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

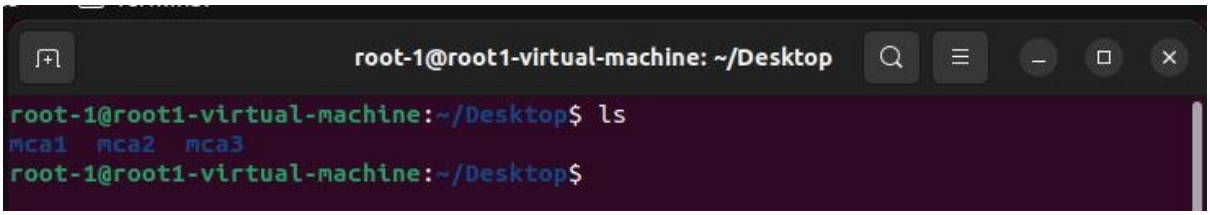
Q1. Command Usages

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

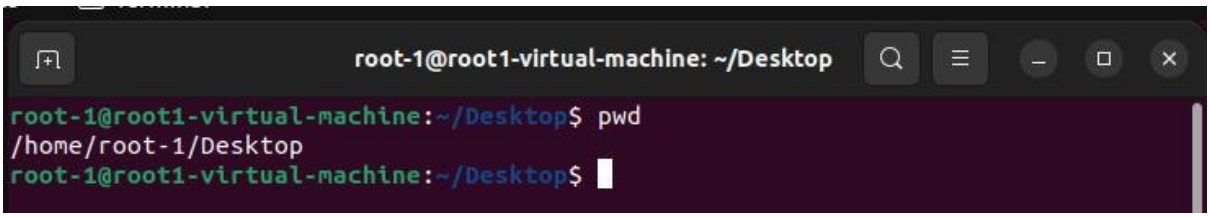
Answer: -

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

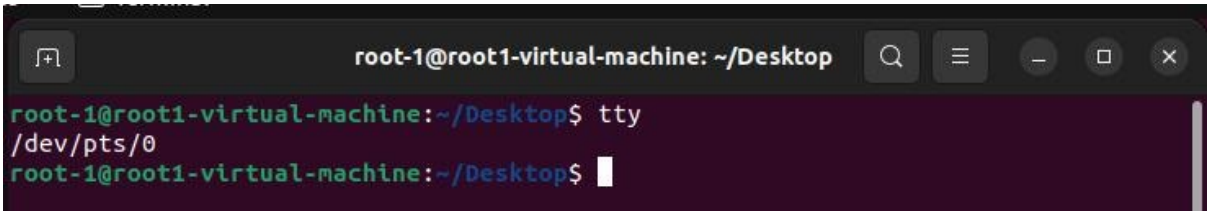
1. **ls**:
   Lists the files and directories in the current directory.
   Example: ls (lists files), ls -l (lists in long format).



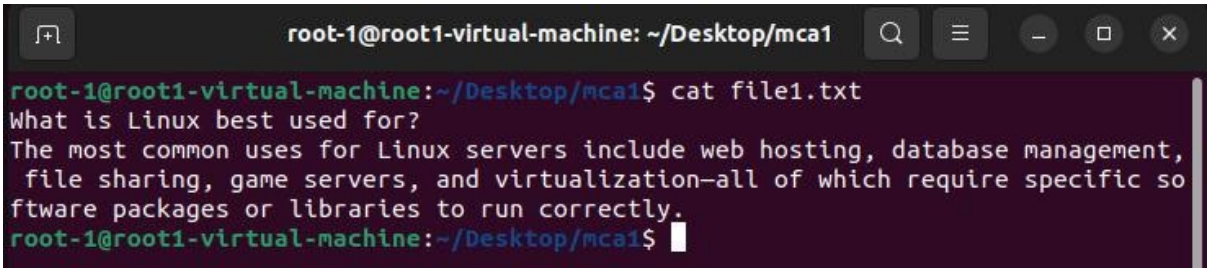
2. **pwd**:
   Prints the current working directory.
   Example: pwd (e.g., /home/user).



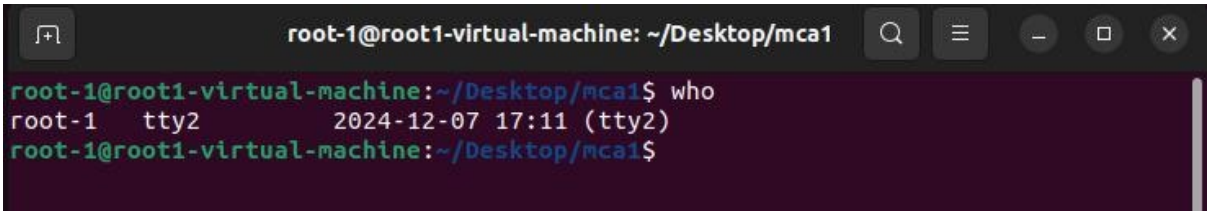
3. **tty**:
   Displays the file name of the terminal connected to the session.
   Example: tty (e.g., /dev/tty1).



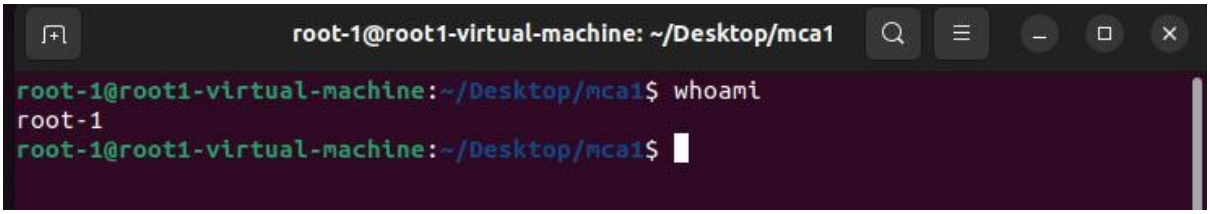
4. **cat**:
   Concatenates and displays the contents of files.
   Example: cat file.txt (displays contents of file.txt).



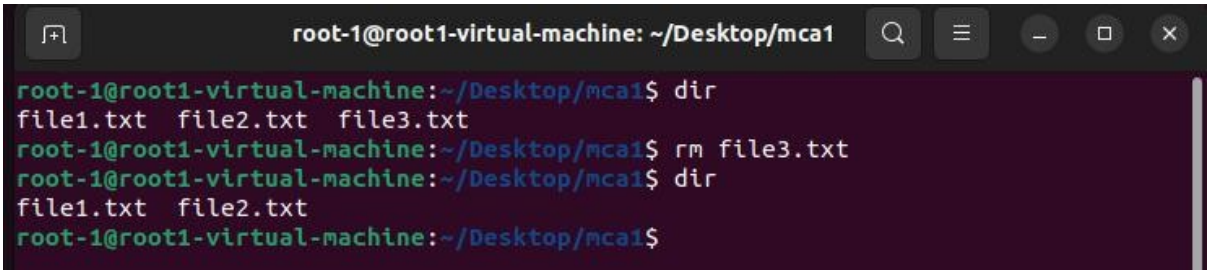
5. **who**:
   Displays who is logged in to the system.
   Example: who (lists usernames, terminal, and login time).



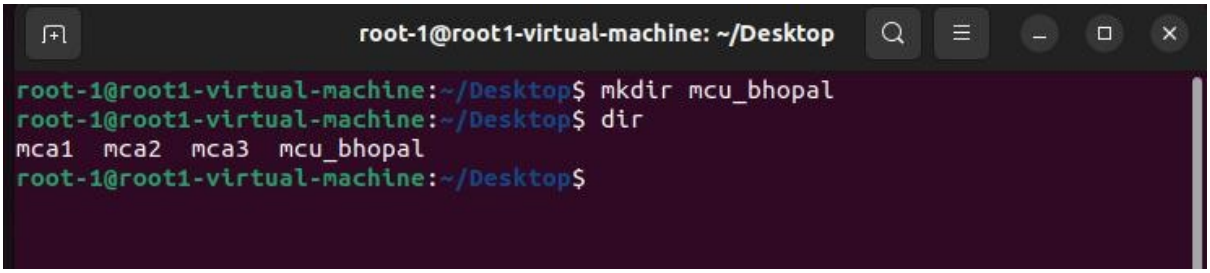
6. **whoami**:
   Displays the current logged-in user.
   Example: whoami (e.g., user).



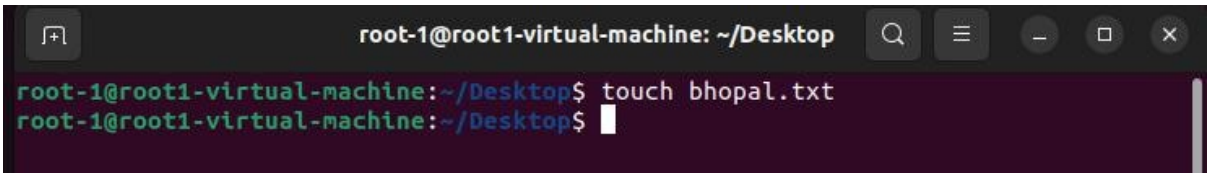
7. **rm**:
   Removes files or directories.
   Example: rm file.txt (removes file.txt), rm -r dir/ (removes directory recursively).



