

**MAKHANLAL CHATURVEDI UNIVERSITY
COMPUTER DEPARTMENT**



LINUX SERVER ADMINISTRATION

CLASS: - MCA (1)

NAME: - ARPIT SHAKYA

SUBMITTING TO: - MR PRAVEEN KUMAR

DATE OF SUBMISSION: - 13-12-2024

MCA1st Linux Practical's List

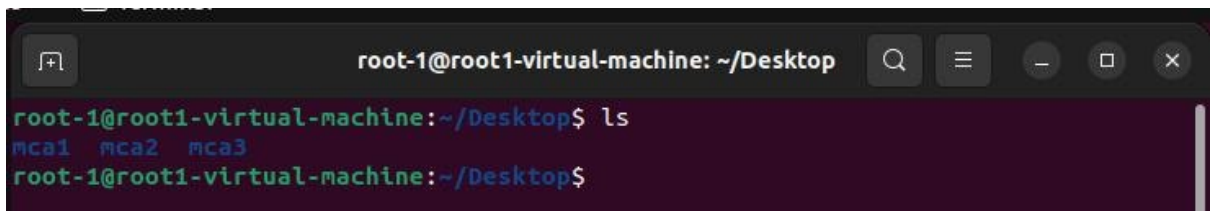
Q1. Command Usages

- o Describe the usage of: ls, pwd, tty, cat, who, whoami, rm, mkdir, touch, cd, cal, cat (append & concatenate), mv, cp, man, date.

Answer: -

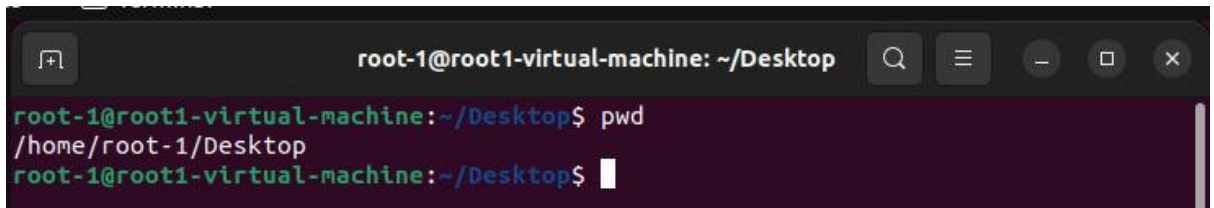
Here's a description of the usage for the **listed Linux commands**:

1. **ls**:
Lists the files and directories in the current directory.
Example: ls (lists files), ls -l (lists in long format).



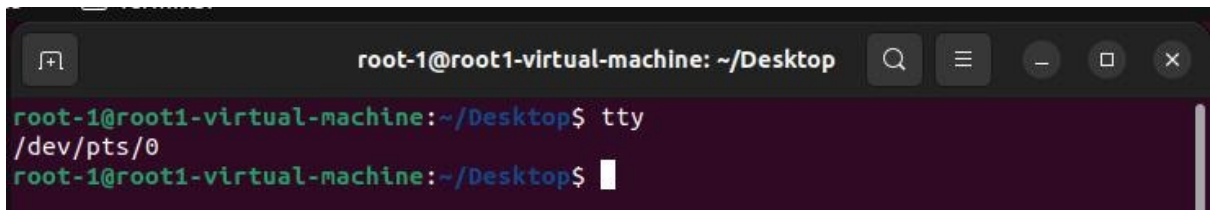
```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ ls
mca1 mca2 mca3
root-1@root1-virtual-machine:~/Desktop$
```

2. **pwd**:
Prints the current working directory.
Example: pwd (e.g., /home/user).



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ pwd
/home/root-1/Desktop
root-1@root1-virtual-machine:~/Desktop$
```

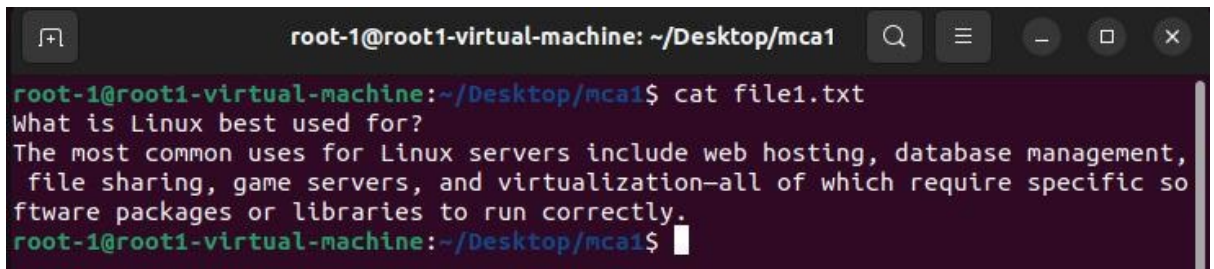
3. **tty**:
Displays the file name of the terminal connected to the session.
Example: tty (e.g., /dev/tty1).



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ tty
/dev/pts/0
root-1@root1-virtual-machine:~/Desktop$
```

4. **cat:**

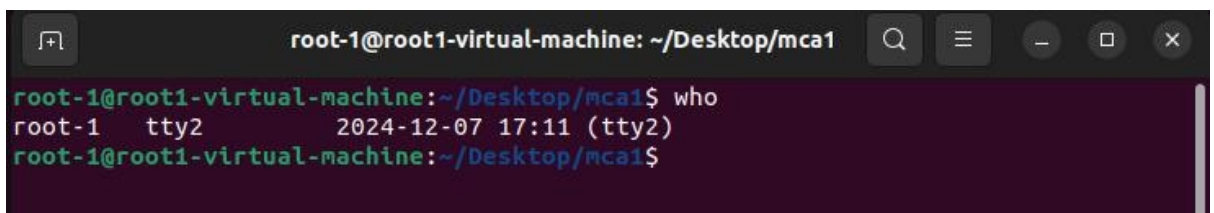
Concatenates and displays the contents of files.
Example: `cat file.txt` (displays contents of file.txt).



```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ cat file1.txt
What is Linux best used for?
The most common uses for Linux servers include web hosting, database management,
file sharing, game servers, and virtualization—all of which require specific so
ftware packages or libraries to run correctly.
root-1@root1-virtual-machine:~/Desktop/mca1$
```

5. **who:**

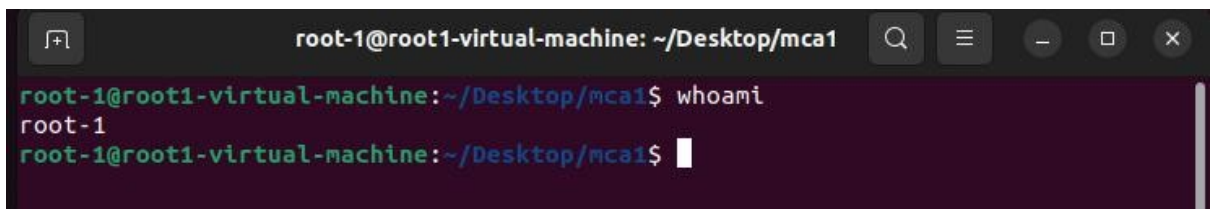
Displays who is logged in to the system.
Example: `who` (lists usernames, terminal, and login time).



```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ who
root-1  tty2          2024-12-07 17:11 (tty2)
root-1@root1-virtual-machine:~/Desktop/mca1$
```

6. **whoami:**

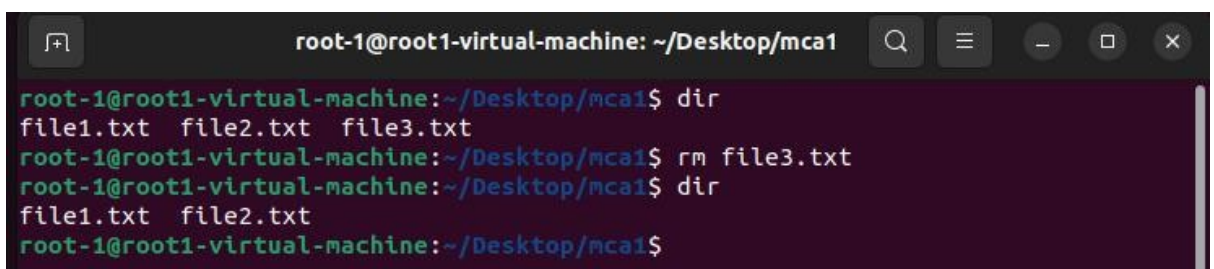
Displays the current logged-in user.
Example: `whoami` (e.g., user).



```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ whoami
root-1
root-1@root1-virtual-machine:~/Desktop/mca1$
```

7. **rm:**

Removes files or directories.
Example: `rm file.txt` (removes file.txt), `rm -r dir/` (removes directory recursively).



```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ dir
file1.txt file2.txt file3.txt
root-1@root1-virtual-machine:~/Desktop/mca1$ rm file3.txt
root-1@root1-virtual-machine:~/Desktop/mca1$ dir
file1.txt file2.txt
root-1@root1-virtual-machine:~/Desktop/mca1$
```

8. **mkdir:**

Creates a new directory.
Example: `mkdir new_directory` (creates new_directory).

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ mkdir mcu_bhopal
root-1@root1-virtual-machine:~/Desktop$ dir
mca1 mca2 mca3 mcu_bhopal
root-1@root1-virtual-machine:~/Desktop$
```

9. **touch:**

Creates an empty file or updates the timestamp of an existing file.

Example: touch newfile.txt (creates an empty file named newfile.txt).

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ touch bhopal.txt
root-1@root1-virtual-machine:~/Desktop$
```

10. **cd:**

Changes the current directory.

Example: cd /home/user (moves to /home/user directory).

```
root-1@root1-virtual-machine: ~/Desktop/mcu_bhopal
root-1@root1-virtual-machine:~/Desktop$ cd mcu_bhopal
root-1@root1-virtual-machine:~/Desktop/mcu_bhopal$
```

11. **cal:**

Displays a calendar.

Example: cal (shows the current month), cal 2024 (shows calendar for the year 2024).

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ cal
December 2024
Su Mo Tu We Th Fr Sa
 1  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 31
root-1@root1-virtual-machine:~/Desktop$
```

12. **cat (append & concatenate):**

- o **Concatenate:** Combines multiple files into one.
- o **Append:** Appends content to an existing file.
Example: cat file1.txt file2.txt > merged.txt (concatenates files into merged.txt),
>> file.txt (appends text to file.txt).

