Sri Lanka Institute of Information Technology

B.Sc. (Hons) Information Technology-

Cyber security

Y2 S1

Systems and Network Programming – IE 2012

Assignment

M. A. AMANTHA

IT 23 1843 12

Introduction

This assignment aims to provide in-depth knowledge about system and network programming through hands-on assignments based on Linux. It is presumed that students, while doing these assignments, will develop the necessary practical skills in the configuration of core system components and basic services such as DHCP, DNS, and NTP. The students will also be introduced to shell scripting and security practices. Specifically, this will entail activities that will enable the students to establish skills in configuring Linux-based systems, user management, system information gathering, and ensuring security in network administration. This assignment also focuses on enhancing problem-solving skills in system automation, firewall configuration, and implementation of best security practices for effective learning.

Declaration

I hereby declare that the work presented here is entirely my original work. I, myself, have carried out the practical work in the setup of the virtual environment, configuration of Linux services, and scripting of automation solutions. All contents and configurations documented are of my personal engagement in the said assignment tasks, and whatever resources utilized have appropriately been acknowledged. This report is hereby submitted in partial fulfillment of my academic requirements in the course, and I hereby attest that it is performed in compliance with the guidelines provided.

Table of Contents

1. VM installation steps

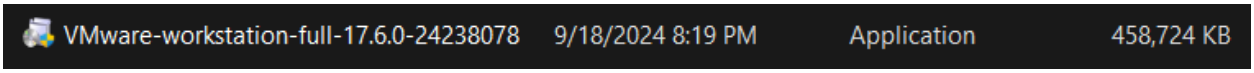
Virtual Machine Setup

Step 01: Download VMware Workstation Pro (Emulator).

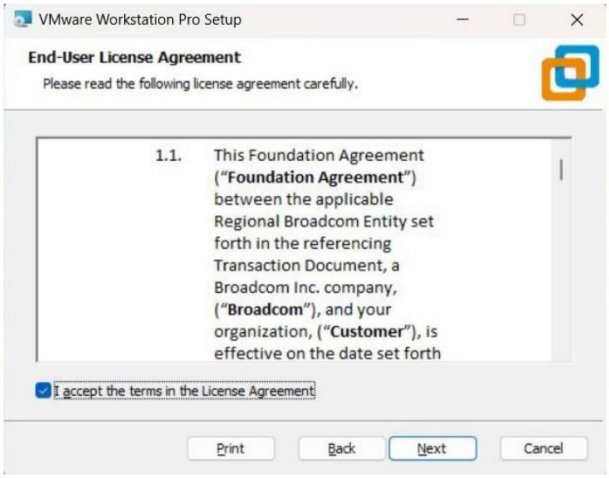
1. Visit VMware website.
    - Go to https://blogs.vmware.com

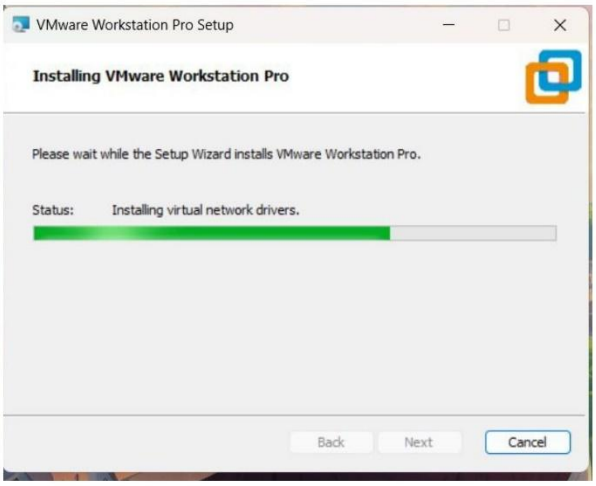
2. Download VMware Workstation Pro.
    - Choose the appropriate version based on your operating system.

- Downloaded file.



3. Install Vmware Workstation Pro.
    - Once the download is complete, Open the installer file.
    - Install the software accepting the terms and conditions.

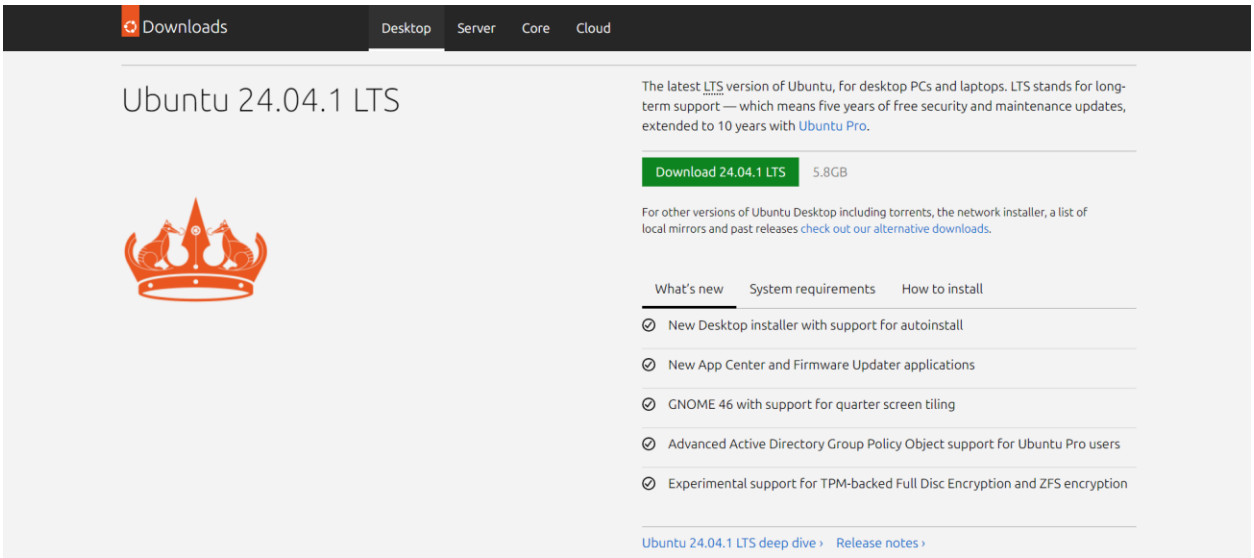




Step 02: Download Ubuntu (Linux Distribution).

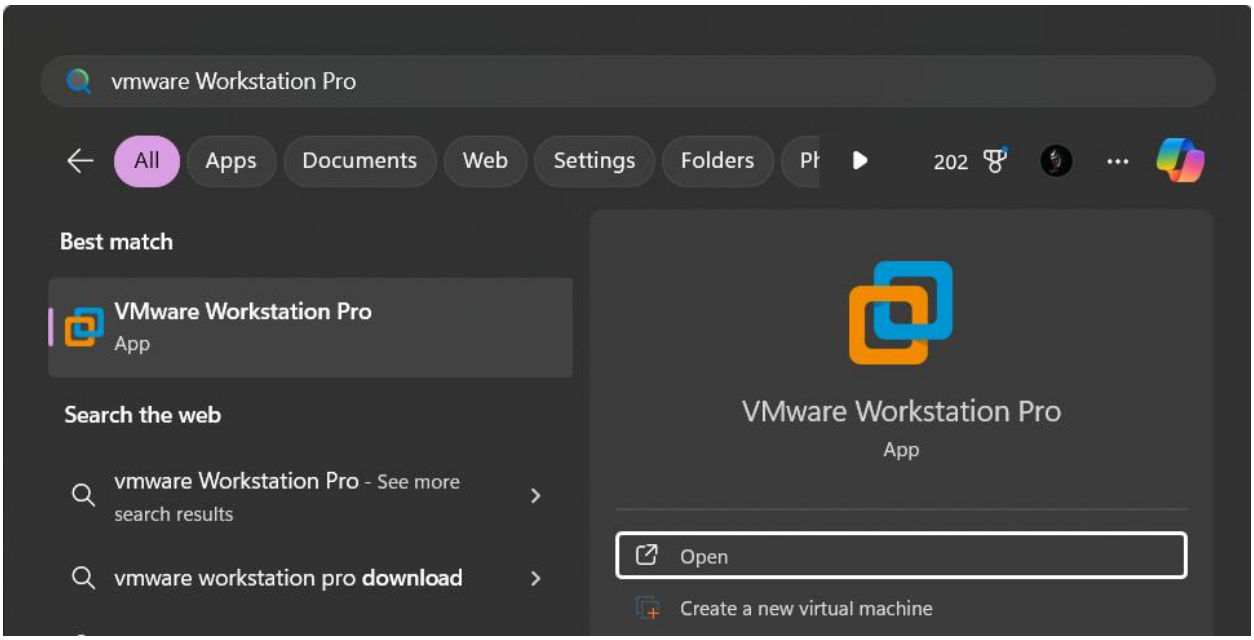
1. Visit Ubuntu Downloads page.
    - Go to https://ubuntu.com/download/desktop

2. Download Ubuntu ISO.
    - Choose the latest LTS (Long Term Support) version and click the download button.



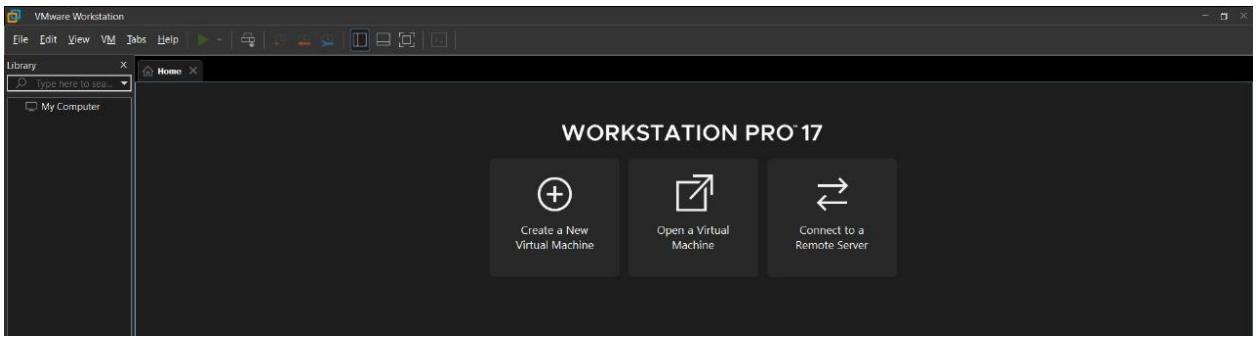
Step 03: Create a New Virtual machine in VMware Workstation pro.

