache2

- **If it is not working !**

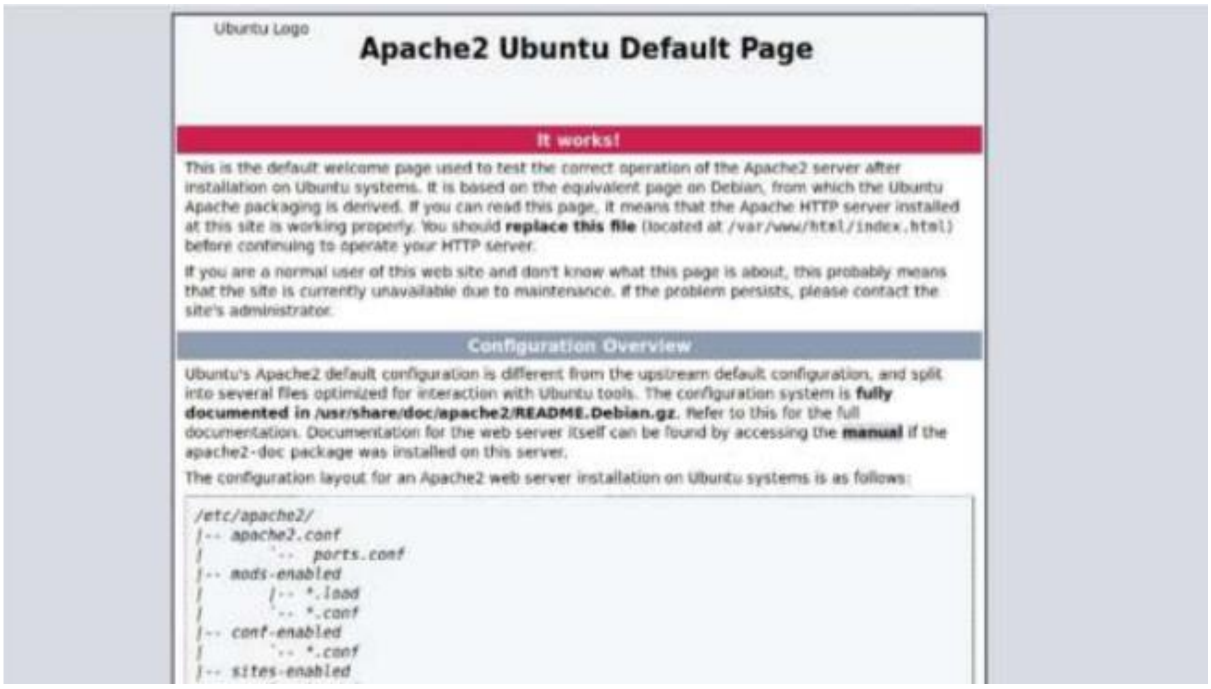
sudo systemctl stop apache2 # to stop if running

sudo systemctl start apache2 # to start if not running

- **Once installed, test by accessing your server's IP in your browser:**

http://127.0.0.1/

http://localhost/


Ubuntu Logo

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```

Install mariadb

sudo apt install mariadb-server mariadb-client

sudo systemctl status mysql # to check status

sudo systemctl start mysql # if not running

sudo mysql_secure_installation # Secure your newly installed MariaDB

Install PHP and commonly used modules

- sudo apt install php libapache2-mod-php php-apcache php-cli php-gd php- curl php-mysql
- sudo systemctl restart apache2
- **Test PHP Processing on Web Server**
- sudo nano /var/www/html/phpinfo.php
- **Inside the file, type in the valid PHP code:**

```
<?php

phpinfo ();

?>
```

- **Press CTRL + X to save and close the file. Press y and ENTER to confirm Open a browser and type in your IP address/phpinfo.php** http://127.0.0.1/phpinfo.php

Install phpmyadmin

sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-

curl

sudo systemctl restart apache2

- **Open a browser: http://localhost/phpmyadmin**

username: root

password: yourpassword If php my admin page not found:

nano /etc/apache2/apache2.conf

- **Add this line to last of the file.**

Press CTRL + X to save and close the file. Press y and ENTER to confirm

Include /etc/phpmyadmin/apache.conf

- **restart apache2 - now try: http://localhost/phpmyadmin**

sudo systemctl restart apache2

- **If any problem for login run the following command**

sudo mysql

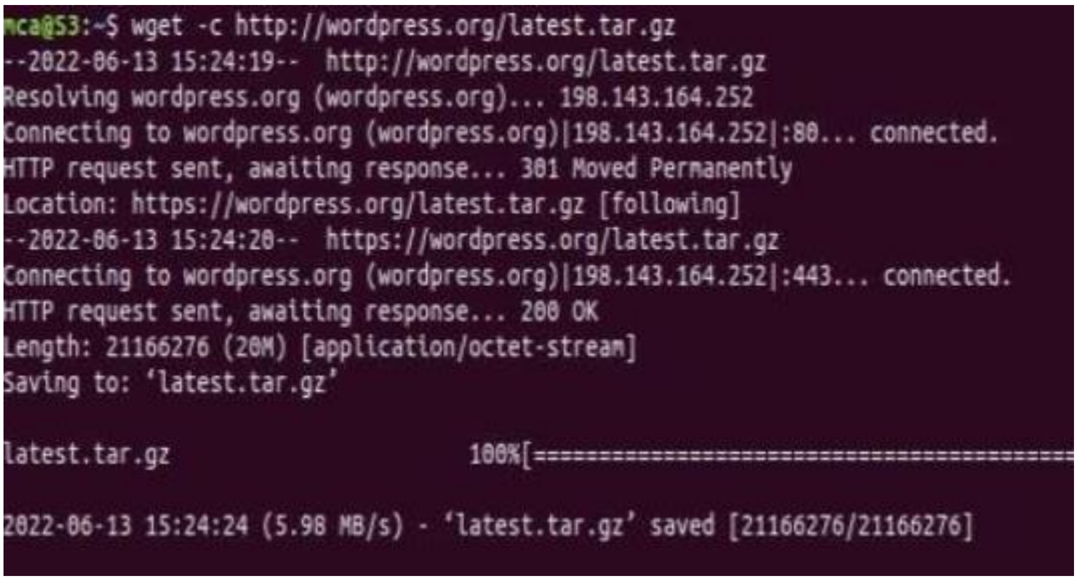
ALTER USER root@localhost IDENTIFIED BY "yourpassword";

Install WordPress with LAMP on Ubuntu 18.04

Step 1 – Download WordPress

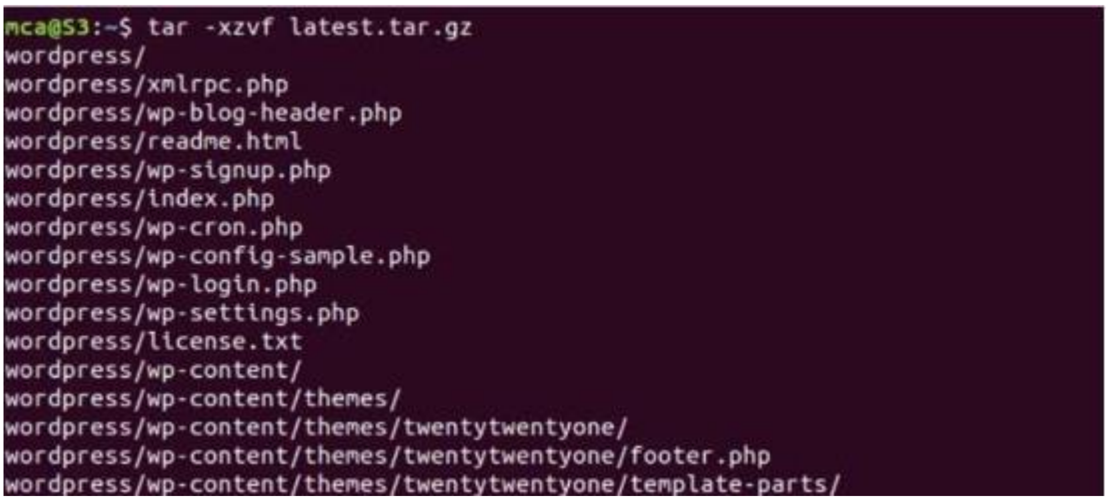
Download the latest version of the WordPress package and extract it by issuing the commands below on the terminal:

- wget -c http://wordpress.org/latest.tar.gz



```
mca@S3:~$ wget -c http://wordpress.org/latest.tar.gz
--2022-06-13 15:24:19--  http://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://wordpress.org/latest.tar.gz [following]
--2022-06-13 15:24:20--  https://wordpress.org/latest.tar.gz
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21166276 (20M) [application/octet-stream]
Saving to: ‘latest.tar.gz’

latest.tar.gz                 100%[============================================

2022-06-13 15:24:24 (5.98 MB/s) - ‘latest.tar.gz’ saved [21166276/21166276]
```

- tar -xzvf latest.tar.gz



```
mca@S3:~$ tar -xzvf latest.tar.gz
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
wordpress/index.php
wordpress/wp-cron.php
wordpress/wp-config-sample.php
wordpress/wp-login.php
wordpress/wp-settings.php
wordpress/license.txt
wordpress/wp-content/
wordpress/wp-content/themes/
wordpress/wp-content/themes/twentytwentyone/
wordpress/wp-content/themes/twentytwentyone/footer.php
wordpress/wp-content/themes/twentytwentyone/template-parts/
```

Then move the WordPress files from the extracted folder to the Apache default root directory, /var/www/html/:

- sudo mv wordpress/* /var/www/html/

Next, set the correct permissions on the website directory, that is give ownership of the WordPress files to the webserver as follows:

- sudo chown -R www-data:www-data /var/www/html/
- sudo chmod -R 755 /var/www/html/

Step 2 – Creating a MySQL Database and User for WordPress

The first step you will take is a preparatory one. Even though MySQL is already installed, you still need to create a database to manage and store the user information for WordPress to use. To get started, log into the MySQL root(administrative) account by issuing the following command:

- sudo mysql

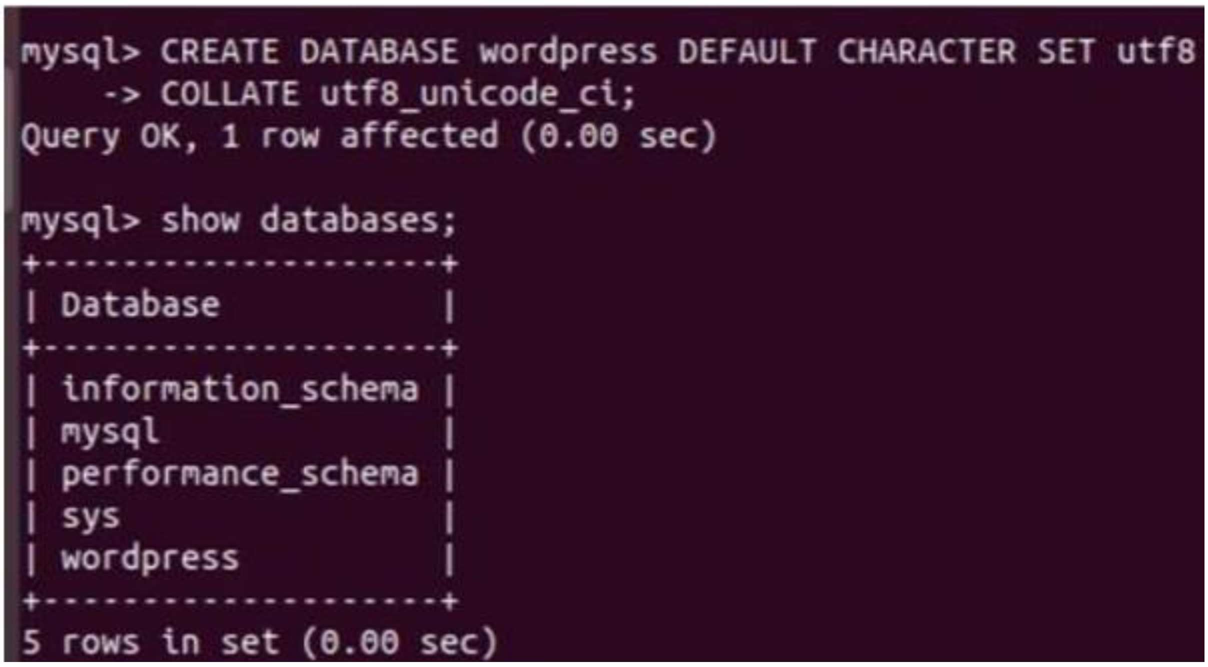
You will be prompted for the password you set for the MySQL root account when you installed the software. However, if you have password authentication enabled for your root user, you can run the following command and enter your password information when prompted:

- mysql -u root –p

From there, you will create a new database that WordPress will control. You can call this whatever you would like, but we will be using WordPress in this guide an example. Create the database for WordPress by writing the following:

- CREATE DATABASE WordPress DEFAULT CHARACTER

SET utf8 COLLATE utf8_unicode_ci;



```
mysql> CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8
    -> COLLATE utf8_unicode_ci;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
5 rows in set (0.00 sec)
```

Next, you are going to create a separate MySQL user account that you will use exclusively to operate on the new database. Creating one-function databases and accounts is a good idea from a management and security standpoint. We will use the name WordPress user as an example in this guide. Feel free to change this if you would like. You can create this account, set a password for it, and then grant it access to thedatabase you created all by running the following command. Remember tochoose a strong password here for your database user:

- GRANT ALL ON wordpress.* TO

'wordpressuser'@'localhost'IDENTIFIED BY 'password';

After creating this user, flush the privileges to ensure that the current instance ofMySQL knows about the recent changes you've made:

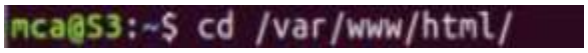
- FLUSH PRIVILEGES;

Exit out of MySQL:

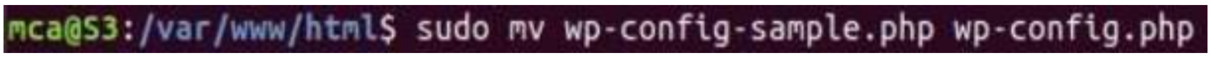
- EXIT

You now have a database and user account in MySQL, each made specifically for WordPress. Go the /var/www/html/ directory and rename existing wp-config-sample.php to wpconfig.php. Also, make sure to remove the default Apache index page.

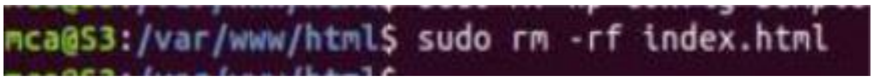
- cd /var/www/html/
- sudo mv wp-config-sample.php wp-config.php
- sudo rm -rf index.html



```
mca@S3:~$ cd /var/www/html/
```



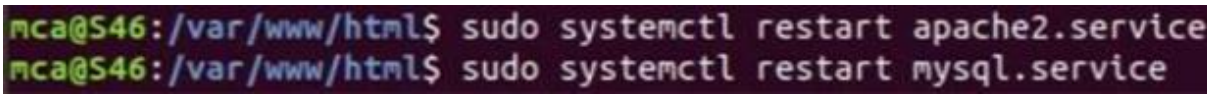
```
mca@S3:/var/www/html$ sudo mv wp-config-sample.php wp-config.php
```



```
mca@S3:/var/www/html$ sudo rm -rf index.html
```

Then update it with your database information under the MySQL settings section (refer to the highlighted boxes in the image below): This setting can be added after the database connection settings, or anywhere else in the file: Save and close the file when you are finished. Restart the web server and mysql service using the commands below:

- sudo systemctl restart apache2.service
- sudo systemctl restart mysql.service



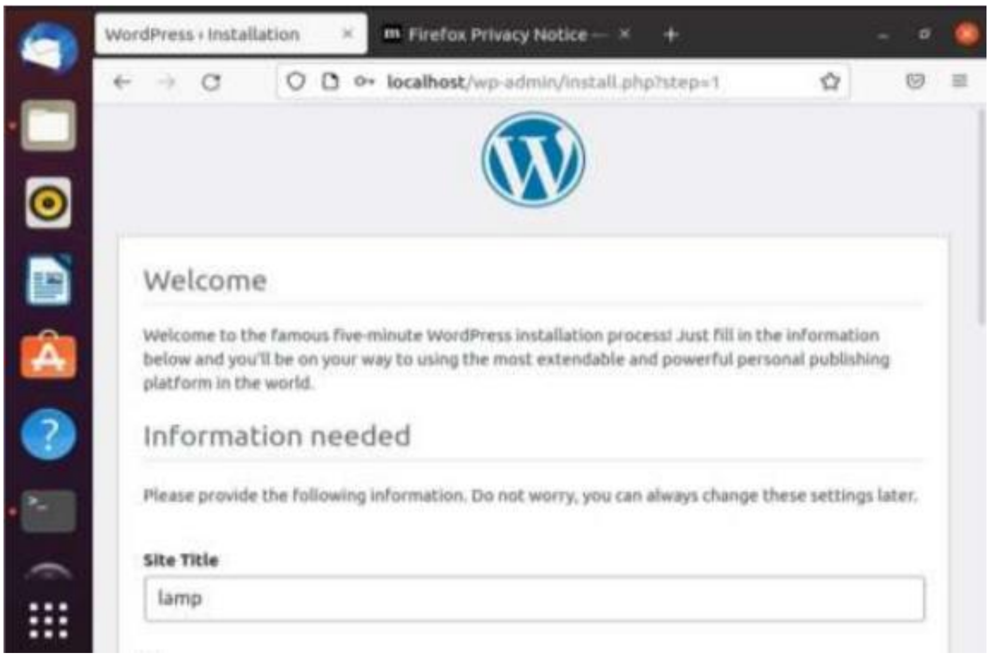
```
mca@S46:/var/www/html$ sudo systemctl restart apache2.service
mca@S46:/var/www/html$ sudo systemctl restart mysql.service
```

Step 3 – Completing the Installation Through the Web Interface

Now that the server configuration is complete, you can complete the installation through the web interface. In your web browser, navigate to your server's domain name or public IP address:

- https://server_domain_or_IP

Select the language you would like to use: Next you will be directed to the main setup page. Select a name for your WordPress site and choose a username (it is recommended not to choose something like "admin" for security purposes). A strong password is generated automatically. Save this password or select an alternative strong password. Enter your email address and select whether you want to discourage search engines from indexing your site: Once you log in, you will be taken to the WordPress administration dashboard: From there, you can begin using and customizing your WordPress site



Welcome

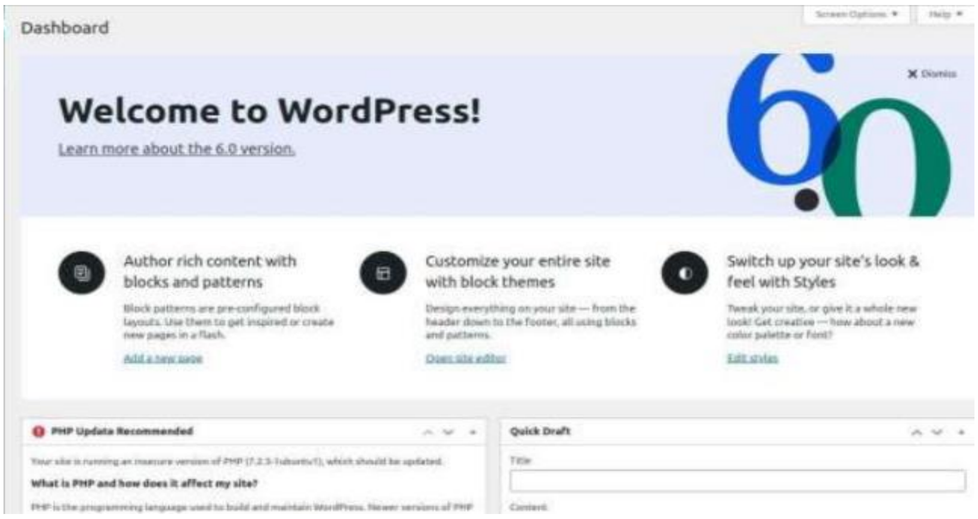
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title: lamp

Once you log in, you will be taken to the WordPress administration dashboard: From there, you can begin using and customizing your WordPress site.



