Linux - Crash Course

By : Bilal Mazhar

Outline

1. Introduction to Linux
2. Linux Installation
3. Directory Structure
4. Basic Commands
5. Administration
6. Linux Projects

Before we start

Some myths about Linux !

- Linux is not user friendly !
- Linux is hard to learn !
- Linux is not for home user !
- Linux is command line only !
- Linux is for Veterans only !

Introduction to Linux

Just like Windows, iOS, and Mac OS, Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system.

An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn't function.

Open source

Linux is also distributed under an open source license. Open source follows these key tenets:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and change it to make it do what you wish.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to distribute copies of your modified versions to others.

Why Linux ?

- Open Source
- Community support
- Customization
- Most of the server are using Linux
- DevOps Most of the tools are available for linux
- Automation
- Secure

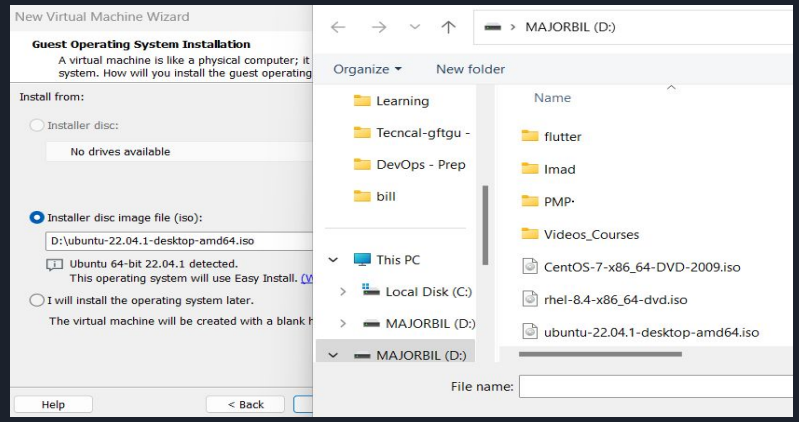
Popular Linux distributions include:

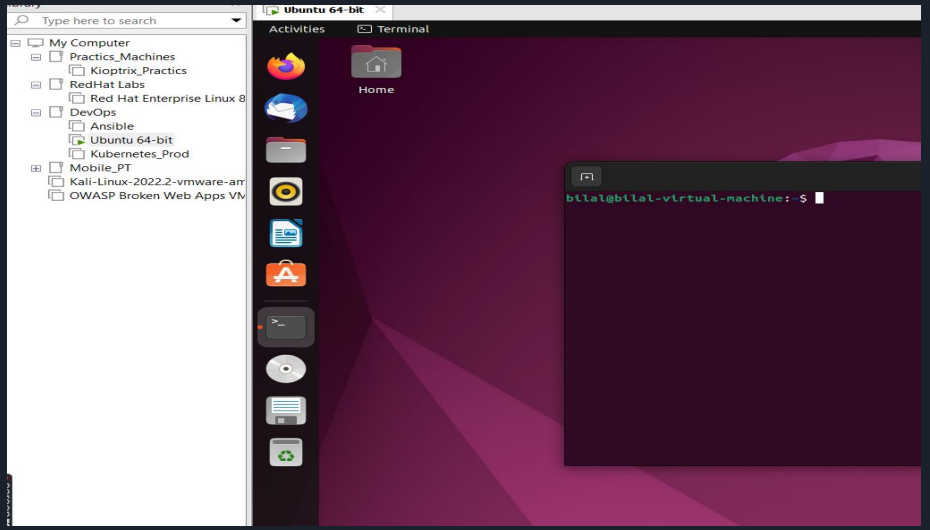
- LINUX MINT
- MANJARO
- DEBIAN
- UBUNTU
- ANTERGOS
- SOLUS
- FEDORA
- ELEMENTARY OS
- OPENSUSE
- Red Hat Enterprise Linux
- Ubuntu Server
- Centos
- SUSE Enterprise Linux

Most used Linux distros currently in IT industry

- RPM Based : - RHEL & Centos
- Debian Based : - Ubuntu Server

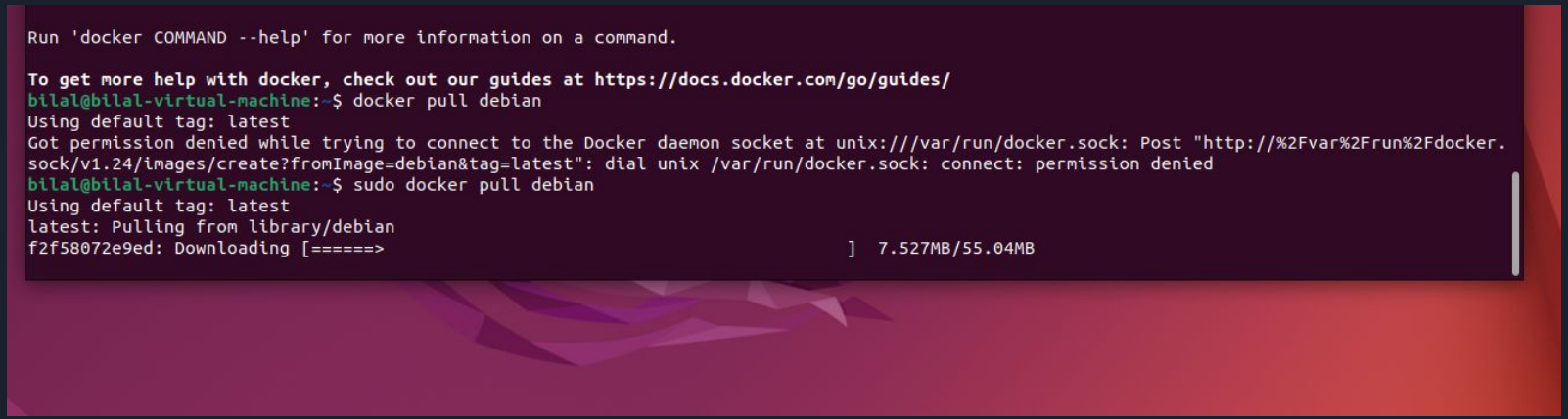
Installation Using VM





Installation Using Docker

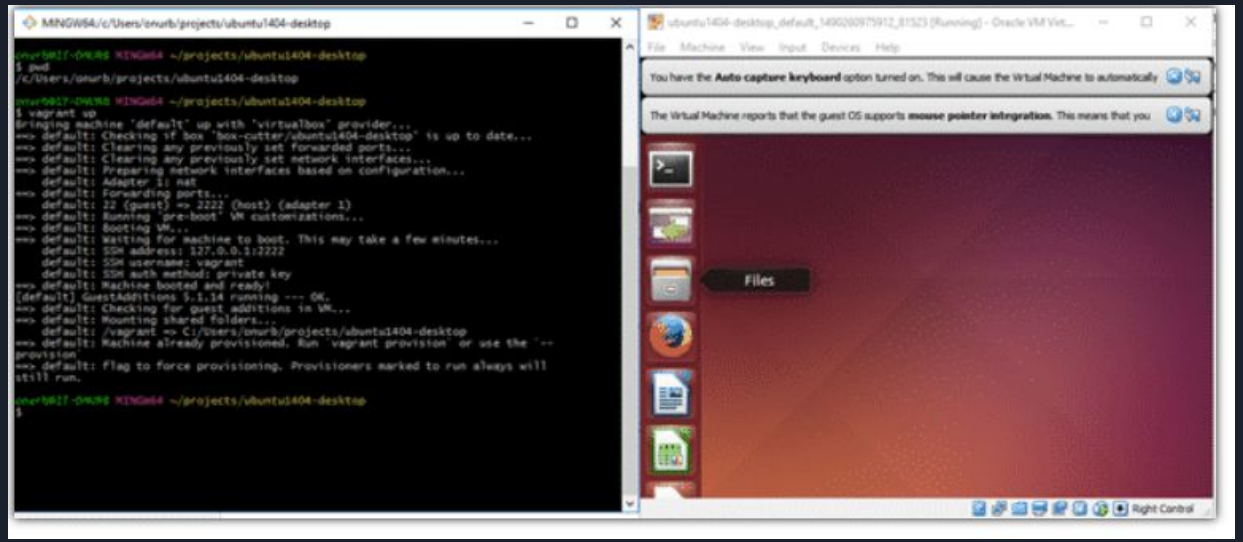
1. Go to https://hub.docker.com/_/debian
2. Search “ **Linux** “ ,
3. Install docker in your linux ,
4. Type command “ **docker pull debian** “ .



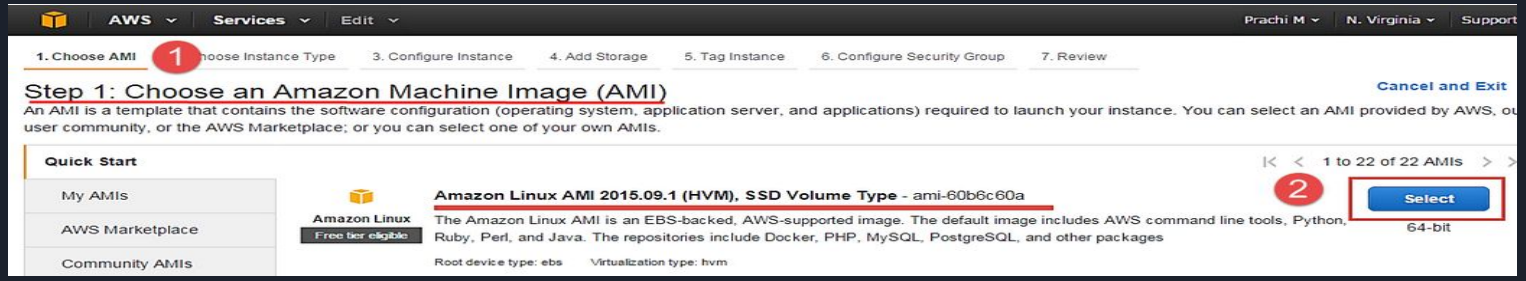
Installation Using Vagrant

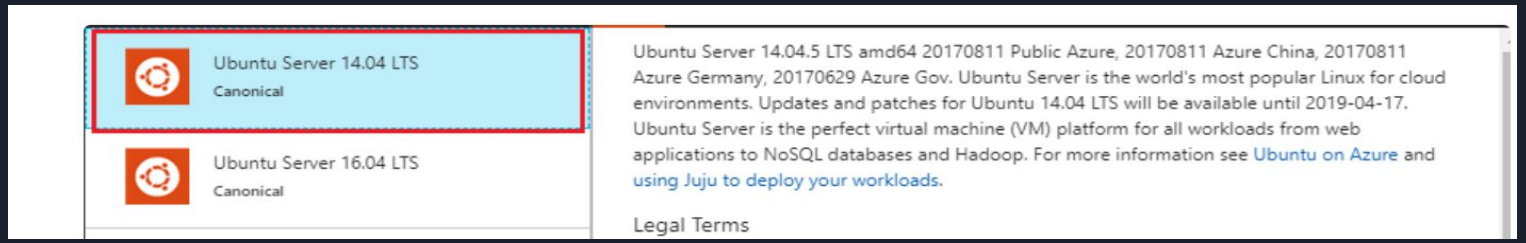
1. Create two directory “ **Project** “ , “ **Ubuntu**” in your linux environment ,
2. Navigate to the directory **Project /Ubuntu,**
3. **Type these Commands**

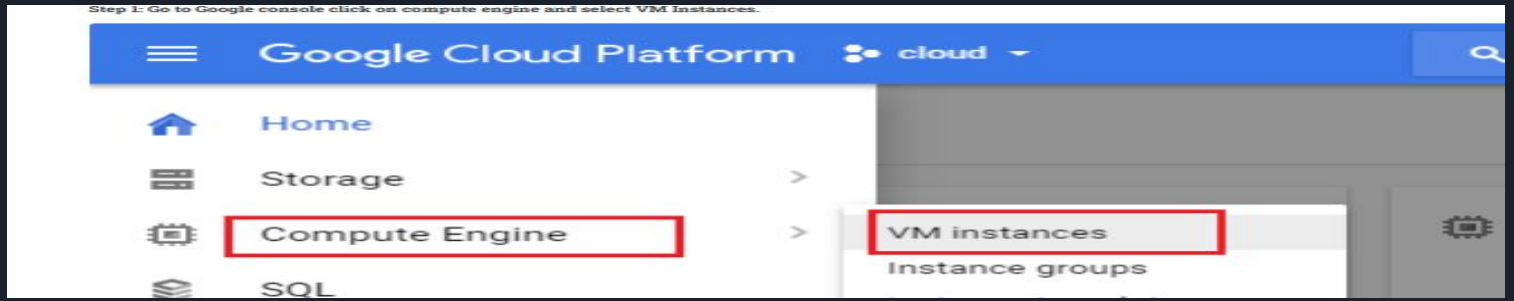
- `vagrant init box-cutter/ubuntu1404-desktop`
- `vagrant up –provider virtualbox`
- `vagrant up`



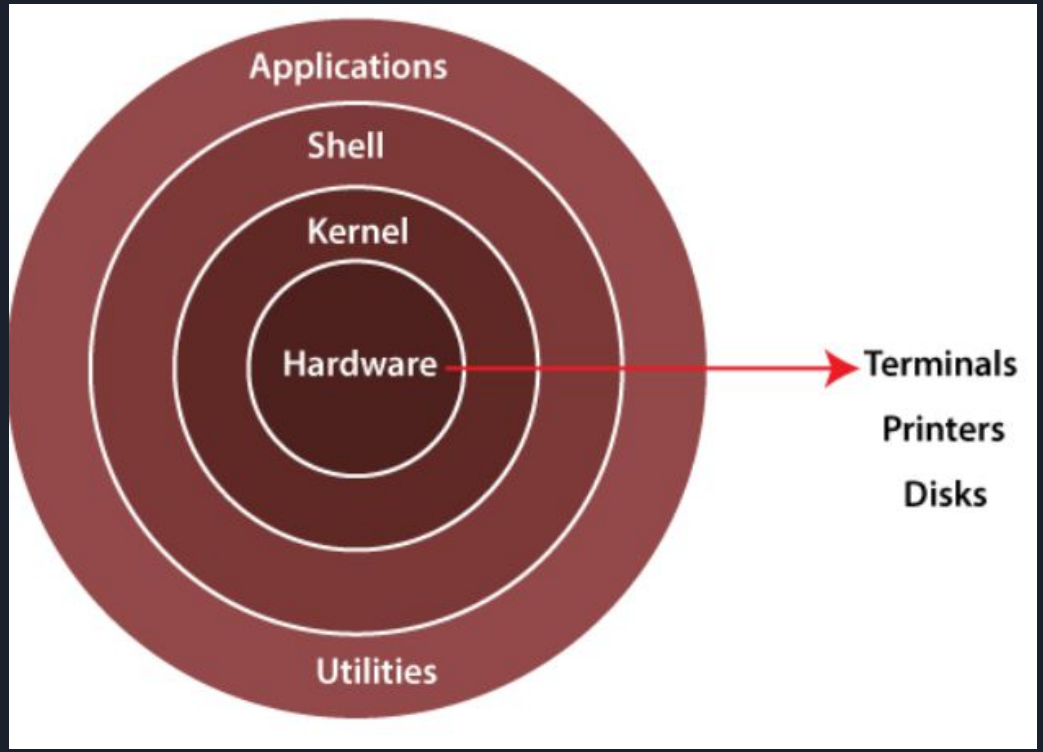
Installation in Cloud [AWS , Azure , GCP]