1. Open VMware Workstation Pro.
    - Launch VMware Workstation Pro from your desktop or start menu.

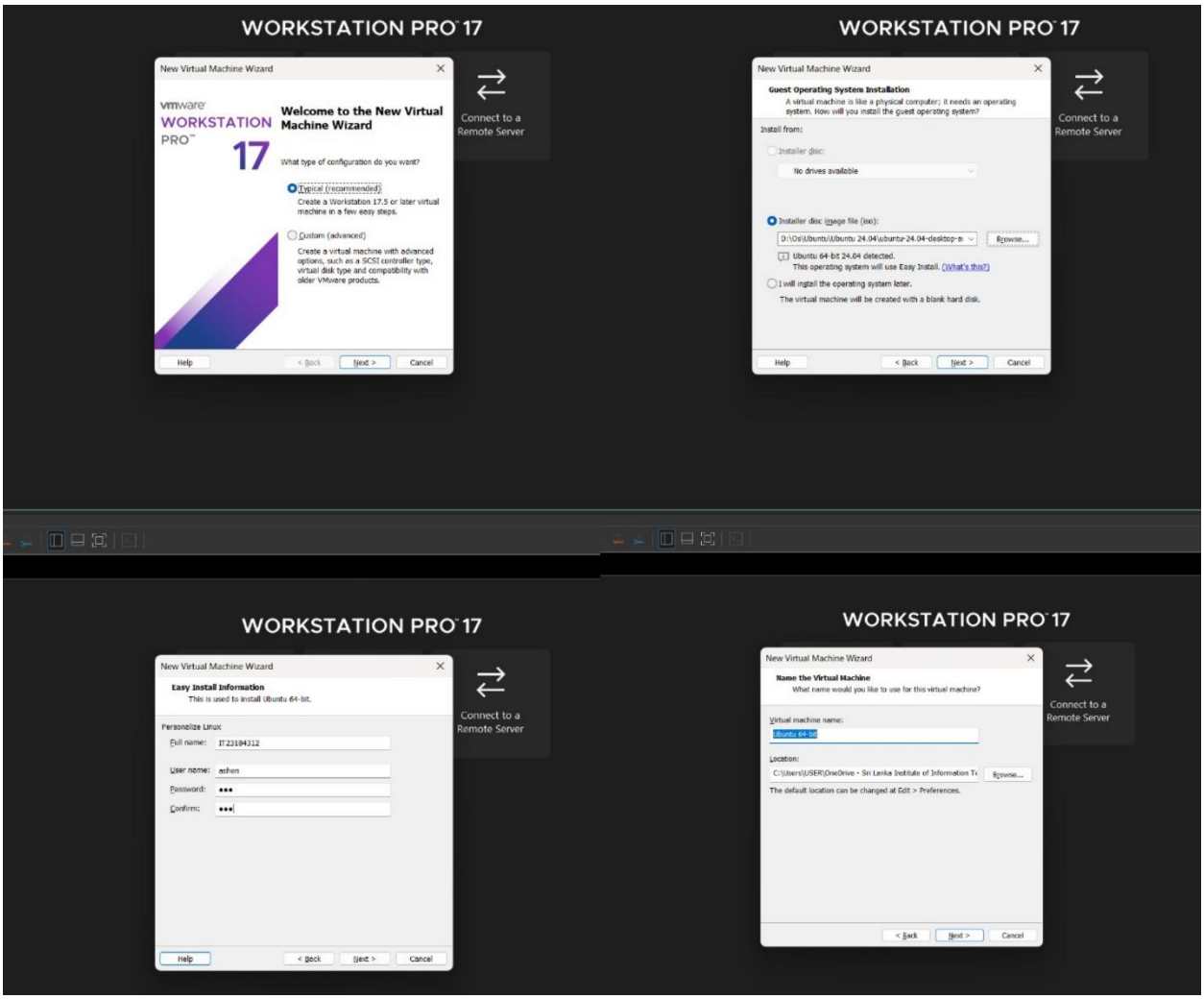


2. Create a new VM.

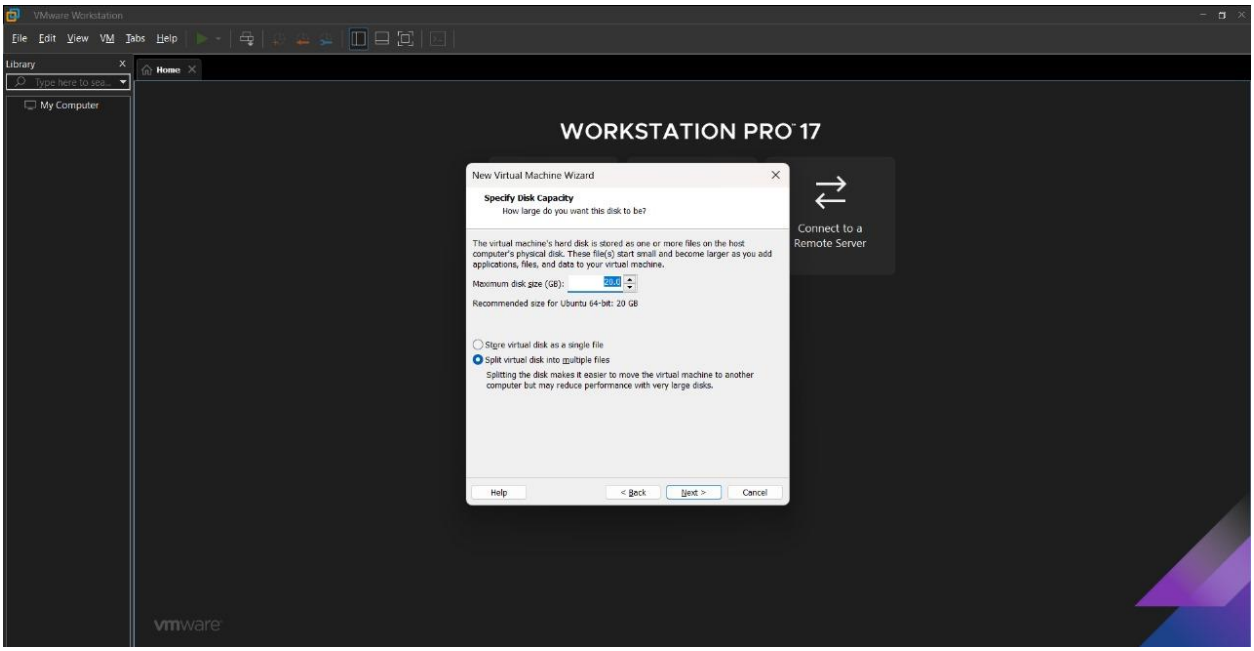
- Click on ‘Create a New Virtual Machine’.



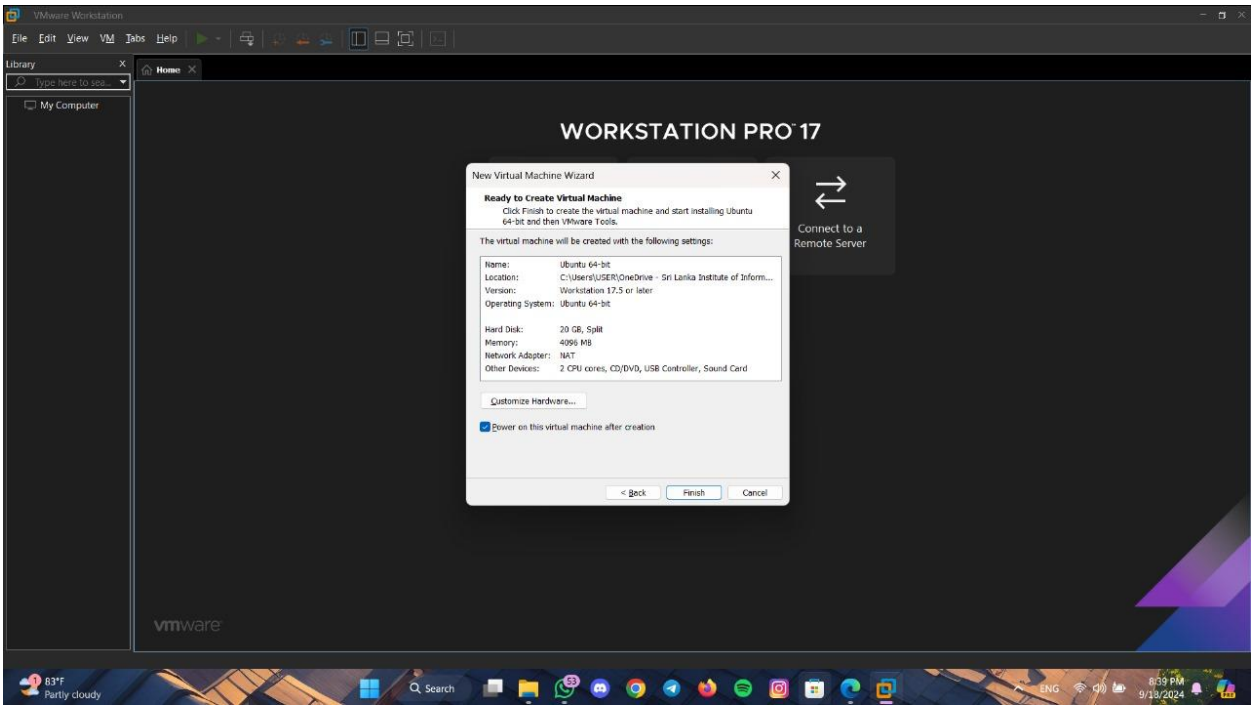
- Select ‘Typical (recommended)’ and click ‘Next’.
- Browse the ISO file.
- Give a name and a password.
- Name the Virtual machine (Set it to "Ubuntu (64-bit)" (or choose the correct version based on the ISO file downloaded).) and click next.



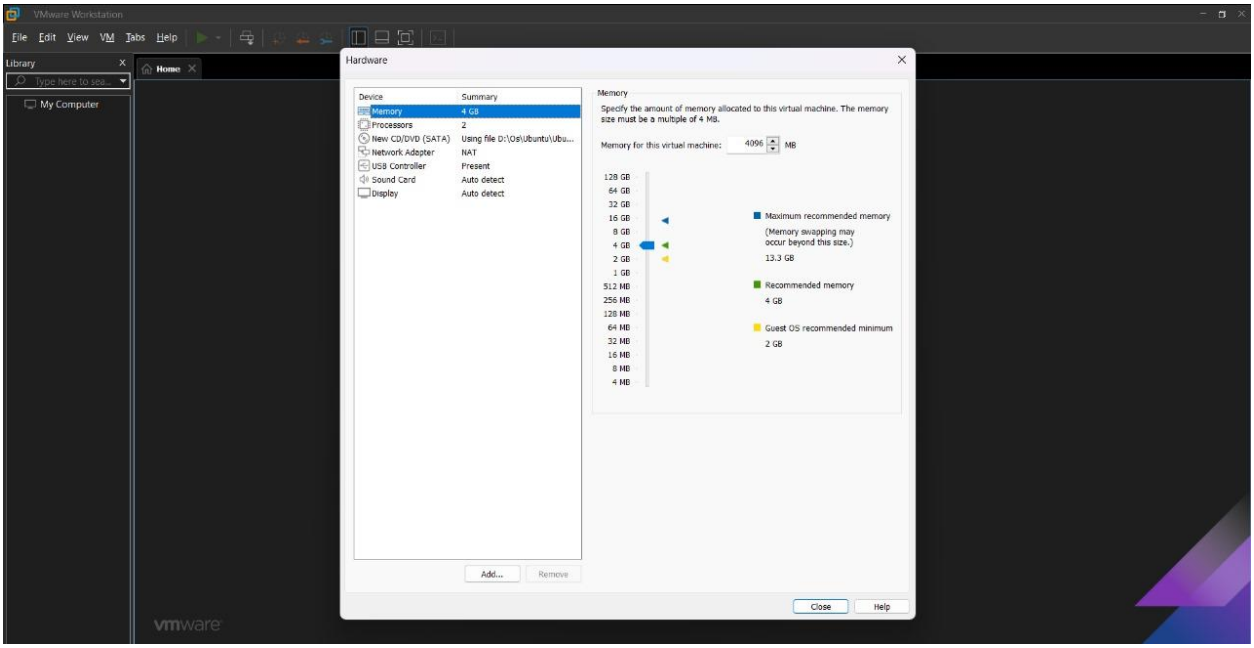
- Allocate storage.