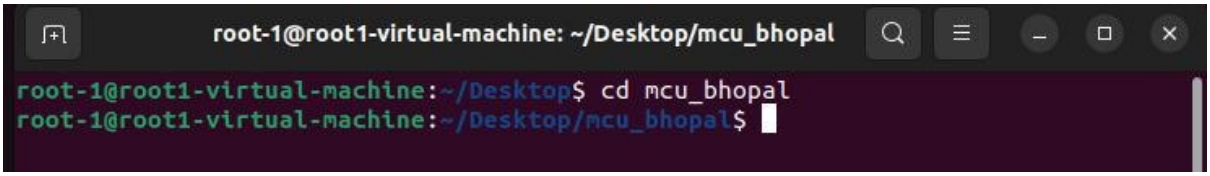
8. **mkdir**:
   Creates a new directory.
   Example: mkdir new_directory (creates new_directory).



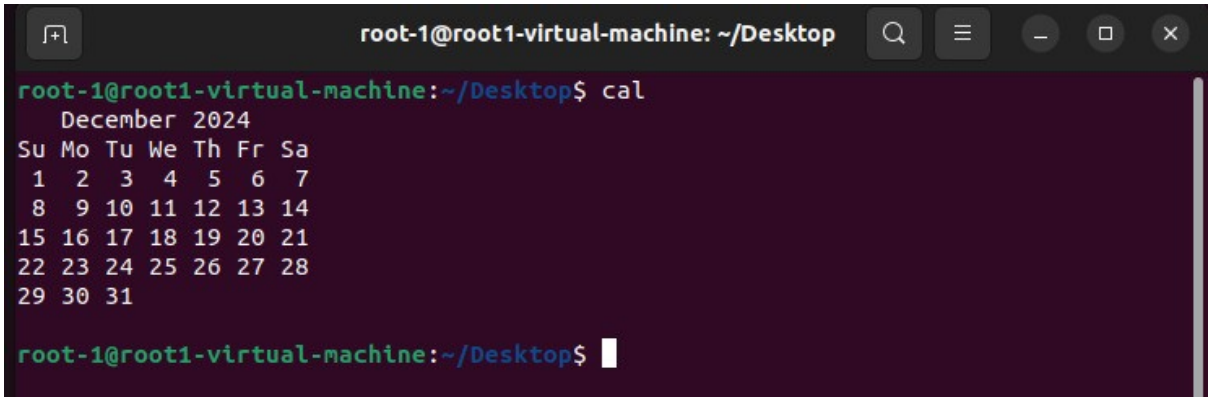
9. **touch**:
   Creates an empty file or updates the timestamp of an existing file.
   Example: touch newfile.txt (creates an empty file named newfile.txt).



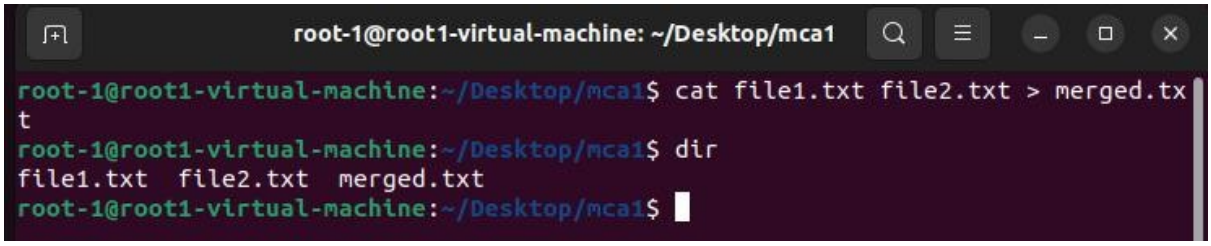
10. **cd**:
    Changes the current directory.
    Example: cd /home/user (moves to /home/user directory).



11. **cal**:
    Displays a calendar.
    Example: cal (shows the current month), cal 2024 (shows calendar for the year 2024).

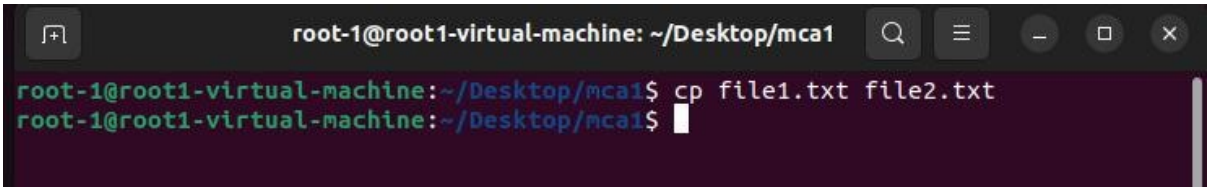


12. **cat (append & concatenate)**:
    - **Concatenate**: Combines multiple files into one.
    - **Append**: Appends content to an existing file.
      Example: cat file1.txt file2.txt > merged.txt (concatenates files into merged.txt), >> file.txt (appends text to file.txt).

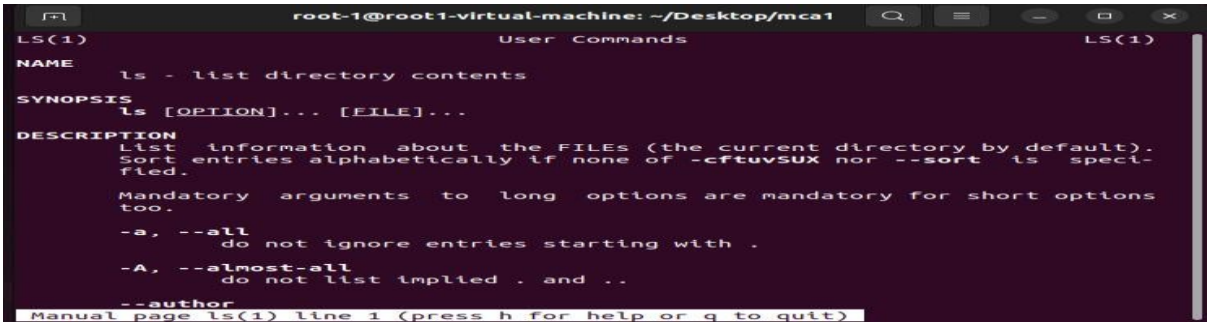


13. **mv**:
    Moves or renames files or directories.
    Example: mv oldname.txt newname.txt (renames file), mv file.txt /home/user/ (moves fi
    another directory).

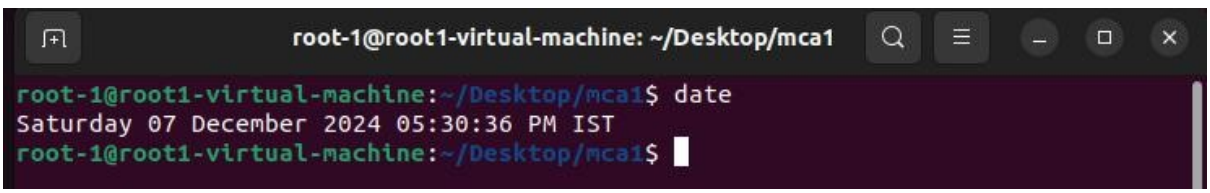
14. **cp**:
    Copies files or directories.
    Example: cp file.txt copy.txt (copies file.txt to copy.txt).



15. **man**:
    Displays the manual or help documentation for a command.
    Example: man ls (shows the manual page for ls).



16. **date**:
    Displays or sets the current date and time.
    Example: (formats the date output).



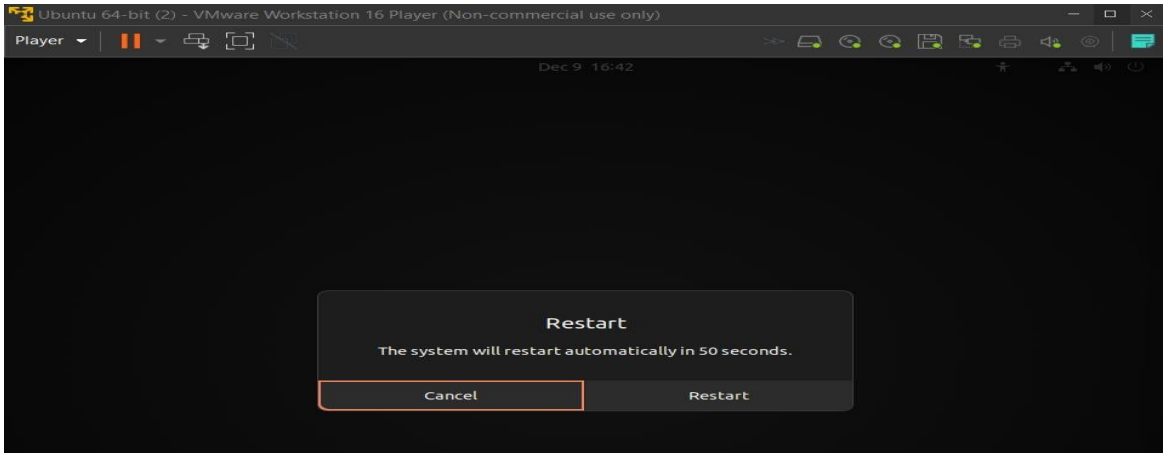
Q2. Root Password Recovery

- Explain the steps to reset the root password.