```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ cat file1.txt file2.txt > merged.txt
root-1@root1-virtual-machine:~/Desktop/mca1$ dir
file1.txt file2.txt merged.txt
root-1@root1-virtual-machine:~/Desktop/mca1$
```

13. **mv:**

Moves or renames files or directories.

Example: mv oldname.txt newname.txt (renames file), mv file.txt /home/user/ (moves file to another directory).

```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ mv file1.txt file2.txt
root-1@root1-virtual-machine:~/Desktop/mca1$
```

14. **cp:**

Copies files or directories.

Example: cp file.txt copy.txt (copies file.txt to copy.txt).

```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ cp file1.txt file2.txt
root-1@root1-virtual-machine:~/Desktop/mca1$
```

15. **man:**

Displays the manual or help documentation for a command.

Example: man ls (shows the manual page for ls).

```
root-1@root1-virtual-machine: ~/Desktop/mca1
LS(1) User Commands LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvsUX nor --sort is specified.
  Mandatory arguments to long options are mandatory for short options too.
  -a, --all
    do not ignore entries starting with .
  -A, --almost-all
    do not list implied . and ..
  --author
  Manual page ls(1) line 1 (press h for help or q to quit)
```

16. **date:**

Displays or sets the current date and time.

Example: (formats the date output).

```
root-1@root1-virtual-machine: ~/Desktop/mca1
root-1@root1-virtual-machine:~/Desktop/mca1$ date
Saturday 07 December 2024 05:30:36 PM IST
root-1@root1-virtual-machine:~/Desktop/mca1$
```

Q2. Root Password Recovery

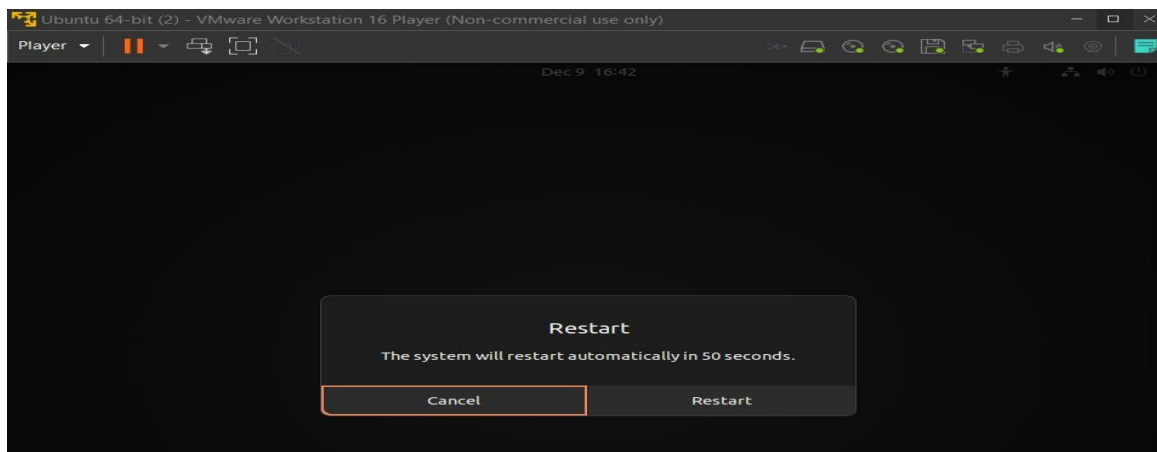
- Explain the steps to reset the root password.

Answer: -

Steps to Recover Root Password in Ubuntu

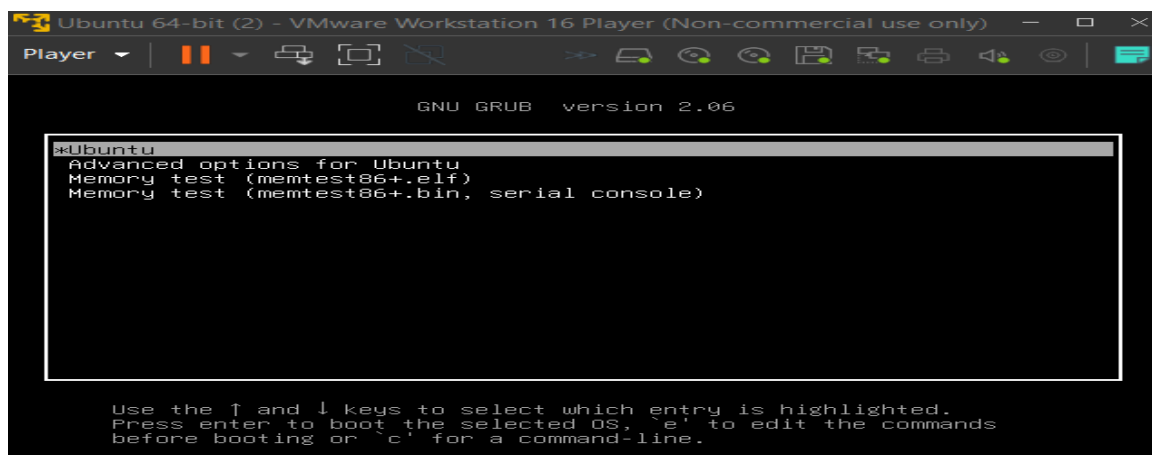
1. Power on the VM and access the GRUB menu:

- o When starting the VM, hold down the **Shift** key (or repeatedly press the **Esc** key) to access the GRUB menu.



2. Edit GRUB boot options:

*The first step is to reboot into the GRUB menu. If Ubuntu 20.04 is the only operating system installed you need to keep pressing **SHIFT** for the GRUB menu to show up. Next, while your **Ubuntu** boot menu is highlighted press **e** to edit the Grub's boot prompt.*



3. Modify the boot commands:

Using your navigational arrows locate the line containing the following string **ro quiet splash \$vt_handoff** & replace the string **ro quiet splash \$vt_handoff** with the following text **init=/bin/bash**. Once you have made the change press **F10** to initiate the regular boot sequence.

```

insmod ext2
set root='hd0,gpt3'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt3 -\
--hint-efi=hd0,gpt3 --hint-baremetal=ahci0,gpt3 9a6dfd40-a553-4dc1-a7c8-\
69b6a8d35ee1
else
  search --no-floppy --fs-uuid --set=root 9a6dfd40-a553-4dc1-a7c\
8-69b6a8d35ee1
fi
linux /boot/vmlinuz-6.8.0-49-generic root=UUID=9a6dfd40-a\
553-4dc1-a7c8-69b6a8d35ee1 rw init=/bin/bash_
initrd /boot/initrd.img-6.8.0-49-generic

```

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return to the GRUB menu.

4. After the successful boot you should be welcomed with a root's shell without a need to enter the root password:

```

form2 cdda tray
[ 11.975008] sr 4:0:0:0: Attached scsi generic sg2 type 5
[ 12.049525] e1000 0000:02:01:0 ens33: renamed from eth0
[ 12.506102] usbcore: registered new interface driver usbhid
[ 12.508922] usbhid: USB HID core driver
[ 12.518450] input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/000\
0:00:11.0/0000:02:00.0/usb1/1-1/1-1:1.0/0003:000F:0003.0001/input/input5
[ 12.521781] hid-generic 0003:000F:0003.0001: input,hidraw0: USB HID v1.10 Mou\
se [VMware VMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0
Begin: Loading essential drivers ... done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
Begin: Running /scripts/local-premount ... done.
Begin: Mounting /usr ... done.
Begin: Will now check root file system ... fsck from util-linux 2.37.2
[/usr/sbin/fsck.ext4 (1) -- /dev/sda3] fsck.ext4 -a -C0 /dev/sda3
/dev/sda3: clean, 220066/1277952 files, 3254582/5110784 blocks
done.
[ 12.698403] EXT4-fs (sda3): mounted filesystem 9a6dfd40-a553-4dc1-a7c8-69b6a8\
d35ee1 r/w with ordered data mode. Quota mode: none.
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
root@(none):/#

```

5. Confirm that the root (/) partition is mounted as **rw**. To do so execute the **mount | grep /** command.

```

[ 12.506102] usbcore: registered new interface driver usbhid
[ 12.508922] usbhid: USB HID core driver
[ 12.518450] input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/000\
0:00:11.0/0000:02:00.0/usb1/1-1/1-1:1.0/0003:000F:0003.0001/input/input5
[ 12.521781] hid-generic 0003:000F:0003.0001: input,hidraw0: USB HID v1.10 Mou\
se [VMware VMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0
Begin: Loading essential drivers ... done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
Begin: Running /scripts/local-premount ... done.
Begin: Mounting /usr ... done.
Begin: Will now check root file system ... fsck from util-linux 2.37.2
[/usr/sbin/fsck.ext4 (1) -- /dev/sda3] fsck.ext4 -a -C0 /dev/sda3
/dev/sda3: clean, 220066/1277952 files, 3254582/5110784 blocks
done.
[ 12.698403] EXT4-fs (sda3): mounted filesystem 9a6dfd40-a553-4dc1-a7c8-69b6a8\
d35ee1 r/w with ordered data mode. Quota mode: none.
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
root@(none):/#
root@(none):/# mount | grep -w /
/dev/sda3 on / type ext4 (rw,relatime)
root@(none):/#

```

6. At this point we are ready to reset the root password. To do so simply execute the **passwd** command and follow the instructions. In case you need to reset your user password, simply execute the **passwd** followed by your username.