- Allocate memory and set finish.

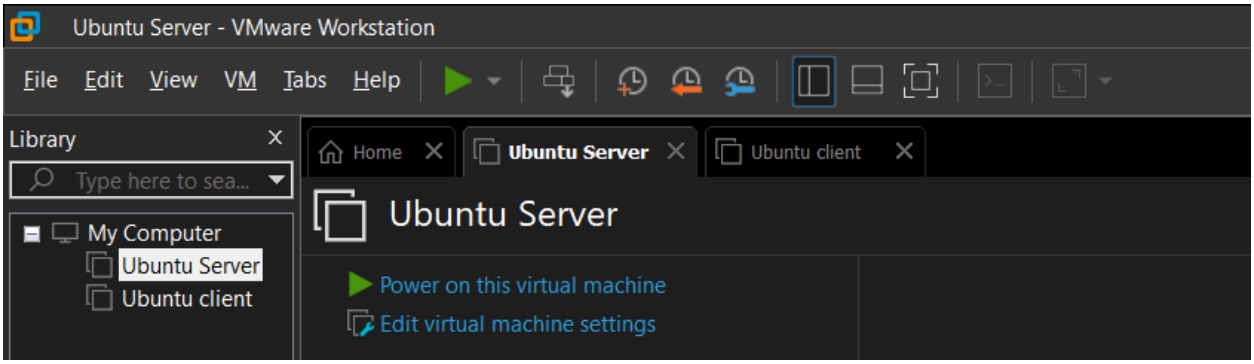


- We can change the settings from the settings tab.
    - Memory settings
    - Processors allocating
    - Network settings, etc.



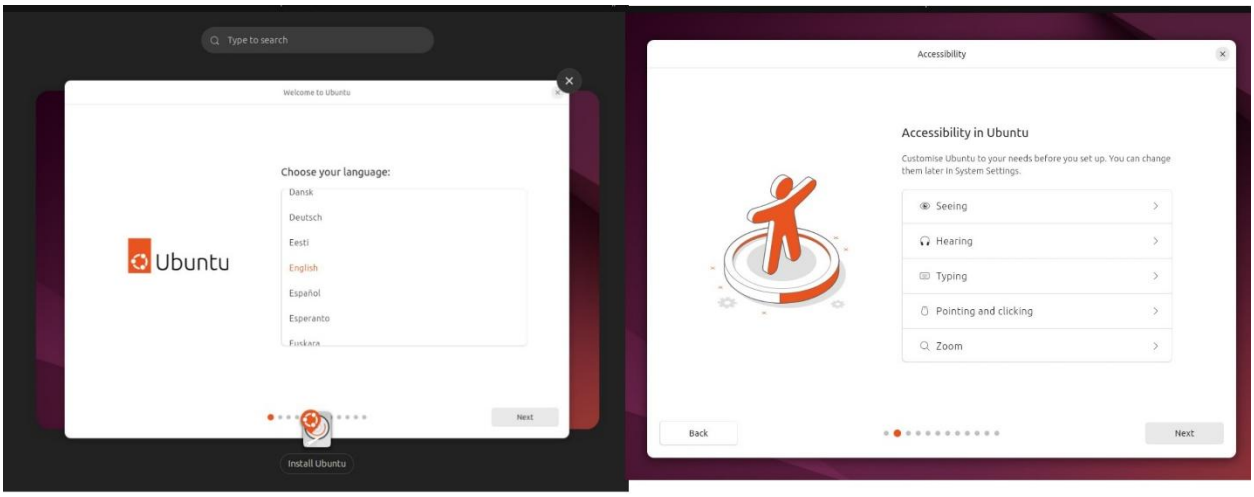
Step 04: Install Ubuntu on the virtual machine.

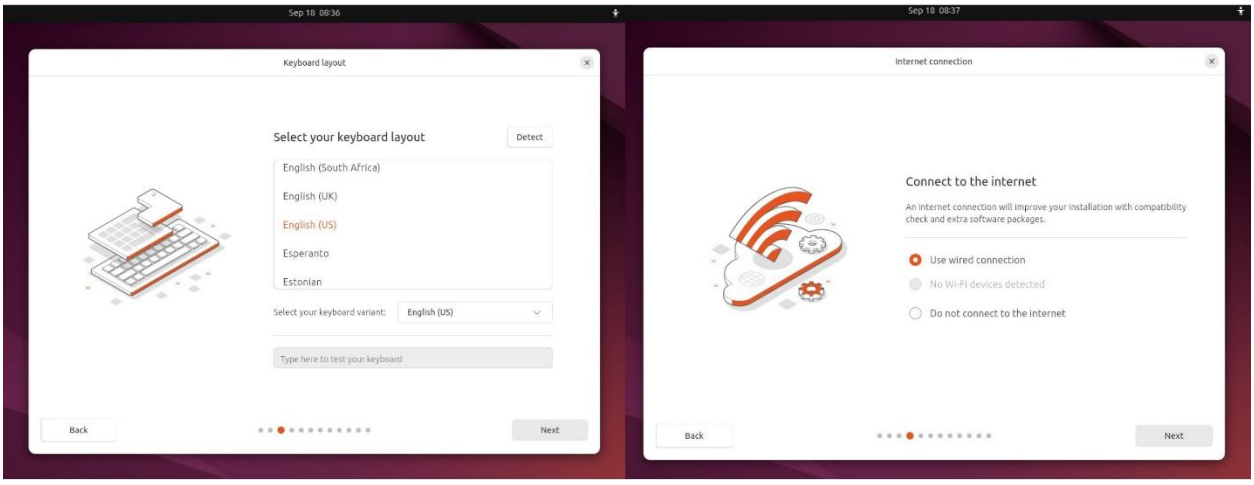
1. After creating the virtual machine, select it from the library tab and click 'Power on this virtual machine'.



2. The VM will boot from the Ubuntu ISO. Follow the on-screen prompts to install Ubuntu:

- **Language Selection:** Choose your language and click "Continue."
- **Keyboard Layout:** Select the appropriate layout and click "Continue."

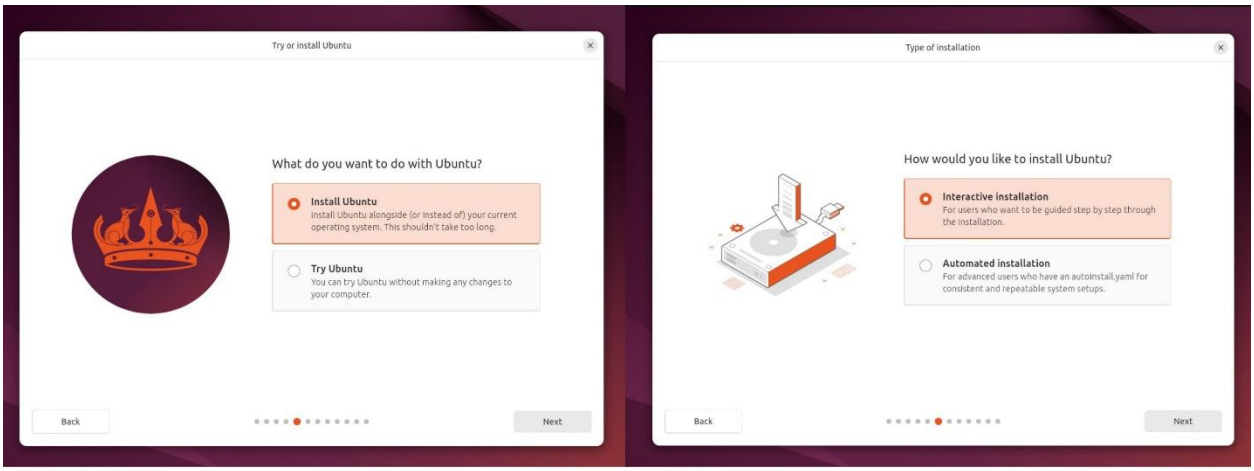


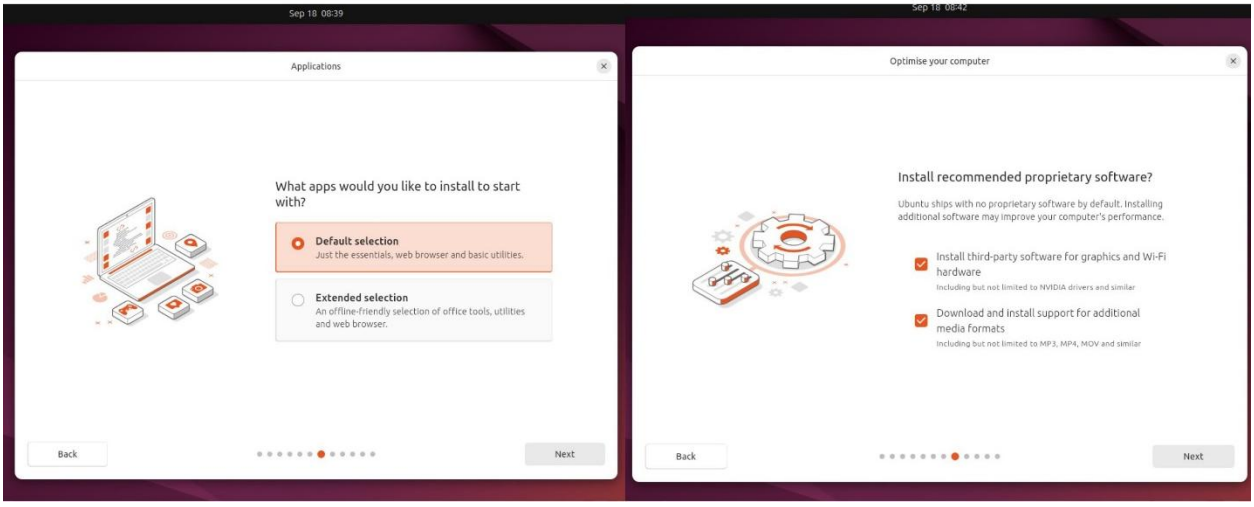


- Install Ubuntu : Select Install Ubuntu and ‘Next’.