Welcome to WordPress!

Learn more about the 6.0 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

PHP Update Recommended

Your site is running an insecure version of PHP (7.2.3-1ubuntu1), which should be updated.

What is PHP and how does it affect my site?

PHP is the programming language used to build and maintain WordPress. Newer versions of PHP

Quick Draft

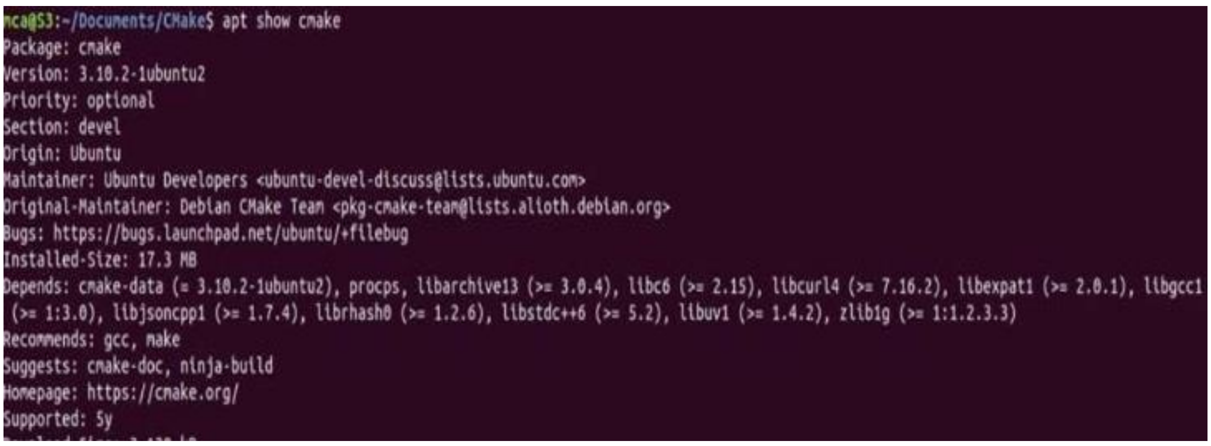
Title

Content

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

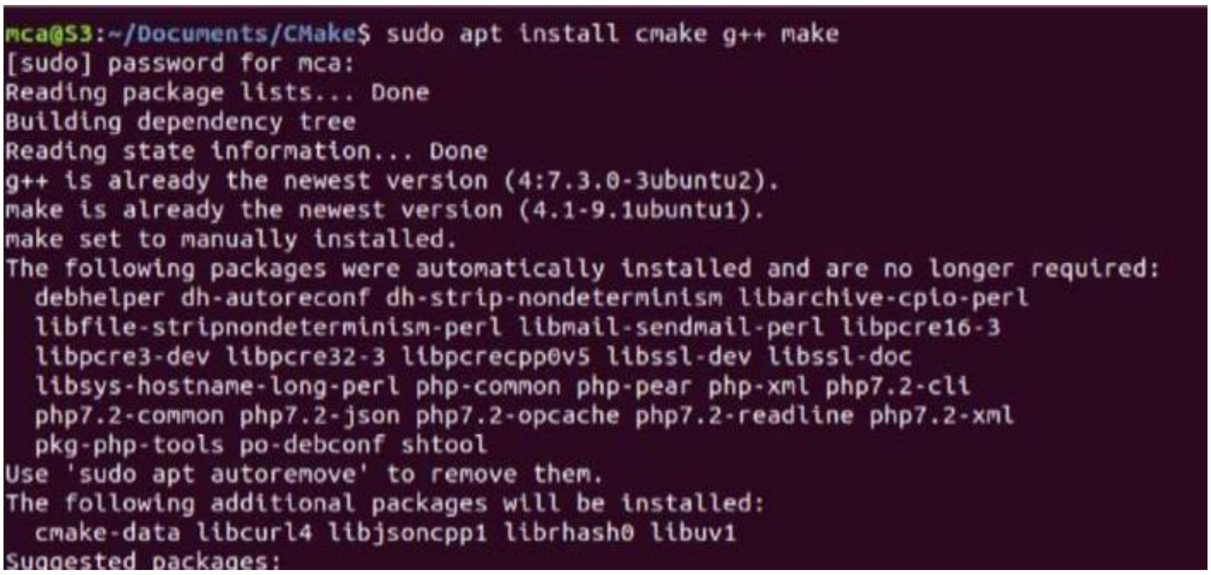
Install the cmake

Apt show cmake



```
mca@S3:~/Documents/CMake$ apt show cmake
Package: cmake
Version: 3.10.2-1ubuntu2
Priority: optional
Section: devel
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian CMake Team <pkg-cmake-team@lists.alioth.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 17.3 MB
Depends: cmake-data (= 3.10.2-1ubuntu2), procps, libarchive13 (>= 3.0.4), libc6 (>= 2.15), libcurl4 (>= 7.16.2), libexpat1 (>= 2.0.1), libgcc1
 (>= 1:3.0), libjsoncpp1 (>= 1.7.4), librhash0 (>= 1.2.6), libstdc++6 (>= 5.2), libuv1 (>= 1.4.2), zlib1g (>= 1:1.2.3.3)
Recommends: gcc, make
Suggests: cmake-doc, ninja-build
Homepage: https://cmake.org/
Supported: 5y
```

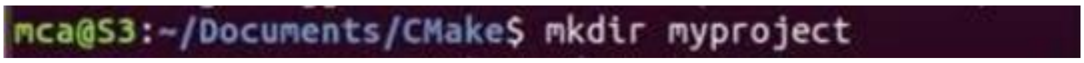
- **$sudo apt install cmake g++ make:** To install cmake , g++ and make using the apt command.



```
mca@S3:~/Documents/CMake$ sudo apt install cmake g++ make
[sudo] password for mca:
Reading package lists... Done
Building dependency tree
Reading state information... Done
g++ is already the newest version (4:7.3.0-3ubuntu2).
make is already the newest version (4.1-9.1ubuntu1).
make set to manually installed.
The following packages were automatically installed and are no longer required:
  debhelper dh-autoreconf dh-strip-nondeterminism libarchive-cpio-perl
  libfile-stripnondeterminism-perl libmail-sendmail-perl libpcre16-3
  libpcre3-dev libpcre32-3 libpcrecpp0v5 libssl-dev libssl-doc
  libsys-hostname-long-perl php-common php-pear php-xml php7.2-cli
  php7.2-common php7.2-json php7.2-opcache php7.2-readline php7.2-xml
  pkg-php-tools po-debconf shtool
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  cmake-data libcurl4 libjsoncpp1 librhash0 libuv1
Suggested packages:
```

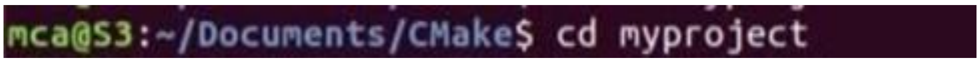
Create directory

- **mkdir cmake**: creating a different directory for our project using the mkdir and cd commands.



```
mca@S3:~/Documents/CMake$ mkdir myproject
```

- **Cd cmake**



```
mca@S3:~/Documents/CMake$ cd myproject
```

- **gedit Helloworld.cpp**

Now create a C++ source file named Hello_world.cpp and add the following **:**

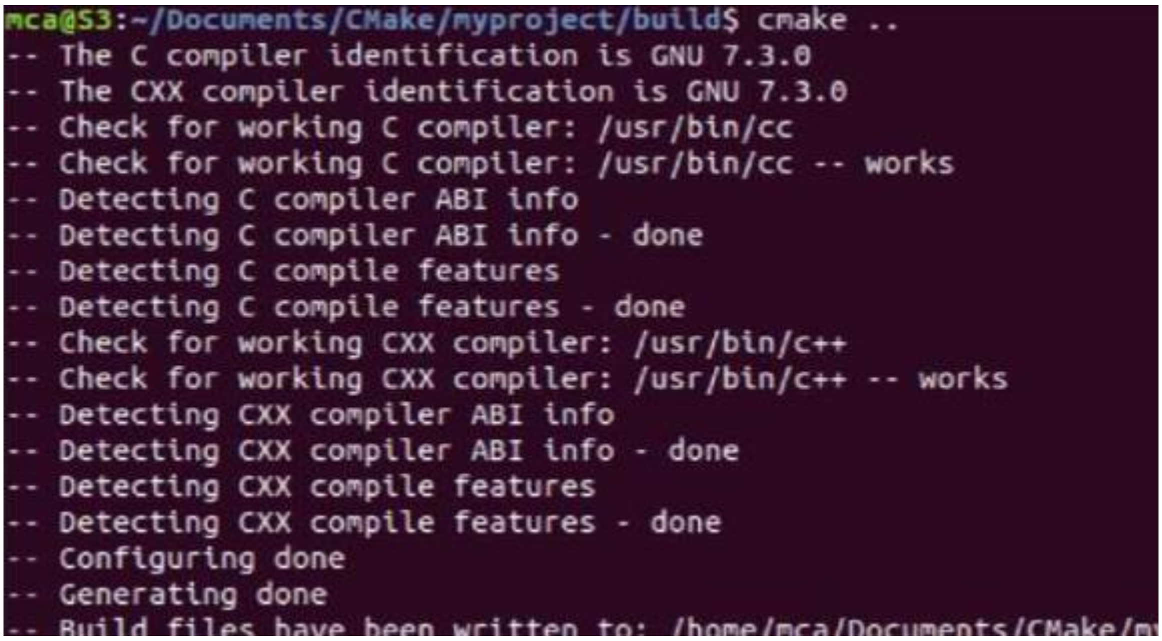
- **gedit CmakeLists.txt**

Create a CMakeLists.txt file(with this exact capitalization) which is required by CMake:

Create directory called Mkdir build:

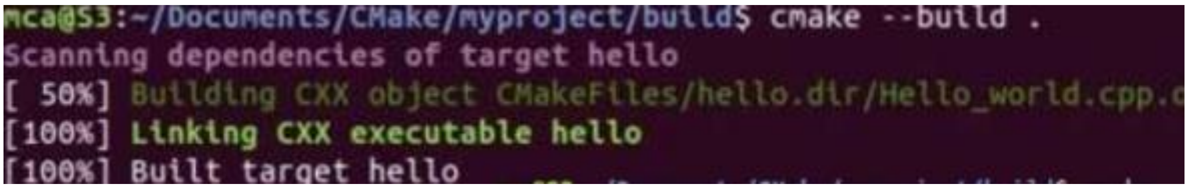
To run cmake we need to change into the build directory:

Cmake..



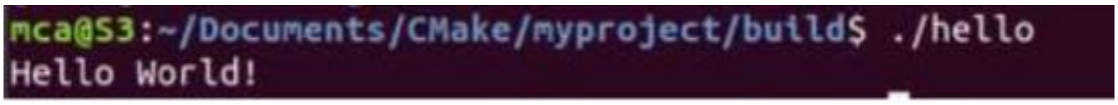
```
mca@S3:~/Documents/CMake/myproject/build$ cmake ..
-- The C compiler identification is GNU 7.3.0
-- The CXX compiler identification is GNU 7.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/mca/Documents/CMake/m
```

- Cmake –build : To generate the executable simply by typing: run hello



```
mca@S3:~/Documents/CMake/myproject/build$ cmake --build .
Scanning dependencies of target hello
[ 50%] Building CXX object CMakeFiles/hello.dir/Hello_world.cpp.
[100%] Linking CXX executable hello
[100%] Built target hello
```

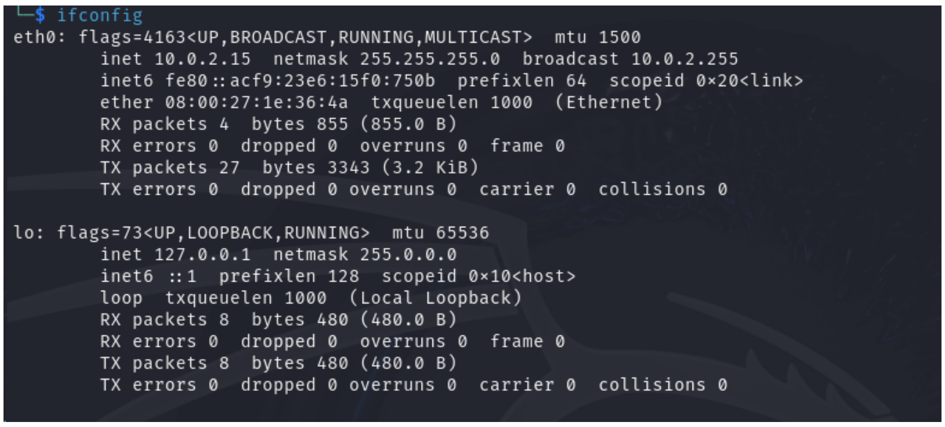
- ./hello: Run the executable by typing:



```
mca@S3:~/Documents/CMake/myproject/build$ ./hello
Hello World!
```

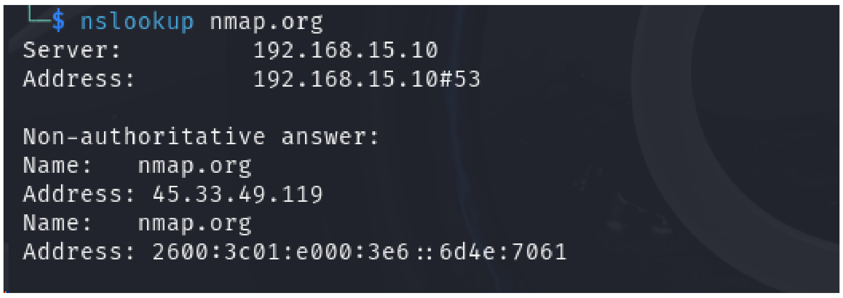
8. Introduction to command line tools for networking IPv4 networking, network commands

1. **ifconfig**: This commands in windows allows you to see a summarized information of your network such as Ip address, subnet mask, server address etc.



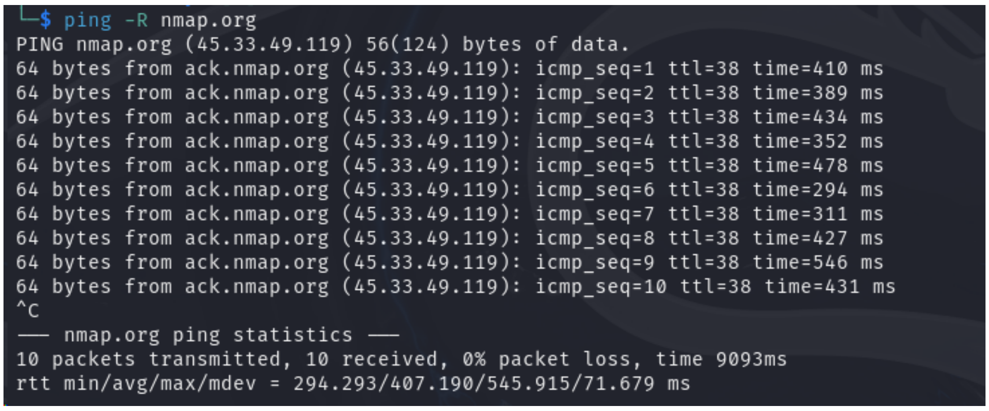
```
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::acf9:23e6:15f0:750b  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:1e:36:4a  txqueuelen 1000  (Ethernet)
        RX packets 4  bytes 855 (855.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 3343 (3.2 KiB)
        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 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. **nslookup**: To show the server to which the system is connected by default. If we want to find the Ip address of a particular domain name, we can also use nslookup



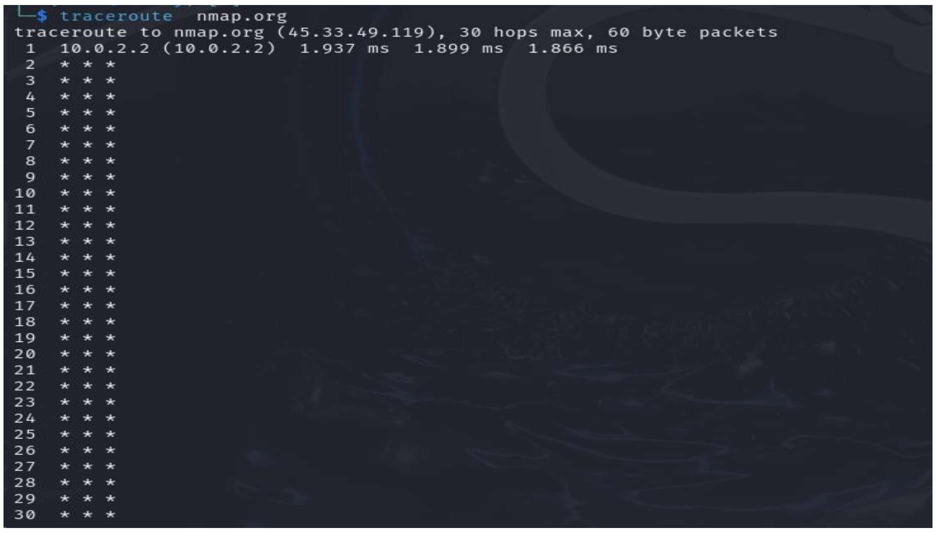
```
└─$ nslookup nmap.org
Server:         192.168.15.10
Address:        192.168.15.10#53