```

Ubuntu 64-bit (2) - VMware Workstation 16 Player (Non-commercial use only)
Player
se [UMware UMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0
Begin: Loading essential drivers ... done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
Begin: Running /scripts/local-premount ... done.
Begin: Will now check root file system ... fsck from util-linux 2.37.2
[/usr/sbin/fsck.ext4 (1) -- /dev/sda3] fsck.ext4 -a -C0 /dev/sda3
/dev/sda3: clean, 220066/1277952 files, 3254582/5110704 blocks
done.
[ 12.698403] EXT4-fs (sda3): mounted filesystem 9a6dfd40-a553-4dc1-a7c8-69b6a8
d35ee1 r/w with ordered data mode. Quota mode: none.
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none):/#
root@(none):/# mount | grep -w /
/dev/sda3 on / type ext4 (rw,relatime)
root@(none):/# passwd
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: password updated successfully
root@(none):/#

```

7. All done. Your root password should be now recovered. All what remains is to reboot your Ubuntu 20.04 system. To do so execute the **exec /sbin/init**

```

Ubuntu 64-bit (2) - VMware Workstation 16 Player (Non-commercial use only)
Player
se [UMware UMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0
Begin: Loading essential drivers ... done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
Begin: Running /scripts/local-premount ... done.
Begin: Will now check root file system ... fsck from util-linux 2.37.2
[/usr/sbin/fsck.ext4 (1) -- /dev/sda3] fsck.ext4 -a -C0 /dev/sda3
/dev/sda3: clean, 220066/1277952 files, 3254582/5110784 blocks
done.
[ 12.698403] EXT4-fs (sda3): mounted filesystem 9a6dfd40-a553-4dc1-a7c8-69b6a8
d35ee1 r/w with ordered data mode. Quota mode: none.
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none):/#
root@(none):/# mount | grep -w /
/dev/sda3 on / type ext4 (rw,relatime)
root@(none):/# passwd
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: password updated successfully
root@(none):/# exec /sbin/init

```

Q3. User, Group, and Membership Tasks ○ Create:

- (a) A group named admin.
- (b) A user harry with admin as a secondary group.
- (c) A user natasha with admin as a secondary group.
- (d) A user sarah without an interactive shell and not in admin. (e) Set the password for all users to password.

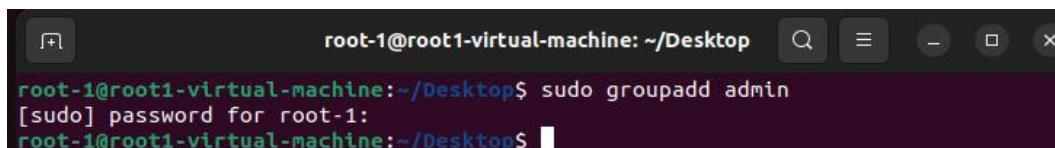
Answer: -

Here is how to complete the tasks step by step using commands in Linux:

(a) Create a Group Named admin

To create a group named admin, use the groupadd command:

```
sudo groupadd admin
```



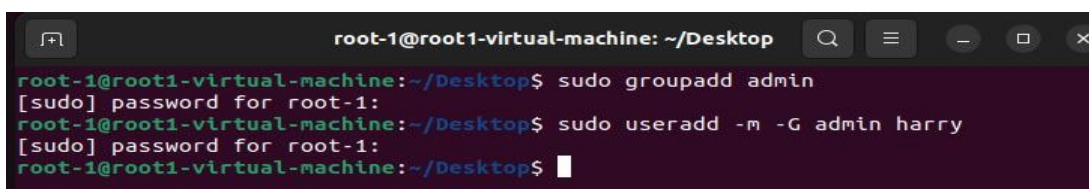
```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo groupadd admin
[sudo] password for root-1:
root-1@root1-virtual-machine:~/Desktop$
```

(b) Create User harry with admin as a Secondary Group

Add user harry and include them in the admin group:

```
sudo useradd -m -G admin harry
```

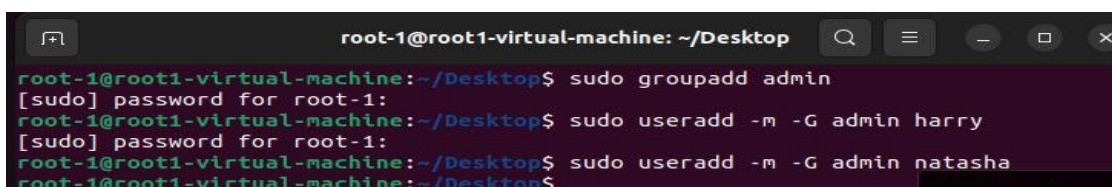
- -m: Creates a home directory for the user.
- -G admin: Adds harry to the admin group as a secondary group.



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo groupadd admin
[sudo] password for root-1:
root-1@root1-virtual-machine:~/Desktop$ sudo useradd -m -G admin harry
[sudo] password for root-1:
root-1@root1-virtual-machine:~/Desktop$
```

(c) Create User natasha with admin as a Secondary Group Similarly, add user natasha and include them in the admin group:

```
sudo useradd -m -G admin natasha
```

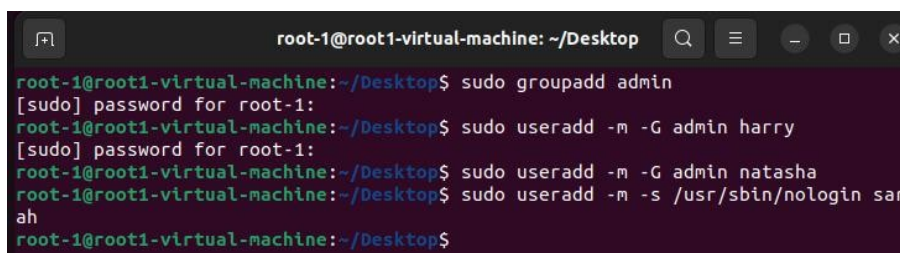


```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo groupadd admin
[sudo] password for root-1:
root-1@root1-virtual-machine:~/Desktop$ sudo useradd -m -G admin harry
[sudo] password for root-1:
root-1@root1-virtual-machine:~/Desktop$ sudo useradd -m -G admin natasha
root-1@root1-virtual-machine:~/Desktop$
```

(d) Create User sarah Without an Interactive Shell and Not in admin Add user sarah with a restricted shell (/usr/sbin/nologin), preventing login:

```
sudo useradd -m -s /usr/sbin/nologin sarah
```

- -s /usr/sbin/nologin: Sets the shell to nologin, disabling login.



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo groupadd admin
[sudo] password for root-1:
root-1@root1-virtual-machine:~/Desktop$ sudo useradd -m -G admin harry
[sudo] password for root-1:
root-1@root1-virtual-machine:~/Desktop$ sudo useradd -m -G admin natasha
root-1@root1-virtual-machine:~/Desktop$ sudo useradd -m -s /usr/sbin/nologin sarah
root-1@root1-virtual-machine:~/Desktop$
```

(e) Set Password for All Users to password

Set the same password (password) for all users using the passwd command.

For harry:

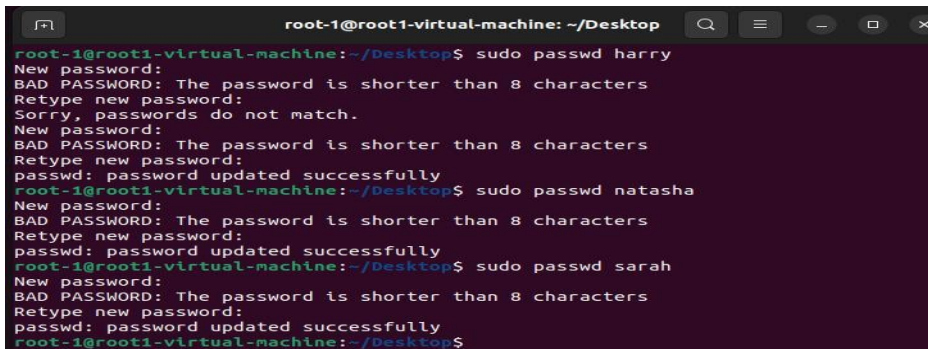
sudo passwd harry

For natasha:

sudo passwd natasha

For sarah:

sudo passwd sarah

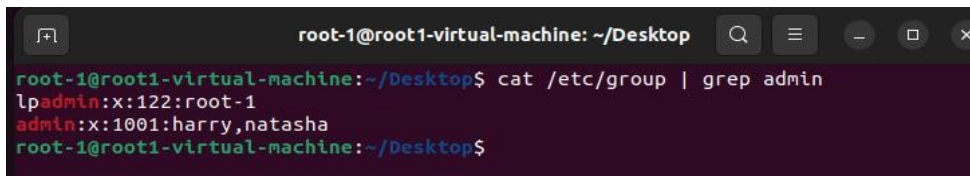


```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo passwd harry
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
root-1@root1-virtual-machine:~/Desktop$ sudo passwd natasha
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
root-1@root1-virtual-machine:~/Desktop$ sudo passwd sarah
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
root-1@root1-virtual-machine:~/Desktop$
```

Verification Commands

1. Verify the admin group exists:

cat /etc/group | grep admin

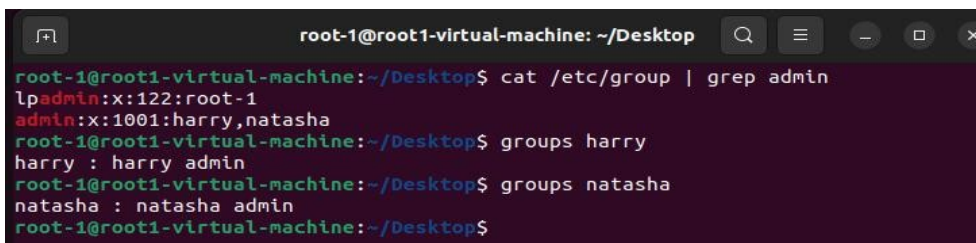


```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ cat /etc/group | grep admin
lpadmin:x:122:root-1
admin:x:1001:harry,natasha
root-1@root1-virtual-machine:~/Desktop$
```

2. Check group memberships for harry and natasha:

groups harry

groups natasha



```
root-1@root1-virtual-machine:~/Desktop$ cat /etc/group | grep admin
lpadmin:x:122:root-1
admin:x:1001:harry,natasha
root-1@root1-virtual-machine:~/Desktop$ groups harry
harry : harry admin
root-1@root1-virtual-machine:~/Desktop$ groups natasha
natasha : natasha admin
root-1@root1-virtual-machine:~/Desktop$
```

3. Verify sarah's shell is set to nologin:

cat /etc/passwd | grep sarah

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ cat /etc/passwd | grep sarah
sarah:x:1003:1004::/home/sarah:/usr/sbin/nologin
root-1@root1-virtual-machine:~/Desktop$
```

Q4. FTPServer Setup:

o Create an FTP server and access it using FileZilla, PuTTY, and MobaXterm.