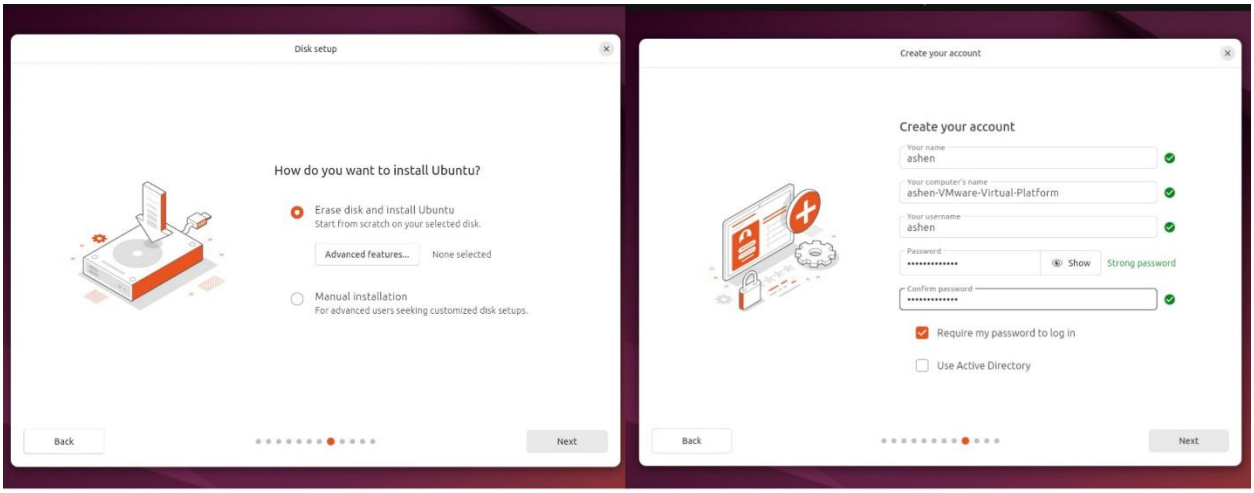
- **Updates and Other Software:** Choose your preference (select "Interactive installation" for a full desktop setup) and click "Next".

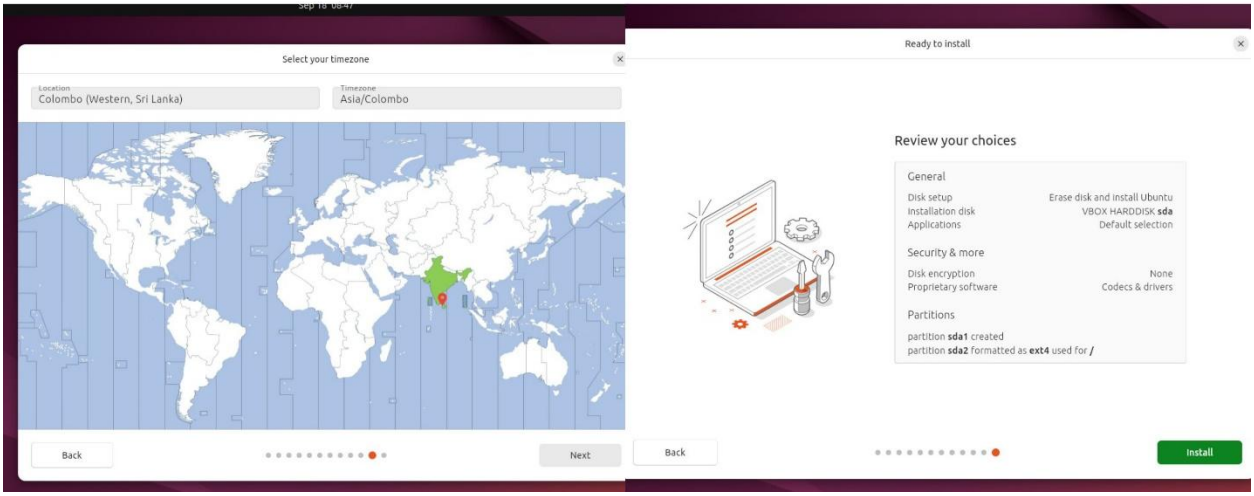
-Default Selection and click ‘Next’.





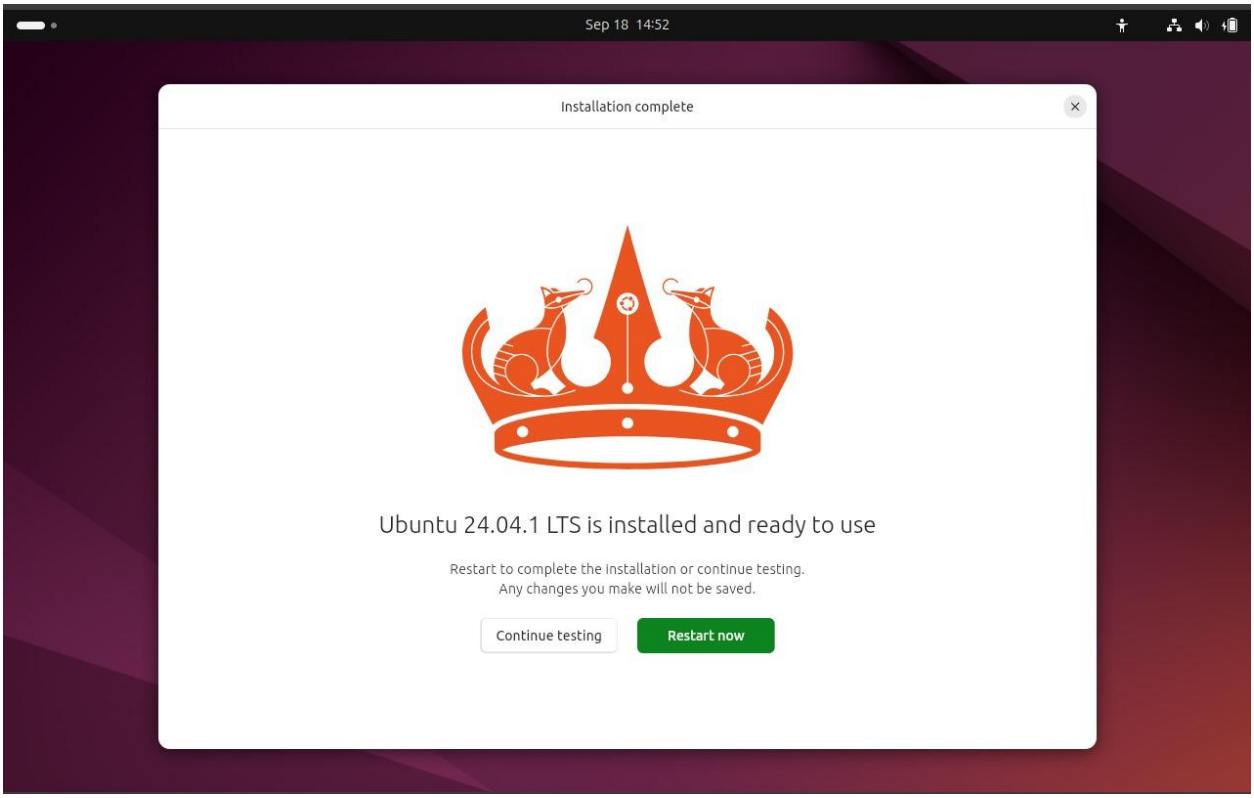
- **Installation Type:** Choose "Erase disk and install Ubuntu" (this applies to the VM, not your host system). Click "Install Now."
- **Time Zone:** Choose your time zone and click "Next."
- **User Details:** Set up your username, computer name, and password. Click "Next".





3. Complete Installation:

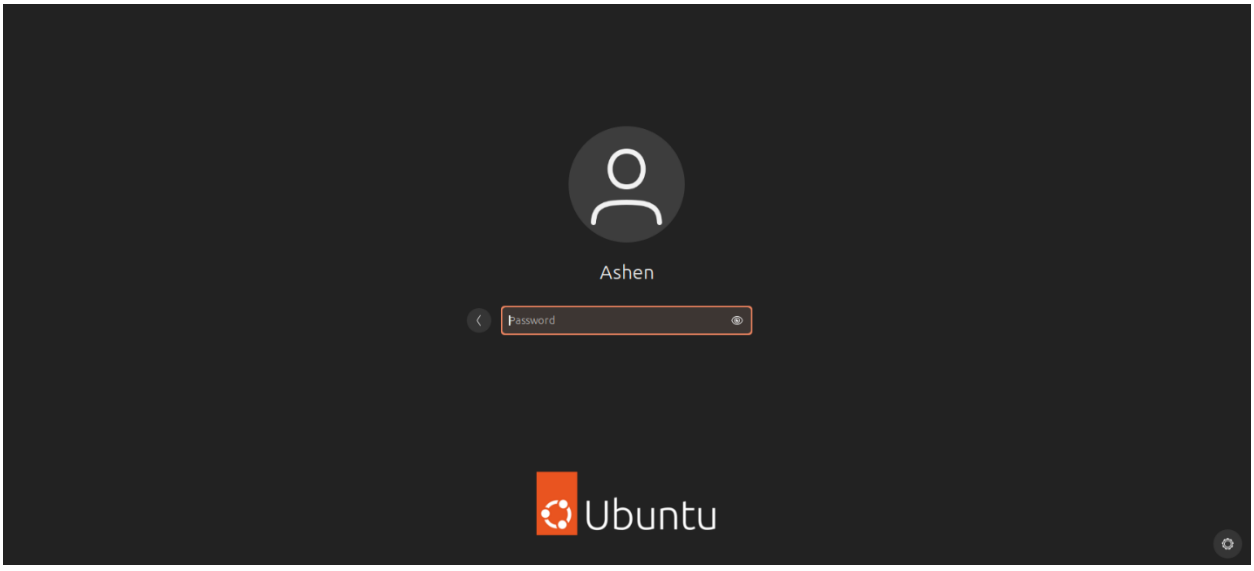
- Once the installation is complete, you'll be prompted to restart the VM. Click "Restart Now"



Step 04: Post-Installation Configuration.

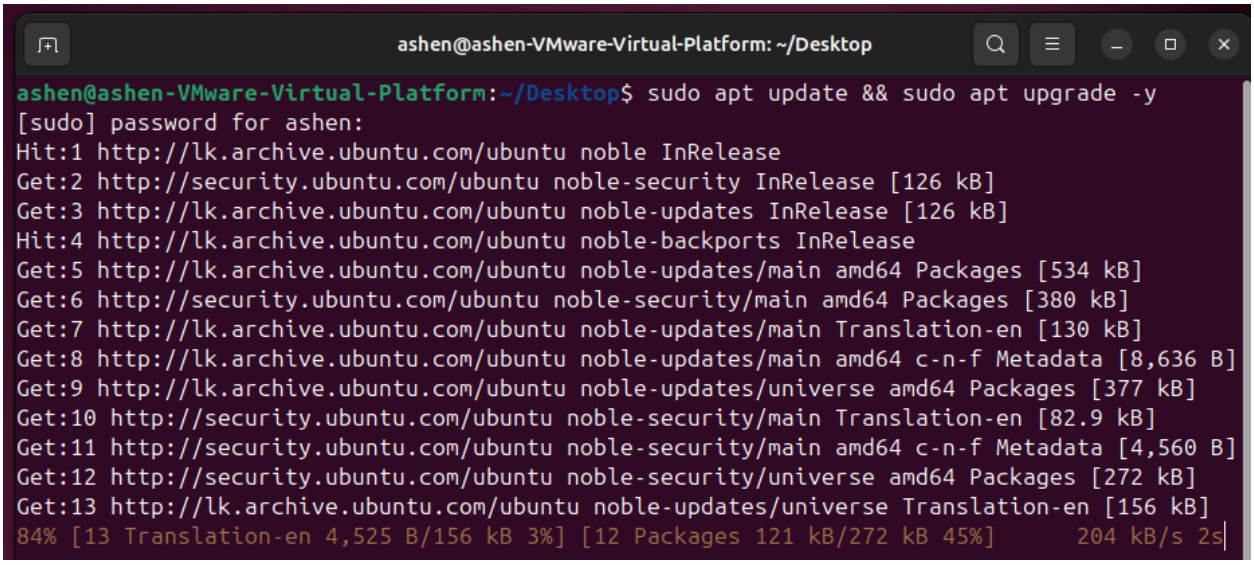
1. First Boot:

- log in with the username and password you set up during installation.