Non-authoritative answer:
Name:   nmap.org
Address: 45.33.49.119
Name:   nmap.org
Address: 2600:3c01:e000:3e6::6d4e:7061
```

3. **ping:** The command used to check the availability of a host. The response shows the URL you are pinging, the Ip address associated with the URL and the size of packets being sent on the first line. The next four lines shows the replies from each individual packets including the time (in milliseconds) for the response and the time to live (TLL) of the packet, that is the amount of time that must pass before the packet discarded.



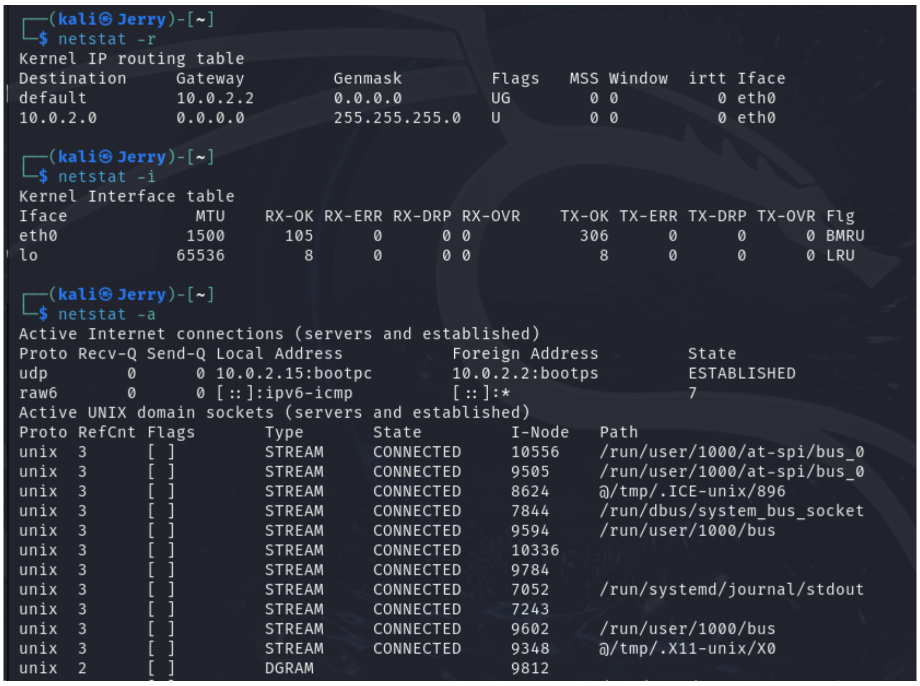
```
└─$ ping -R nmap.org
PING nmap.org (45.33.49.119) 56(124) bytes of data.
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=1 ttl=38 time=410 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=2 ttl=38 time=389 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=3 ttl=38 time=434 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=4 ttl=38 time=352 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=5 ttl=38 time=478 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=6 ttl=38 time=294 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=7 ttl=38 time=311 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=8 ttl=38 time=427 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=9 ttl=38 time=546 ms
64 bytes from ack.nmap.org (45.33.49.119): icmp_seq=10 ttl=38 time=431 ms
^C
— nmap.org ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9093ms
rtt min/avg/max/mdev = 294.293/407.190/545.915/71.679 ms
```

4. **traceroute:** traceroute is a command-line utility in Linux and other Unix-like operating systems that allows you to track the path that packets take from your computer to a destination host on a network. It is used for troubleshooting network connectivity issues and identifying network delays.



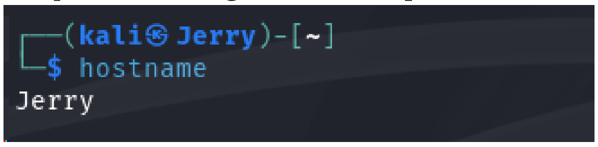
```
└─$ traceroute  nmap.org
traceroute to nmap.org (45.33.49.119), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  1.937 ms  1.899 ms  1.866 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

5. **netstat:** netstat is a command-line utility in Linux and other Unix-like operating systems that provides information about network connections, routing tables, interface statistics, masquerade connections, and more. It's used for monitoring network-related information and diagnosing network issues.



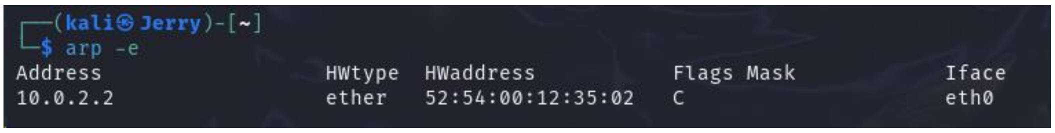
```
┌──(kali㉿Jerry)-[~]
└─$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         10.0.2.2        0.0.0.0         UG        0 0          0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0

┌──(kali㉿Jerry)-[~]
└─$ netstat -i
Kernel Interface table
Iface             MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0             1500      105      0      0 0           306      0      0      0 BMRU
lo              65536        8      0      0 0             8      0      0      0 LRU

┌──(kali㉿Jerry)-[~]
└─$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 10.0.2.15:bootpc        10.0.2.2:bootps         ESTABLISHED
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     10556    /run/user/1000/at-spi/bus_0
unix  3      [ ]         STREAM     CONNECTED     9505     /run/user/1000/at-spi/bus_0
unix  3      [ ]         STREAM     CONNECTED     8624     @/tmp/.ICE-unix/896
unix  3      [ ]         STREAM     CONNECTED     7844     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     9594     /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     10336
unix  3      [ ]         STREAM     CONNECTED     9784
unix  3      [ ]         STREAM     CONNECTED     7052     /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7243
unix  3      [ ]         STREAM     CONNECTED     9602     /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     9348     @/tmp/.X11-unix/X0
unix  2      [ ]         DGRAM                    9812
```

6. **hostname:** The hostname command is a command-line utility in Linux and other Unix-like operating systems that allow you to view or set the hostname of the system. The hostname is the unique name assigned to a computer within a network.



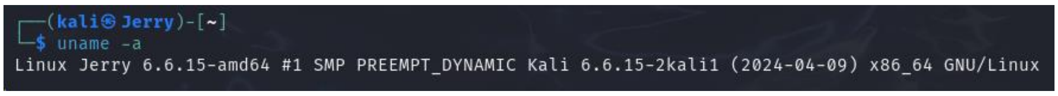
```
┌──(kali㉿Jerry)-[~]
└─$ hostname
Jerry
```

7. **arp:** The arp command is a command-line utility in Linux and other Unix-like operating systems that allows you to view and manipulate the Address Resolution Protocol (ARP) cache, which is used to map IP addresses to MAC addresses on a local network. ARP is essential for communication between devices within the same subnet.



```
┌──(kali㉿Jerry)-[~]
└─$ arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.2                 ether   52:54:00:12:35:02   C                     eth0
```

8. **uname:** The uname command is a command-line utility in Linux and other Unix-like operating systems that provides information about the system's kernel and operating system. It's used to retrieve information about the system's architecture, release version, and other details.



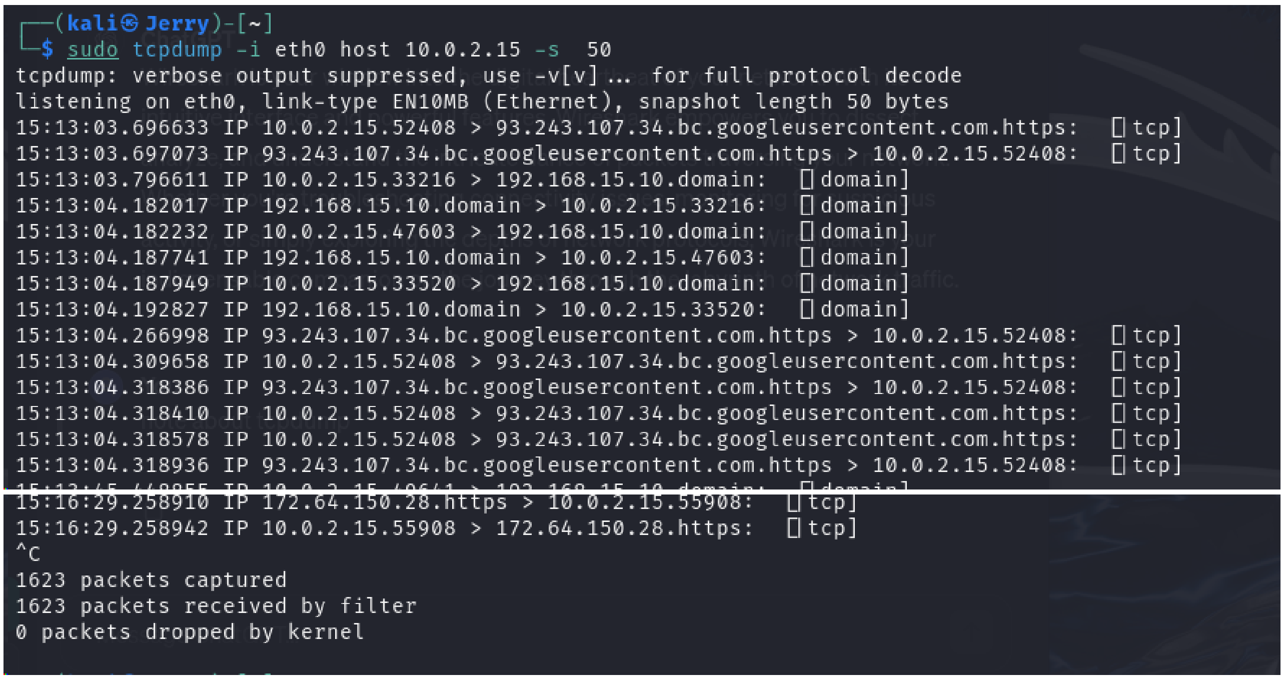
```
┌──(kali㉿Jerry)-[~]
└─$ uname -a
Linux Jerry 6.6.15-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-04-09) x86_64 GNU/Linux
```

9. Analyzing network packet stream using tcpdump and Wireshark.

Tcpdump

The tcpdump utility runs on the Linux command line. Tcpdump is a simple application that works well in Linux servers without Linux-based network devices, a GUI or various IoT nodes. These attributes give tcpdump an advantage over more powerful GUI-based analyzers, like Wireshark. Tcpdump is also scriptable, which means it can enable scheduled captures.