Answer: -

Setting up an FTP server in Ubuntu and accessing it using **FileZilla**, **PuTTY**, and **MobaXterm** involves the following steps. Let's break it down for clarity:

Step 1: Install FTP Server (vsftpd)

1. Open the terminal in Ubuntu.
2. Install the **vsftpd** package:

```
sudo apt update
```

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [569 k
```

```
sudo apt install vsftpd
```

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt install vsftpd
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontent. It is he
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontent. It is he
ld by process 3965 (unattended-upgr)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontent. It is he
```

3. Check if the service is running:

```
sudo systemctl status vsftpd
```

```
-virtual-machine:~/Desktop$ sudo systemctl status vsftpd
ice - vsftpd FTP server
loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset=
active (running) since Tue 2024-12-10 00:58:02 IST; 5min ago
145483 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exit
145485 (vsftpd)
1 (limit: 2217)
860.0K
21ms
/system.slice/vsftpd.service
└─145485 /usr/sbin/vsftpd /etc/vsftpd.conf

02 mukesh-virtual-machine systemd[1]: Starting vsftpd FTP server...
02 mukesh-virtual-machine systemd[1]: Started vsftpd FTP server.
(END)
```

If it's not running, start it:

```
sudo systemctl start vsftpd
```

```
virtual-machine:~/Desktop$ sudo systemctl start vsftpd
virtual-machine:~/Desktop$
```

Step 2: Configure vsftpd

1. Open the vsftpd configuration file:

```
sudo nano /etc/vsftpd.conf
```

```
virtual-machine:~/Desktop$ sudo systemctl start vsftpd
virtual-machine:~/Desktop$ sudo nano /etc/vsftpd.conf
virtual-machine:~/Desktop$
```

2. Modify/add the following settings:

- o Enable local users to log in:
- o local_enable=YES
- o Enable write access:
- o write_enable=YES
- o Optional: Disable anonymous access for better security:
- o anonymous_enable=NO
- o Uncomment to allow local users to upload files:
- o chroot_local_user=YES

3. Save and close the file (Ctrl+O, Enter, Ctrl+X).

4. Restart the **vsftpd** service:

```
sudo systemctl restart vsftpd
```

```
virtual-machine:~/Desktop$ sudo systemctl start vsftpd
virtual-machine:~/Desktop$ sudo nano /etc/vsftpd.conf
virtual-machine:~/Desktop$ sudo systemctl restart vsftpd
virtual-machine:~/Desktop$
```

Step 3: Set Up a Local FTP User

1. Create a new user for FTP access:

```
sudo adduser ftpuser
```

Follow the prompts to set up the password.

2. Assign the home directory permissions (optional):

```
sudo chmod -R 755 /home/ftpuser
```

```
with serial 5
rManagementInhibitor: Request successful, cookie is 1137538397
rManagementInhibitor: Requesting idle
with serial 6
rManagementInhibitor: Request successful

virtual-machine:~/Desktop$ sudo chmod a-w /home/ftpuser
virtual-machine:~/Desktop$ sudo mkdir /home/ftpuser/uploads
user:ftpuser /home/ftpuser/uploads
virtual-machine:~/Desktop$ sudo systemctl restart vsftpd
virtual-machine:~/Desktop$ filezilla
option from /home/mukesh/.config/filezilla/filezilla.xml
l from /org/freedesktop/DBus, member NameAcquired
with serial 3
```

Step 4: Access FTP Server

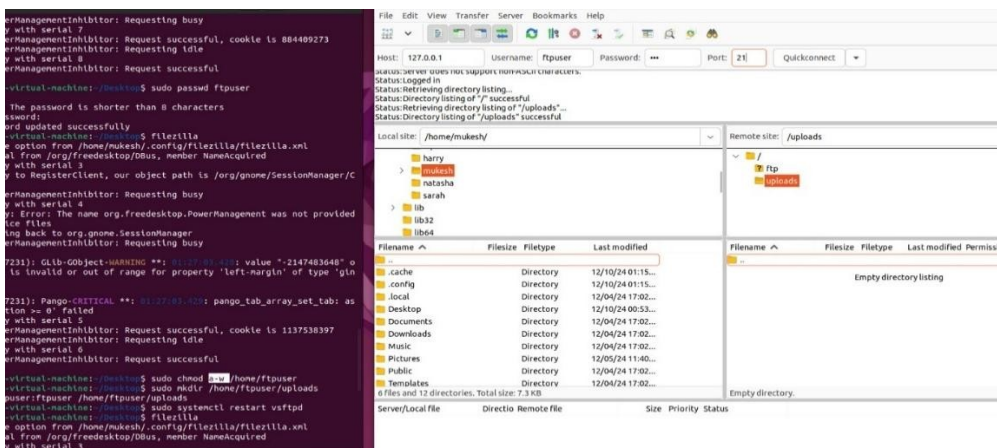
Using FileZilla

1. Install FileZilla:

```
sudo apt install filezilla
```

2. Open FileZilla and enter:

- o **Host:** ftp://127.0.0.1
- o **Username:** ftpuser
- o **Password:** 123
- o **Port:** 21



3. Click **Quickconnect** to access the server.

Using PuTTY

1. Open PuTTY.

2. Select the **FTP** protocol and specify the hostname/IP address.

3. Enter the username (ftpuuser) and password in the session.

Using MobaXterm

1. Open MobaXterm.
2. Create a new FTP session:
 - o Go to **Session > FTP**.
 - o Enter the server IP, port, username (ftpuuser), and password.
3. Connect to the FTP server and browse files.

Step 5: Test the Setup

1. Create a sample file on your local machine.
2. Upload it to the FTP server using FileZilla, PuTTY, or MobaXterm.
3. Check the file in the corresponding directory on the server:
4. `ls /home/ftpuuser`

Q5. Website Hosting:

- Host a website using httpd.

Answer: -

Hosting a website using **httpd** (Apache HTTP Server) on Ubuntu involves several steps. Here's a by-step guide to perform this practical:

Step 1: Install Apache HTTP Server

1. Open the terminal and update your package list:
`sudo apt update`
2. Install Apache:
`sudo apt install apache2`

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [356 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [5,316 B]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.7 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Fetched 739 kB in 6s (121 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
474 packages can be upgraded. Run 'apt list --upgradable' to see them.
root-1@root1-virtual-machine:~/Desktop$ sudo apt install apache2
Waiting for cache lock: could not get lock /var/lib/dpkg/lock-frontent. It is heWaiting for cache lock:
dpkg/lock-frontent, It is held by process 3965 (unattended-upgr)
Waiting for cache lock: could not get lock /var/lib/dpkg/lock-frontent. It is heWaiting for cache lock:
dpkg/lock-frontent, It is held by process 3965 (unattended-upgr)
Waiting for cache lock: could not get lock /var/lib/dpkg/lock-frontent. It is heWaiting for cache lock:
```

3. Confirm Apache is running:
`sudo systemctl status apache2`

```
ggers for libc-bin (2.35-0ubuntu3.8) ...
virtual-machine:~/Desktop$ sudo systemctl status apache2
Service apache2.service - The Apache HTTP Server
Loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
Active (running) since Tue 2024-12-10 01:54:57 IST; 27s ago
https://httpd.apache.org/docs/2.4/
148586 (apache2)
55 (limit: 2217)
5.9M
0.5ms
/system.slice/apache2.service
-148586 /usr/sbin/apache2 -k start
-148587 /usr/sbin/apache2 -k start
-148588 /usr/sbin/apache2 -k start
```

4. Enable Apache to start on boot:
`sudo systemctl enable apache2`

```
virtual-machine:~/Desktop$ sudo systemctl enable apache2
state of apache2.service with SysV service script with /lib/system
-install.
b/systemd/systemd-sysv-install enable apache2
virtual-machine:~/Desktop$
```

Step 2: Set Up Your Website

Option 1: Use the Default Web Directory

1. The default web directory for Apache is `/var/www/html`.
2. Navigate to this directory:
`cd /var/www/html`
3. Replace the default `index.html` with your website files:
 - o Remove the default file:
`sudo rm index.html`
 - o Create a new `index.html` file:
`sudo nano index.html`

```
virtual-machine:~/Desktop$ sudo systemctl enable apache2
state of apache2.service with SysV service script with /lib/system
-install.
b/systemd/systemd-sysv-install enable apache2
virtual-machine:~/Desktop$ cd /var/www/html
virtual-machine:/var/www/html$ sudo rm index.html
virtual-machine:/var/www/html$ sudo nano index.html
virtual-machine:/var/www/html$
```

Add the following example content:

```
<html>
<head><title>My Website</title></head>
<body><h1>Welcome to My Website!</h1></body>
```

</html>

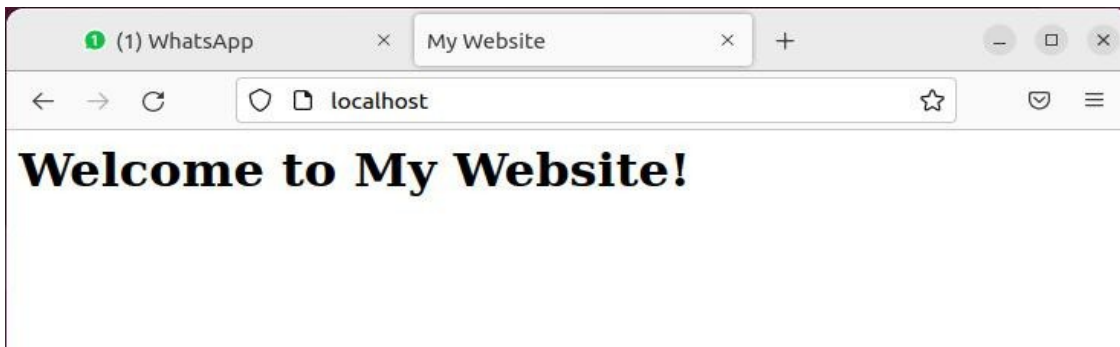


```
GNU nano 6.2 index.html
<html>
<head><title>My Website</title></head>
<body><h1>Welcome to My Website!</h1></body>
</html>
```

- o Save the file (Ctrl+O, Enter, Ctrl+X).

4. Test the setup by opening a browser and visiting:

<http://localhost>



You should see the "Welcome to My Website!" message.

Q6. Virtual Hosting

- o Configure virtual hosting for `web1.example.com` and `web2.example.com`.

Answer: -

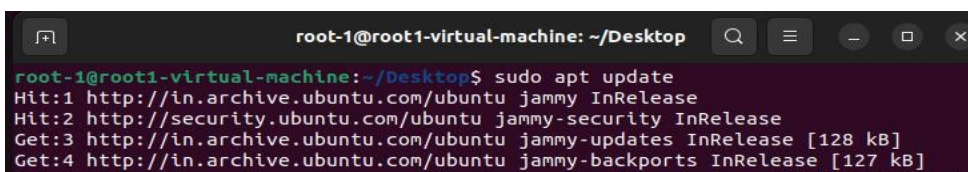
To configure virtual hosting for **web1.example.com** and **web2.example.com** on Ubuntu, you set up **Apache** as the web server and configure virtual hosts. Here's a step-by-step guide to this practical:

Step 1: Install Apache Web Server

1. Open the terminal and update your system:

```
sudo apt update
```

```
sudo apt upgrade
```



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine: ~/Desktop$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
```


2. Install the Apache2 package:

```
sudo apt install apache2
```

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [356 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [5,316 B]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.7 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Fetched 739 kB in 6s (121 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
474 packages can be upgraded. Run 'apt list --upgradable' to see them.
root-1@root1-virtual-machine:~/Desktop$ sudo apt install apache2
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 3965 (unattended-upgr)
dpkg/lock-frontend. It is held by process 3965 (unattended-upgr)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 3965 (unattended-upgr)
dpkg/lock-frontend. It is held by process 3965 (unattended-upgr)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 3965 (unattended-upgr)
dpkg/lock-frontend. It is held by process 3965 (unattended-upgr)
```

3. Enable and start Apache:

```
sudo systemctl enable apache2
```

```
sudo systemctl start apache2
```

4. Check Apache status to confirm it's running:

```
sudo systemctl status apache2
```

```
virtual-machine:~/Desktop$ sudo systemctl enable apache2
● start apache2
   state of apache2.service with SysV service script with /lib/systemd
   /install.
   lib/systemd/systemd-sysv-install enable apache2
virtual-machine:~/Desktop$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   active (running) since Tue 2024-12-10 01:54:57 IST; 33min ago
     https://httpd.apache.org/docs/2.4/
    148586 (apache2)
     65 (limit: 2217)
     1.7M
     33ms
   /system.slice/apache2.service
     —148586 /usr/sbin/apache2 -k start
     —148587 /usr/sbin/apache2 -k start
     —148588 /usr/sbin/apache2 -k start
17 mukesh-virtual-machine systemd[1]: Starting The Apache HTTP Ser
17 mukesh-virtual-machine apachectl[148584]: AH00558: apache2: Cou
17 mukesh-virtual-machine systemd[1]: Started The Apache HTTP Serv
(END)
```

Step 2: Create Document Root Directories for Virtual Hosts

1. Create directories for both websites (e.g., /var/www/web1.example.com and /var/www/web2.example.com):

```
sudo mkdir -p /var/www/web1.example.com
```

```
sudo mkdir -p /var/www/web2.example.com
```

2. Set proper permissions for the directories:

```
sudo chown -R www-data:www-data /var/www/web1.example.com
```

```
sudo chown -R www-data:www-data /var/www/web2.example.com
```

3. Create simple index.html files for testing:

```
echo "<h1>Welcome to Web1</h1>" | sudo tee /var/www/web1.example.com/index.html
```

```
echo "<h1>Welcome to Web2</h1>" | sudo tee /var/www/web2.example.com/index.html
```

```
virtual-machine:~/Desktop$ sudo mkdir -p /var/www/web1.example.com
/var/www/web2.example.com
virtual-machine:~/Desktop$ sudo chown -R www-data:www-data /var/www
www-data:www-data /var/www/web2.example.com
virtual-machine:~/Desktop$ echo "<h1>Welcome to Web1</h1>" | sudo tee
web1.example.com/index.html
ome to Web2</h1>" | sudo tee /var/www/web2.example.com/index.html
Web1</h1>
Web2</h1>
virtual-machine:~/Desktop$
```

Step 3: Configure Apache Virtual Hosts

1. Create virtual host configuration files for **web1.example.com** and **web2.example.com** in `/etc/apache2/sites-available/`.

For **web1.example.com**:

```
sudo nano /etc/apache2/sites-available/web1.example.com.conf
```

Add the following configuration:

```
<VirtualHost *:80>
    ServerAdmin webmaster@web1.example.com
    ServerName web1.example.com
    DocumentRoot /var/www/web1.example.com
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
GNU nano 6.2 /etc/apache2/sites-available/web1.example.com.conf
<VirtualHost *:80>
  ServerAdmin webmaster@web1.example.com
  ServerName web1.example.com
  DocumentRoot /var/www/web1.example.com
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

For **web2.example.com**:

```
sudo nano /etc/apache2/sites-available/web2.example.com.conf
```

Add the following configuration:

```
<VirtualHost *:80>
    ServerAdmin webmaster@web2.example.com
    ServerName web2.example.com
    DocumentRoot /var/www/web2.example.com
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
</VirtualHost>
```

Step 4: Enable the Sites and Restart Apache

1. Enable both virtual hosts:
sudo a2ensite web1.example.com.conf
sudo a2ensite web2.example.com.conf
2. Disable the default site (optional):
sudo a2dissite 000-default.conf
3. Reload Apache to apply the changes:
sudo systemctl reload apache2

```
virtual-machine:~/Desktop$ sudo a2ensite web1.example.com.conf
web2.example.com.conf
web1.example.com.
e new configuration, you need to run:
load apache2
web2.example.com.
e new configuration, you need to run:
load apache2
virtual-machine:~/Desktop$ sudo a2dissite 000-default.conf
lt disabled.
e new configuration, you need to run:
load apache2
virtual-machine:~/Desktop$ ^C
virtual-machine:~/Desktop$ sudo systemctl reload apache2
virtual-machine:~/Desktop$
```

Step 5: Edit /etc/hosts for Local Testing

1. Since you're configuring this locally, add the server names to your /etc/hosts file for test
sudo nano /etc/hosts
2. Add the following lines (replace 127.0.0.1 with your local IP if necessary):
127.0.0.1 web1.example.com
127.0.0.1 web2.example.com

```
GNU nano 6.2 /etc/hosts *
127.0.0.1 web1.example.com
127.0.0.1 web2.example.com

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

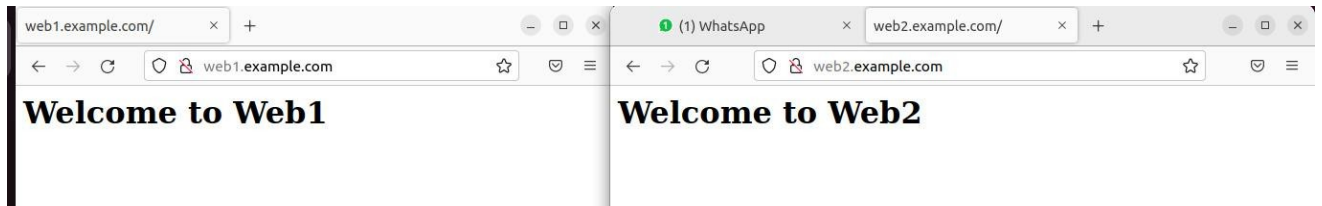
3. Save and close the file.

Step 6: Test the Virtual Hosts

1. Open a web browser on your Ubuntu machine and visit the URLs:

http://web1.example.com → Should display "Welcome to Web1".

http://web2.example.com → Should display "Welcome to Web2".



Q7. NFS Setup

- Create and access an NFS share.

Answer: -

To set up and access an NFS (Network File System) share in Ubuntu, follow these steps:

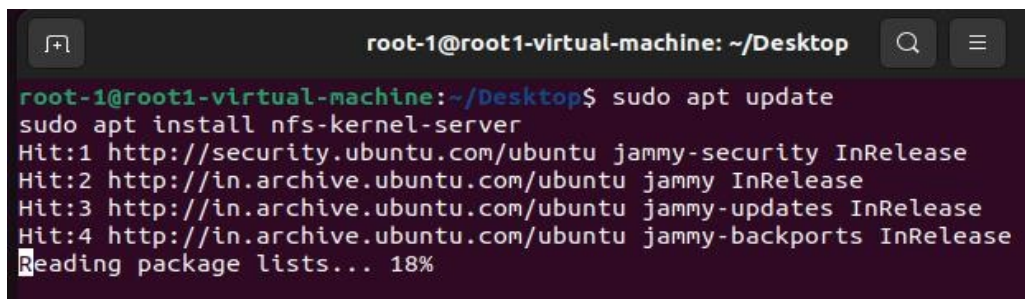
Step 1: Install NFS Server and Client

On the NFS Server:

1. Open the terminal and install the NFS server package:

```
sudo apt update
```

```
sudo apt install nfs-kernel-server
```

A terminal window screenshot showing the installation of nfs-kernel-server. The prompt is root-1@root1-virtual-machine: ~/Desktop. The commands entered are 'sudo apt update' and 'sudo apt install nfs-kernel-server'. The output shows four hits for security, jammy, updates, and backports, followed by 'Reading package lists... 18%'.

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt update
root-1@root1-virtual-machine:~/Desktop$ sudo apt install nfs-kernel-server
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... 18%
```

2. Enable and start the NFS server:

```
sudo systemctl enable nfs-server
```

```
sudo systemctl start nfs-server
```

On the NFS Client:

1. Open the terminal and install the NFS client package:

```
sudo apt update
```

```
sudo apt install nfs-common
```

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo systemctl enable nfs-server
sudo systemctl start nfs-server
root-1@root1-virtual-machine:~/Desktop$ sudo apt update
sudo apt install nfs-common
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
256 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-common is already the newest version (1:2.6.1-1ubuntu1.2).
nfs-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 256 not upgraded.
root-1@root1-virtual-machine:~/Desktop$
```

Step 2: Create a Directory to Share

1. Create a directory on the server that you want to share:
`sudo mkdir -p /mnt/nfs_share`
2. Set the appropriate permissions (optional, based on your use case):
`sudo chmod 777 /mnt/nfs_share`