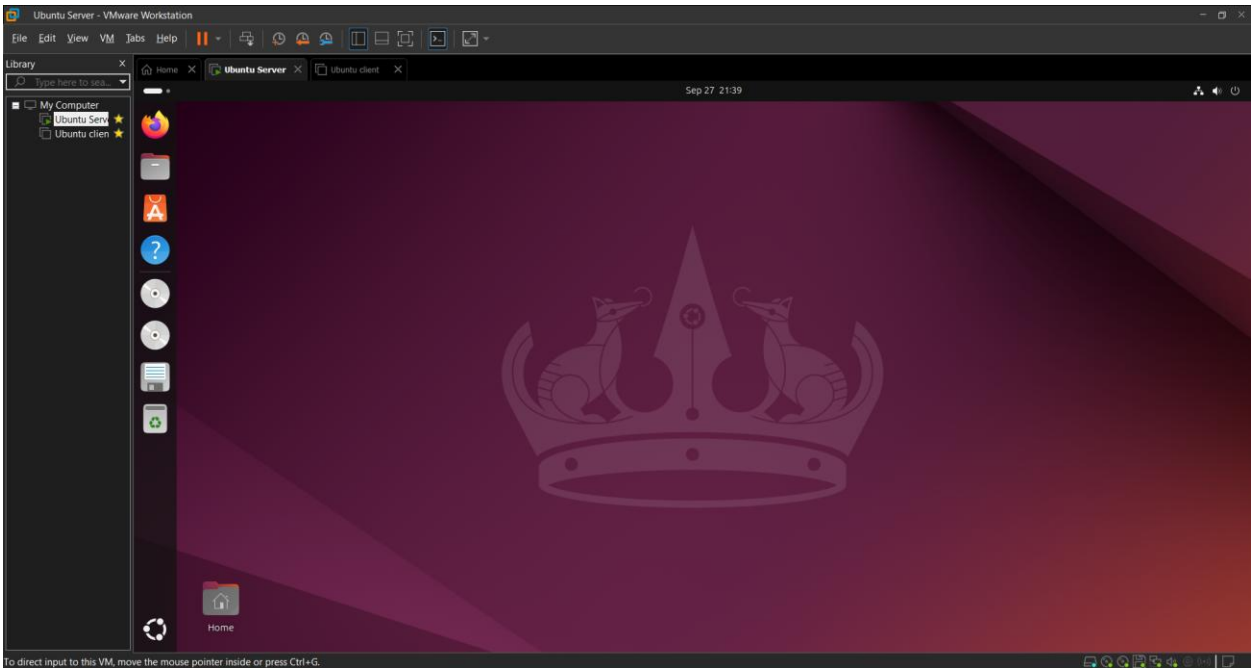


2. Update Ubuntu.

   - sudo apt update && sudo apt upgrade -y
   - Enter the password.

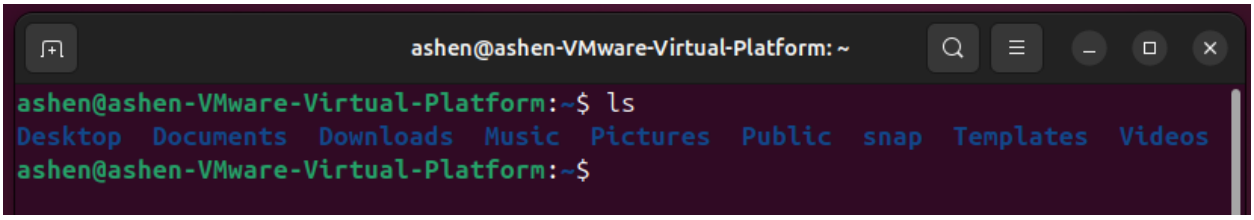


Installation successful.

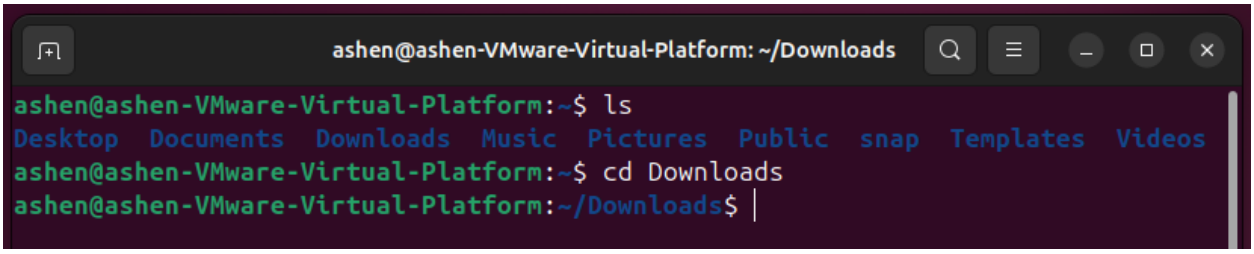


2. Basic Navigation commands and File manipulation commands

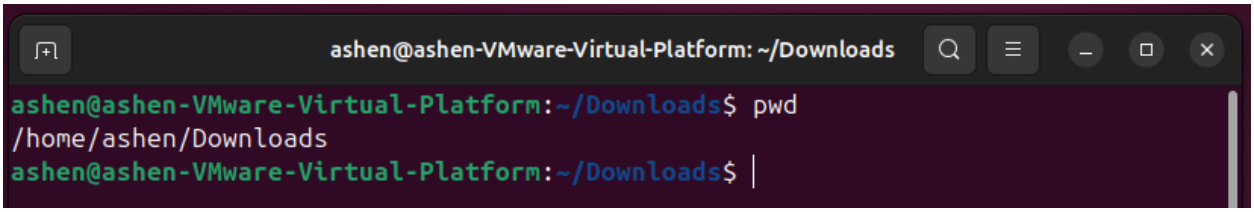
1. **ls :** This command is used to display the contents of a directory.



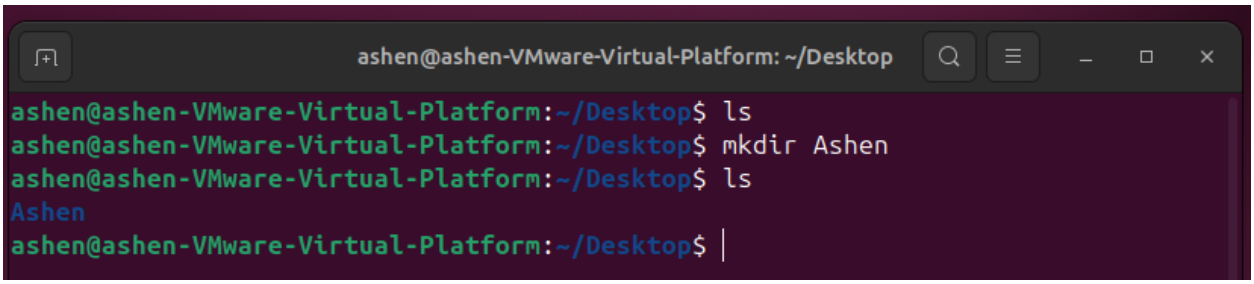
2. **cd :** This command is used to navigate between directories in the filesystem.



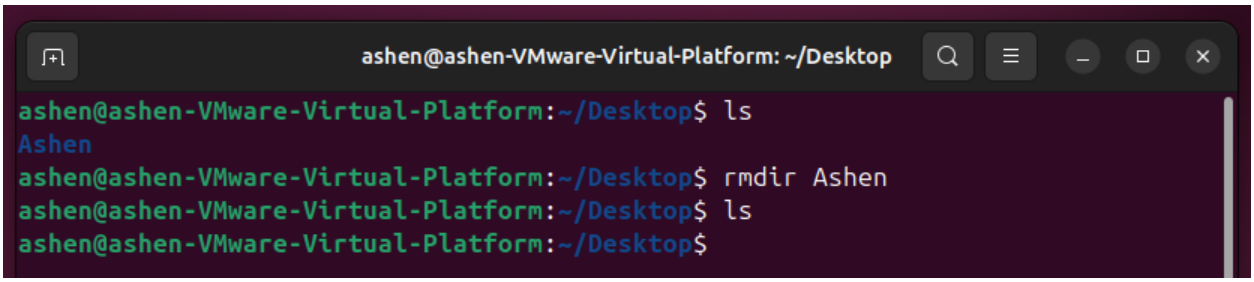
3. **pwd :** This command is used to display the full path of the current directory in which you are working.



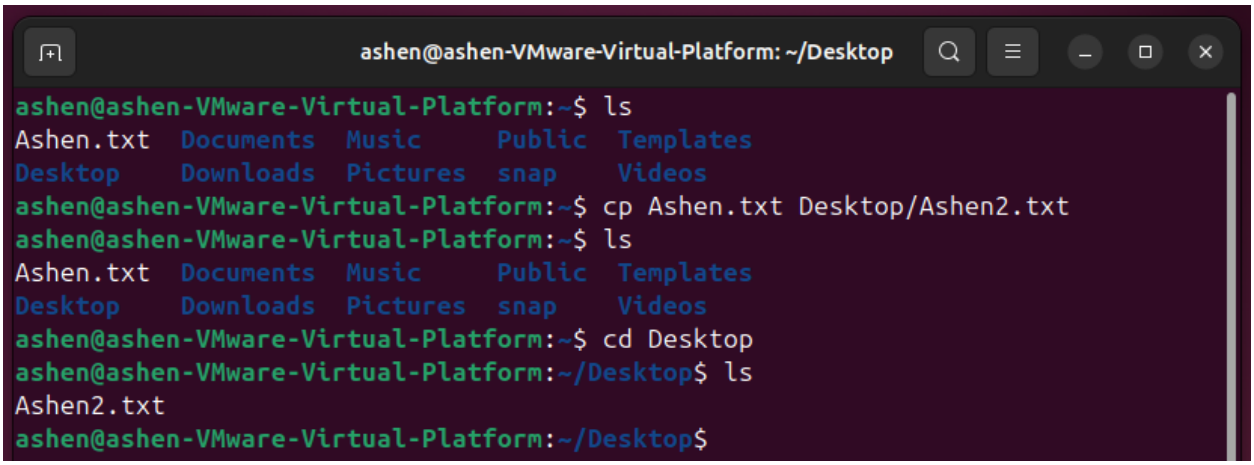
4. **mkdir :** This command is used to create a new directory(folder) in the file system. For example, if you want to create a directory named "Ashen" you have to type: **mkdir** Ashen.