Answer: -

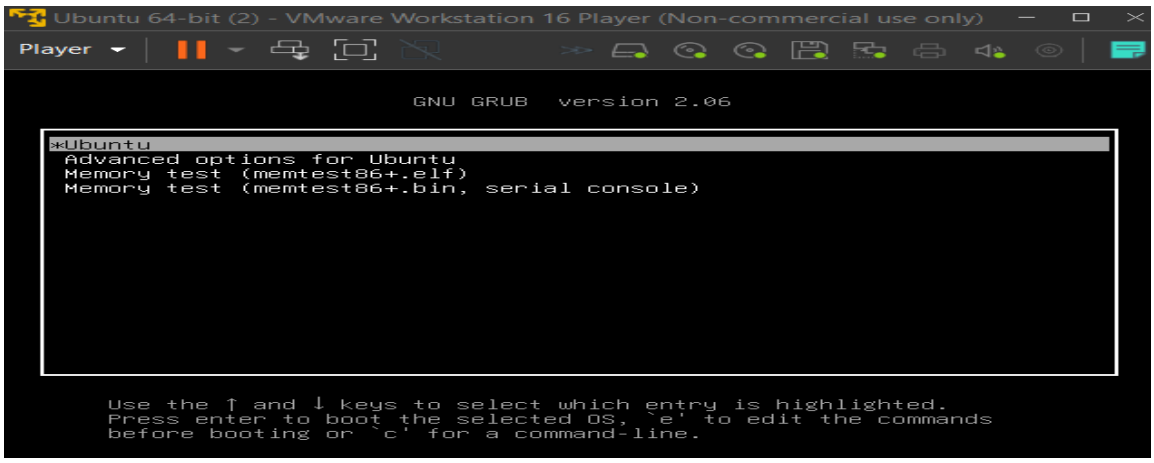
Steps to Recover Root Password in Ubuntu

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

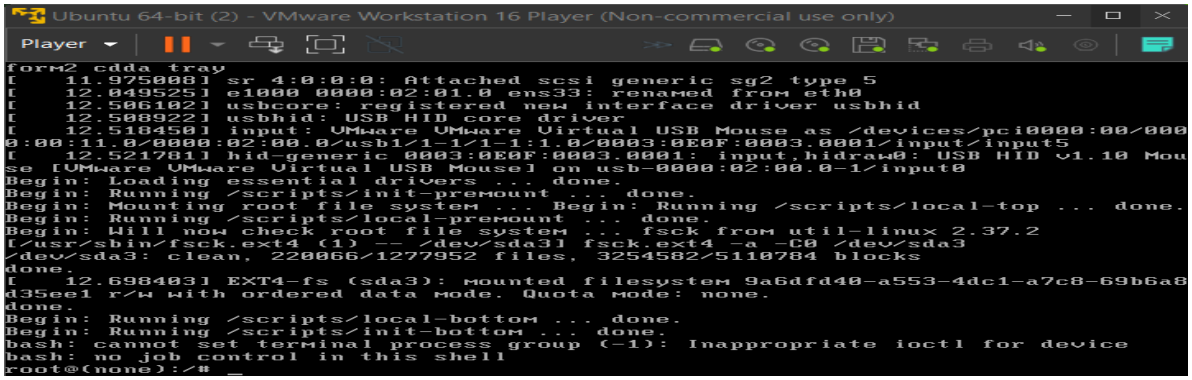
    The first step is to reboot into the GRUB menu. If Ubuntu 20.04 is the only operating sys installed you need to keep pressing **SHIFT** *for the GRUB menu to show up. Next, while your* **Ubuntu** *boot menu is highlighted press* **e** *to edit the Grub's boot prompt.*



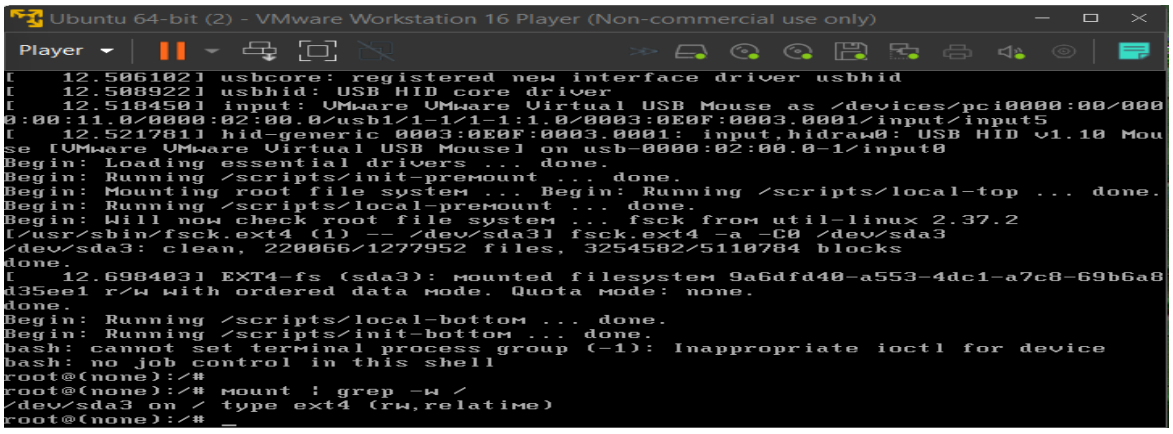
3. **Modify the boot commands:**

    Using your navigational arrows locate the line containing the following string **ro quiet s $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.

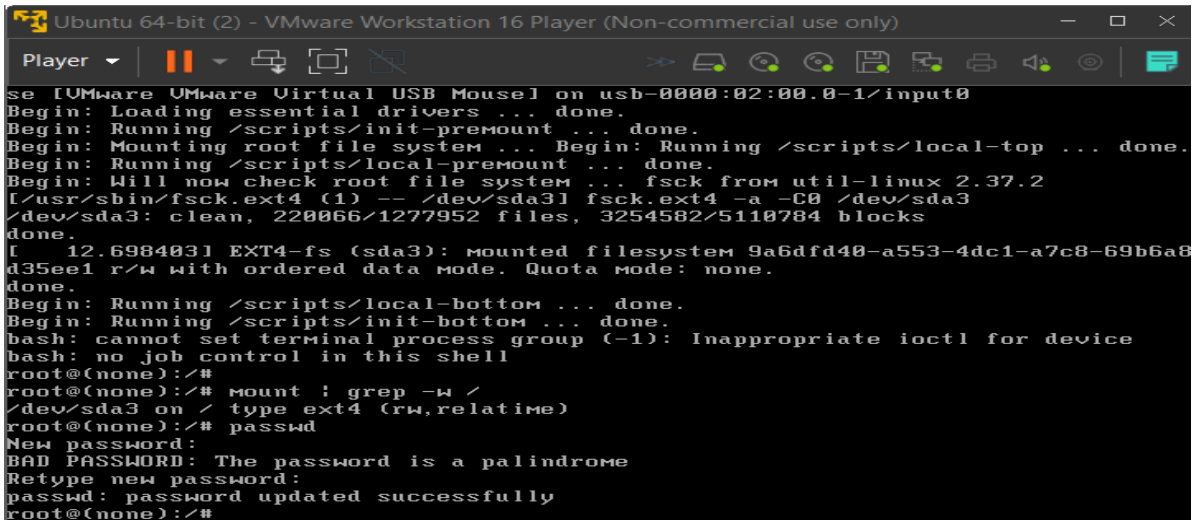
4. After the successful boot you should be welcomed with a root's shell without a need to enter the root password:



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

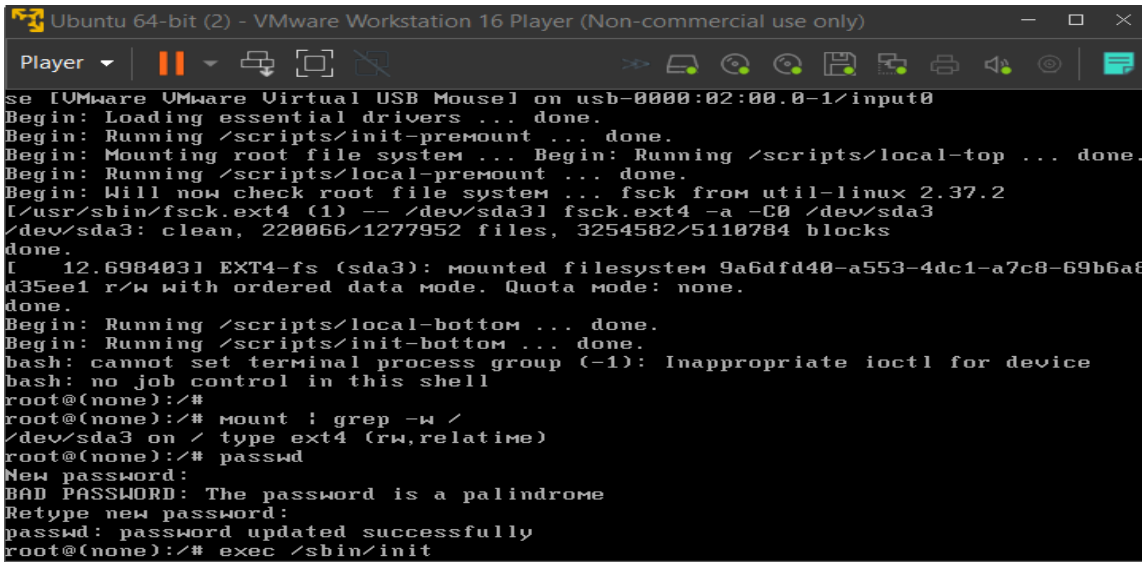


6. At this point we are ready to reset the root password. To do so simply execute the **passwd** command and follow the instructions. In case you need to reset your user password, simply execute the **passwd** followed by your username.



7. All done. Your root password should be now recovered. All what remains is to reboot you

   Ubuntu 20.04 system. To do so execute the **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.