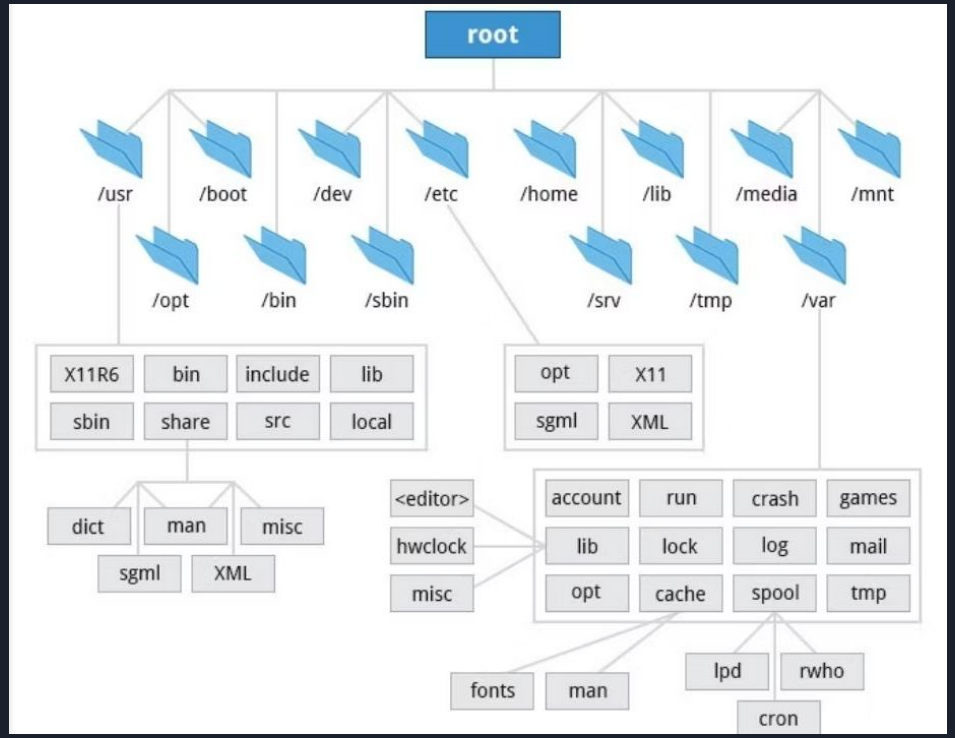
Architecture of Linux system



Architecture of Linux system

Component	***Description***
Kernel	The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS.
System Libraries	These libraries can be specified as some special functions. These are applied for implementing the operating system's functionality
System Utility Programs	It is responsible for doing specialized level and individual activities.
Hardware layer	Linux operating system contains a hardware layer that consists of several peripheral devices like 1. 1. CPU 2. HDD 3. RAM
Shell	It is an interface among the kernel and user

Directory Structure



Some Important Directories

- Home Directories: `/root`,`/home/`*`username`*
- User Executable: `/bin`, `/usr/bin`, `/usr/local/bin`
- System Executables: `/sbin`, `/usr/sbin`, `/usr/local/sbin`
- Other Mountpoints: `/media`, `/mnt`
- Configuration: `/etc`
- Temporary Files: `/tmp`
- Kernels and Bootloader: `/boot`
- Server Data: `/var`, `/srv`
- System Information: `/proc`, `/sys`
- Shared Libraries: `/lib`, `/usr/lib`, `/usr/local/lib`

Difference between Absolute and relative path

→ What is a path?

A path is a unique location to a file or a folder in a file system of an OS. A path to a file is

a combination of / and alphanumeric characters

→ is an absolute path?

An absolute path is defined as the specifying the location of a file or directory from the root

directory(/). In other words we can say absolute path is a complete path from start of actual

filesystem from / directory

→ Some examples of absolute path:

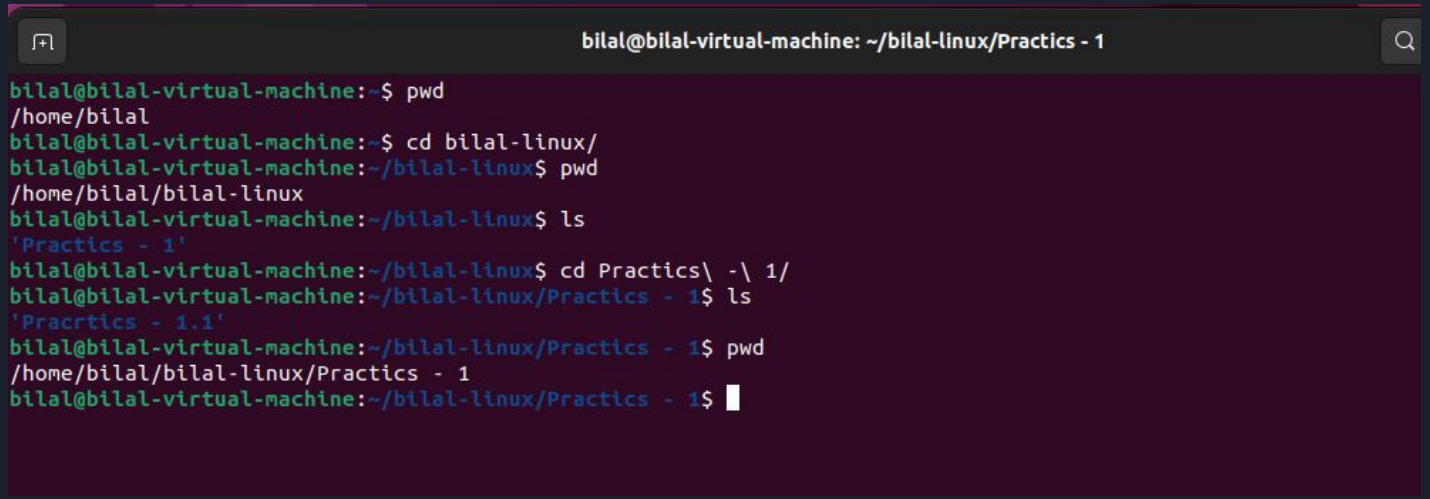
- /home/imran/linux-practices/
- /var/ftp/pub
- /etc/samba.smb.conf
- /boot/grub/grub.conf

If you see all these paths started from / directory which is a root directory for every Linux/Unix machines.

Difference between Absolute and relative path

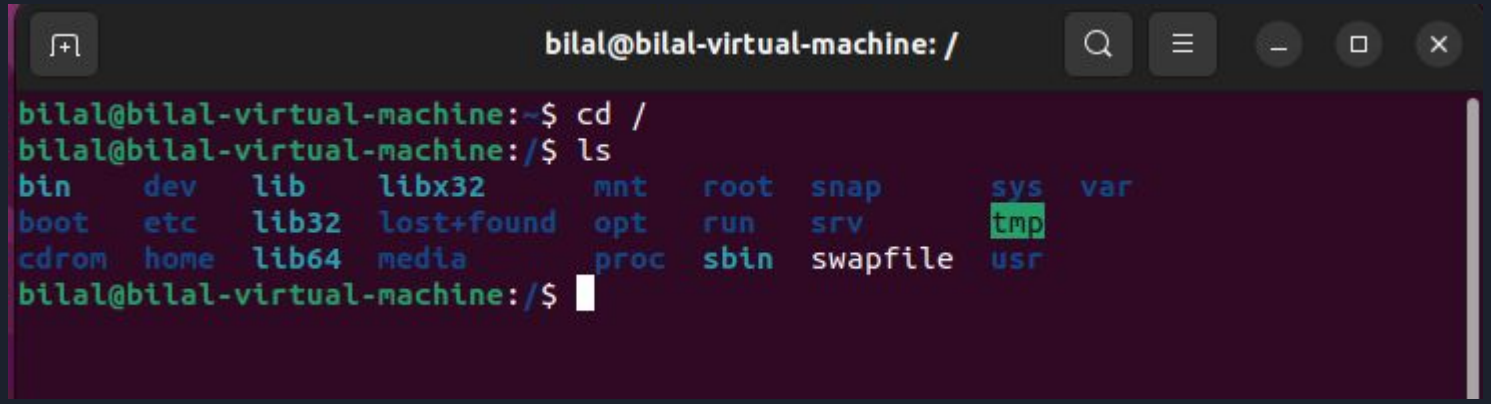
What is the relative path?

Relative path is defined as path related to the present working directory(pwd). Suppose I am located in /home/imran and I want to change directory to /home/imran/linux-practices. I can use relative path concept to change directory to linux-practices and also devops dir directory.



Root Directory (/)

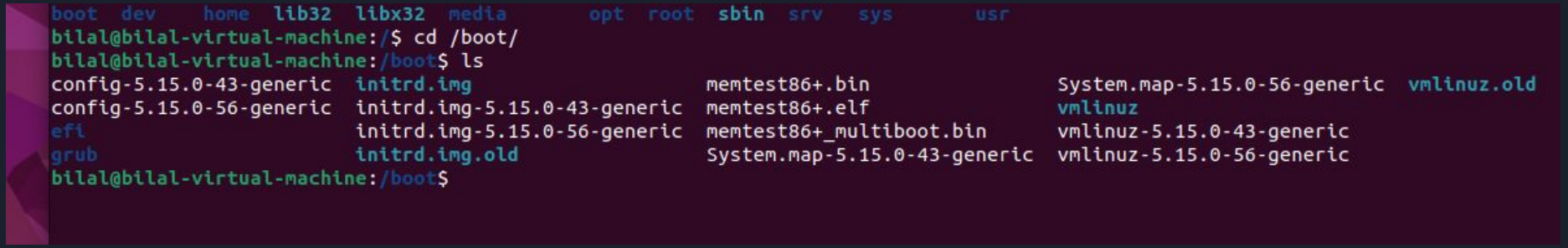
The root directory, represented by a forward slash (/), stores all the directories in Linux. If you cd into this directory and use the ls command, you will find a list of all the directories on your machine. The absolute path of every file passes through the root directory as it is the parent to all other directories.



As you can see, the “/$” sign indicates that you’re in the root directory now.

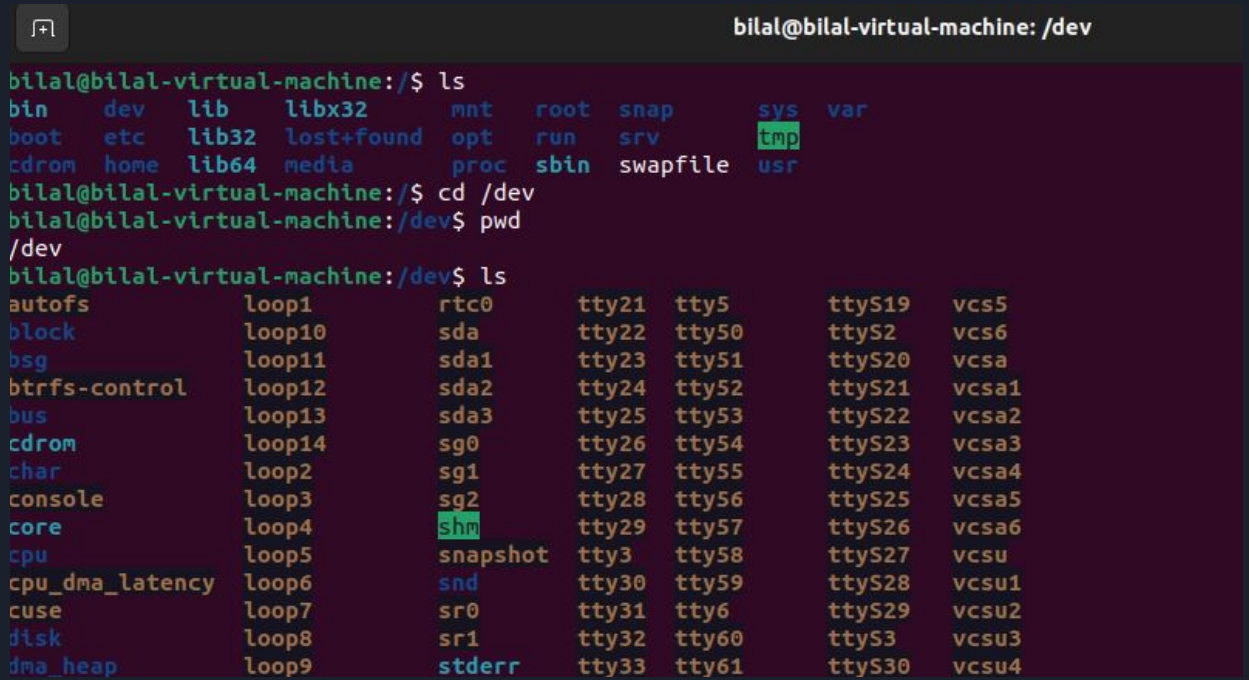
/boot

The boot directory contains important files needed by the boot loader. The initial ram file system or initramfs is also stored here along with the kernel. You shouldn't tinker with this directory on your primary machine. If you wish to experiment, feel free to do so in a virtual machine.



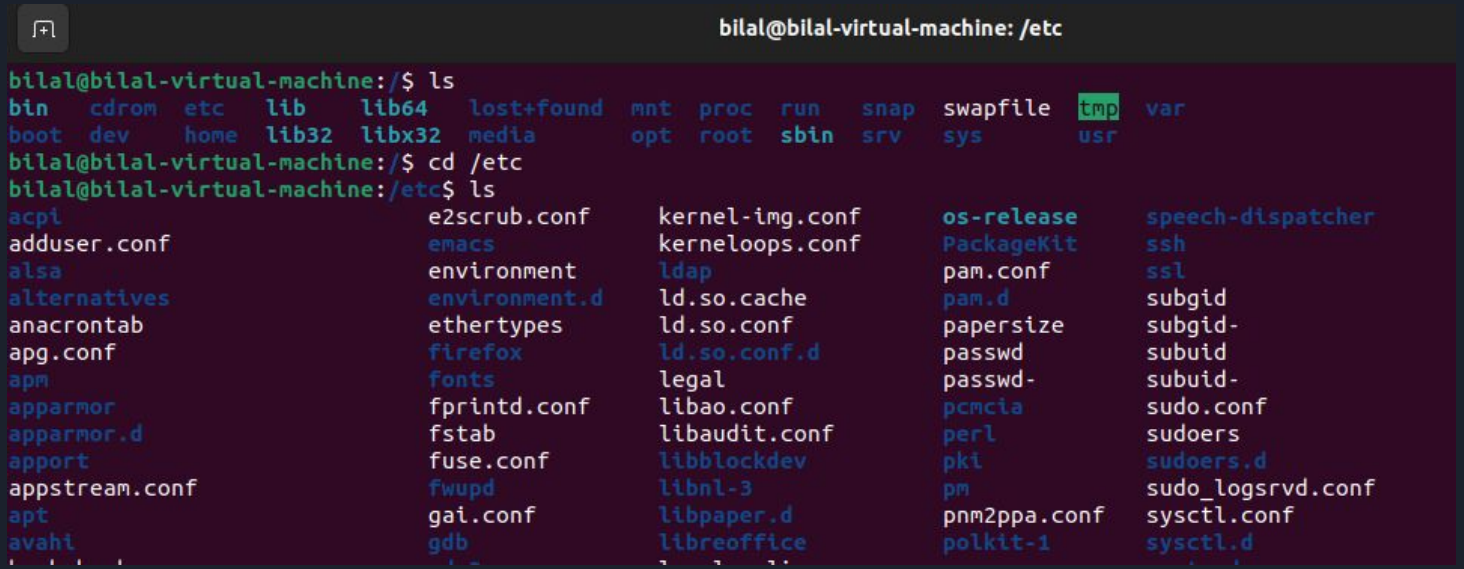
/dev

Linux treats everything as a file, and hardware is not an exception to this rule. The /dev directory contains special, virtual files representing hardware components like a mouse, keyboard, storage devices, etc., connected to your system.