5. **rmdir :** This command is used to remove or delete an empty directory from the file system. For example, if you want to delete a directory named "Ashen" you have to type: **rmdir** Ashen.

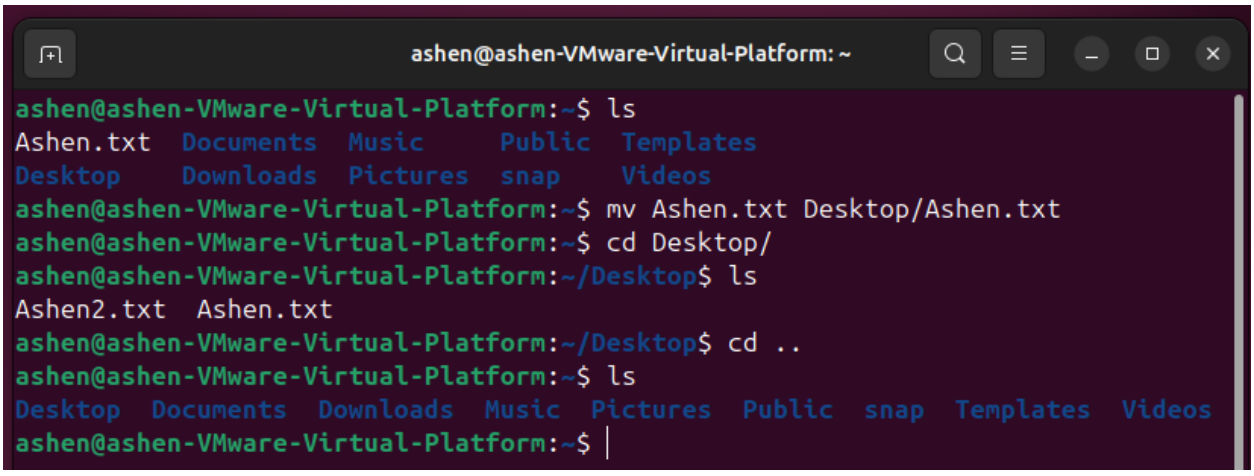


6. **cp :** This command is used for copying and rename files or directories to a different location. For example **cp *source_file* destination_file**

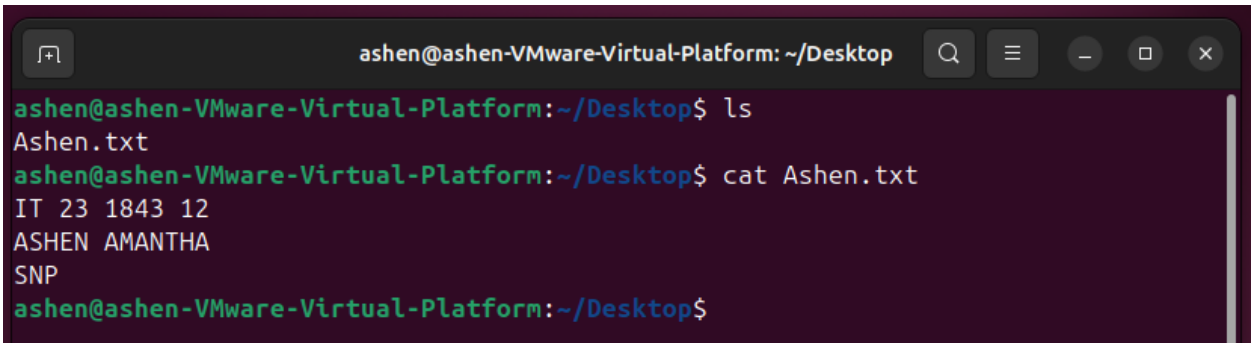


7. **mv :** This command is used to move the files or directories to another location. It also allows you to rename files or subdirectories by operating via the same directory.

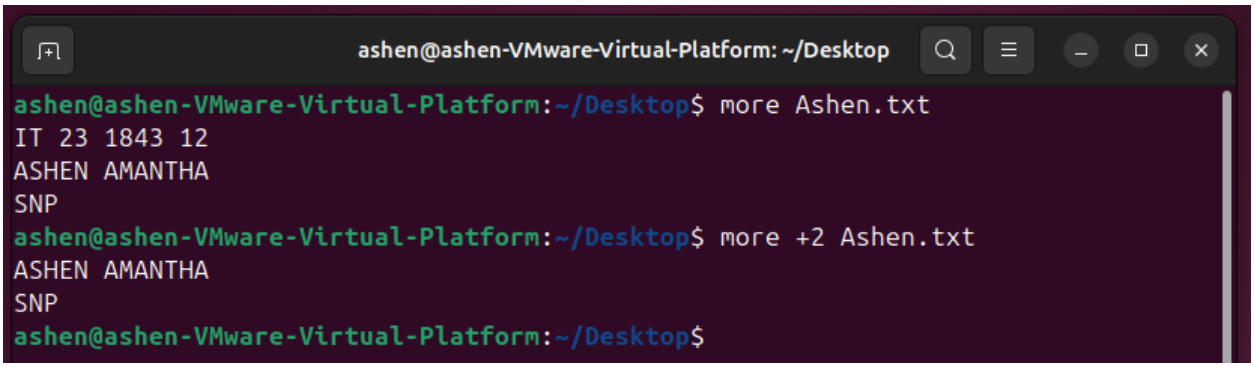
- Simple syntax is **mv *source_file* destination_file** to move a file.



8. **cat :** (concatenate) This command is used to display the contents of a file, most commonly used in text files.

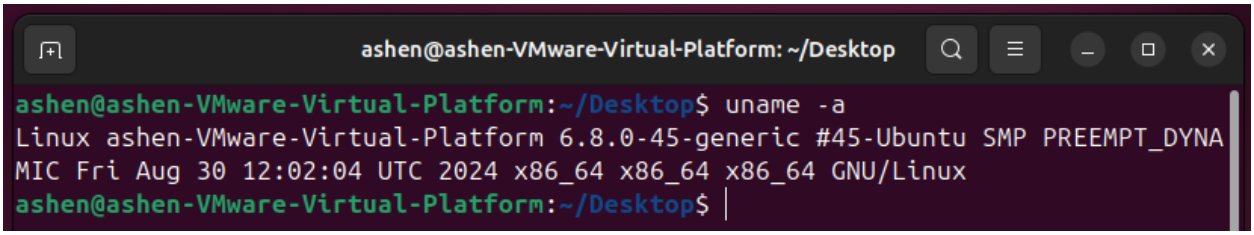


9. **more :** This command is used to view the contents of a text file one screen at a time(but not to edit) It's useful for viewing large files in a terminal window because it allows you to navigate through the file page by page, rather than displaying the entire content at once.

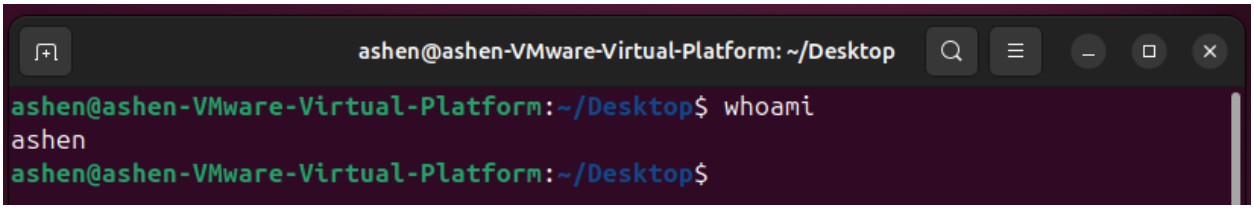


3. 15 Basic commands with brief Description

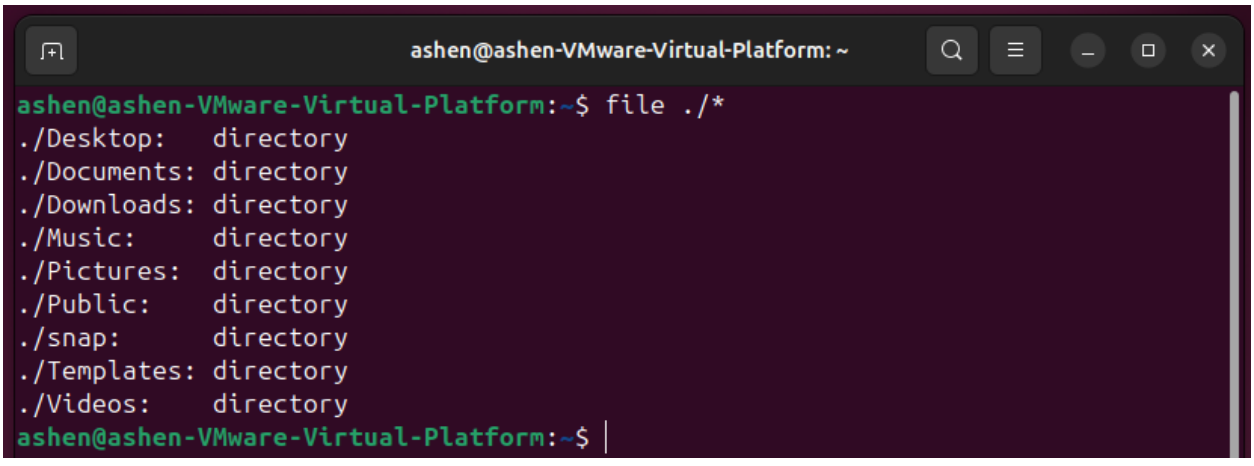
1. **uname -a :** This command shows system details. Will give out details about the operating system like the kernel version, machine hardware name and the operating system name.



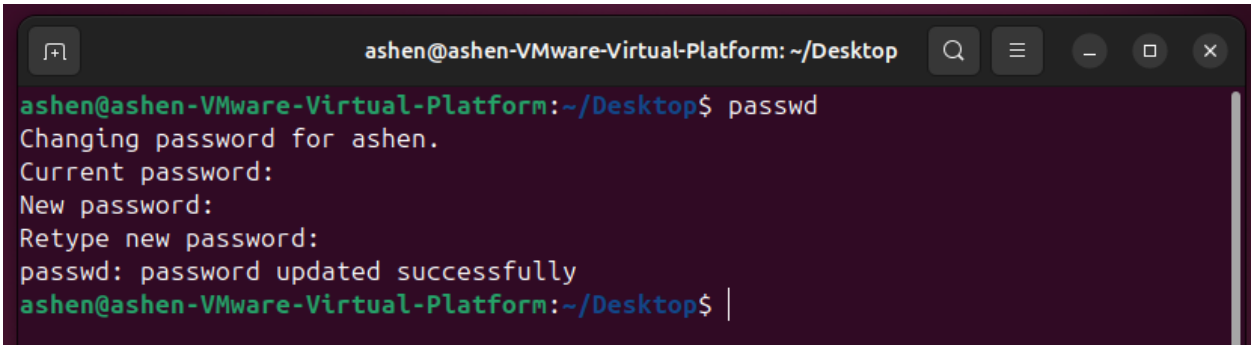
2. **whoami :** This command will print the name of the currently logged in use.