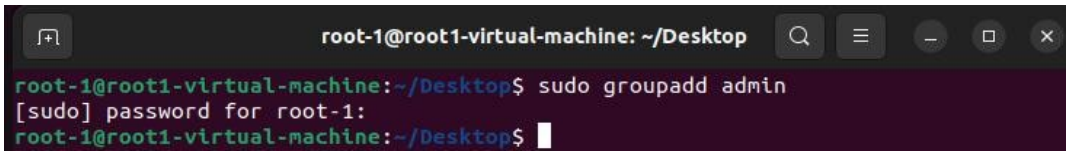
Answer: -

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

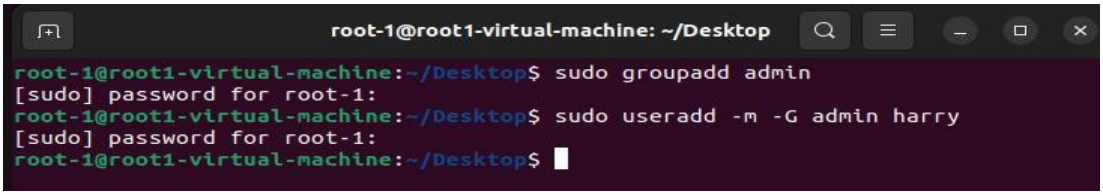


(b) Create User harry with admin as a Secondary Group

Add user harry and include them in the admin group:

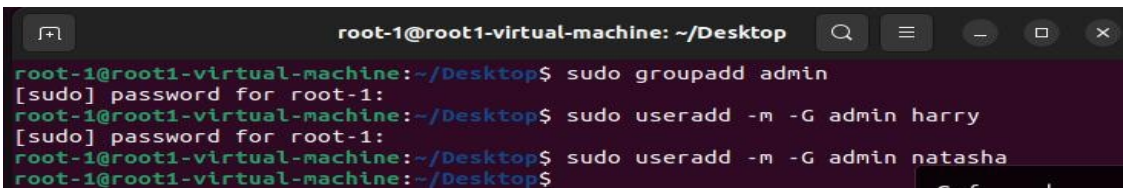
sudo useradd -m -G admin harry

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



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

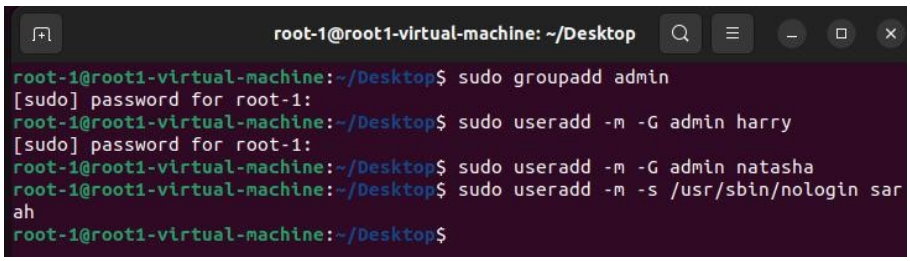
sudo useradd -m -G admin natasha



(d) Create User sarah Without an Interactive Shell and Not in admin Add user sarah wit 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.



(e) Set Password for All Users to password

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

For harry:

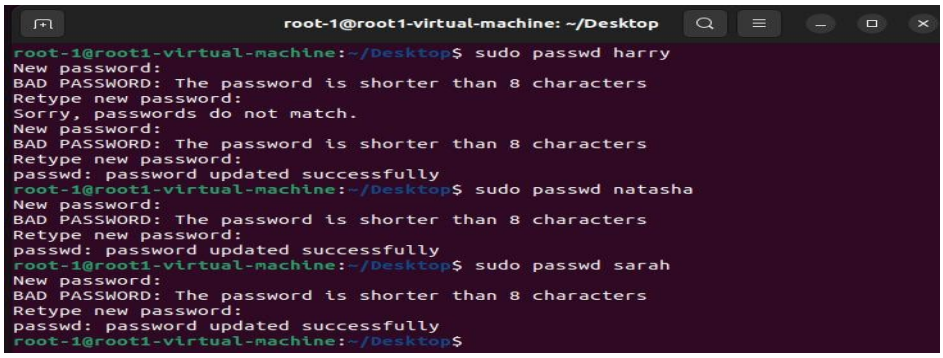
sudo passwd harry

For natasha:

sudo passwd natasha

For sarah:

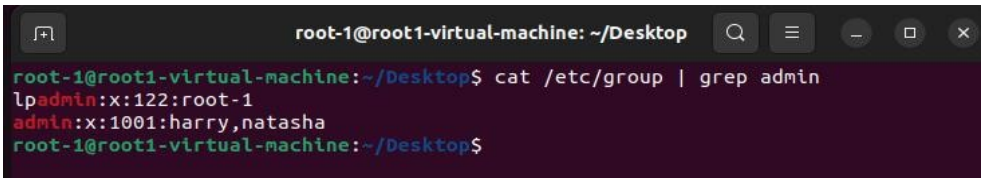
sudo passwd sarah



Verification Commands

1. **Verify the admin group exists:**

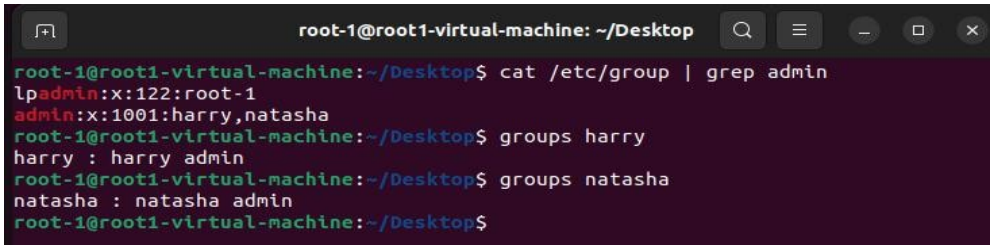
   cat /etc/group | grep admin



2. **Check group memberships for harry and natasha:**

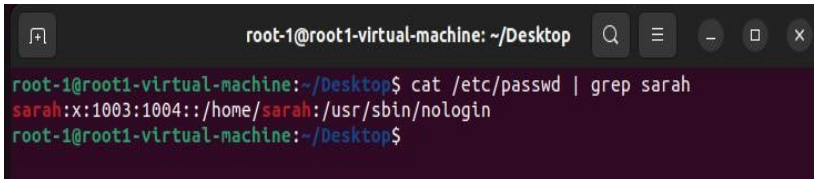
   groups harry

   groups natasha



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

   cat /etc/passwd | grep sarah



Q4. FTPServer Setup:

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

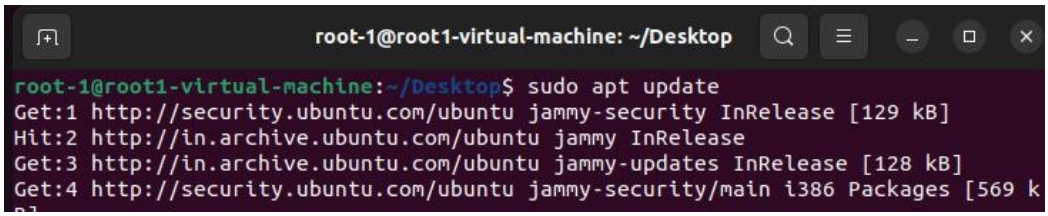
Answer: -

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:

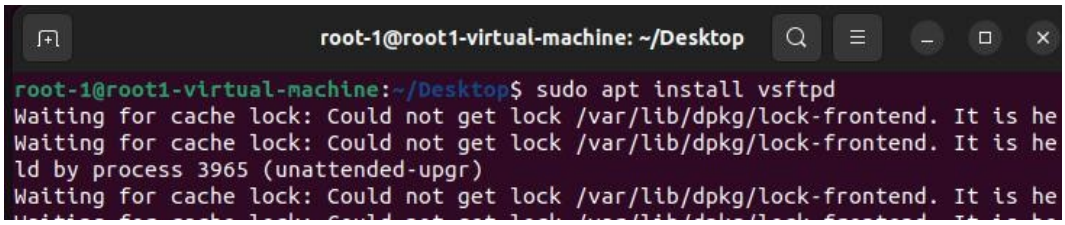
Step 1: Install FTP Server (vsftpd)

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

   sudo apt update

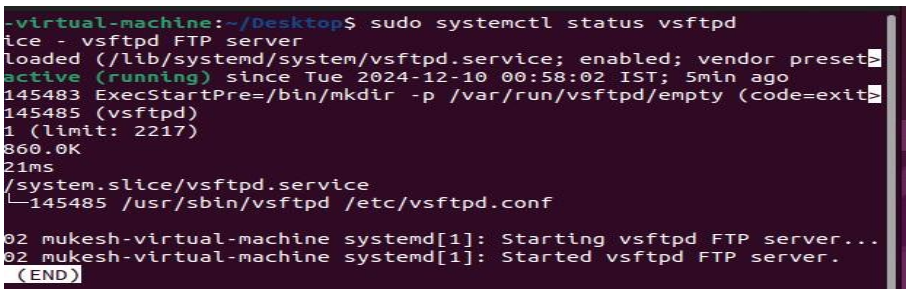


   sudo apt install vsftpd



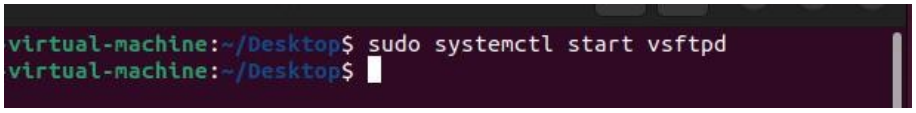
3. Check if the service is running:

   sudo systemctl status vsftpd



If it’s not running, start it:

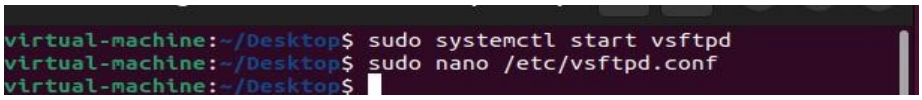
sudo systemctl start vsftpd



Step 2: Configure vsftpd

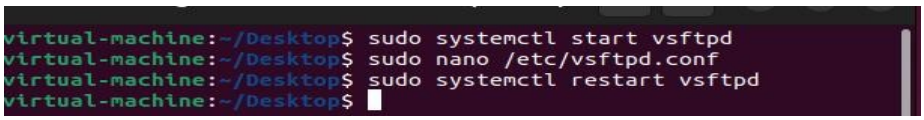
1. Open the vsftpd configuration file:

   sudo nano /etc/vsftpd.conf



2. Modify/add the following settings:
   - Enable local users to log in:
   - local_enable=YES
   - Enable write access:
   - write_enable=YES
   - Optional: Disable anonymous access for better security:
   - anonymous_enable=NO
   - Uncomment to allow local users to upload files:
   - chroot_local_user=YES
3. Save and close the file (Ctrl+O, Enter, Ctrl+X).
4. Restart the **vsftpd** service:

   sudo systemctl restart vsftpd



Step 3: Set Up a Local FTP User

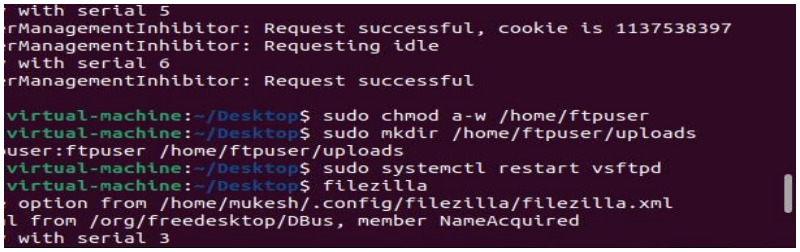
1. Create a new user for FTP access:

   sudo adduser ftpuser

Follow the prompts to set up the password.

2. Assign the home directory permissions (optional):

   sudo chmod -R 755 /home/ftpuser



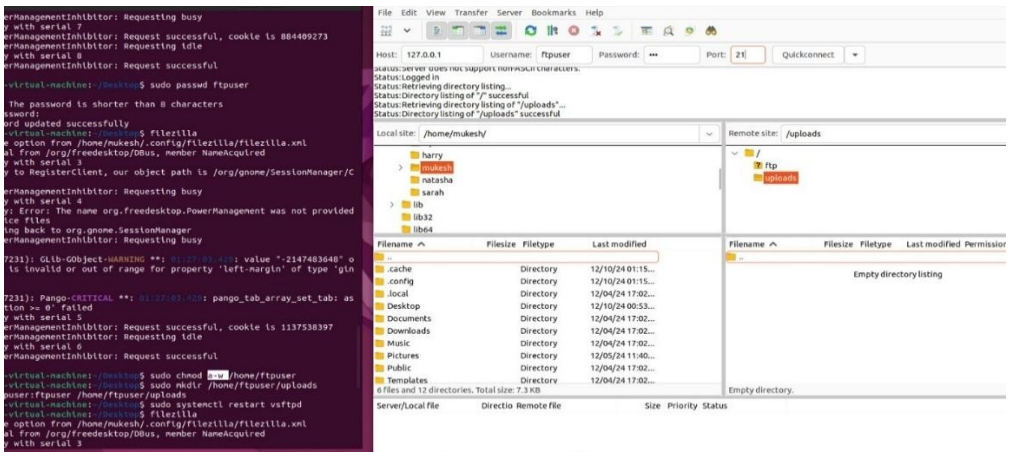
Step 4: Access FTP Server

Using FileZilla

1. Install FileZilla:

   sudo apt install filezilla

2. Open FileZilla and enter:
   - **Host**: ftp://127.0.0.1
   - **Username**: ftpuser
   - **Password**: 123
   - **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 (ftpuser) and password in the session.

Using MobaXterm

1. Open MobaXterm.
2. Create a new FTP session:
    - Go to **Session > FTP**.
    - Enter the server IP, port, username (ftpuser), 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/ftpuser

Q5. Website Hosting:

- Host a website using httpd.

Answer: -

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

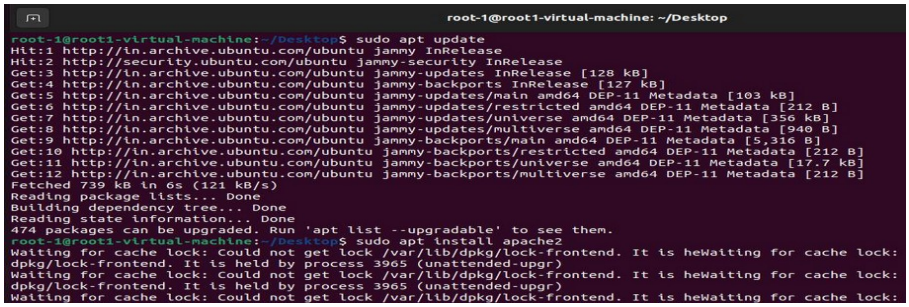
Step 1: Install Apache HTTP Server

1. Open the terminal and update your package list:

    sudo apt update

2. Install Apache:

    sudo apt install apache2



3. Confirm Apache is running:

    sudo systemctl status apache2

```
iggers for libc-bin (2.35-0ubuntu3.8) ...
-virtual-machine:~/Desktop$ sudo systemctl status apache2
/ice - The Apache HTTP Server
loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
ctive (running) since Tue 2024-12-10 01:54:57 IST; 27s ago
ttps://httpd.apache.org/docs/2.4/
148586 (apache2)
55 (limit: 2217)
5.9M
55ms
/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:
   - Remove the default file:

     sudo rm index.html
   - 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>

- 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

- Configure virtual hosting for web1.example.com and

  web2.example.com.

Answer: -

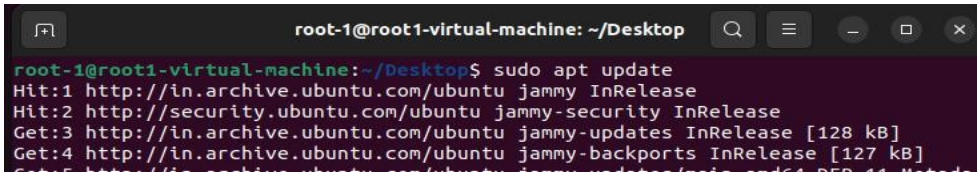
To configure virtual hosting for **web1.example.com** and **web2.example.com** on Ubuntu, yo
set up **Apache** as the web server and configure virtual hosts. Here's a step-by-step guide to p
this practical:

Step 1: Install Apache Web Server

1. Open the terminal and update your system:

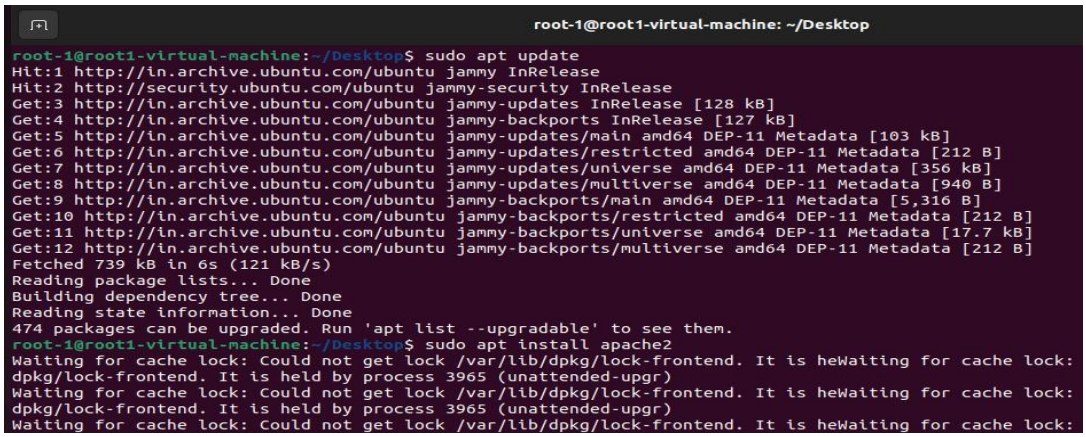
   sudo apt update

   sudo apt upgrade



2. Install the Apache2 package:

   sudo apt install apache2



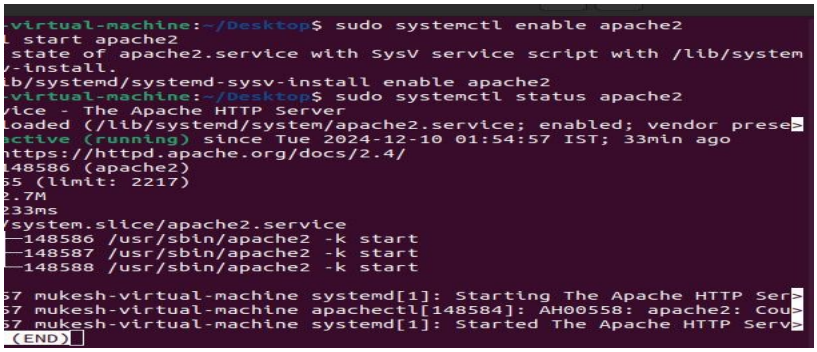
3. Enable and start Apache:

   sudo systemctl enable apache2

   sudo systemctl start apache2

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

   sudo systemctl status apache2



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.ht

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

```
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/ww
.com
www-data:www-data /var/www/web2.example.com
virtual-machine:~/Desktop$ echo "<h1>Welcome to Web1</h1>" | sudo
eb1.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.co** /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