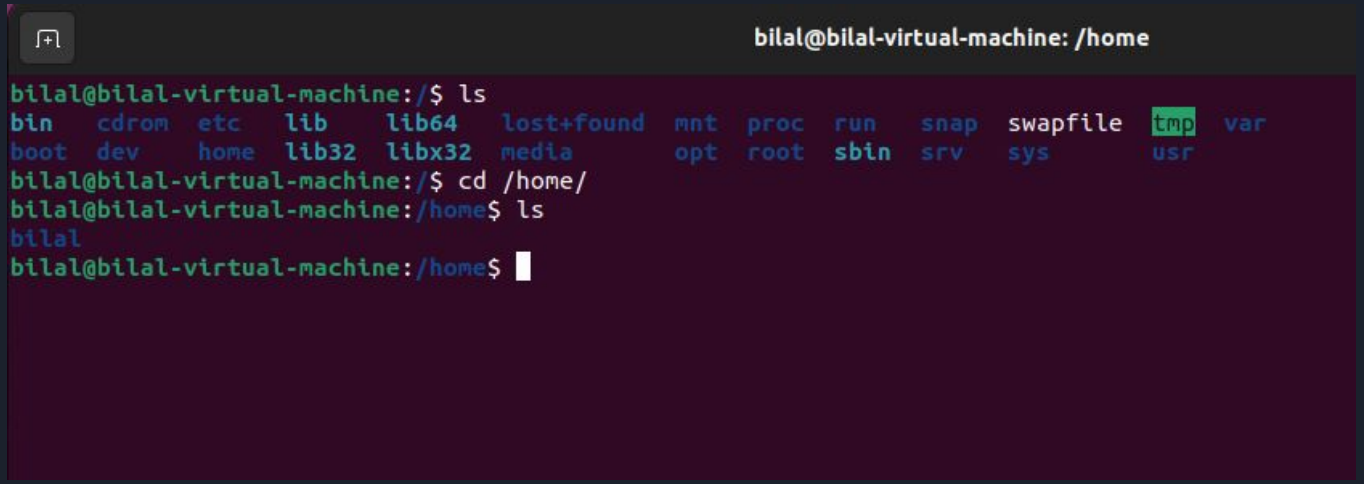
/etc

The /etc directory contains vital system configuration files such as startup scripts, networking files, user account-related files, etc. You have to edit configuration files in the /etc directory to make any system-wide changes.



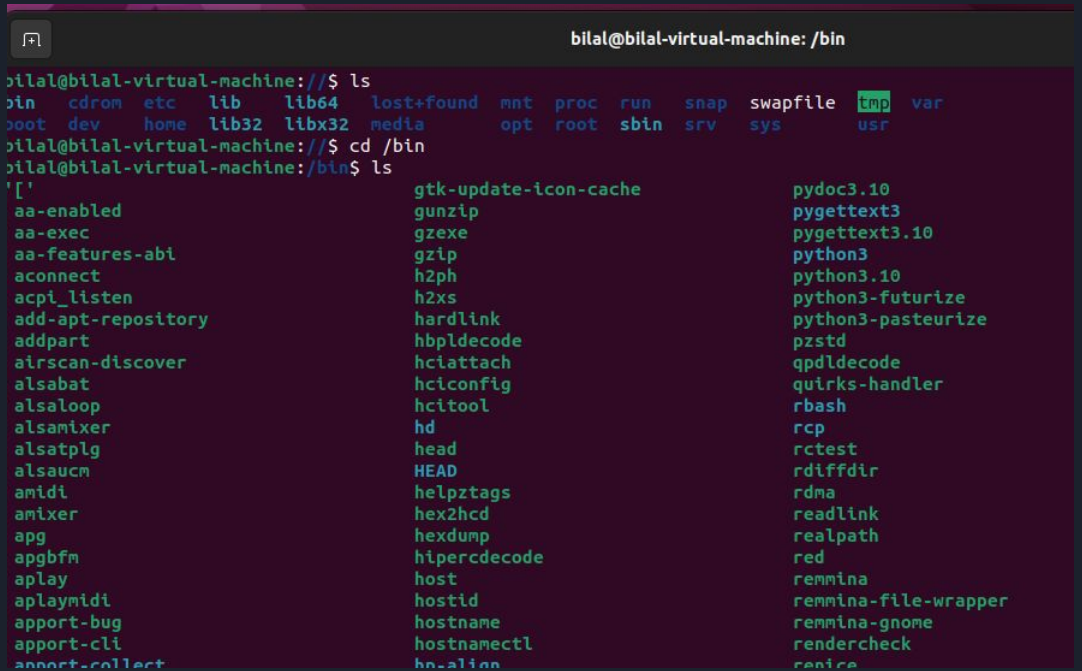
/home

The /home directory stores an individual user's home directory. If you're switching from a Windows environment, you will find the /home directory akin to the C:/Users directory. It contains user-specific configurations inside each user's directory.



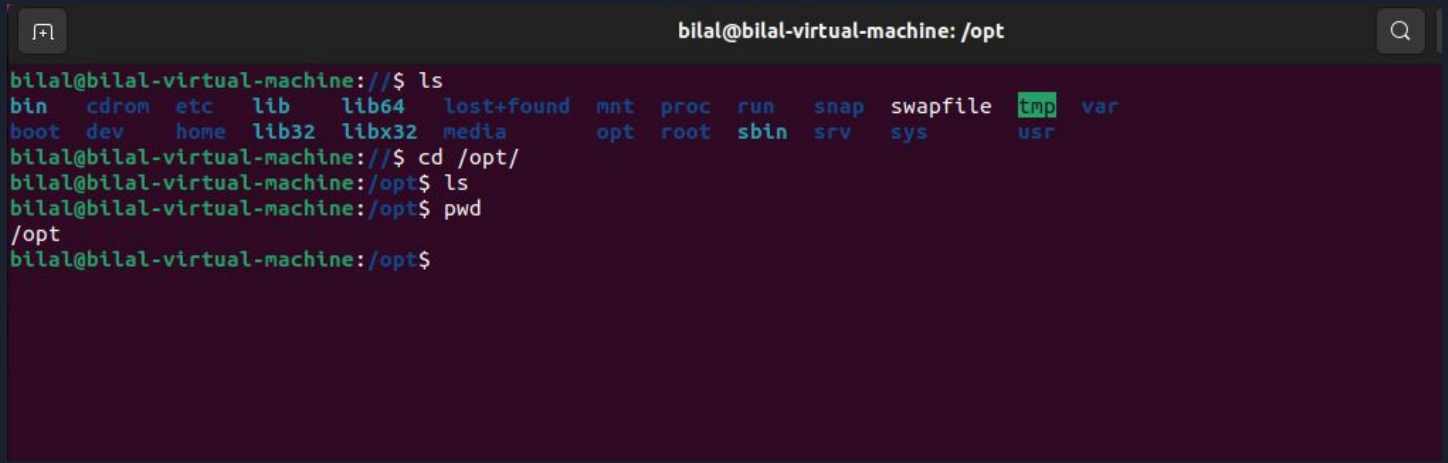
/bin

The /bin directory contains system commands and other executable programs. The ls command that you use to list out the subdirectories along with many other useful commands is located within the /bin directory.



/opt

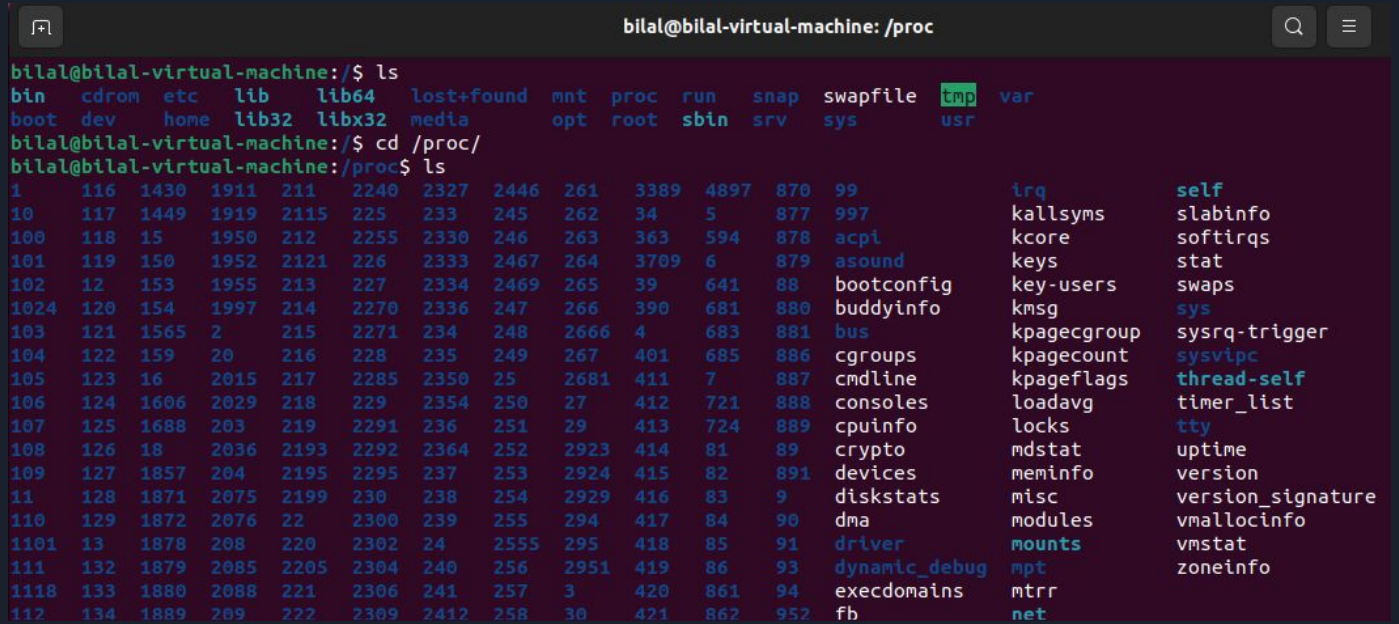
The /opt directory contains optional software packages to facilitate better compatibility of certain applications. When you install a third-party application that is not available in the official distribution repository, its software code gets stored in the /opt directory.



/proc

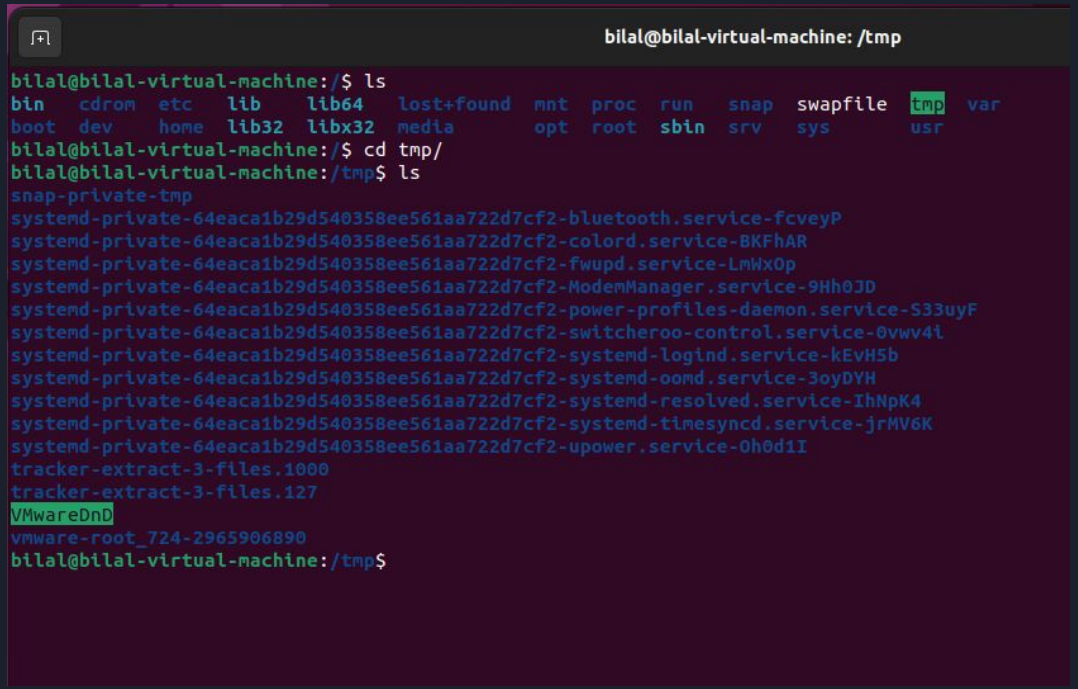
The /proc directory is a pseudo-filesystem containing information about processes and kernel parameters. It is populated with data during boot-up and is cleaned when you shut down your Linux machine.

The /proc directory is also home to system information such as memory usage, processor information, and so on.



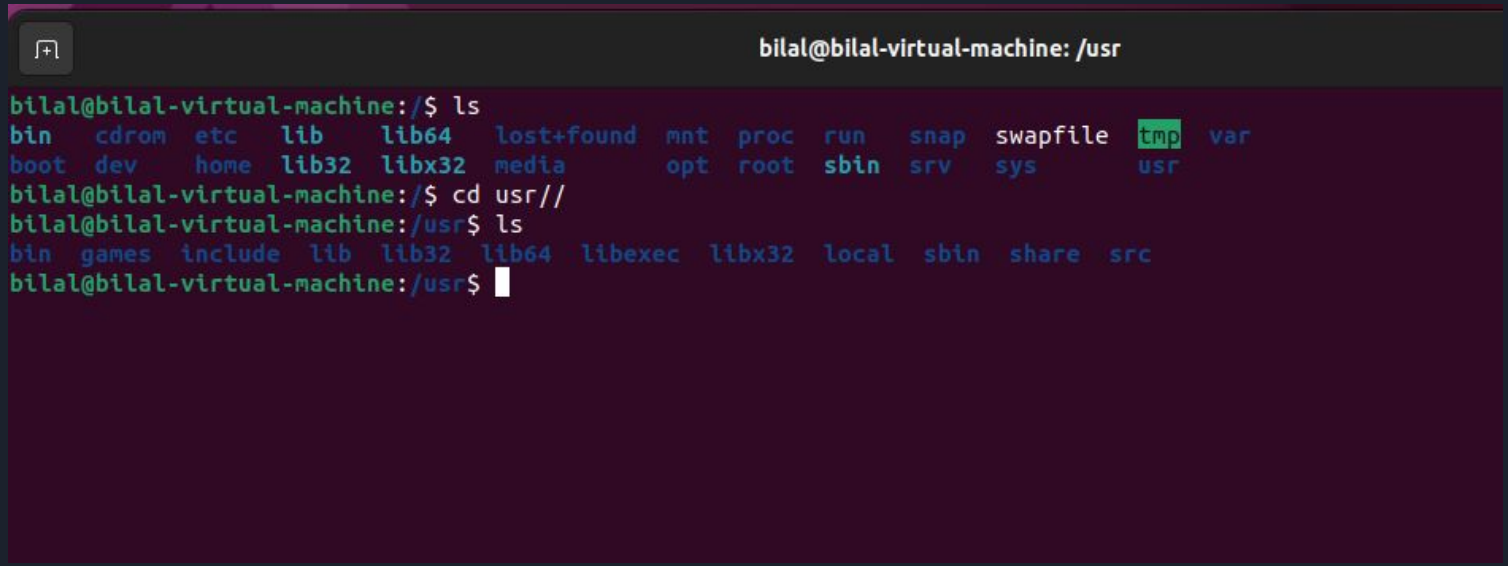
/tmp

The /tmp directory is used by the system and its applications to store temporary files. You can also store temporary data in this folder, but remember that the data will be deleted upon rebooting your system.