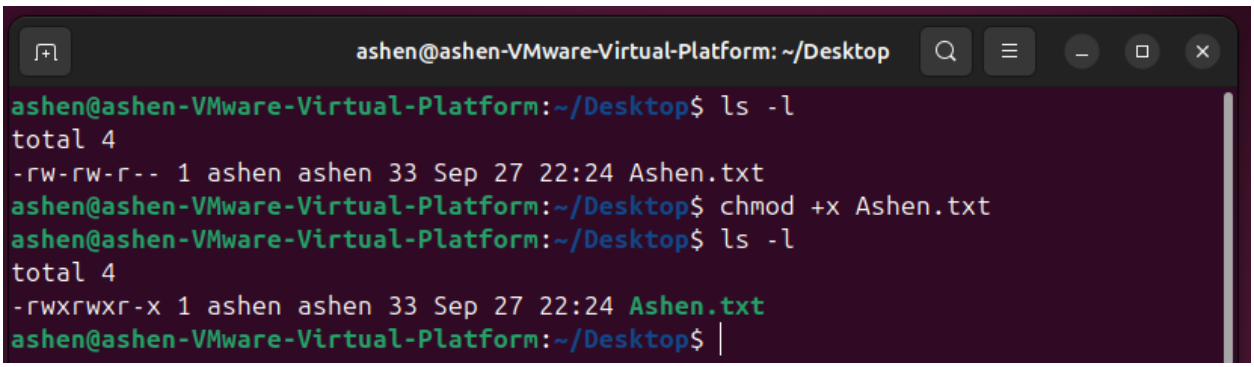
3. **file ./* :** this command determine the type of each file and directory in the current directory.



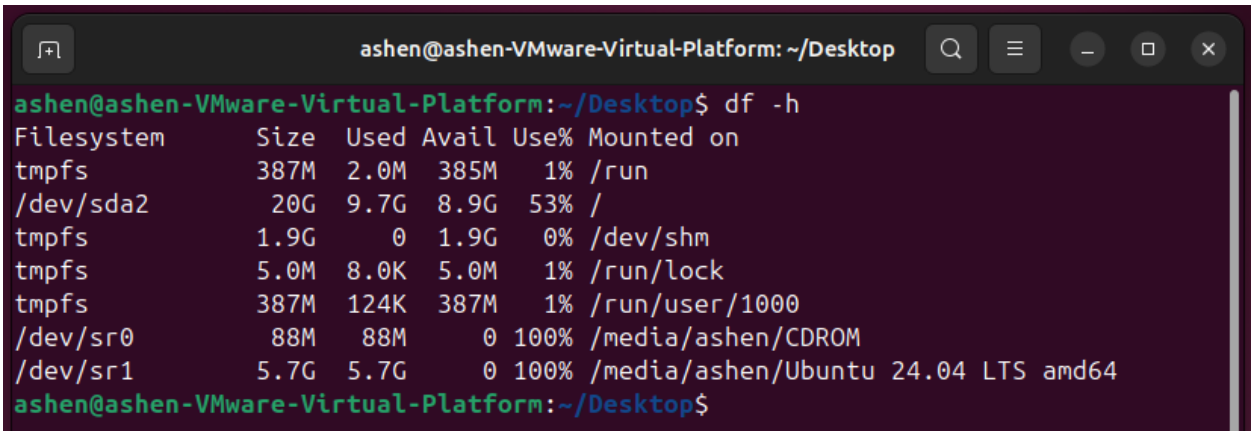
4. **passwd :** This command is used to change the password of the already logged in account.



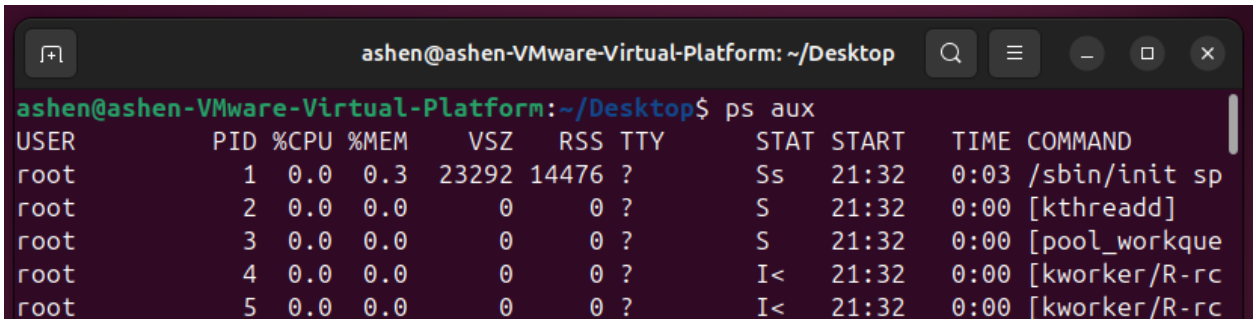
5. **chmod :** This command is used to change file or directory permissions.



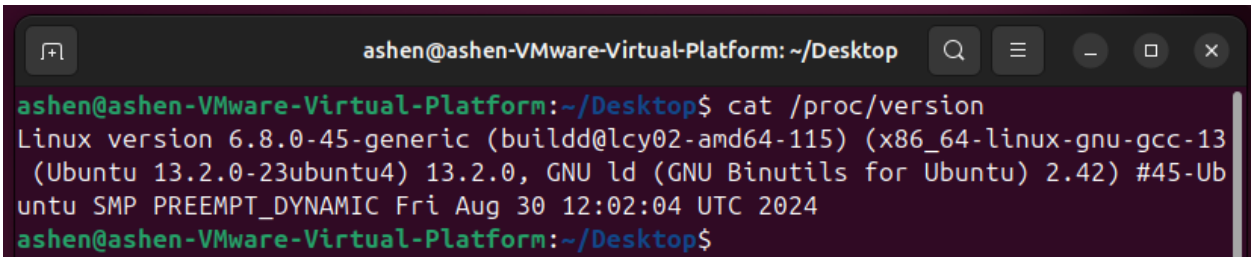
6. **df -h :** This command shows the disk space usage in a human-readable format.



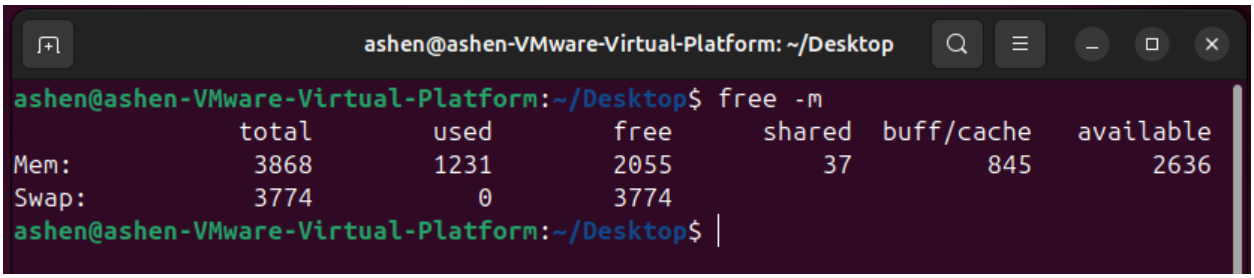
7. **ps aux :**This command lists currently running processes on the system .



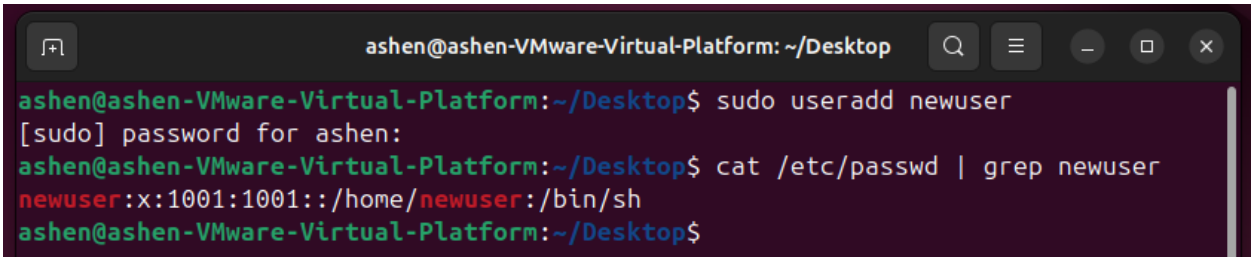
8. **cat /proc/version :** This command displays the version information of the Linux kernel and related details about the system. The **/proc/version** file contains the kernel version, the compiler used to build the kernel etc.



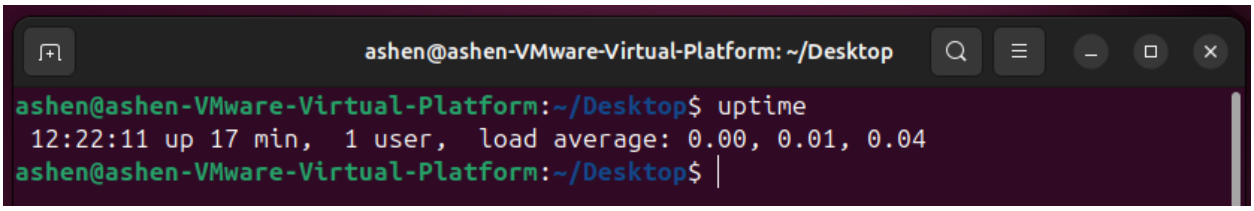
9. **free -m :** This command in Linux displays information about the system's memory usage, both RAM and swap space, in megabytes



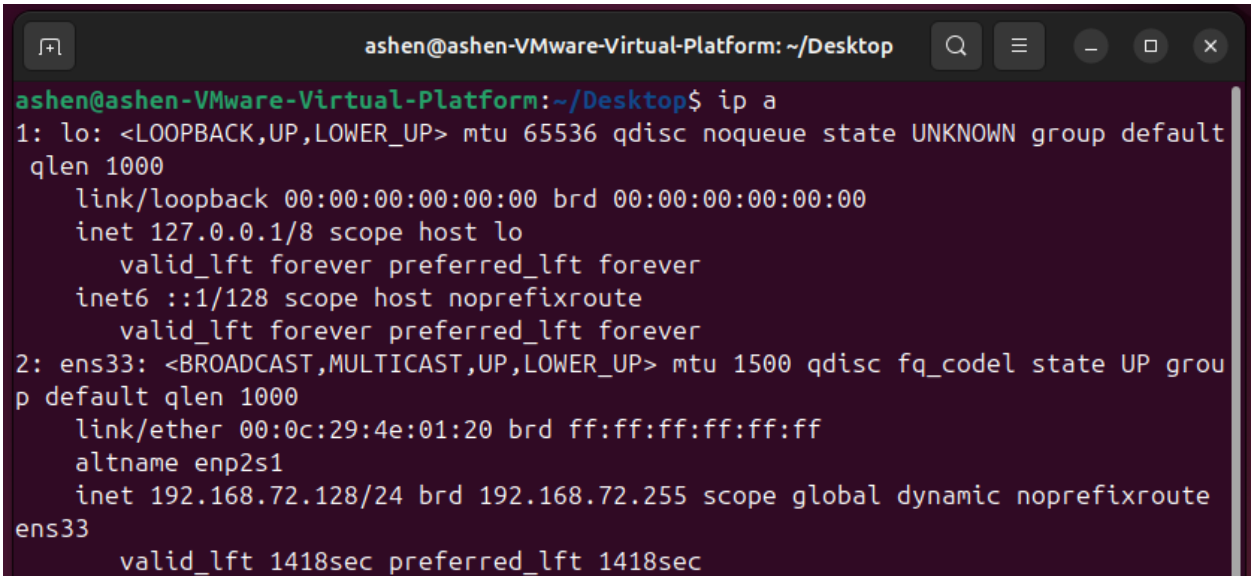
10. **useradd :** This command in Linux is used to create a new user account on the system.