```
root-1@root1-virtual-machine:~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo mkdir -p /mnt/nfs_share
root-1@root1-virtual-machine:~/Desktop$ sudo chmod 777 /mnt/nfs_share
root-1@root1-virtual-machine:~/Desktop$
```

Step 3: Configure the NFS Export

1. Open the NFS exports configuration file:
`sudo nano /etc/exports`
2. Add an entry for the directory you created. For example:
`/mnt/nfs_share *(rw,sync,no_subtree_check)`

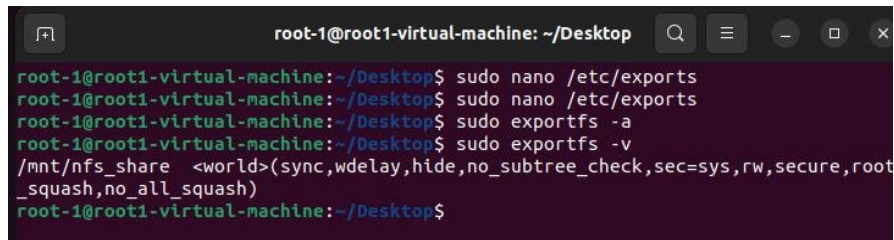
```
GNU nano 6.2 /etc/exports
/etc/exports: the access control list for filesystems which may be exported
to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/mnt/nfs_share *(rw,sync,no_subtree_check)
```

- o *: Allows all clients to connect (replace * with a specific client IP or subnet for more security).
 - o rw: Allows read-write access.
 - o sync: Ensures data is written synchronously to disk.
 - o no_subtree_check: Improves performance by not checking file permissions for subdirectories.
3. Save and exit the file (Ctrl+O, Enter, Ctrl+X).
 4. Apply the changes:

```
sudo exportfs -a
```

5. Verify the shared directory:

```
sudo exportfs -v
```

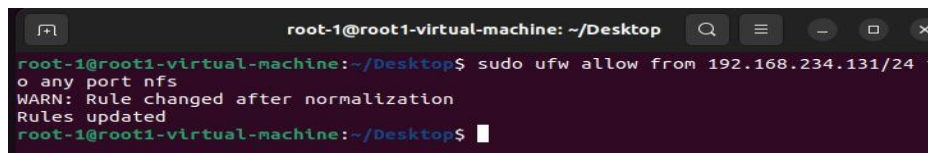


```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo nano /etc/exports
root-1@root1-virtual-machine:~/Desktop$ sudo nano /etc/exports
root-1@root1-virtual-machine:~/Desktop$ sudo exportfs -a
root-1@root1-virtual-machine:~/Desktop$ sudo exportfs -v
/mnt/nfs_share <world>(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
root-1@root1-virtual-machine:~/Desktop$
```

Step 4: Configure Firewall (Optional)

If you're using a firewall, allow NFS traffic:

```
sudo ufw allow from 192.168.234.131/24 to any port nfs
```



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo ufw allow from 192.168.234.131/24 to any port nfs
WARN: Rule changed after normalization
Rules updated
root-1@root1-virtual-machine:~/Desktop$
```

Step 5: Access the NFS Share from the Client

1. On the client machine, create a directory to mount the NFS share:

```
sudo mkdir -p /mnt/nfs_client_share
```

2. Mount the NFS share using the mount command:

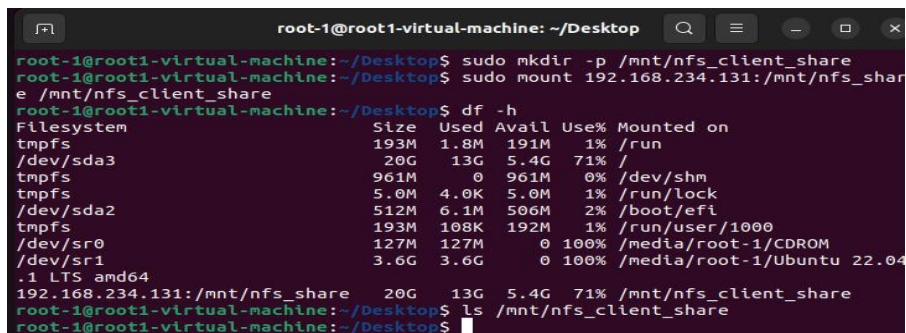
```
sudo mount 192.168.234.131:/mnt/nfs_share /mnt/nfs_client_share
```

Replace <server_IP> with the IP address of the NFS server.

3. Verify the mounted share:

```
df -h
```

```
ls /mnt/nfs_client_share
```



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo mkdir -p /mnt/nfs_client_share
root-1@root1-virtual-machine:~/Desktop$ sudo mount 192.168.234.131:/mnt/nfs_share /mnt/nfs_client_share
root-1@root1-virtual-machine:~/Desktop$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            193M  1.8M  191M   1% /run
/dev/sda3        20G   13G  5.4G  71% /
tmpfs            961M   0    961M   0% /dev/shm
tmpfs            5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2        512M  6.1M  506M   2% /boot/efi
tmpfs            193M  108K  192M   1% /run/user/1000
/dev/sr0         127M  127M   0 100% /media/root-1/CDROM
/dev/sr1         3.6G  3.6G   0 100% /media/root-1/Ubuntu 22.04
.1 LTS amd64
192.168.234.131:/mnt/nfs_share 20G  13G  5.4G  71% /mnt/nfs_client_share
root-1@root1-virtual-machine:~/Desktop$ ls /mnt/nfs_client_share
root-1@root1-virtual-machine:~/Desktop$
```

Step 6: Make the Mount Permanent (Optional)

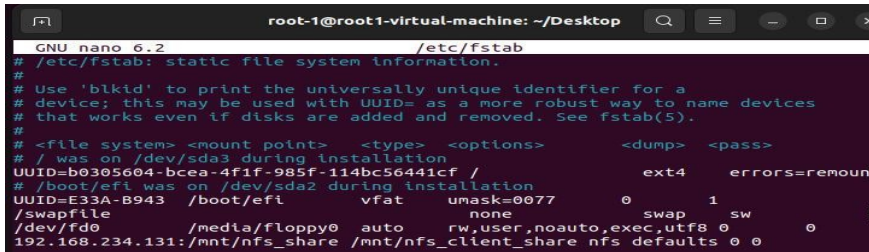
To ensure the NFS share is mounted automatically at boot, add an entry to the /etc/fstab file on the client:

1. Open the file:

```
sudo nano /etc/fstab
```

2. Add the following line:

```
192.168.234.131:/mnt/nfs_share /mnt/nfs_client_share nfs defaults 0 0
```

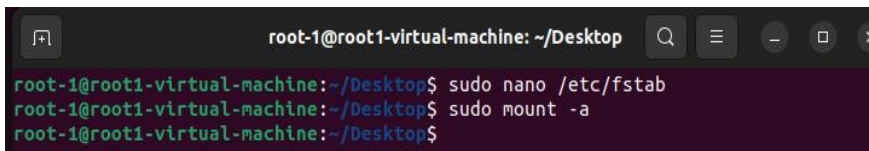


```
root-1@root1-virtual-machine: ~/Desktop
GNU nano 6.2 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda3 during installation
UUID=b0305604-bcea-4f1f-985f-114bc56441cf / ext4 errors=remount-ro
# /boot/efi was on /dev/sda2 during installation
UUID=E33A-B943 /boot/efi vfat umask=0077 0 swap 1 sw
/swapfile /swapfile none none 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0
192.168.234.131:/mnt/nfs_share /mnt/nfs_client_share nfs defaults 0 0
```

3. Save and exit the file (Ctrl+O, Enter, Ctrl+X).

4. Test the configuration:

```
sudo mount -a
```



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine: ~/Desktop$ sudo nano /etc/fstab
root-1@root1-virtual-machine: ~/Desktop$ sudo mount -a
root-1@root1-virtual-machine: ~/Desktop$
```

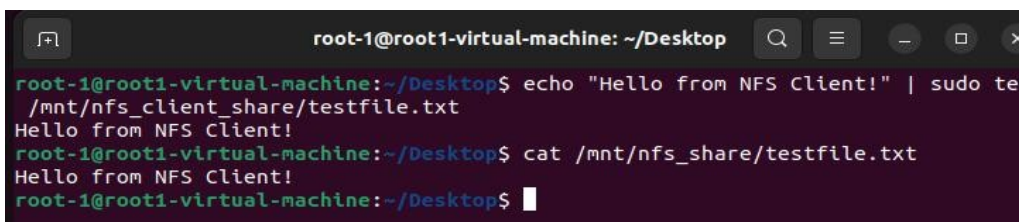
Testing the NFS Share

1. Create or edit files in the NFS share on the client:

```
echo "Hello from NFS Client!" | sudo tee /mnt/nfs_client_share/testfile.txt
```

2. Check the file on the server:

```
cat /mnt/nfs_share/testfile.txt
```



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine: ~/Desktop$ echo "Hello from NFS Client!" | sudo tee
/mnt/nfs_client_share/testfile.txt
Hello from NFS Client!
root-1@root1-virtual-machine: ~/Desktop$ cat /mnt/nfs_share/testfile.txt
Hello from NFS Client!
root-1@root1-virtual-machine: ~/Desktop$
```

Q8. MariaDB Setup

- Setup MariaDB server.

Answer: -

To set up a MariaDB server on Ubuntu, follow these steps:

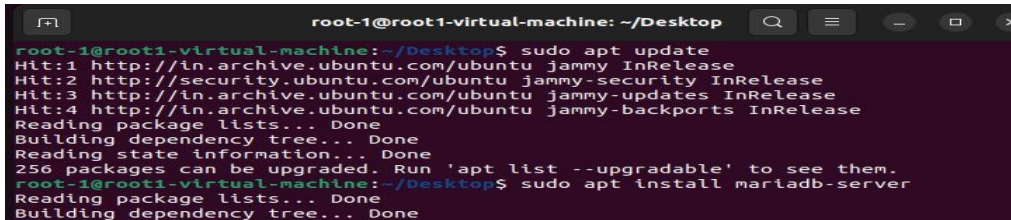
Step 1: Update the Package Repository

1. Open a terminal in Ubuntu.

2. Update the package repository: `sudo apt update`

Step 2: Install MariaDB Server

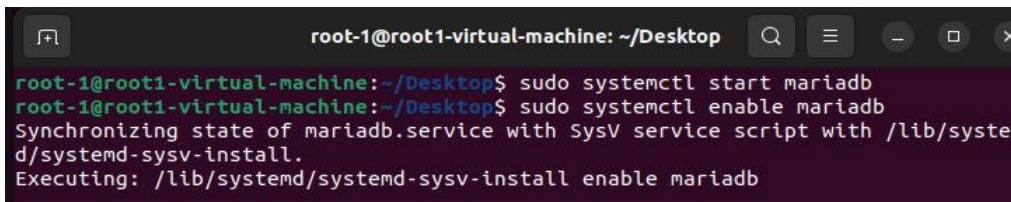
1. Install MariaDB: `sudo apt install mariadb-server`
2. Confirm the installation by typing Y when prompted.



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
256 packages can be upgraded. Run 'apt list --upgradable' to see them.
root-1@root1-virtual-machine:~/Desktop$ sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree... Done
```