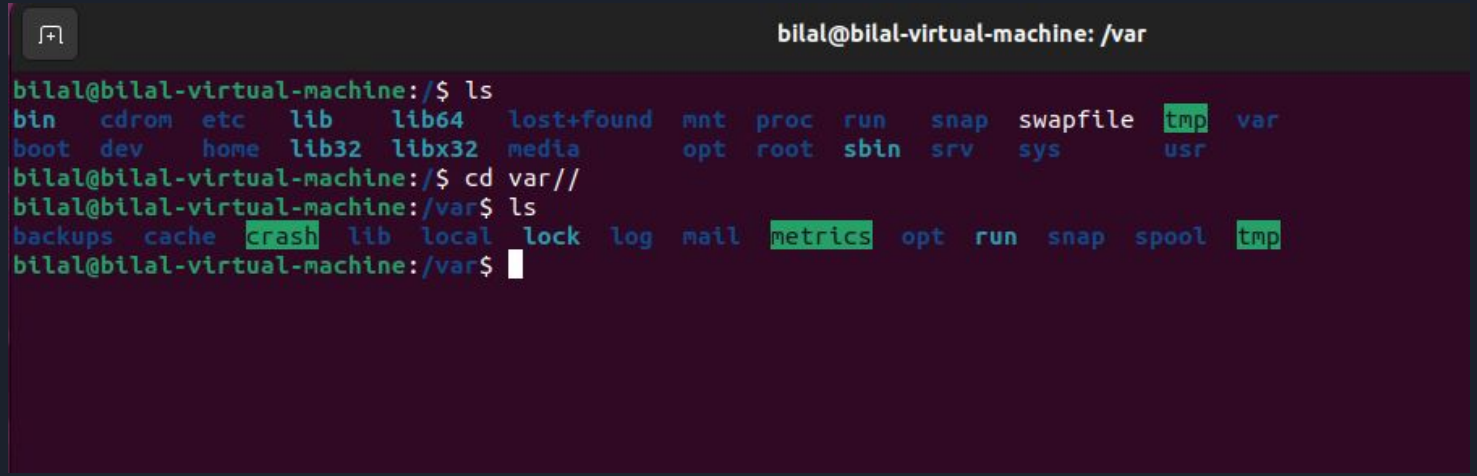
/usr

The /usr directory contains most of the files, libraries, programs, and system utilities. The /bin folder is symbolically linked to /usr/bin. The same goes for the /sbin and /lib directories.



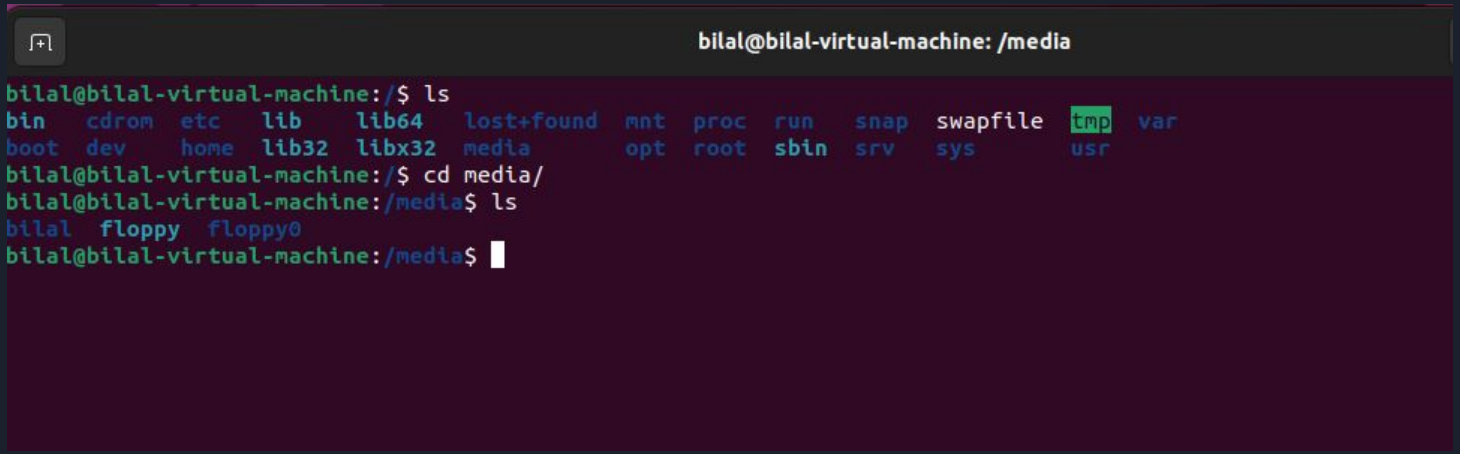
/var

The /var directory is the storage space for system-generated variable files, and it includes logs, caches, and spool files. The data in /var isn't automatically deleted, so sysadmins can collect and investigate system logs if need be.



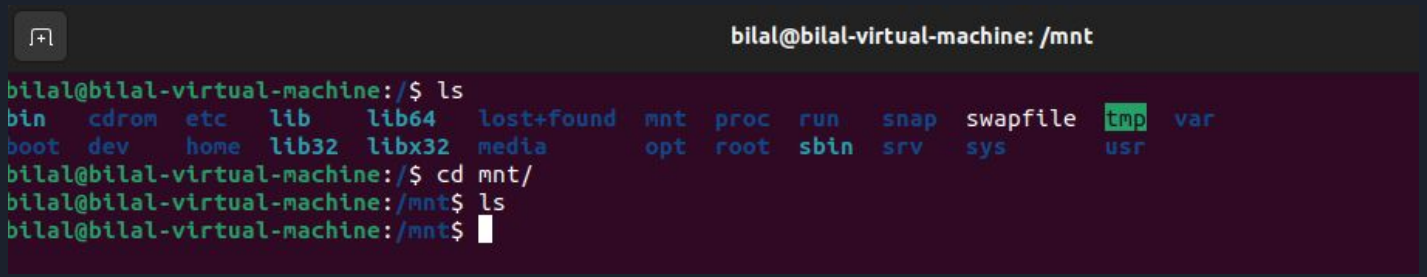
/media

When you connect any removable media device such as a USB thumb drive, CD, or DVD, Linux creates a subdirectory under /media where the contents of the device are laid out. This is usually done automatically by the system as soon as you plug the device in. When you remove the device, the system deletes the corresponding subdirectory.



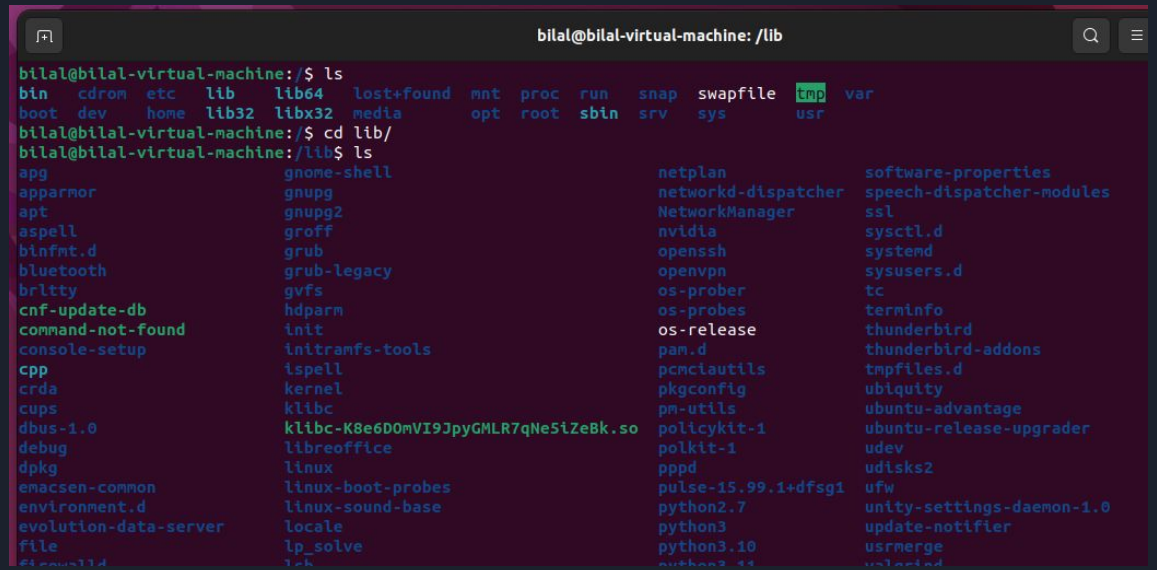
/mnt

The /mnt directory is used to mount storage devices in the system temporarily. However, some Linux distributions also use /mnt as a permanent storage solution. Unlike /media, the storage device isn't automatically mounted at /mnt by the system. Sysadmins have to manually mount a storage device and populate the file system table accordingly.



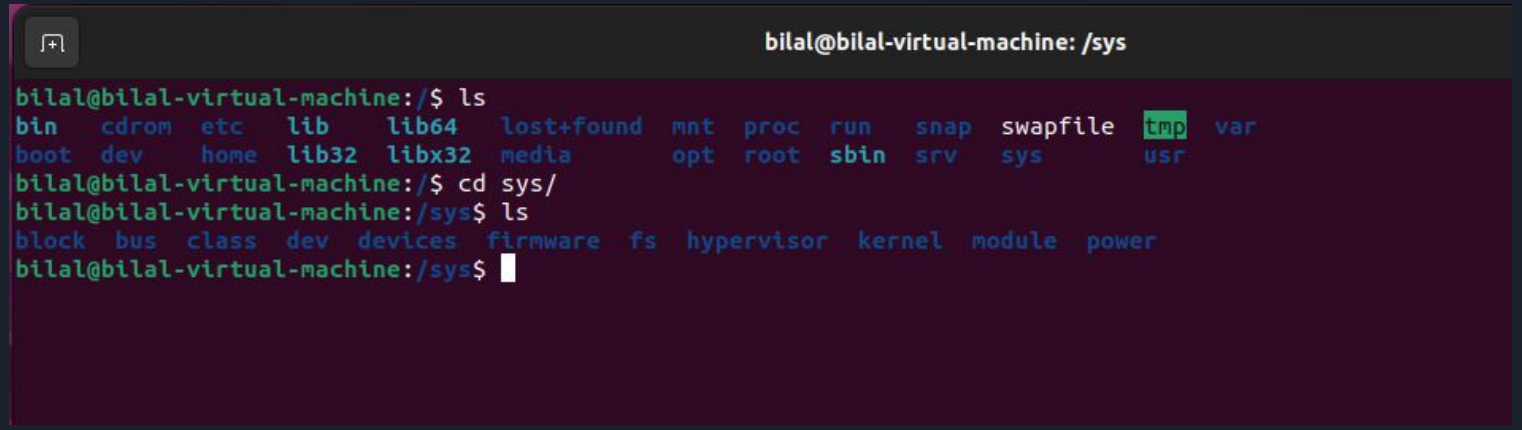
/lib

A library is a collection of pre-compiled code that executable binaries can use. In Linux, the /lib directory serves as the storage space for all libraries needed by the binaries in the /bin directory.



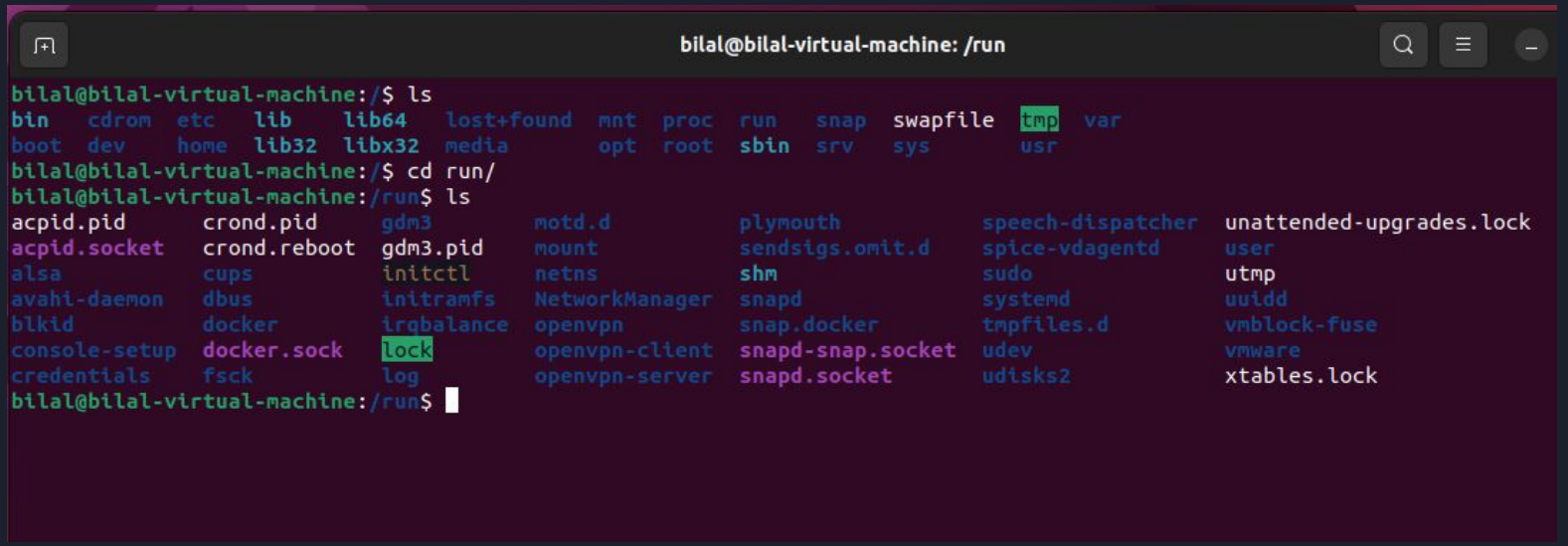
/sys

The /sys directory contains information about the various system components and drivers. It's akin to /proc but structured differently. Sysadmins use /proc and /sys interchangeably to collect data.