- **sudo tcpdump -i eth0 host 192.168.1.100**



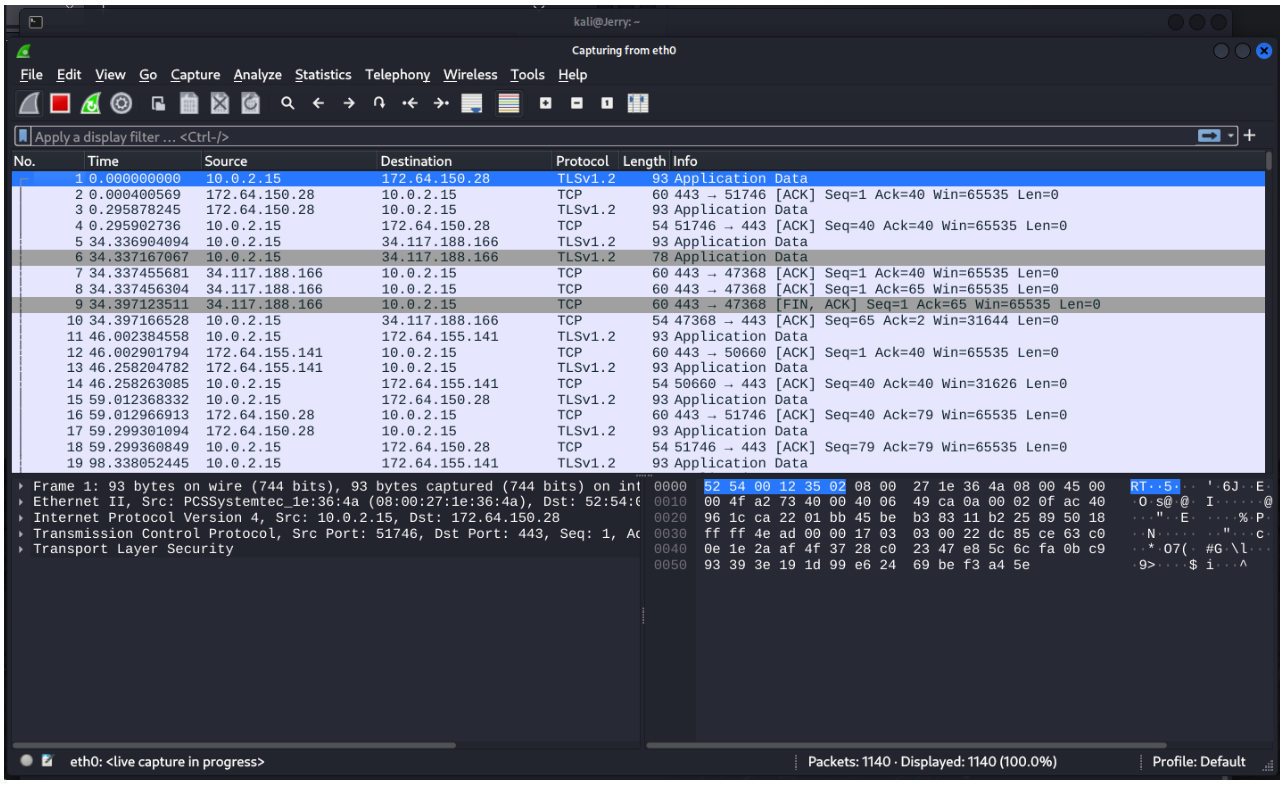
```
┌──(kali㉿Jerry)-[~]
└─$ sudo tcpdump -i eth0 host 10.0.2.15 -s  50
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 50 bytes
15:13:03.696633 IP 10.0.2.15.52408 > 93.243.107.34.bc.googleusercontent.com.https:  □tcp]
15:13:03.697073 IP 93.243.107.34.bc.googleusercontent.com.https > 10.0.2.15.52408:  □tcp]
15:13:03.796611 IP 10.0.2.15.33216 > 192.168.15.10.domain:  □domain]
15:13:04.182017 IP 192.168.15.10.domain > 10.0.2.15.33216:  □domain]
15:13:04.182232 IP 10.0.2.15.47603 > 192.168.15.10.domain:  □domain]
15:13:04.187741 IP 192.168.15.10.domain > 10.0.2.15.47603:  □domain]
15:13:04.187949 IP 10.0.2.15.33520 > 192.168.15.10.domain:  □domain]
15:13:04.192827 IP 192.168.15.10.domain > 10.0.2.15.33520:  □domain]
15:13:04.266998 IP 93.243.107.34.bc.googleusercontent.com.https > 10.0.2.15.52408:  □tcp]
15:13:04.309658 IP 10.0.2.15.52408 > 93.243.107.34.bc.googleusercontent.com.https:  □tcp]
15:13:04.318386 IP 93.243.107.34.bc.googleusercontent.com.https > 10.0.2.15.52408:  □tcp]
15:13:04.318410 IP 10.0.2.15.52408 > 93.243.107.34.bc.googleusercontent.com.https:  □tcp]
15:13:04.318578 IP 10.0.2.15.52408 > 93.243.107.34.bc.googleusercontent.com.https:  □tcp]
15:13:04.318936 IP 93.243.107.34.bc.googleusercontent.com.https > 10.0.2.15.52408:  □tcp]
15:13:45.448855 IP 10.0.2.15.49641 > 192.168.15.10.domain:  □domain]
15:16:29.258910 IP 172.64.150.28.https > 10.0.2.15.55908:  □tcp]
15:16:29.258942 IP 10.0.2.15.55908 > 172.64.150.28.https:  □tcp]
^C
1623 packets captured
1623 packets received by filter
0 packets dropped by kernel
```

Wireshark

Wireshark is a powerful network protocol analyser that allows you to capture and interactively browse the traffic running on a computer network. Here are some key points to note about Wireshark:

1. **Packet Capture**: Wireshark captures packets from a network interface in real-time or from a previously saved capture file. It supports a wide range of network protocols and can dissect and display their contents.
2. **Packet Analysis**: Wireshark provides detailed packet analysis capabilities, allowing you to inspect packet headers, payloads, and other protocol-specific information. It can decode hundreds of protocols and display them in a human-readable format.
3. **Filtering**: Wireshark offers powerful filtering capabilities, allowing you to focus on specific types of traffic based on various criteria such as IP addresses, ports, protocols, and packet contents. Filters help in efficiently analysing network traffic.
4. **Statistics**: Wireshark provides various statistics tools to analyse network traffic patterns, including endpoint statistics, protocol hierarchy, packet length distribution, and more. These statistics help in understanding network behaviour and troubleshooting network issues.
5. **Protocol Support**: Wireshark supports a vast number of network protocols, including common ones like TCP, UDP, HTTP, DNS, and more obscure or proprietary protocols. It can dissect and decode protocol data units (PDUs) for a comprehensive analysis.
6. **Graphical Interface**: Wireshark features a user-friendly graphical interface that allows you to interactively explore captured packets. It provides various views, including packet list, packet details, and packet bytes, along with customizable colorization and layout options.
7. **Cross-Platform**: Wireshark is cross-platform and runs on Windows, macOS, and Linux. It offers consistent features and capabilities across different operating systems, making it a versatile tool for network analysis.
8. **Community Support**: Wireshark has a large and active user community that contributes to its development and provides support through forums, mailing lists, and documentation. The community also maintains a repository of capture files for learning and reference purposes.

Overall, Wireshark is an essential tool for network administrators, security professionals, and anyone involved in network troubleshooting, analysis, or monitoring. Its comprehensive features, protocol support, and user-friendly interface make it an asset for understanding and debugging network communications.


kali@Jerry: ~
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time          Source          Destination     Protocol Length Info
   1 0.000000000   10.0.2.15       172.64.150.28   TLSv1.2  93 Application Data
   2 0.000400569   172.64.150.28   10.0.2.15       TCP      60 443 → 51746 [ACK] Seq=1 Ack=40 Win=65535 Len=0
   3 0.295878245   172.64.150.28   10.0.2.15       TLSv1.2  93 Application Data
   4 0.295902736   10.0.2.15       172.64.150.28   TCP      54 51746 → 443 [ACK] Seq=40 Ack=40 Win=65535 Len=0
   5 34.336904094  10.0.2.15       34.117.188.166  TLSv1.2  93 Application Data
   6 34.337167067  10.0.2.15       34.117.188.166  TLSv1.2  78 Application Data
   7 34.337455681  34.117.188.166  10.0.2.15       TCP      60 443 → 47368 [ACK] Seq=1 Ack=40 Win=65535 Len=0
   8 34.337456304  34.117.188.166  10.0.2.15       TCP      60 443 → 47368 [ACK] Seq=1 Ack=65 Win=65535 Len=0
   9 34.397123511  34.117.188.166  10.0.2.15       TCP      60 443 → 47368 [FIN, ACK] Seq=1 Ack=65 Win=65535 Len=0
  10 34.397166528  10.0.2.15       34.117.188.166  TCP      54 47368 → 443 [ACK] Seq=65 Ack=2 Win=31644 Len=0
  11 46.002384558  10.0.2.15       172.64.155.141  TLSv1.2  93 Application Data
  12 46.002901794  172.64.155.141  10.0.2.15       TCP      60 443 → 50660 [ACK] Seq=1 Ack=40 Win=65535 Len=0
  13 46.258204782  172.64.155.141  10.0.2.15       TLSv1.2  93 Application Data
  14 46.258263085  10.0.2.15       172.64.155.141  TCP      54 50660 → 443 [ACK] Seq=40 Ack=40 Win=31626 Len=0
  15 59.012368332  10.0.2.15       172.64.150.28   TLSv1.2  93 Application Data
  16 59.012966913  172.64.150.28   10.0.2.15       TCP      60 443 → 51746 [ACK] Seq=40 Ack=79 Win=65535 Len=0
  17 59.299301094  172.64.150.28   10.0.2.15       TLSv1.2  93 Application Data
  18 59.299360849  10.0.2.15       172.64.150.28   TCP      54 51746 → 443 [ACK] Seq=79 Ack=79 Win=65535 Len=0
  19 98.338052445  10.0.2.15       172.64.155.141  TLSv1.2  93 Application Data