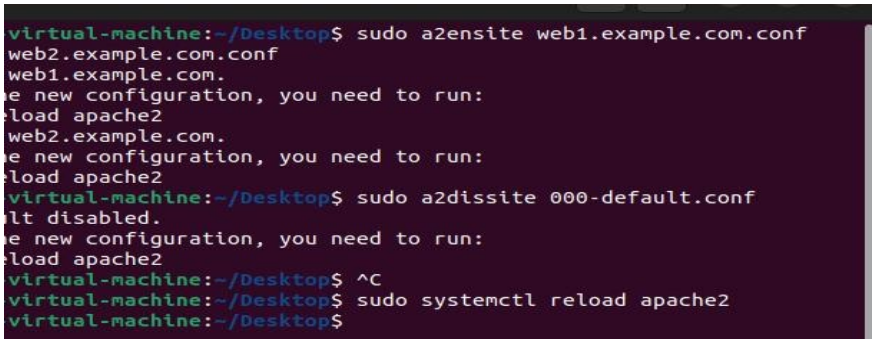
   sudo a2ensite web2.example.com.conf

2. Disable the default site (optional):

   sudo a2dissite 000-default.conf

3. Reload Apache to apply the changes:

   sudo systemctl reload apache2



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

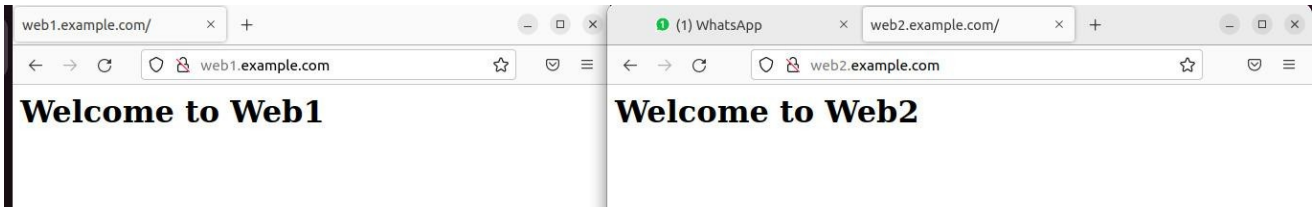
3. Save and close the file.

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:

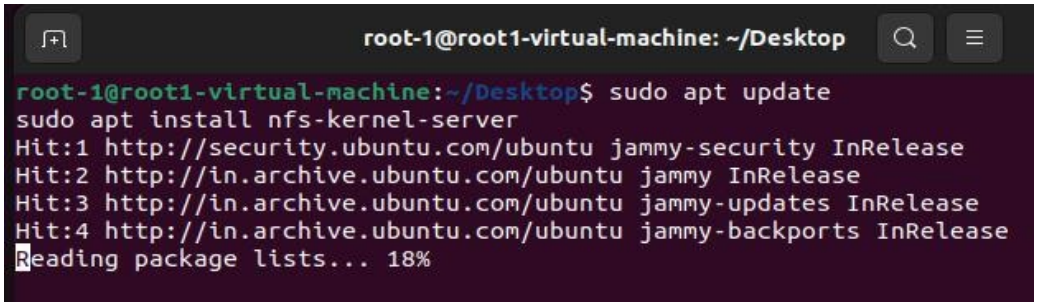
Step 1: Install NFS Server and Client

On the NFS Server:

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

   sudo apt update

   sudo apt install nfs-kernel-server



2. Enable and start the NFS server:

   sudo systemctl enable nfs-server

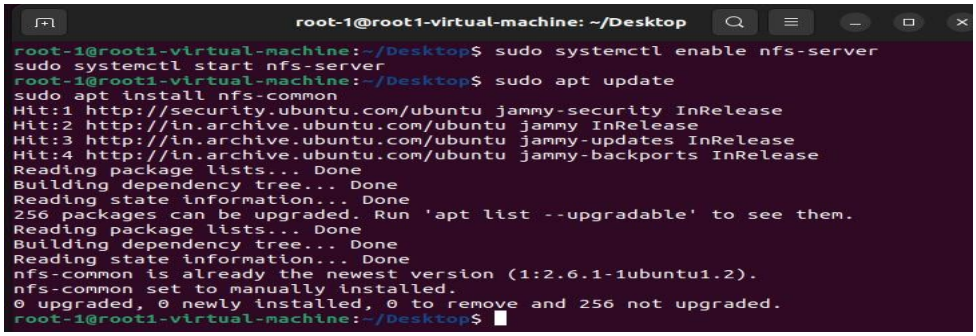
   sudo systemctl start nfs-server

On the NFS Client:

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

   sudo apt update

   sudo apt install nfs-common



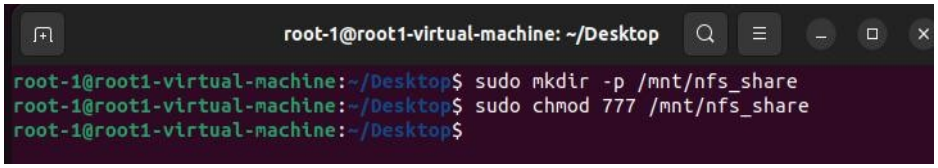
Step 2: Create a Directory to Share

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



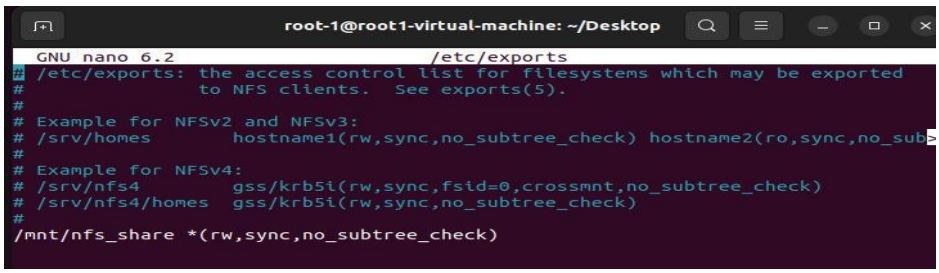
Step 3: Configure the NFS Export

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)



   - *: Allows all clients to connect (replace * with a specific client IP or subnet for mor security).
   - rw: Allows read-write access.
   - sync: Ensures data is written synchronously to disk.
   - no_subtree_check: Improves performance by not checking file permissions for subdirectories.

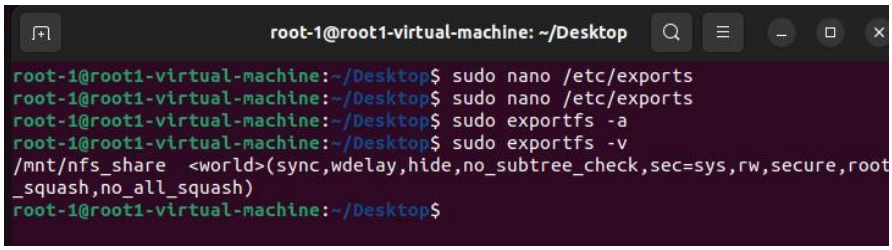
3. Save and exit the file (Ctrl+O, Enter, Ctrl+X).

4. Apply the changes:

sudo exportfs -a

5. Verify the shared directory:

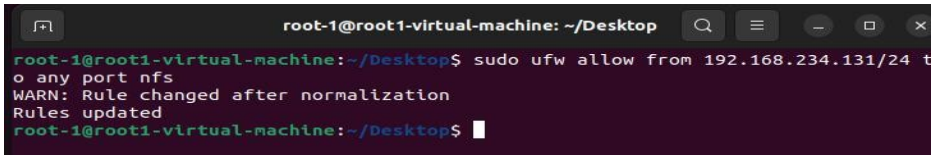
   sudo exportfs -v



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



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:

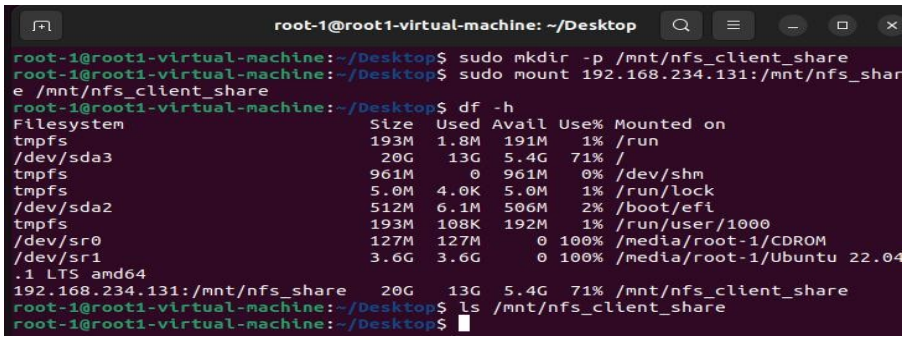
   sudo mount 192.168.234.131:/mnt/nfs_share /mnt/nfs_client_share

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

3. Verify the mounted share:

   df -h

   ls /mnt/nfs_client_share



Step 6: Make the Mount Permanent (Optional)

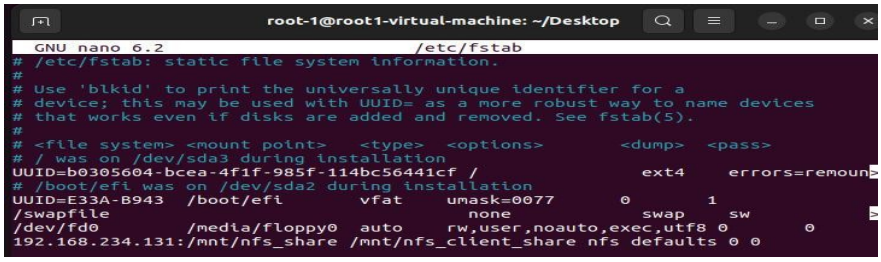
To ensure the NFS share is mounted automatically at boot, add an entry to the /etc/fstab fil
the client:

1. Open the file:

   sudo nano /etc/fstab

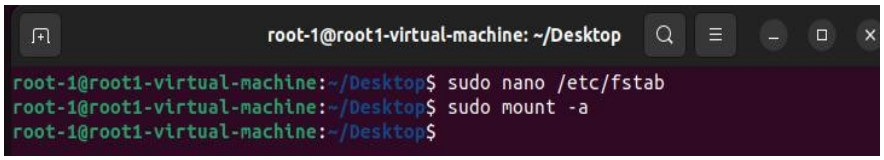
2. Add the following line:

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



3. Save and exit the file (Ctrl+O, Enter, Ctrl+X).
4. Test the configuration:

   sudo mount -a



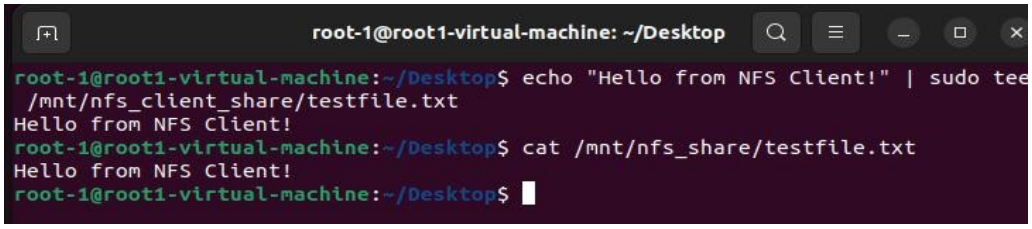
Testing the NFS Share

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



Q8. MariaDB Setup

- SetupaMariaDB server.

Answer: -

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

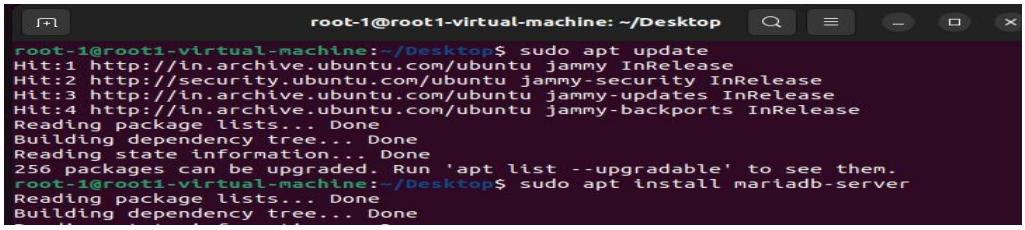
Step 1: Update the Package Repository

1. Open a terminal in Ubuntu.

2. Update the package repository: sudo apt update

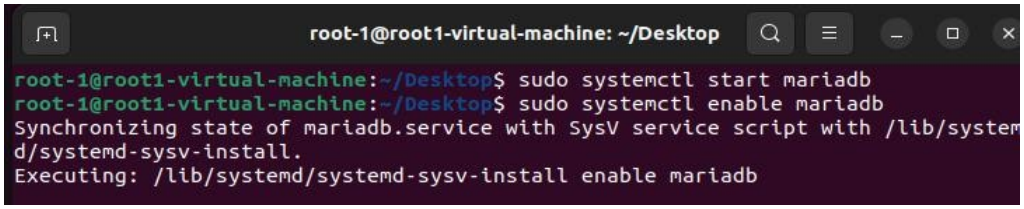
Step 2: Install MariaDB Server

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



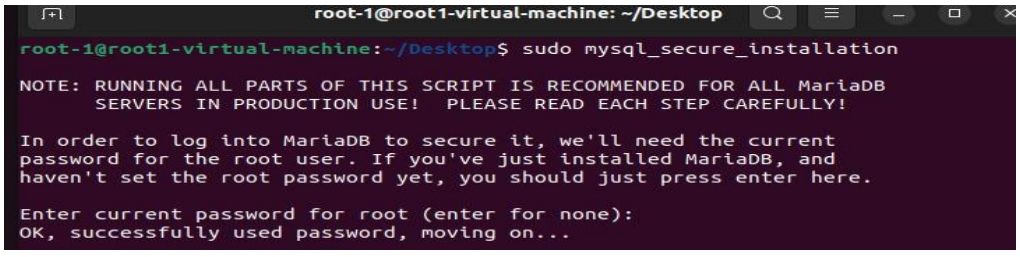
Step 3: Start and Enable MariaDB Service

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



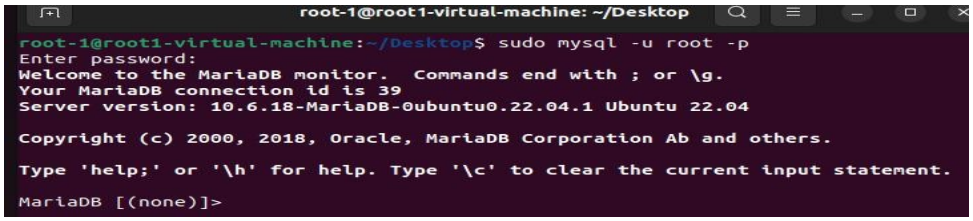
Step 4: Secure MariaDB Installation

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



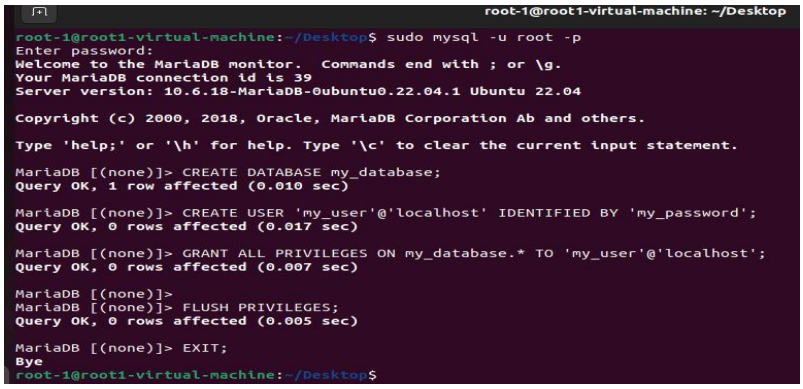
Step 5: Access MariaDB

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



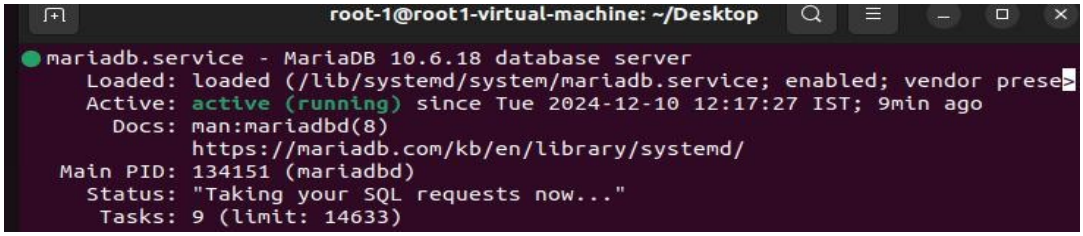
Step 6: Create a New Database and User (Optional)

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.* 'my_user'@'localhost';
4. Reload privileges: FLUSH PRIVILEGES;
5. Exit MariaDB: EXIT;

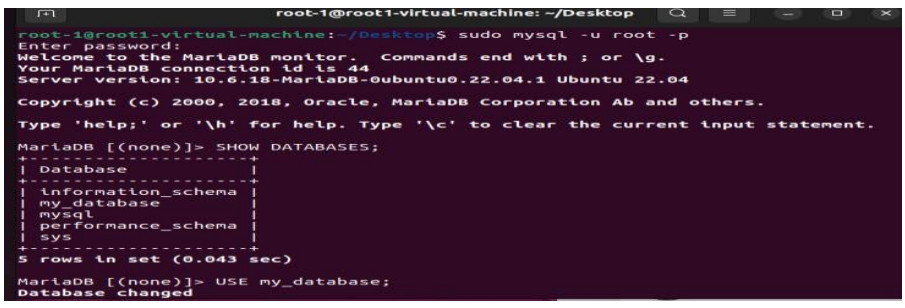


Step 7: Test MariaDB Server

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



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



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** these steps. This includes installing the necessary software (Apache, MySQL, PHP), setting up WordPress, and configuring basic optimization.