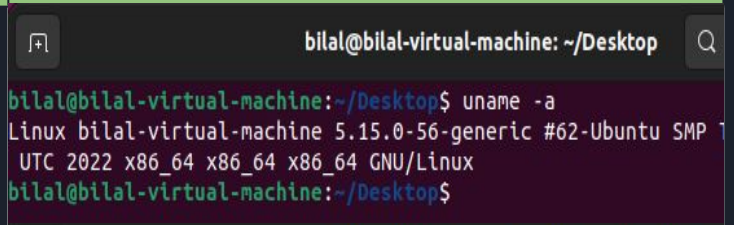
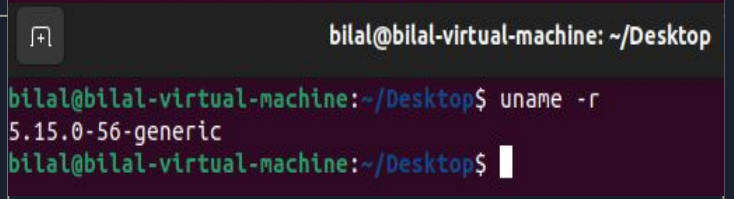
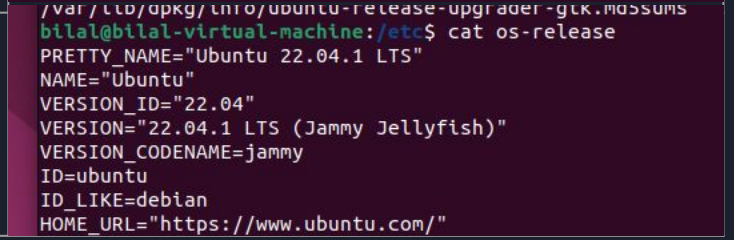


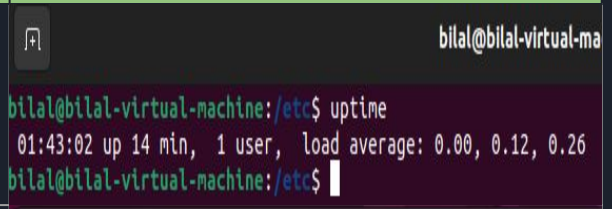
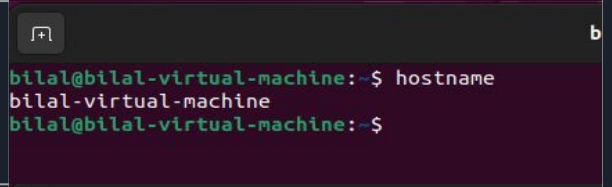
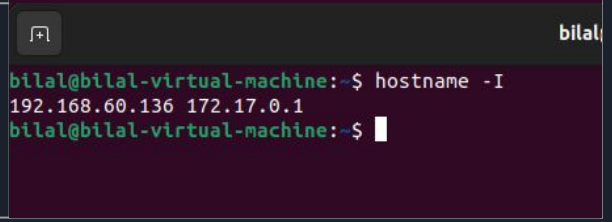
/run

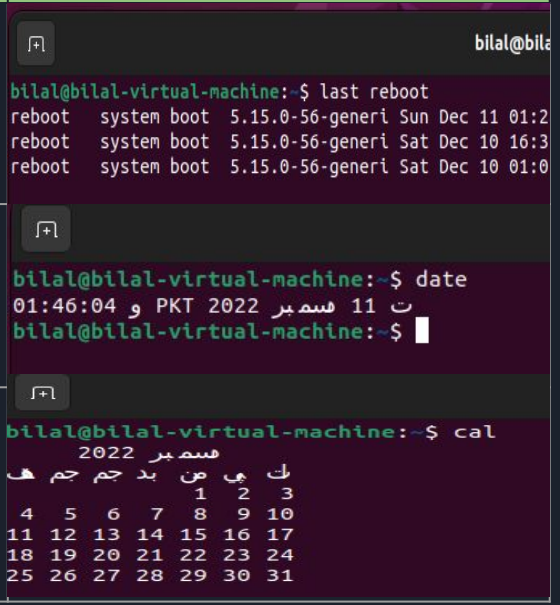
The /run directory logs system information since boot time. You can find information about the daemons that are running, logged-in users, and more. The data stored in the /run directory can give you an idea of how the system resources are being utilized since startup.



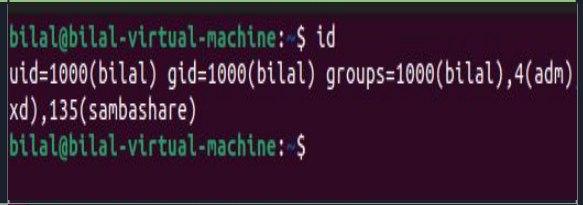
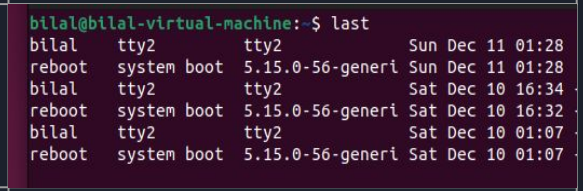
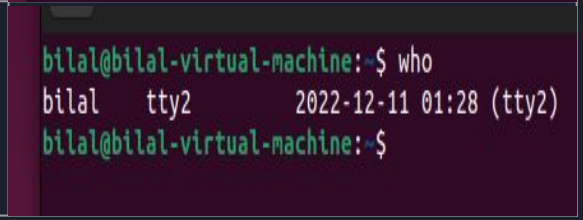
SYSTEM INFORMATION

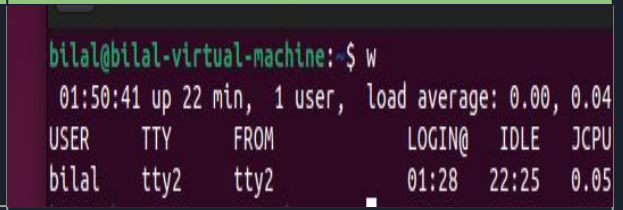
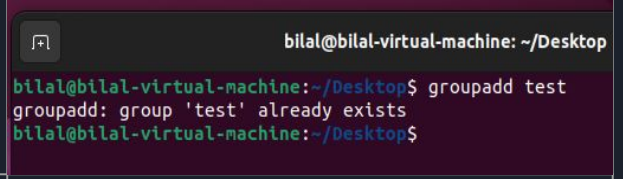
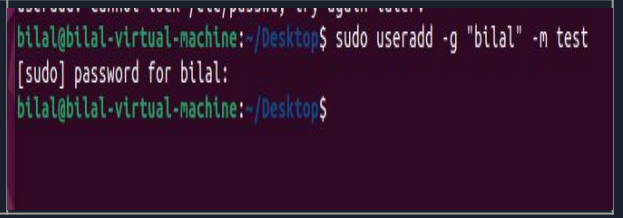
Commands	Description	POC
uname -a	Display Linux system information	
uname -r	Display kernel release information	
cat /etc/redhat-release	Show which version of Red Hat installed	

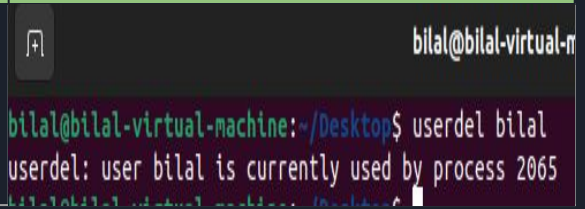
Commands	Description	POC
uptime	Show how long the system has been running + load	
hostname	Show system host name	
hostname -I	Display all local IP addresses of the host.	

Commands	Description	POC
last reboot	Show system reboot history	
date	Show the current date and time	
cal	Show this month's calendar	

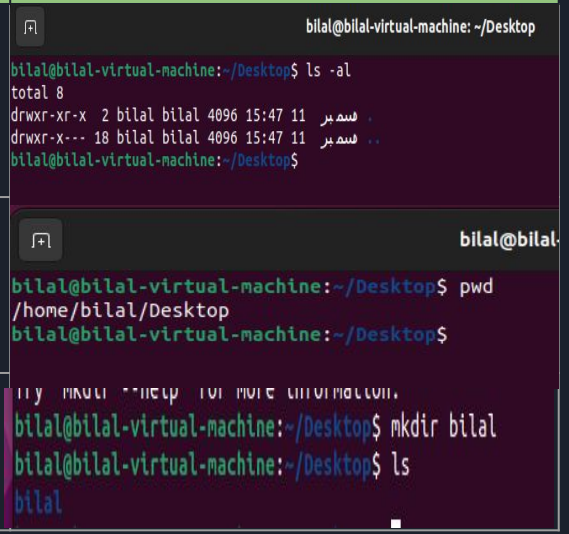
USER INFORMATION AND MANAGEMENT

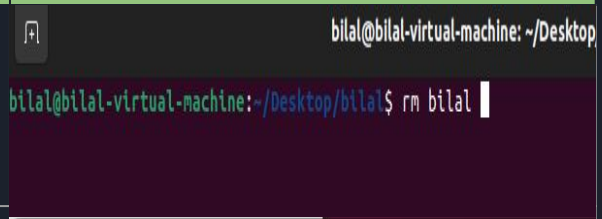
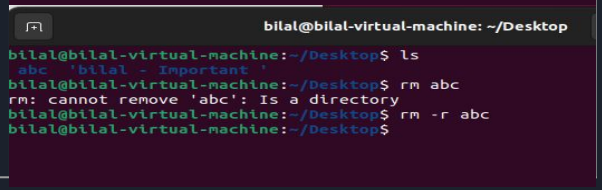
Commands	***Description***	***POC***
id	Display the user and group ids of your current user.	
last	Display the last users who have logged onto the system.	
who	Show who is logged into the system.	

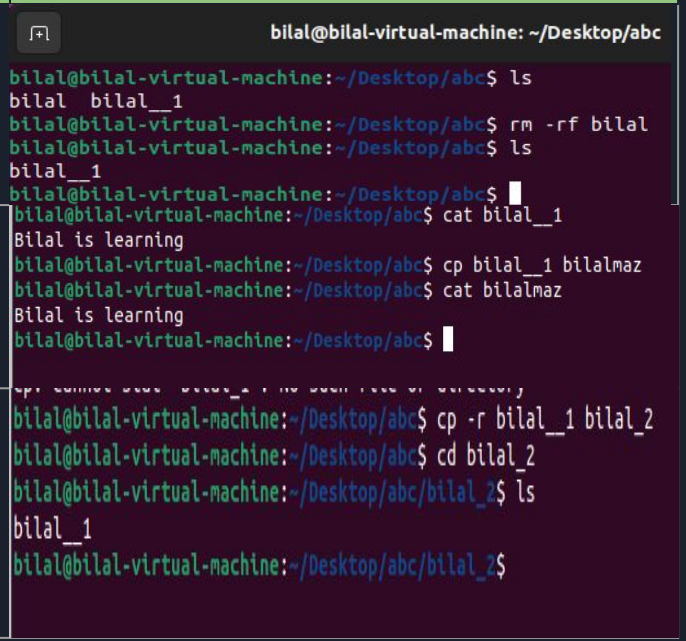
Commands	***Description***	***POC***
w	Show who is logged in and what they are doing.	
groupadd test	Create a group named "test".	
useradd -g "John Smith" -m john	Create an account named john, with a comment of "John Smith" and create the user's home directory.	

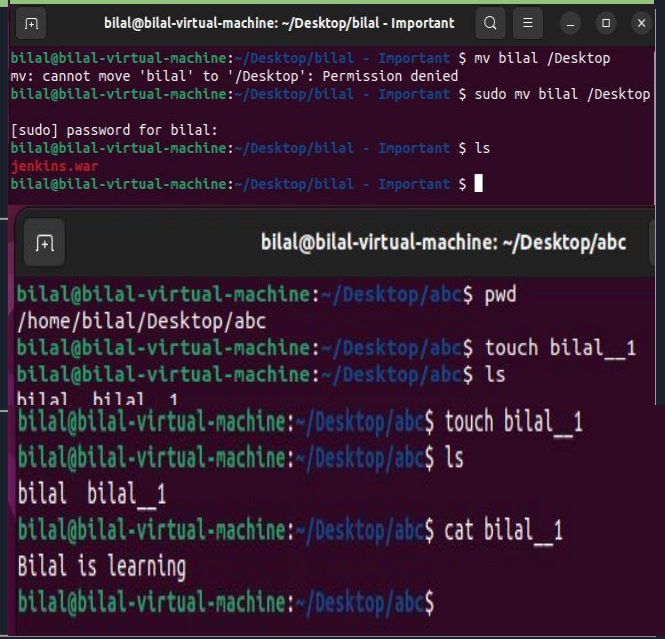
Commands	***Description***	***POC***
userdel john	Delete the john account.	

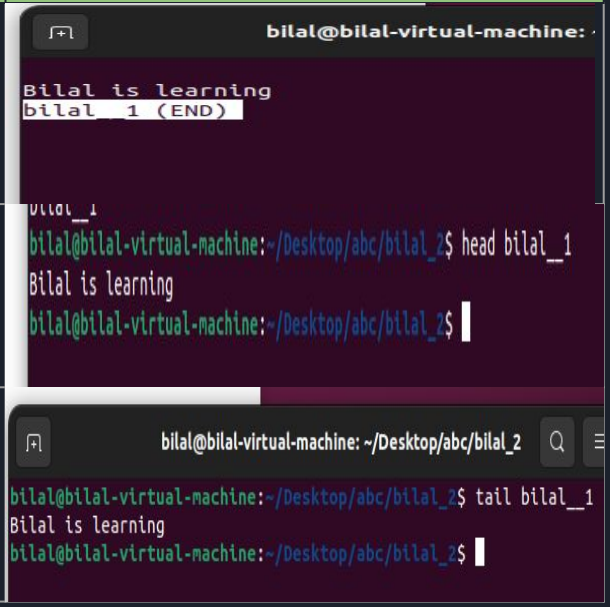
FILE AND DIRECTORY COMMANDS

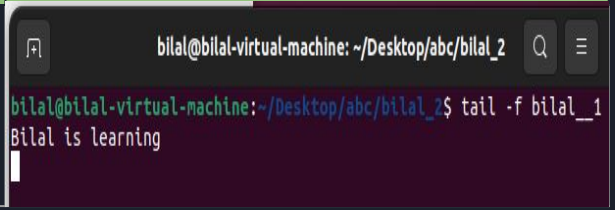
Commands	***Description***	***POC***
ls -al	List all files in a long listing (detailed) format	
pwd	Display the present working directory	
mkdir directory	Create a directory	

Commands	Description	POC
rm file	Remove (delete) file	
rm -r directory	Remove the directory and its contents recursively	

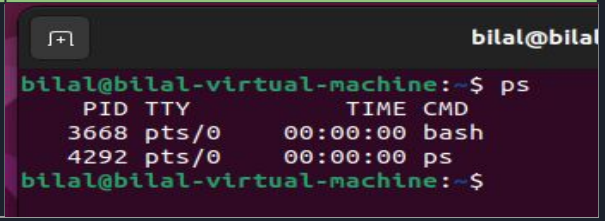
Commands	Description	POC
rm -rf directory	forcefully remove directory recursively	
cp file1 file2	Copy file1 to file2	
cp -r source_directory destination	Copy source_directory recursively to destination. If destination exists, copy source_directory into destination, otherwise create destination with the contents of source_directory.	