Step 3: Start and Enable MariaDB Service

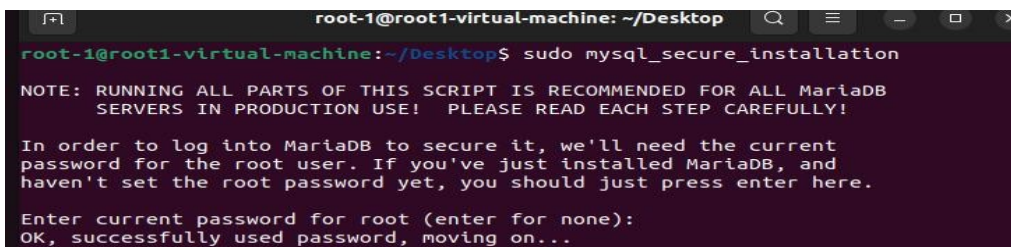
1. Start the MariaDB service: `sudo systemctl start mariadb`
2. Enable the service to start automatically on boot: `sudo systemctl enable mariadb`



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo systemctl start mariadb
root-1@root1-virtual-machine:~/Desktop$ sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mariadb
```

Step 4: Secure MariaDB Installation

1. Run the following script to secure your MariaDB installation: `sudo mysql_secure_installation`
2. Follow the prompts:
 - o Set the root password (if not already set).
 - o Remove anonymous users: Y
 - o Disallow root login remotely: Y
 - o Remove test database: Y
 - o Reload privilege tables: Y



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...
```

Step 5: Access MariaDB

1. Log in to MariaDB using the terminal: `sudo mysql -u root -p`
2. Enter the root password when prompted.


```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.6.18-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)]>
```

Step 6: Create a New Database and User (Optional)

1. Create a new database: `CREATE DATABASE my_database;`
2. Create a new user with a password: `CREATE USER 'my_user'@'localhost' IDENTIFIED BY 'my_password';`
3. Grant privileges to the user for the database: `GRANT ALL PRIVILEGES ON my_database.* TO 'my_user'@'localhost';`
4. Reload privileges: `FLUSH PRIVILEGES;`
5. Exit MariaDB: `EXIT;`

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.6.18-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 my_database;
Query OK, 1 row affected (0.010 sec)

MariaDB [(none)]> CREATE USER 'my_user'@'localhost' IDENTIFIED BY 'my_password';
Query OK, 0 rows affected (0.017 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON my_database.* TO 'my_user'@'localhost';
Query OK, 0 rows affected (0.007 sec)

MariaDB [(none)]>
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> EXIT;
Bye
root-1@root1-virtual-machine:~/Desktop$
```

Step 7: Test MariaDB Server

1. Verify MariaDB service is running: `sudo systemctl status mariadb`

```
root-1@root1-virtual-machine: ~/Desktop
● mariadb.service - MariaDB 10.6.18 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Tue 2024-12-10 12:17:27 IST; 9min ago
     Docs: man:mariadbd(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 134151 (mariadbd)
   Status: "Taking your SQL requests now..."
     Tasks: 9 (limit: 14633)
```

2. Test login with the new user: `mysql -u my_user -p`

```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.6.18-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)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| my_database |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.043 sec)

MariaDB [(none)]> USE my_database;
Database changed
```

MariaDB server is now set up.

Q9. CMS Hosting

- Host and optimize a CMS using WordPress.

Answer: -

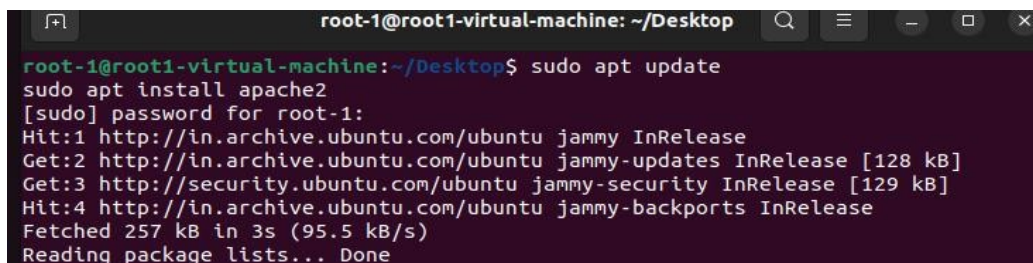
To host and optimize a **Content Management System (CMS)** using **WordPress** on **Ubuntu**, follow these steps. This includes installing the necessary software (Apache, MySQL, PHP), setting up WordPress, and configuring basic optimization.

Step 1: Install LAMP Stack

The **LAMP stack** includes **Linux**, **Apache**, **MySQL**, and **PHP**, which are the prerequisites for WordPress.

1. Install Apache (Web Server):

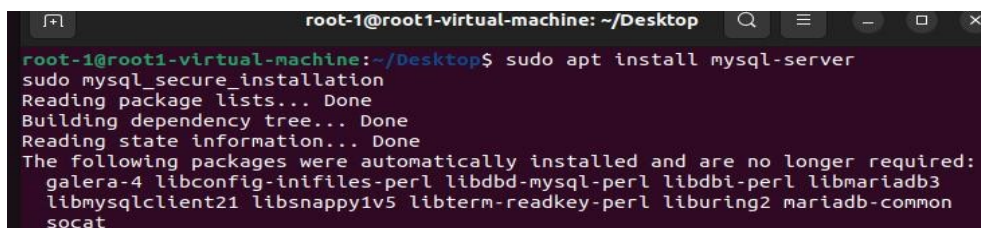
```
sudo apt update
sudo apt install apache2
```



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt update
sudo apt install apache2
[sudo] password for root-1:
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 257 kB in 3s (95.5 kB/s)
Reading package lists... Done
```

2. Install MySQL (Database Server):

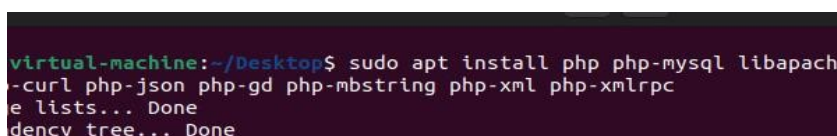
```
sudo apt install mysql-server
sudo mysql_secure_installation
```



```
root-1@root1-virtual-machine: ~/Desktop
root-1@root1-virtual-machine:~/Desktop$ sudo apt install mysql-server
sudo mysql_secure_installation
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
galera-4 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libmariadb3
libmysqlclient21 libsnappy1v5 libterm-readkey-perl liburing2 mariadb-common
socat
```

3. Install PHP (Programming Language): Install PHP and necessary PHP extensions for WordPress:

```
sudo apt install php php-mysql libapache2-mod-php php-curl php-json php-gd php-mbstring php-xml php-xmlrpc
```



```
virtual-machine:~/Desktop$ sudo apt install php php-mysql libapache2-mod-php php-curl php-json php-gd php-mbstring php-xml php-xmlrpc
Reading package lists... Done
Building dependency tree... Done
```

4. Restart Apache to apply changes: sudo systemctl restart apache2

```
-virtual-machine:~/Desktop$ sudo systemctl restart apache2
-virtual-machine:~/Desktop$
```

Step 2: Create MySQL Database for WordPress

1. Log in to MySQL: `sudo mysql -u root -p`
2. Create a new database and user for WordPress:

```
CREATE DATABASE wordpress;
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'your_password';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

```
-virtual-machine:~/Desktop$ sudo systemctl restart apache2
-virtual-machine:~/Desktop$ sudo mysql -u root -p
mysql>
mysql: [Warning] Using a password on the command line to connect is insecure.
mysql> CREATE DATABASE wordpress;
mysql> CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'your_password';
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';
mysql> FLUSH PRIVILEGES;
mysql> EXIT;
```

Step 3: Download and Install WordPress

1. Change to the web server's root directory: `cd /var/www/html`
2. Download WordPress: `sudo wget https://wordpress.org/latest.tar.gz`
3. Extract the WordPress files: `sudo tar -xzf latest.tar.gz`

```
-virtual-machine:~/Desktop$ cd /var/www/html
-virtual-machine:/var/www/html$ sudo wget https://wordpress.org/latest.tar.gz
14:32:12-- https://wordpress.org/latest.tar.gz
wordpress.org (wordpress.org)... 198.143.164.252
wordpress.org (wordpress.org)[198.143.164.252]:443... connected.
sent, awaiting response... 200 OK
653 (26M) [application/octet-stream]
atest.tar.gz'

   36%[=====>                ] 9.47M  129KB/s   in 80s
14:33:35 (122 KB/s) - Read error at byte 9932505/26931653 (Connection reset by peer)
14:33:36-- (try: 2) https://wordpress.org/latest.tar.gz
wordpress.org (wordpress.org)[198.143.164.252]:443... connected.
sent, awaiting response... 206 Partial Content
653 (26M), 16999148 (16M) remaining [application/octet-stream]
atest.tar.gz'

  100%[++++++=====] 25.68M  90.8KB/s   in 99s
14:35:16 (167 KB/s) - 'latest.tar.gz' saved [26931653/26931653]
-virtual-machine:/var/www/html$ sudo tar -xzf latest.tar.gz
rpc.php
blog-header.php
dme.html
signup.php
```

4. Move the WordPress files into the current directory: `sudo mv wordpress/* /var/www/html/`

5. Set correct permissions:

```
sudo chown -R www-data:www-data /var/www/html/
```

```
sudo chmod -R 755 /var/www/html/
```

```
trackback.php
comments-post.php
-virtual-machine:/var/www/html$ sudo mv wordpress/* /var/www/html/
-virtual-machine:/var/www/html$ sudo chown -R www-data:www-data /va
755 /var/www/html/
-virtual-machine:/var/www/html$
```

Step 4: Configure WordPress

1. Navigate to the WordPress directory: `cd /var/www/html`

2. Copy the sample configuration file to create the actual configuration file: `sudo cp wp-config-sample.php wp-config.php`

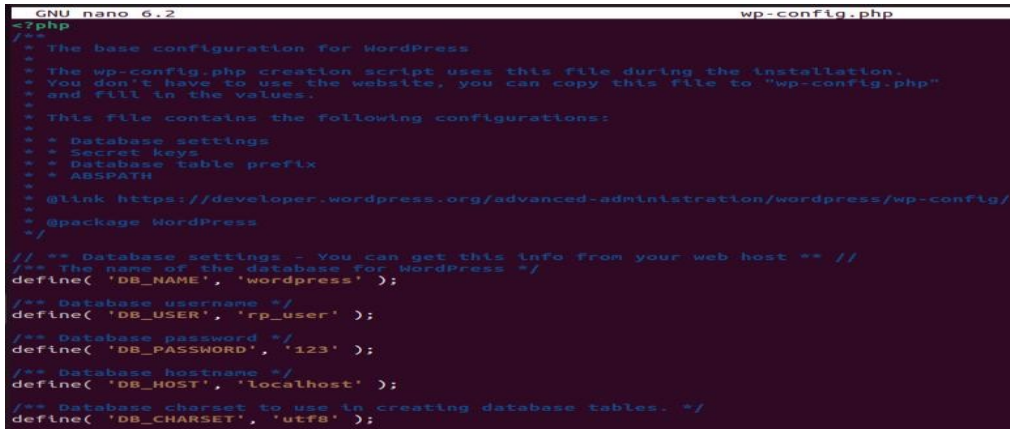
```
virtual-machine:/var/www/html$ cd /var/www/html
virtual-machine:/var/www/html$ sudo cp wp-config-sample.php wp-con
virtual-machine:/var/www/html$ sudo nano wp-config.php
virtual-machine:/var/www/html$
```

3. Edit the `wp-config.php` file to add the database details: `sudo nano wp-config.php`

Modify the following lines:

```
define('DB_NAME', 'wordpress');  
define('DB_USER', 'wp_user');  
define('DB_PASSWORD', '123');  
define('DB_HOST', 'localhost');
```

4. Save the file (Ctrl+O, Enter, Ctrl+X).



```
GNU nano 6.2 wp-config.php  
?php  
/**  
 * The base configuration for WordPress  
 *  
 * The wp-config.php creation script uses this file during the installation.  
 * You don't have to use the website, you can copy this file to "wp-config.php"  
 * and fill in the values.  
 *  
 * This file contains the following configurations:  
 *  
 * Database settings  
 * Secret keys  
 * Database table prefix  
 * ABSPATH  
 *  
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/  
 *  
 * @package WordPress  
 */  
  
/** ** Database settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define( 'DB_NAME', 'wordpress' );  
  
/** Database username */  
define( 'DB_USER', 'wp_user' );  
  
/** Database password */  
define( 'DB_PASSWORD', '123' );  
  
/** Database hostname */  
define( 'DB_HOST', 'localhost' );  
  
/** Database charset to use in creating database tables. */  
define( 'DB_CHARSET', 'utf8' );
```

Step 5: Set Up WordPress via Web Browser

1. Open your web browser and go to `http://<your_server_ip>` (e.g., `http://192.168.1.10`).
2. You should see the WordPress installation page. Select your language and proceed.
3. On the next screen, you'll be prompted to enter information such as:
 - o **Site Title:** Your website's title.
 - o **Username:** Administrator username.
 - o **Password:** Administrator password.
 - o **Email:** Admin email address.
4. Complete the setup by clicking **Install WordPress**.

Step 6: Optimize WordPress for Performance

1. Enable Caching

Install a caching plugin like **W3 Total Cache** or **WP Super Cache** via the WordPress admin panel.

- Log in to the WordPress dashboard.
- Go to **Plugins > Add New**.
- Search for **W3 Total Cache** or **WP Super Cache** and click **Install Now**.
- After installation, activate and configure the plugin for optimal performance.

2. Install and configure an SSL Certificate

If you're running the website publicly, using SSL is essential:

- Install **Certbot**: `sudo apt install certbot python3-certbot-apache`
- Get an SSL certificate: `sudo certbot --apache`
- Follow the prompts to configure HTTPS for your website.

3. Enable GZIP Compression

You can enable GZIP compression for faster load times by adding the following to your Apache configuration: `sudo nano /etc/apache2/mods-enabled/deflate.conf`

Add the following lines: `SetOutputFilter DEFLATE`

`AddOutputFilterByType DEFLATE text/plain text/html text/xml text/css application/x-javascript application/javascript`

Then restart Apache: `sudo systemctl restart apache2`

4. Optimize Images

Use a plugin like **Smush** to optimize images:

- Go to **Plugins > Add New**.
- Search for **Smush** and install it.
- Activate and follow the instructions for image optimization.

5. Use a Content Delivery Network (CDN)

Set up a CDN (e.g., Cloudflare) to serve static files (images, CSS, JS) from multiple locations worldwide, improving website speed.

6. Database Optimization

- Install the **WP-Optimize** plugin for regular database cleanup and optimization.
- Alternatively, you can optimize the database manually through phpMyAdmin or MySQL commands.

Step 7: Verify and Test Your WordPress Site

1. Access the WordPress dashboard at `http://<your_server_ip>/wp-admin` using the admin username and password you set during installation.
2. Test the website's performance using tools like **GTmetrix** or **Google PageSpeed Insights** to check for any further optimizations.

Optional: Secure WordPress

1. **Disable XML-RPC** (if not needed): Edit the `wp-config.php` file to disable XML-RPC for additional security: `define('XMLRPC_REQUEST', false);`
2. **Install a Security Plugin**: Use plugins like **Wordfence Security** or **iThemes Security** to improve WordPress security.

Q10. Network Connection Setup

- Create a network connection named mylab of type Ethernet with the following:
 - IP:192.168.45.155/24
 - Gateway: 192.168.45.1
 - DNS:8.8.8.8, 192.168.45.1

Answer: -

To set up a network connection named mylab with the specified settings in Ubuntu, steps:

Using the Graphical User Interface (GUI)

1. Open Network Settings:

- Click on the network icon in the top-right corner of the screen.
- Select Settings or Network Settings.

2. Add a New Connection:

- In the Network section, click on the + button (Add New Connection).
- Choose Wired (for Ethernet).

3. Configure the Connection:

- Name the connection mylab in the Connection Name field.
- Go to the IPv4 tab.
 - Select Manual.
 - Enter the following details:
 - Address: 192.168.45.155
 - Netmask: 255.255.255.0 (equivalent to /24 CIDR).
 - Gateway: 192.168.45.1
 - DNS: 8.8.8.8, 192.168.45.1.
- Save the settings and activate the connection.

Using the Command Line (CLI)

1. **Open Terminal:** Press Ctrl + Alt + T to open the terminal.
2. **Create the Connection:** Use the nmcli (Network Manager Command Line Interface) command to configure the network:

```
sudo nmcli connection add type ethernet con-name mylab ifname eth0 ipv4.addresses 192.168.45.155/24 ipv4.gateway 192.168.45.1 ipv4.dns "8.8.8.8,192.168.45.1" ipv4.method manual
```

Replace eth0 with your actual Ethernet interface name. To find your interface name, run: ip link

```

-virtual-machine:/var/www/html$ sudo nmcli connection add type ethernet
mylab ifname eth0 ipv4.addresses 192.168.45.155/24 ipv4.gateway 192.168.45.1
ipv4.dns "8.8.8.8,192.168.45.1" ipv4.method manual
Connection 'mylab' (65ce0d15-a354-41fd-9eb4-720a6a45949a) successfully added.
-virtual-machine:/var/www/html$ ip link
LINK: ens33: <ens33: UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
device enp2s1
link/ether 00:0c:29:fa:fd:f0 brd ff:ff:ff:ff:ff:ff

```

3. **Activate the Connection:** sudo nmcli connection up mylab

```

-virtual-machine:/var/www/html$ sudo nmcli connection modify mylab
interface-name ens33
-virtual-machine:/var/www/html$ sudo nmcli connection up mylab
Connection 'mylab' (65ce0d15-a354-41fd-9eb4-720a6a45949a) successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/Connection/2)
-virtual-machine:/var/www/html$ nmcli connection show mylab
mylab
id: 65ce0d15-a354-41fd-9eb4-720a6a45949a
type: ethernet
device: ens33
state: up
interface-name: ens33
autoconnect: yes

```

4. **Verify the Connection:** Check if the network settings are applied correctly:

nmcli connection show mylab

```

-virtual-machine:/var/www/html$ sudo nmcli connection up mylab
Connection 'mylab' (65ce0d15-a354-41fd-9eb4-720a6a45949a) successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/Connection/3)
-virtual-machine:/var/www/html$ nmcli connection show mylab
mylab
id: 65ce0d15-a354-41fd-9eb4-720a6a45949a
type: ethernet
device: ens33
state: up
interface-name: ens33
autoconnect: yes
autoconnect-priority: 0
autoconnect-retries: -1 (default)
delayed-connect: 0 (default)

```

ip addr show ens33

```

-virtual-machine:/var/www/html$ ip addr show ens33
LINK: ens33: <ens33: UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
device enp2s1
link/ether 00:0c:29:fa:fd:f0 brd ff:ff:ff:ff:ff:ff
inet 192.168.45.155/24 brd 192.168.45.255 scope global noprefixroute ens33
    proto kernel scope link src 192.168.45.155
inet6 fe80::8c9a:710:8c4d:7200/64 scope link noprefixroute
    proto kernel scope link src fe80::8c9a:710:8c4d:7200
-virtual-machine:/var/www/html$

```


Verify Network Connectivity

1. Ping the gateway to ensure the network is working:

```
ping -c 4 192.168.45.1
```

```
-virtual-machine:~/Desktop$ ping -c 4 192.168.45.1
45.1 (192.168.45.1) 56(84) bytes of data.
45.155 icmp_seq=1 Destination Host Unreachable
45.155 icmp_seq=2 Destination Host Unreachable
45.155 icmp_seq=3 Destination Host Unreachable
45.155 icmp_seq=4 Destination Host Unreachable

5.1 ping statistics ---
nsmitted, 0 received, +4 errors, 100% packet loss, time 3085ms
-virtual-machine:~/Desktop$
```

2. Test DNS resolution by pinging a domain:

```
ping -c 4 google.com
```

```
-virtual-machine:~/Desktop$ ping -c 4 google.com
com: Temporary failure in name resolution
-virtual-machine:~/Desktop$
```

This should set up the mylab Ethernet connection successfully on your Ubuntu system.