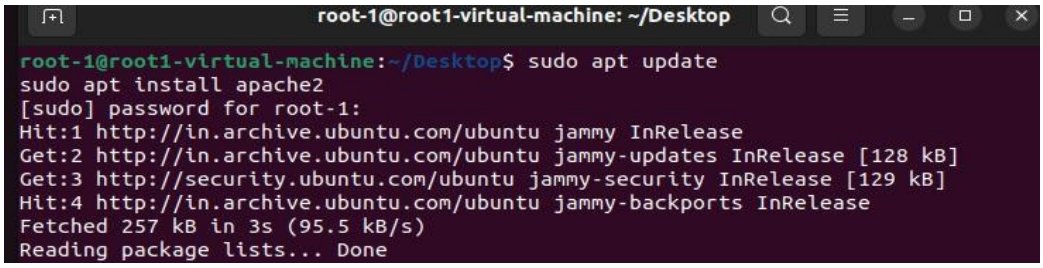
Step 1: Install LAMP Stack

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

1. **Install Apache (Web Server):**

   sudo apt update

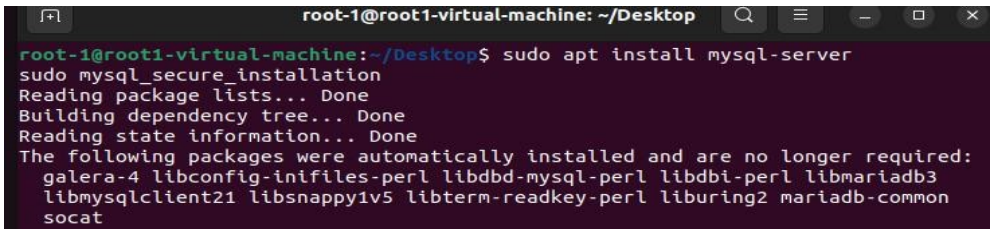
   sudo apt install apache2



2. **Install MySQL (Database Server):**

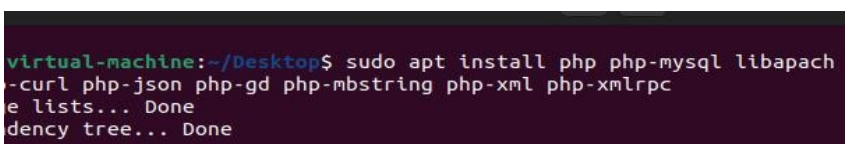
   sudo apt install mysql-server

   sudo mysql_secure_installation

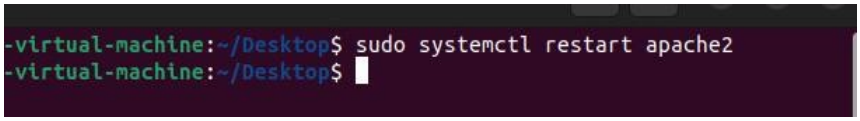


3. **Install PHP (Programming Language):** Install PHP and necessary PHP extensions for WordPress:

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



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



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:

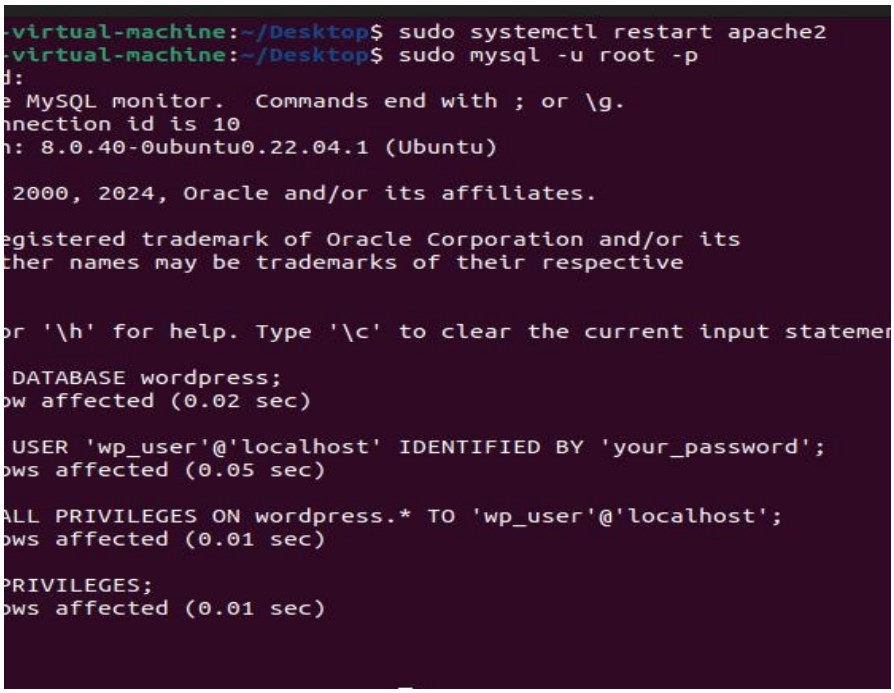
   CREATE DATABASE wordpress;

   CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'your_password';

   GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';

   FLUSH PRIVILEGES;

   EXIT;



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 -xzvf 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
dpress.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

: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

:35:16 (167 KB/s) - 'latest.tar.gz' saved [26931653/26931653]

-virtual-machine:/var/www/html$ sudo tar -xzvf 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/htm
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-co
   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:

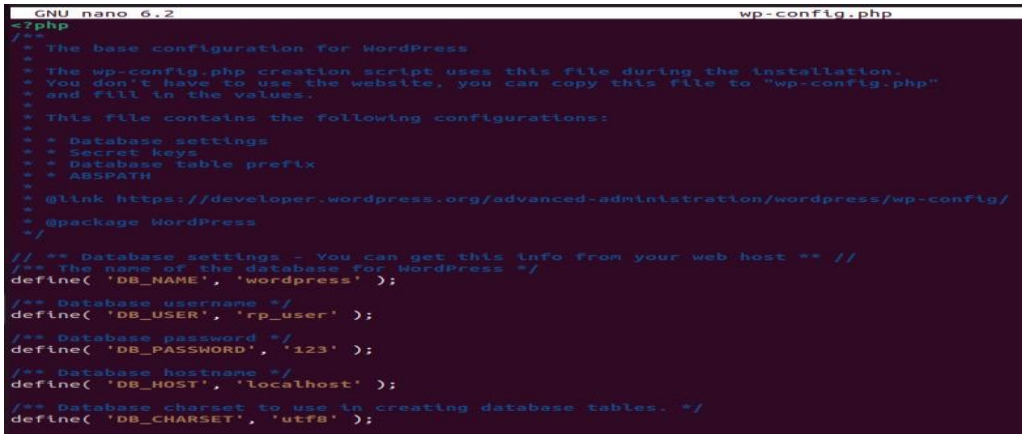
define('DB_NAME', 'wordpress');

define('DB_USER', 'wp_user');

define('DB_PASSWORD', '123');

define('DB_HOST', 'localhost');

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



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:
    - **Site Title**: Your website's title.
    - **Username**: Administrator username.
    - **Password**: Administrator password.
    - **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 p

- 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 Insig** 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 ad security:define('XMLRPC_REQUEST', false);
2. **Install a Security Plugin**: Use plugins like **Wordfence Security** or **iThemes Securit** 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.me manual

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

```
-virtual-machine:/var/www/html$ sudo nmcli connection add type ethe
 mylab ifname eth0 ipv4.addresses 192.168.45.155/24 ipv4.gateway 19
v4.dns "8.8.8.8,192.168.45.1" ipv4.method manual
rd for mukesh:
ylab' (65ce0d15-a354-41fd-9eb4-720a6a45949a) successfully added.
-virtual-machine:/var/www/html$ ip link
ACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
t qlen 1000
back 00:00:00:00:00:00 brd 00:00:00:00:00:00
OADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mod
up default qlen 1000
 00:0c:29:fa:fd:f0 brd ff:ff:ff:ff:ff:ff
np2s1
```

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

```
-virtual-machine:/var/www/html$ sudo nmcli connection modify mylab
terface-name ens33
-virtual-machine:/var/www/html$ sudo nmcli connection up mylab
ccessfully activated (D-Bus active path: /org/freedesktop/NetworkMa
onnection/2)
-virtual-machine:/var/www/html$ nmcli connection show mylab
ens33
:                       mylab
id:                     65ce0d15-a354-41fd-9eb4-720a6a45949a
able-id:                --
pe:                     802-3-ethernet
terface-name:           ens33
toconnect:              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
ccessfully activated (D-Bus active path: /org/freedesktop/NetworkMa
onnection/3)
-virtual-machine:/var/www/html$ nmcli connection show mylab
:                       mylab
id:                     65ce0d15-a354-41fd-9eb4-720a6a45949a
able-id:                --
pe:                     802-3-ethernet
terface-name:           ens33
toconnect:              yes
toconnect-priority:     0
toconnect-retries:      -1 (default)
lti-connect:            0 (default)
```

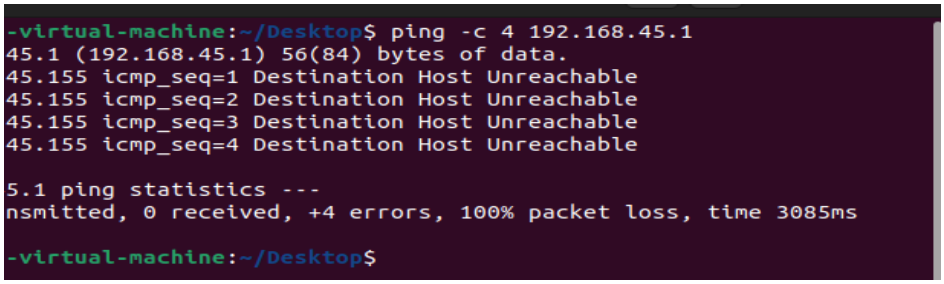
   ip addr show ens33

```
-virtual-machine:/var/www/html$ ip addr show ens33
OADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
en 1000
 00:0c:29:fa:fd:f0 brd ff:ff:ff:ff:ff:ff
np2s1
168.45.155/24 brd 192.168.45.255 scope global noprefixroute ens33
lft forever preferred_lft forever
0::8c9a:710:8c4d:7200/64 scope link noprefixroute
lft forever preferred_lft forever
-virtual-machine:/var/www/html$
```

Verify Network Connectivity

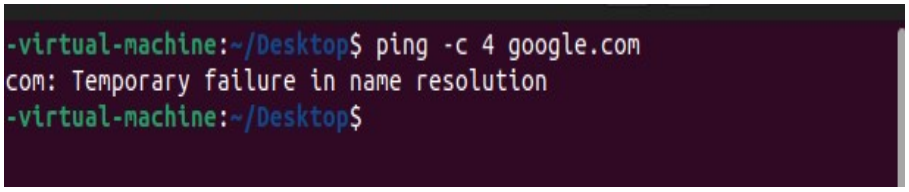
1. Ping the gateway to ensure the network is working:

   ping -c 4 192.168.45.1



2. Test DNS resolution by pinging a domain:

   ping -c 4 google.com



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