› Frame 1: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on int
› Ethernet II, Src: PCSSystemtec_1e:36:4a (08:00:27:1e:36:4a), Dst: 52:54:(
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 172.64.150.28
› Transmission Control Protocol, Src Port: 51746, Dst Port: 443, Seq: 1, Ac
› Transport Layer Security

0000  52 54 00 12 35 02 08 00  27 1e 36 4a 08 00 45 00   RT··5···'·6J··E·
0010  00 4f a2 73 40 00 40 06  49 ca 0a 00 02 0f ac 40   ·O·s@·@·I······@
0020  96 1c ca 22 01 bb 45 be  b3 83 11 b2 25 89 50 18   ···"··E·····%·P·
0030  ff ff 4e ad 00 00 17 03  03 00 22 dc 85 ce 63 c0   ··N·······"···c·
0040  0e 1e 2a af 4f 37 28 c0  23 47 e8 5c 6c fa 0b c9   ··*·O7(·#G·\l···
0050  93 39 3e 19 1d 99 e6 24  69 be f3 a4 5e            ·9>····$i···^

eth0: <live capture in progress>    Packets: 1140 · Displayed: 1140 (100.0%)    Profile: Default
```

10.Introduction to Hypervisors and VMs, Xen or KVM Introduction to Containers: Docker, installation, and deployment.

Procedure

For the Ubuntu system, all packages required to run KVM are available on official upstream repositories.

Install them using the commands:

- sudo apt update
- sudo apt-get install qemu qemu-kvm libvirt-bin bridge-utils virt-manager virtviewer–y

Create Virtual Machine

- You can create virtual machine using virt-manager utility.

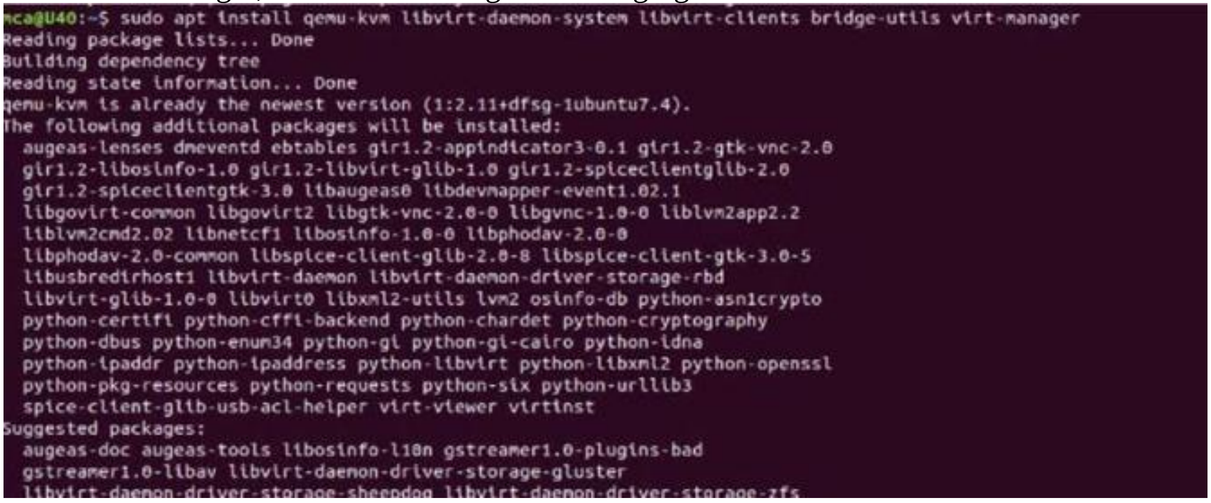
Run the following command to start the virt-manager:

sudo virt-manager

- virsh help
- virsh help
- virsh help list
- Sudo virsh nodeinfo

- Virsh start
- vm virsh start
- virsh start testvm1

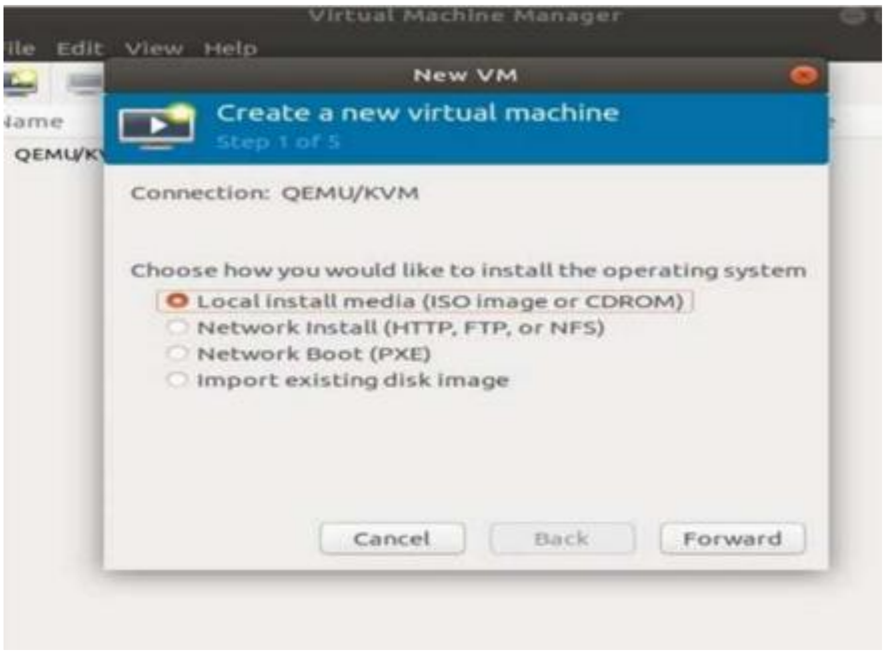
Step 1: Update the repositories
Step 2: Install essential KVM packages
Install virt-manager, a tool for creating and managing VMs