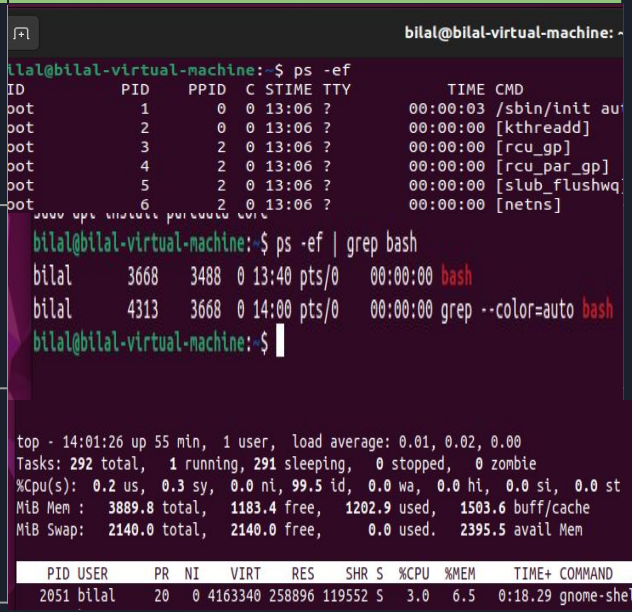
Commands	Description	POC
mv file1 file2	# Rename or move file1 to file2. If file2 is an existing directory, move file1 into directory file2	
touch file	Create an empty file or update the access and modification times of file.	
cat file	View the contents of file	

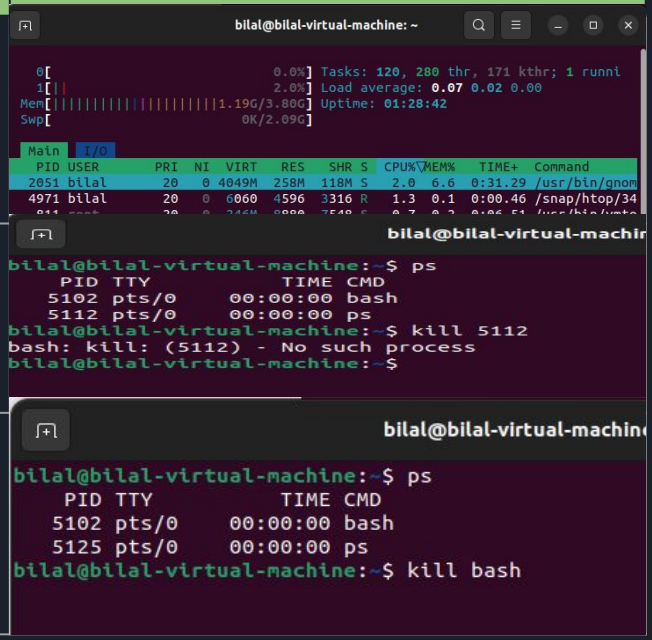
Commands	Description	POC
less file	Browse through a text file	
head file	Display the first 10 lines of file	
tail file	Display the last 10 lines of file	

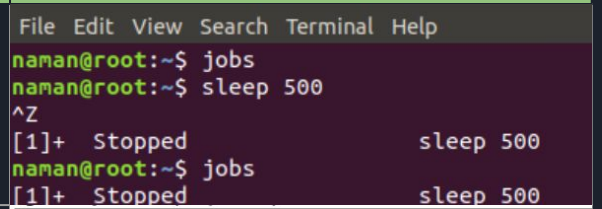
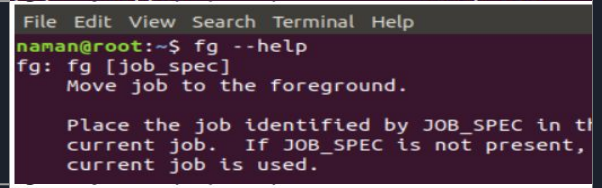
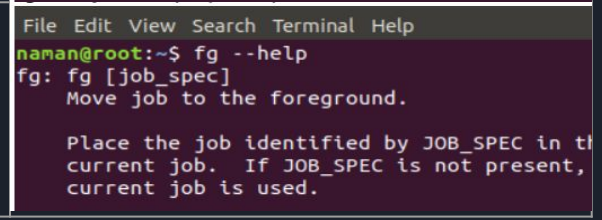
Commands	Description	POC
tail -f file	Display the last 10 lines of file and "follow" the file as it grows.	

PROCESS MANAGEMENT

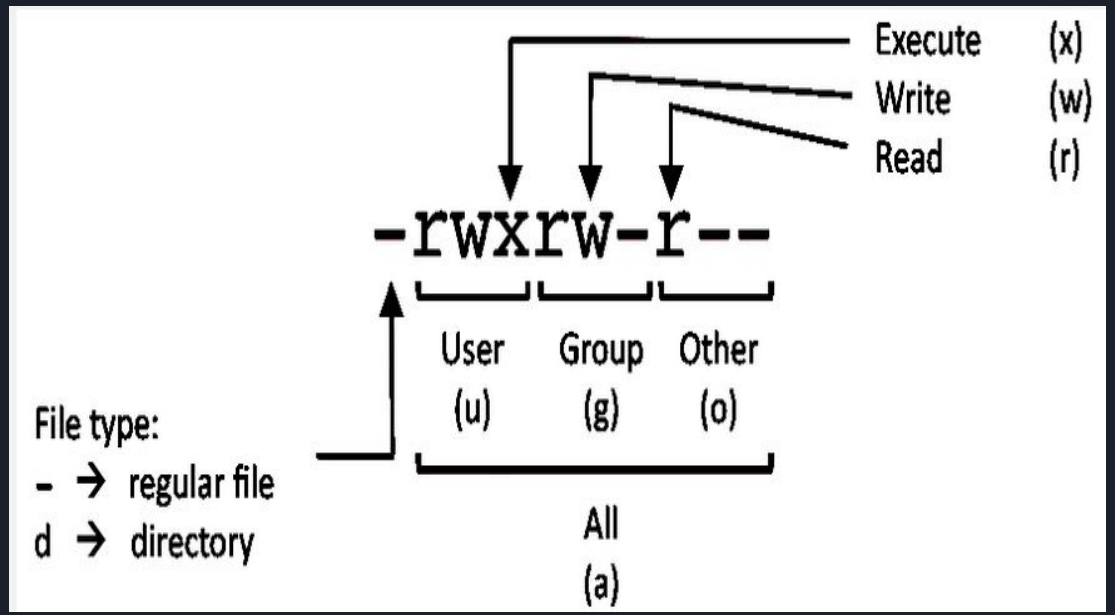
Commands	Description	POC
ps	Display your currently running processes	

Commands	Description	POC
ps -ef	Display all the currently running processes on the system	
ps -ef | grep processname	Display process information for processname	
top	Display and manage the top processes	

Commands	Description	POC
htop	Interactive process viewer (top alternative)	
kill pid	Kill process with process ID of pid	
killall process name	Kill all processes named processname	

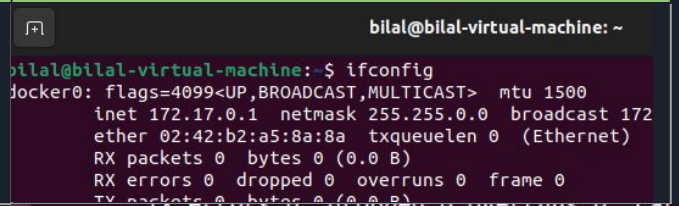
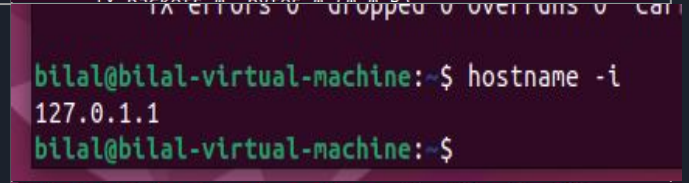
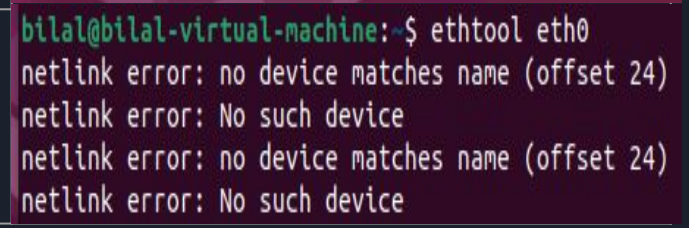
Commands	***Description***	***POC***
bg	Start program in the background	
fg	Brings the most recent background job to foreground	
fg n	Brings job n to the foreground	

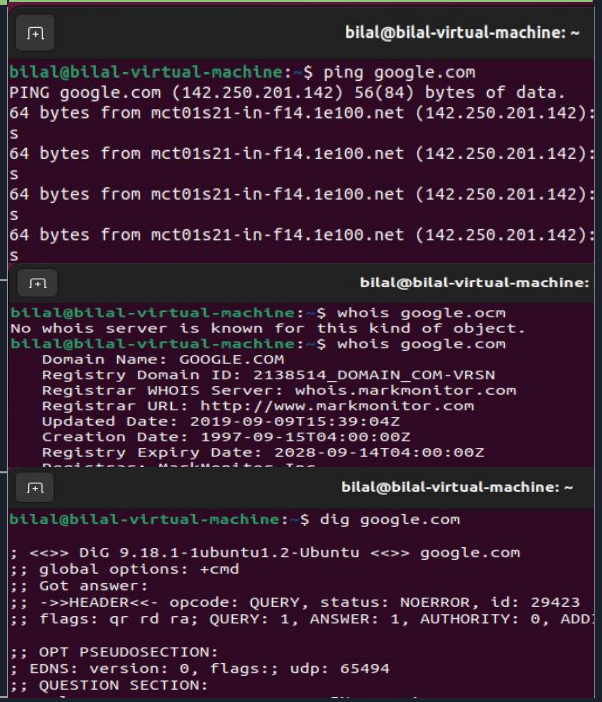
FILE PERMISSIONS

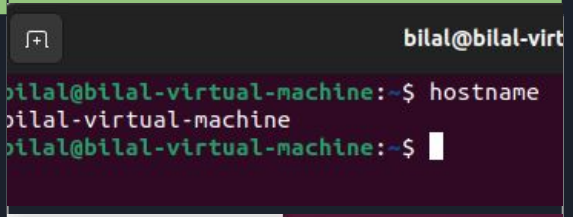
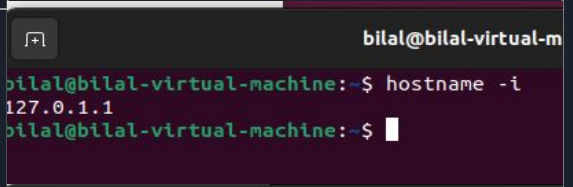
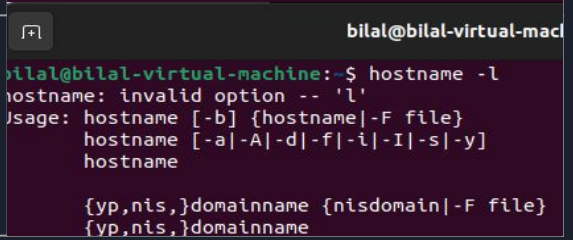


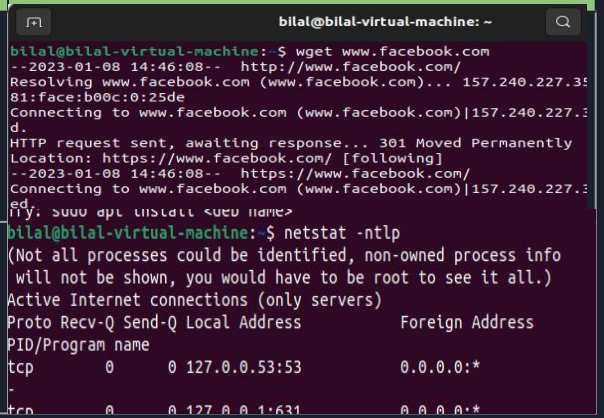

```
        PERMISSION      EXAMPLE

         U   G   W
        rwx rwx rwx     chmod 777 filename
        rwx rwx r-x     chmod 775 filename
        rwx r-x r-x     chmod 755 filename
        rw- rw- r--     chmod 664 filename
        rw- r-- r--     chmod 644 filename

# NOTE: Use 777 sparingly!

        LEGEND
        U = User
        G = Group
        W = World

        r = Read
        w = write
        x = execute
        - = no access
```

NETWORKING

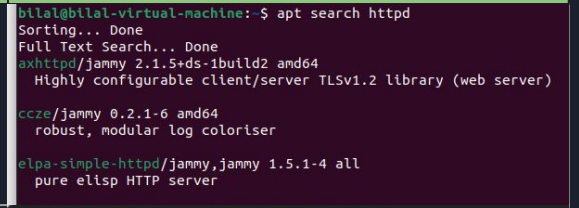
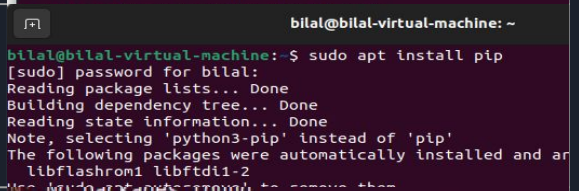
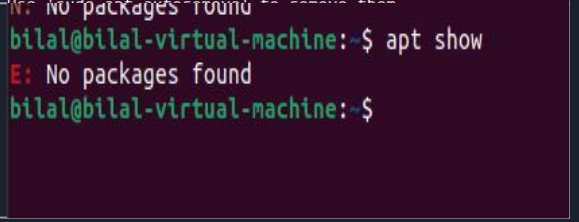
Commands	Description	POC
ip a / If config	Display all network interfaces and IP address	
Hostname -i	Display eth0 address and details	
ethtool eth0	Query or control network driver and hardware settings	

Commands	Description	POC
ping host	Send ICMP echo request to host	
whois domain	Display whois information for domain	
dig domain	Display DNS information for domain	

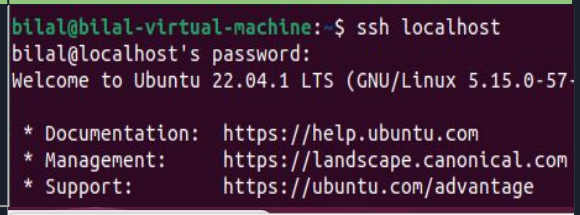
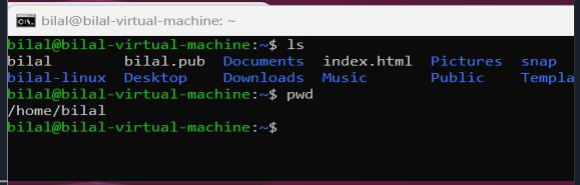
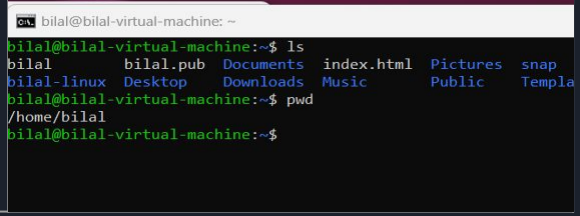
Commands	Description	POC
host domain	Display DNS IP address for domain	
hostname -i	Display the network address of the host name.	
hostname -I	Display all local IP addresses of the host.	

Commands	Description	POC
wget http://domain.com/file	Download http://domain.com/file	
netstat -nutlp	Display listening tcp and udp ports and corresponding programs	

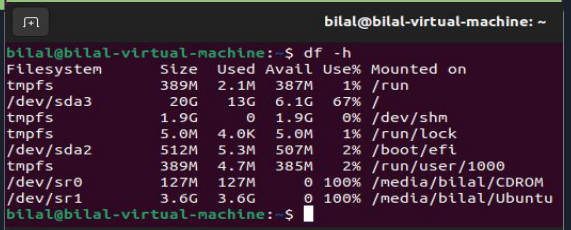
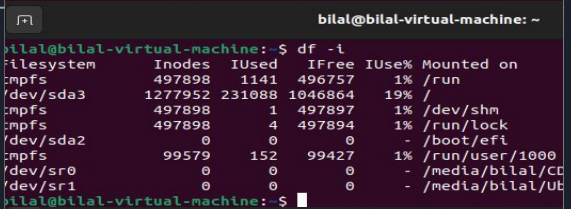
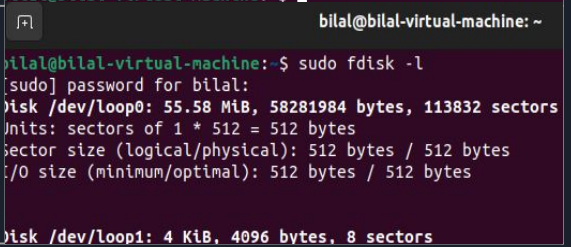
INSTALLING PACKAGES

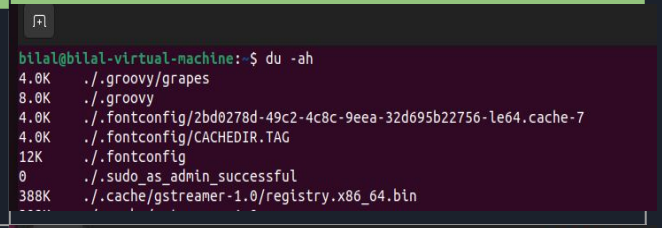
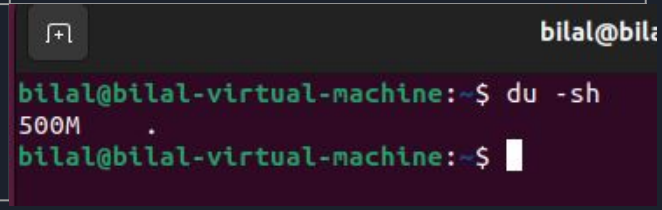
Commands	***Description***	***POC***
yum search keyword	Search for a package by keyword.	
Yum / snap/apt install package	Install package.	
yum info package	Display description and summary information about package.	