```
mca@U40:~$ sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
qemu-kvm is already the newest version (1:2.11+dfsg-1ubuntu7.4).
The following additional packages will be installed:
  augeas-lenses dmeventd ebtables gir1.2-appindicator3-0.1 gir1.2-gtk-vnc-2.0
  gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0 gir1.2-spiceclientglib-2.0
  gir1.2-spiceclientgtk-3.0 libaugeas0 libdevmapper-event1.02.1
  libgovirt-common libgovirt2 libgtk-vnc-2.0-0 libgvnc-1.0-0 liblvm2app2.2
  liblvm2cmd2.02 libnetcf1 libosinfo-1.0-0 libphodav-2.0-0
  libphodav-2.0-common libspice-client-glib-2.0-8 libspice-client-gtk-3.0-5
  libusbredirhost1 libvirt-daemon libvirt-daemon-driver-storage-rbd
  libvirt-glib-1.0-0 libvirt0 libxml2-utils lvm2 osinfo-db python-asn1crypto
  python-certifi python-cffi-backend python-chardet python-cryptography
  python-dbus python-enum34 python-gi python-gi-cairo python-idna
  python-ipaddr python-ipaddress python-libvirt python-libxml2 python-openssl
  python-pkg-resources python-requests python-six python-urllib3
  spice-client-glib-usb-acl-helper virt-viewer virtinst
Suggested packages:
  augeas-doc augeas-tools libosinfo-l10n gstreamer1.0-plugins-bad
  gstreamer1.0-libav libvirt-daemon-driver-storage-gluster
  libvirt-daemon-driver-storage-sheepdog libvirt-daemon-driver-storage-zfs
```

Step 3: Start virt-manager with

Step 4: In the first window, click the computer icon in the upper-left corner, In the dialogue box that opens, select the option to install the VM using an ISO image. Then click Forward.

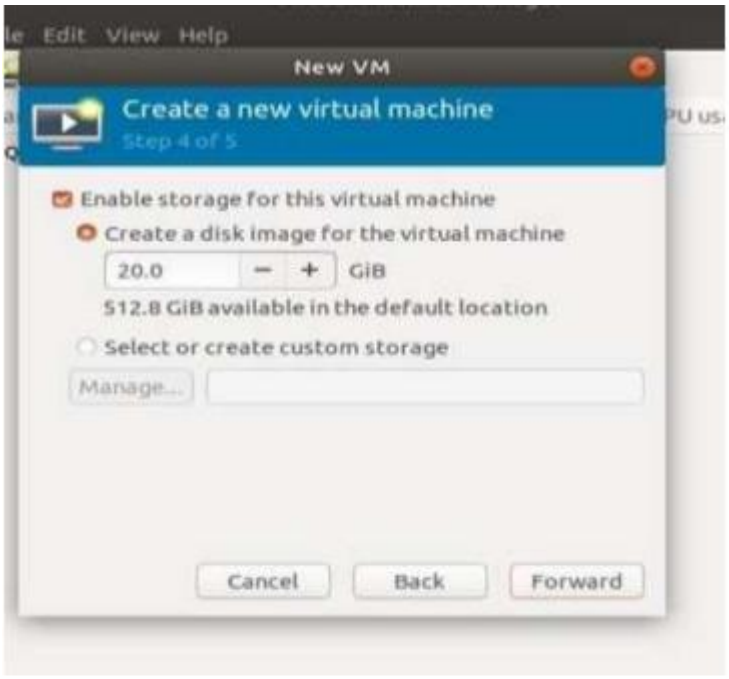


Virtual Machine Manager
ile Edit View Help
New VM
Create a new virtual machine
Step 1 of 5
Connection: QEMU/KVM
Choose how you would like to install the operating system
[✓ Local install media (ISO image or CDROM)]
Network Install (HTTP, FTP, or NFS)
Network Boot (PXE)
Import existing disk image

Step 5: Choose ISO, click Forward

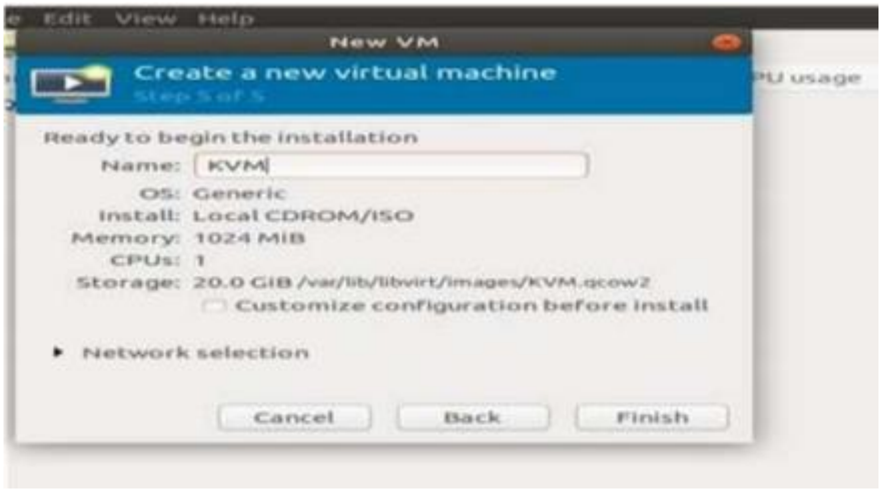
Step 6: Enter the amount of RAM and the number of CPUs you wish to allocate to the VM and proceed to the next step.

Step 7: Allocate hard disk space to the VM. Click Forward to go to the last step.



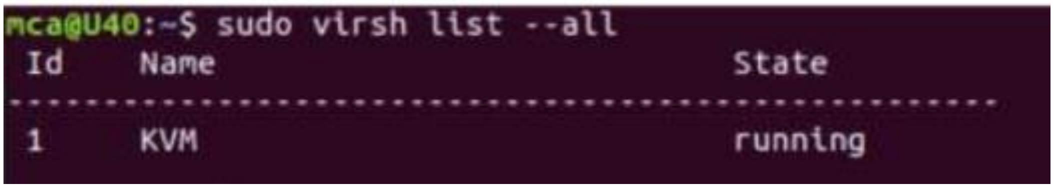
Enable storage for this virtual machine
Create a disk image for the virtual machine
20.0 GiB
512.8 GiB available in the default location
Select or create custom storage

Step 8: Specify the name for your VM and click Finish to complete the setup.



Ready to begin the installation
Name: KVM
OS: Generic
Install: Local CDROM/ISO
Memory: 1024 MiB
CPUs: 1
Storage: 20.0 GiB /var/lib/libvirt/images/KVM.qcow2
Customize configuration before install

Step 9: Select language
Step 10: The VM starts automatically, prompting you to start installing the OS that is on the ISO file.
Step 11: Check the state of KVM



```
mca@U40:~$ sudo virsh list --all
 Id    Name                           State
----------------------------------------------------
 1     KVM                            running
```

Introduction to Containers: Docker installation and deployment

Procedure

Steps for Installing Docker:

Step 1: Open the terminal on Ubuntu.
Step 2: Remove any Docker files that are running in the system, using the following command

- Command: $ sudo apt-get remove docker docker-engine docker.io

After entering the above command, you will need to enter the password of the root and press enter.

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

- Command: $ sudo apt-get update

Step 4: Install Docker using the following command:

- Command: $ sudo apt install docker.io

You will then get a prompt asking you to choose between y/n – choose 'y'

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

- Command: $ sudo snap install docker

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

- Command: $ docker –version

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

- Command: $ sudo docker run hello-world

Here, hello-world is the docker image present on the Docker hub.

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

- Command: $ sudo docker images

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

- Command: $ sudo docker ps -a

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

- Command: $ sudo docker ps

11.Installing and configuring modern frameworks like Laravel

Step 1: Prerequisites

- Install Required Software: Make sure you have a web server (e.g., Apache or Nginx), PHP, Composer (dependency manager), and a database server (e.g., MySQL) installed on your system.
- **Install Composer:** Download and install Composer by following the instructions on

the official Composer website.

Step 2: Install Laravel

1. Create a New Laravel Project: Open a terminal and navigate to the directory where you want to create your Laravel project. Run the following command:

- composer create-project --prefer-dist laravel/laravel myproject

This will create a new Laravel project named "myproject."

Step 3: Configure the Web Server

1. Apache:

- Create a new virtual host configuration for your Laravel project in your Apache configuration.
- Set the Document Root to the public directory of your Laravel project.
- Enable the necessary Apache modules (e.g., rewrite) and restart Apache.

2. Nginx:

- Create a new server block configuration for your Laravel project in your Nginx configuration.
- Set the root directive to the public directory of your Laravel project.
- Configure the necessary location directives and restart Nginx.

Step 4: Configure Laravel

1. Environment Configuration:

- Rename the. env.example file in your Laravel project root to. env.
- Set database connection details, application key, and other settings in the .env file.

2. Generate Application Key: Run the following command in your Laravel project directory:

- php artisan key: generate

3. Run Migrations: If your .env file is configured with database details, run migrations

to create necessary database tables:

- php artisan migrate

Step 5: Testing the Setup

1. **Access the Application**: Open a web browser and visit the URL you configured for your Laravel project. You should see the Laravel welcome page.
2. **Create Routes and Views:** Begin building your application by defining routes and creating views in the resources/views directory.

12.Ansible play book deployment

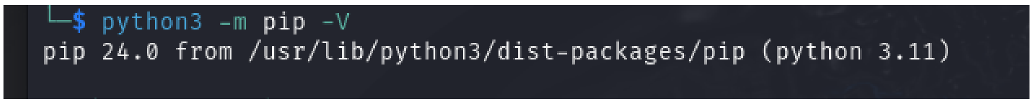
Ansible playbooks are the cornerstones of automation in Ansible. They define a set of instructions (tasks) that Ansible executes on remote systems (managed nodes) to achieve a desired configuration state.

Here are some key points about Ansible playbook deployment:

- **Reusability and Orchestration**: Playbooks are reusable, allowing you to deploy complex applications or configurations across multiple machines with consistency. They can orchestrate the deployment process, running specific tasks on different groups of hosts based on their roles (e.g., web servers, database servers).
- **Declarative Nature**: Playbooks are declarative, meaning they specify the desired end state rather than the specific steps to achieve it. This makes them easier to understand and maintain. Ansible figures out the most efficient way to reach the desired state.
- **Inventory Management**: Playbooks rely on an inventory file that defines the managed nodes and groups of nodes. This allows you to target specific groups for deployment tasks.
- **Roles**: Playbooks can leverage roles, which are reusable modules containing related tasks for specific functionalities. This promotes modularity and simplifies playbook structure.

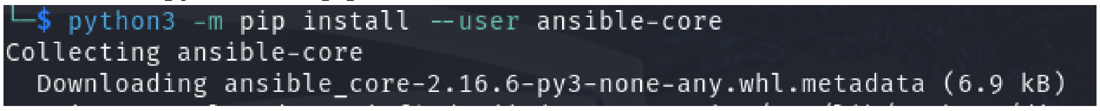
Installation

Prerequisites: pip & python 3
Check version: python3 -m pip -V

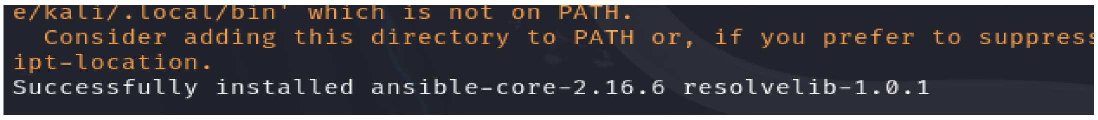


```
└─$ python3 -m pip -V
pip 24.0 from /usr/lib/python3/dist-packages/pip (python 3.11)
```

Installation: python3 -m pip install --user ansible-core

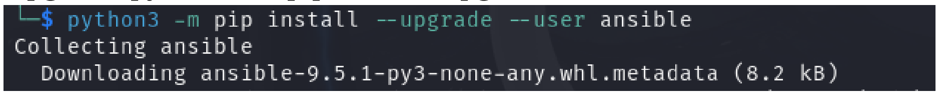


```
└─$ python3 -m pip install --user ansible-core
Collecting ansible-core
  Downloading ansible_core-2.16.6-py3-none-any.whl.metadata (6.9 kB)
```

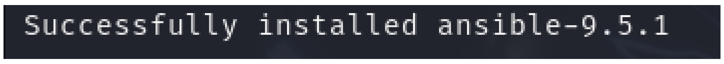


```
e/kali/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress
ipt-location.
Successfully installed ansible-core-2.16.6 resolvelib-1.0.1
```

Upgrade: python3 -m pip install --upgrade --user ansible



```
└─$ python3 -m pip install --upgrade --user ansible
Collecting ansible
  Downloading ansible-9.5.1-py3-none-any.whl.metadata (8.2 kB)
```



```
Successfully installed ansible-9.5.1
```