SSH LOGINS

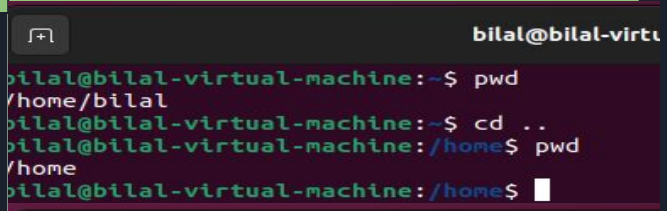
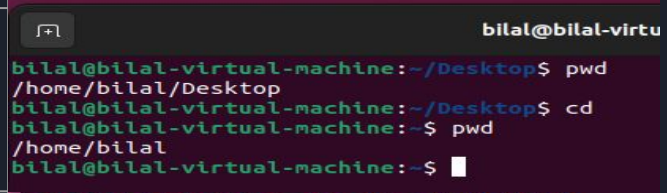
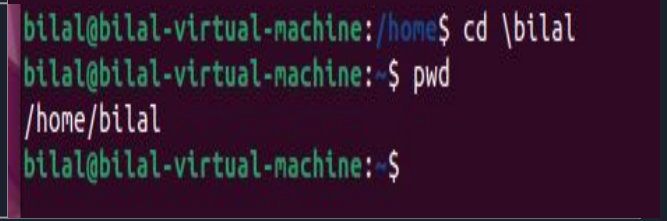
Commands	***Description***	***POC***
ssh host	Connect to host as your local username.	
ssh user@host	Connect to host as user	
ssh -p port user@host	Connect to host using port	

DISK USAGE

Commands	Description	POC
df -h	Show free and used space on mounted filesystems	
df -i	Show free and used inodes on mounted filesystems	
fdisk -l	Display disks partitions sizes and types	

Commands	Description	POC
du -ah	Display disk usage for all files and directories in human readable format	
du -sh	Display total disk usage off the current directory	

DIRECTORY NAVIGATION

Commands	***Description***	***POC***
cd ..	To go up one level of the directory tree. (Change into the parent directory.)	
cd	Go to the $HOME directory	
cd /etc	Change to the /etc directory	

Projects

Project 1 → Working with Vim editor

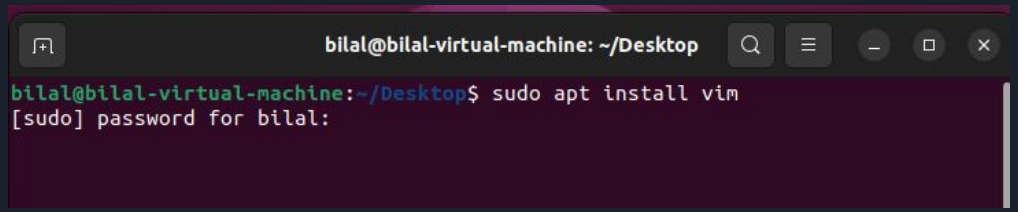
Project 2 → install Apache server in Ubuntu

Project 3 → Configure Firewall

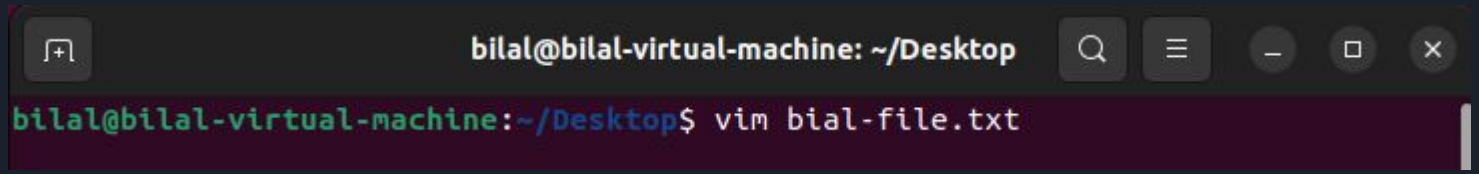
Project 4 → Wordpress Installation

Project 1 : Create test file in linux

→ Install Vim editor

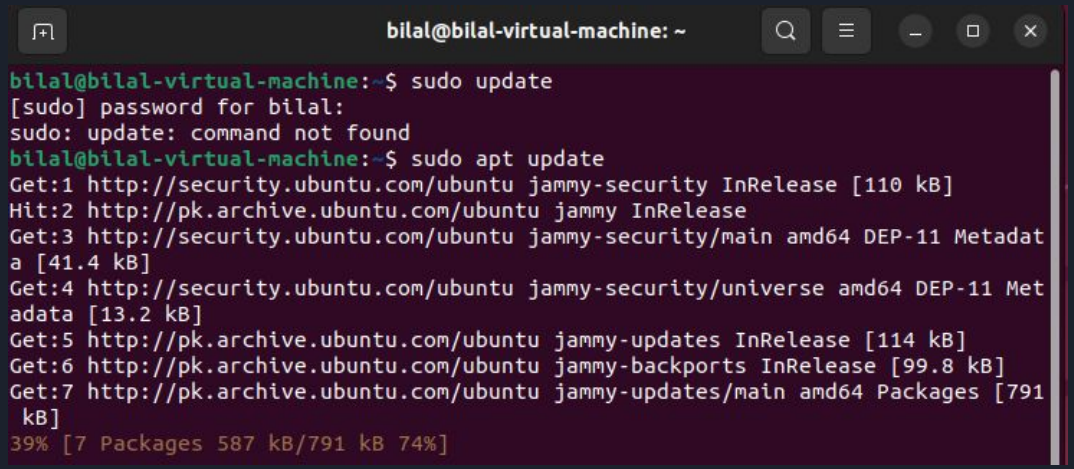


→ File Create



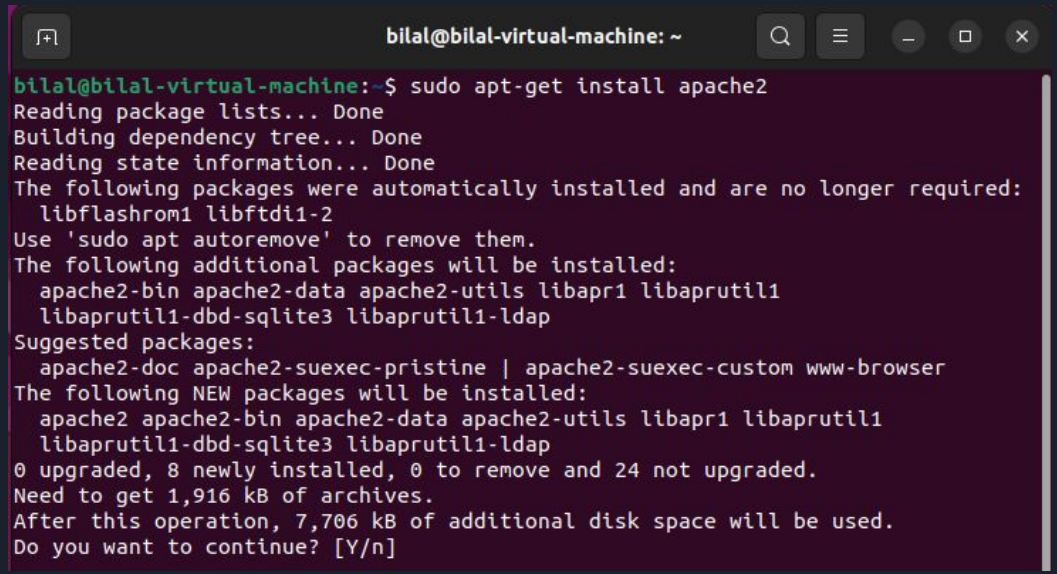
Project 2 : Install Apache server in Ubuntu

→ First, we need to update our system by using the below command:



Project 2 : Install Apache server in Ubuntu

→ After updating the system, we need to run the following command:



Project 2 : Install Apache server in Ubuntu

→ Start Apache Server

```
bilal@bilal-virtual-machine:~$ sudo systemctl start apache2
bilal@bilal-virtual-machine:~$
```

→ We can also see the status of whether Apache has been enabled using the command:

```
bilal@bilal-virtual-machine: ~
bilal-virtual-machine
    State: running
     Jobs: 0 queued
   Failed: 0 units
    Since: Sun 2023-01-08 13:06:16 PKT; 2h 32min ago
   CGroup: /
           ├─1661 bpfilter_umh
           ├─user.slice
           │ └─user-1000.slice
```

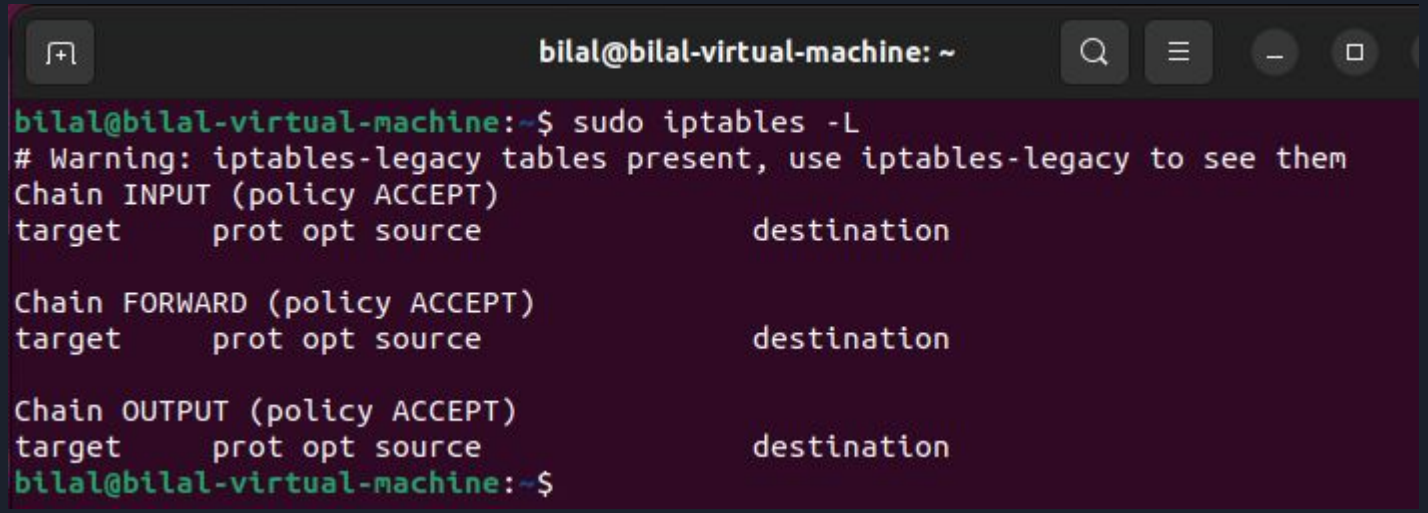
Project 3 : Configure Firewall

→ By default, the iptables is installed on most of the Linux distros. We can use the following command to ensure that iptable is available on our system

```
E: Unable to locate package iptable
bilal@bilal-virtual-machine:~$ sudo apt-get install iptables
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables is already the newest version (1.8.7-1ubuntu5).
iptables set to manually installed.
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
bilal@bilal-virtual-machine:~$
```

Project 3 : Configure Firewall

→ List the iptables current rules



Project 3 : Configure Firewall

→ Simply put, iptables is a firewall program for Linux. It will monitor traffic from and to your server using tables. These tables contain sets of rules, called chains, that will filter incoming and outgoing data packets.

When a packet matches a rule, it is given a target, which can be another chain or one of these special values:

→ **ACCEPT** – will allow the packet to pass through.

→ **DROP** – will not let the packet pass through.

→ **RETURN** – stops the packet from traversing through a chain and tell it to go back to the previous chain.

It will alert iptables that you are adding new rules to a chain. Then, you can combine the command with other options, such

→ **-i (interface)**

→ **-p (protocol)**

→ **-s (source)**

→ **–dport**

→ **-j (target)**

```
sudo iptables -A <chain> -i <interface> -p <protocol (tcp/udp) > -s
<source> --dport <port no.>  -j <target>
```

Project 3 : Configure Firewall

→ Enabling Traffic on Localhost

→ To allow traffic on localhost, type this command:

```
sudo iptables -A INPUT -i lo -j ACCEPT
```

For this iptables tutorial, we use lo or loopback interface. It is utilized for all communications on the localhost. The command above will make sure that the connections between a database and a web application on the same machine are working properly.

Project 3 : Configure Firewall

→ Enabling Connections on HTTP, SSH, and SSL Port

we want http (port 80), https (port 443), and ssh (port 22) connections to work as usual. To do this, we need to specify the protocol (-p) and the corresponding port (–dport). You can execute these commands one by one:

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

→ It's time to check if the rules have been appended in iptables:

```
sudo iptables -L -v
```

```
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:http
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:https
```

Project 3 : Configure Firewall

→ Filtering Packets Based on Source

Iptables allows you to filter packets based on an IP address or a range of IP addresses. You need to specify it after the -s option. For example, to accept packets from 192.168.1.3, the command would be:

You can also reject packets from a specific IP address by replacing the ACCEPT target with DROP.

```
sudo iptables -A INPUT -s 192.168.1.3 -j DROP
```

If you want to drop packets from a range of IP addresses, you have to use the -m option and iprange module. Then, specify the IP address range with –src-range. Remember, a hyphen should separate the range of ip addresses without space, like this:

```
sudo iptables -A INPUT -m iprange --src-range 192.168.1.100-192.168.1.200
-j DROP
```

Project 3 : Configure Firewall

→ Dropping all Other Traffic

It is crucial to use the DROP target for all other traffic after defining –dport rules. This will prevent an unauthorized connection from accessing the server via other open ports. To achieve this, simply type:

```
sudo iptables -A INPUT -j DROP
```

→ Deleting Rules

If you want to remove all rules and start with a clean slate, you can use the -F option (flush):

```
sudo iptables -F
```

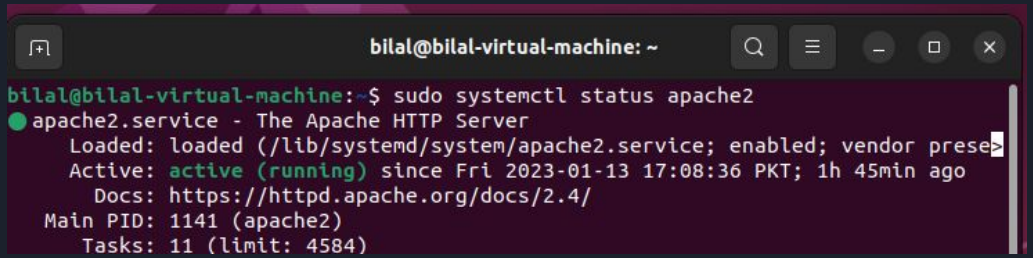
→ Persisting Changes

```
sudo /sbin/iptables-save
```

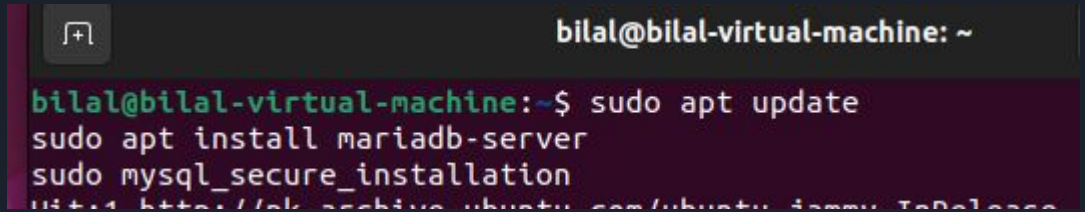
Project 4 : Wordpress installation in linux

→ You need to install apache2 and apache2 status can be found using the following command

→ Commands : sudo apt-get install apache2

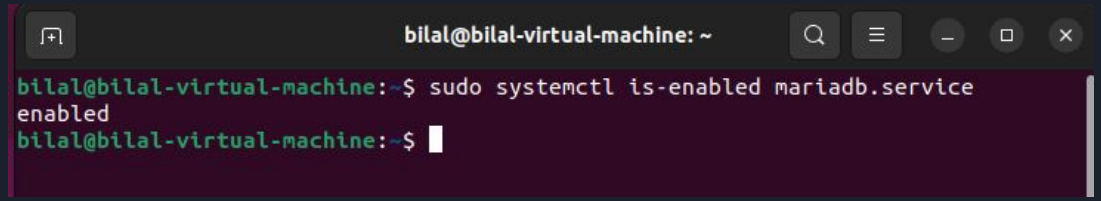


→ Install mariaDB

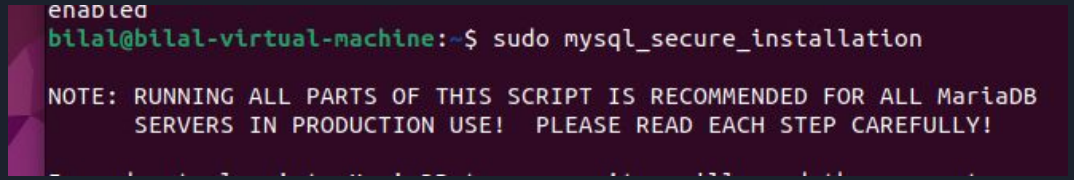


Project 4 : Wordpress installation in linux

→ To enable the service

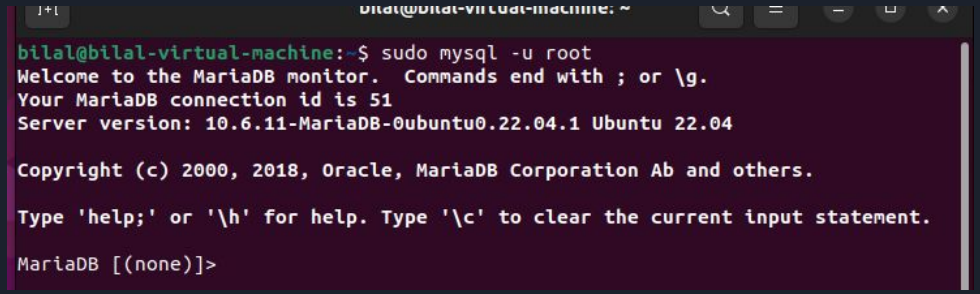


→ To ensure installation of Mysql

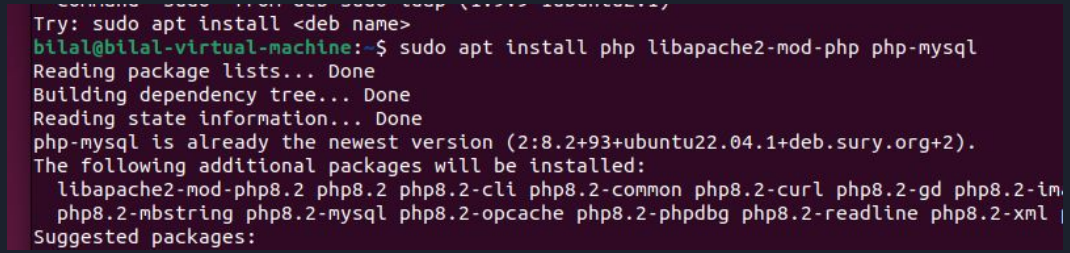


Project 4 : Wordpress installation in linux

→ Connect with Database

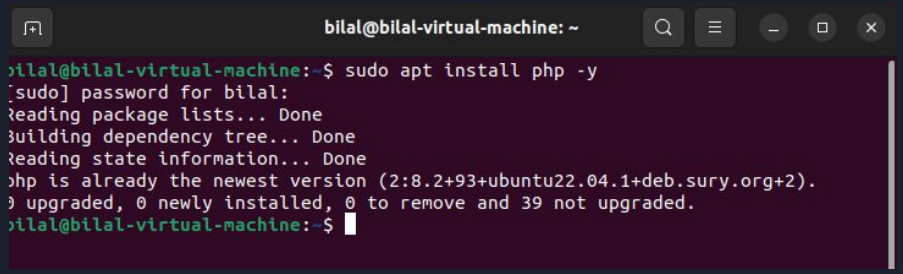


→ Install PHP libraries

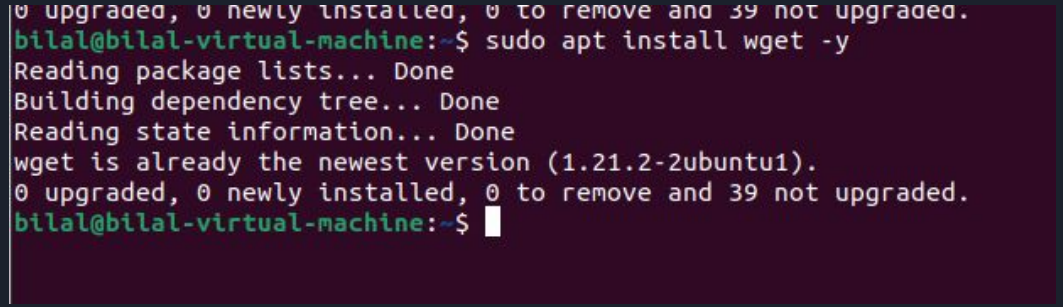


Project 4 : Wordpress installation in linux

→ Install php

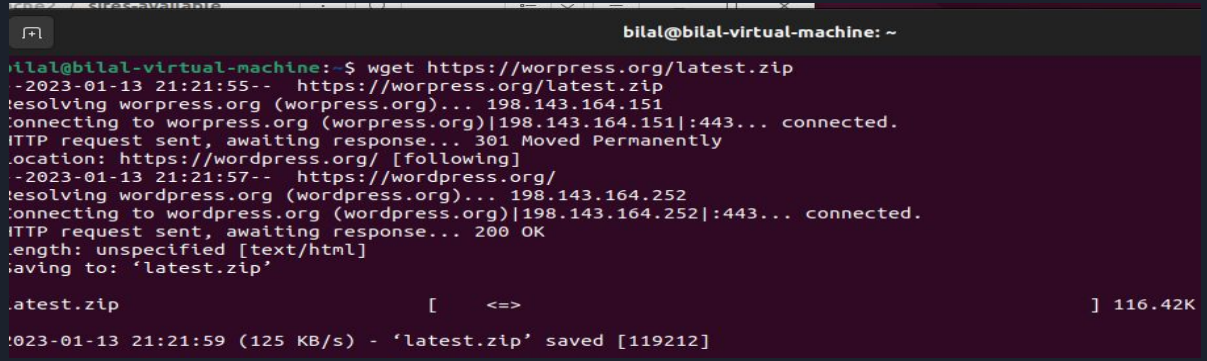


→ Install wget



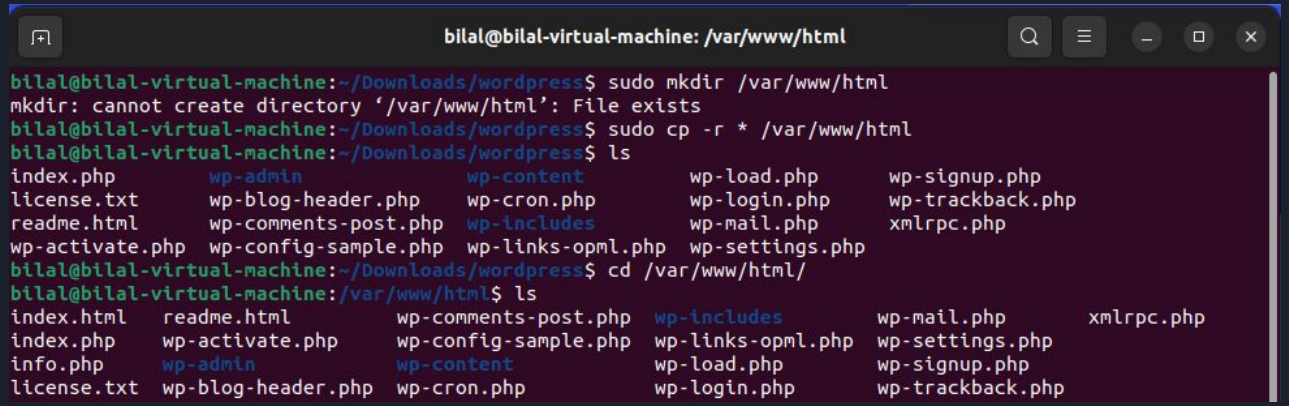
Project 4 : Wordpress installation in linux

→ Download wordpress



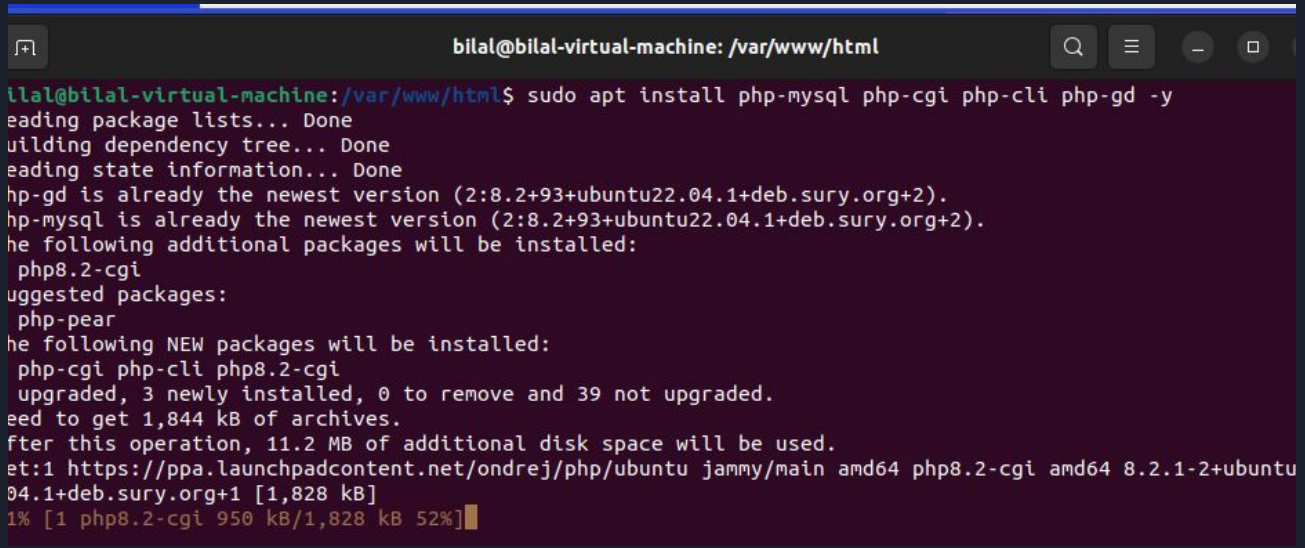
Project 4 : Wordpress installation in linux

→ copy wordpress file /var/www/html



Project 4 : Wordpress installation in linux

→ library installation



Project 4 : Wordpress installation in linux

→ restart apache2

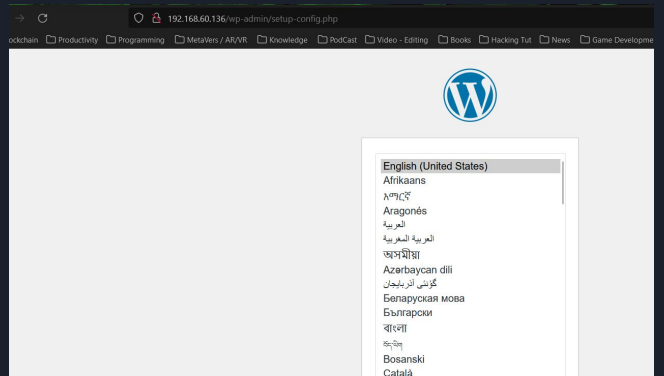
```
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for php8.2-cgi (8.2.1-2+ubuntu22.04.1+deb.sury.org+1) ...
bilal@bilal-virtual-machine:/var/www/html$ sudo systemctl restart apache2
bilal@bilal-virtual-machine:/var/www/html$
```

→ Change permission

```
chown: invalid group: ‘www-data:ww-data’
bilal@bilal-virtual-machine:/var/www/html$ sudo chown -R www-data:www-data /var/www/
bilal@bilal-virtual-machine:/var/www/html$
```

→ Check your IP : ip a or ifconfig

→ open Browser → ip/wp-admin/setup-config.php



Project 4 : Wordpress installation in linux

→ login to mysql

```
       valid_lft forever preferred_lft forever
bilal@bilal-virtual-machine:/var/www/html$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 67
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

→ Create Database

```
MariaDB [(none)]> create database wordpress;
ERROR 1007 (HY000): Can't create database 'wordpress'; database exists
MariaDB [(none)]>
```

Project 4 : Wordpress installation in linux

→ Create user

```
+--------------------+
8 rows in set (0.002 sec)

MariaDB [(none)]> create user "wordpress@123" identified by "password";
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

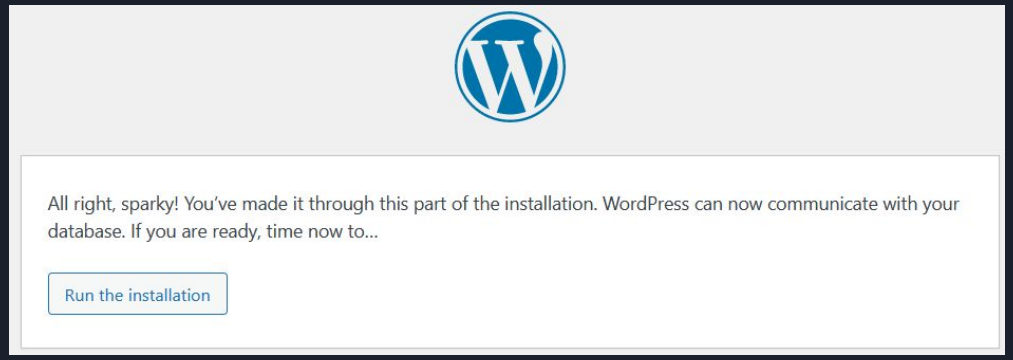
→ Grand privileges

```
 server version for the right syntax to use near 'privilleges on wordpress.*
MariaDB [(none)]> grant all  on wordpress.* to "wordpress@123";
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Project 4 : Wordpress installation in linux

→ add detail in wordpress wizard

→ begin installation



→ We made it !!