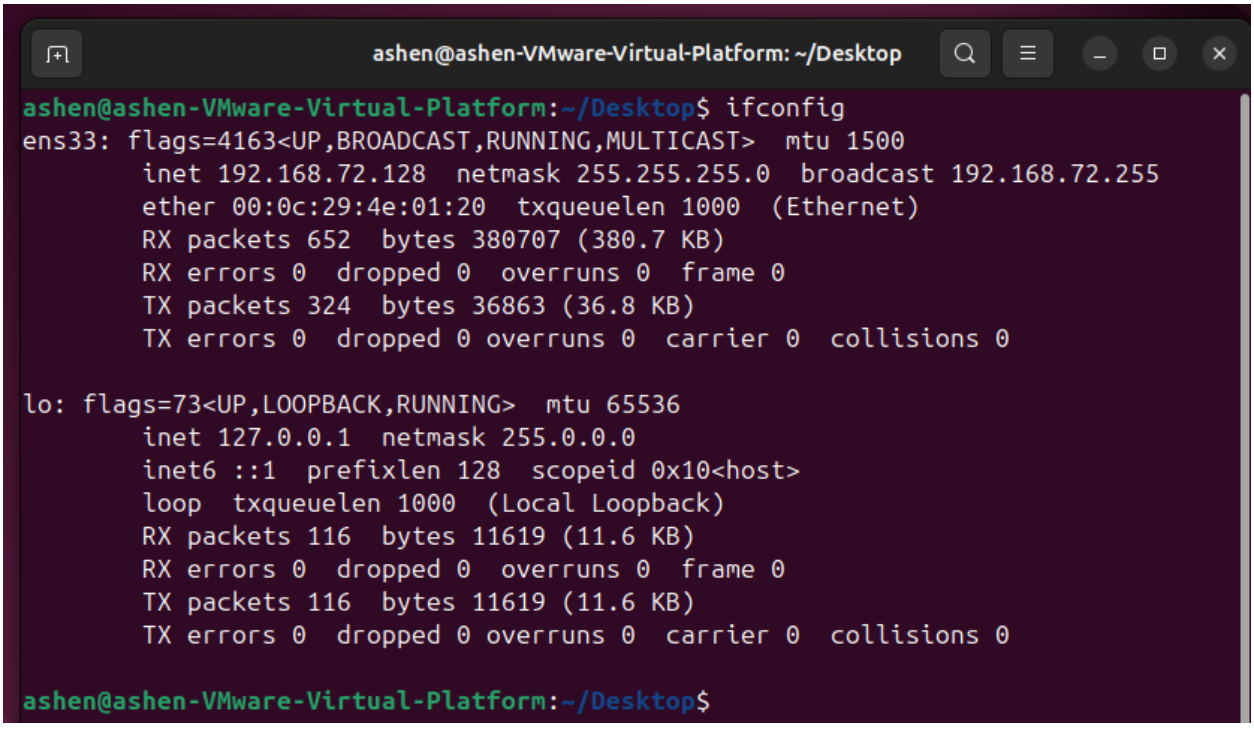
11. **uptime :** This command shows how long the system has been running, along with the number of users and load averages.



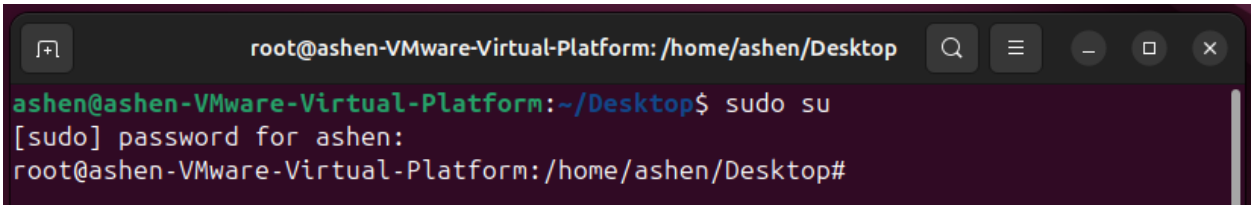
12. **ip a :** This command used to display all network interfaces and their IP addresses.



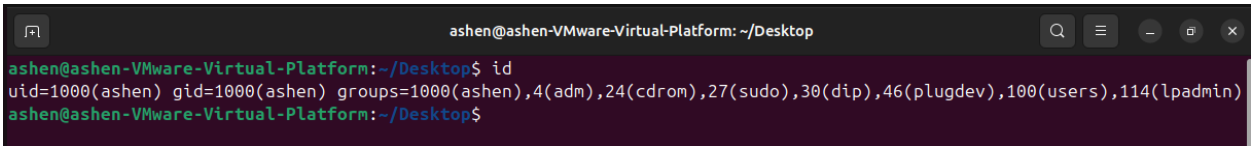
13. **ifconfig :** This command used to display network interface configurations



14. **sudo su :** This command is used to switch to Root User



15. **id :** This command is useful for quickly checking user and group IDs, which can help in managing permissions and understanding user roles in a Linux system.

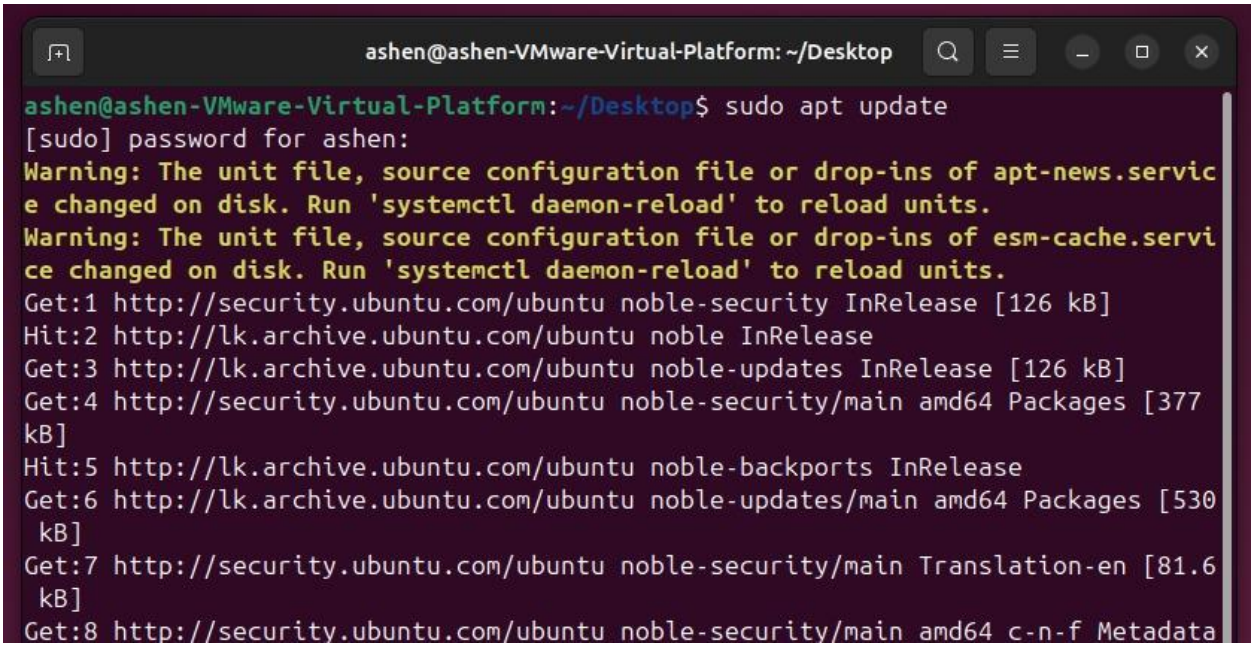


4.Configuration steps for DHCP, DNS and NTP Services

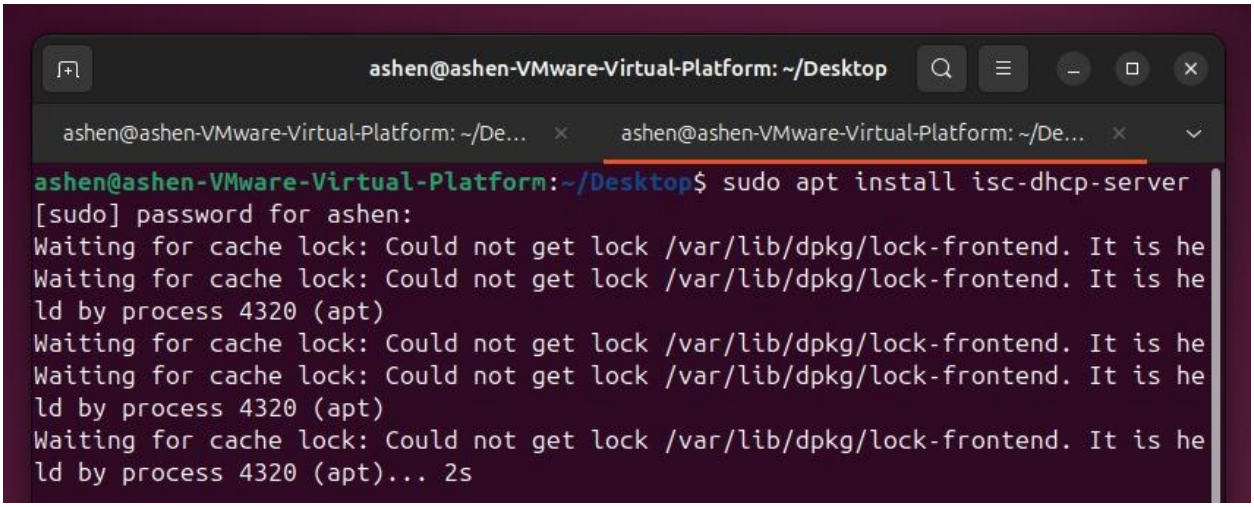
DHCP(Dynamic Host Configuration Protocol) Installation and Configuration

➢ *DHCP Server Configuration*

1. Update the package list.

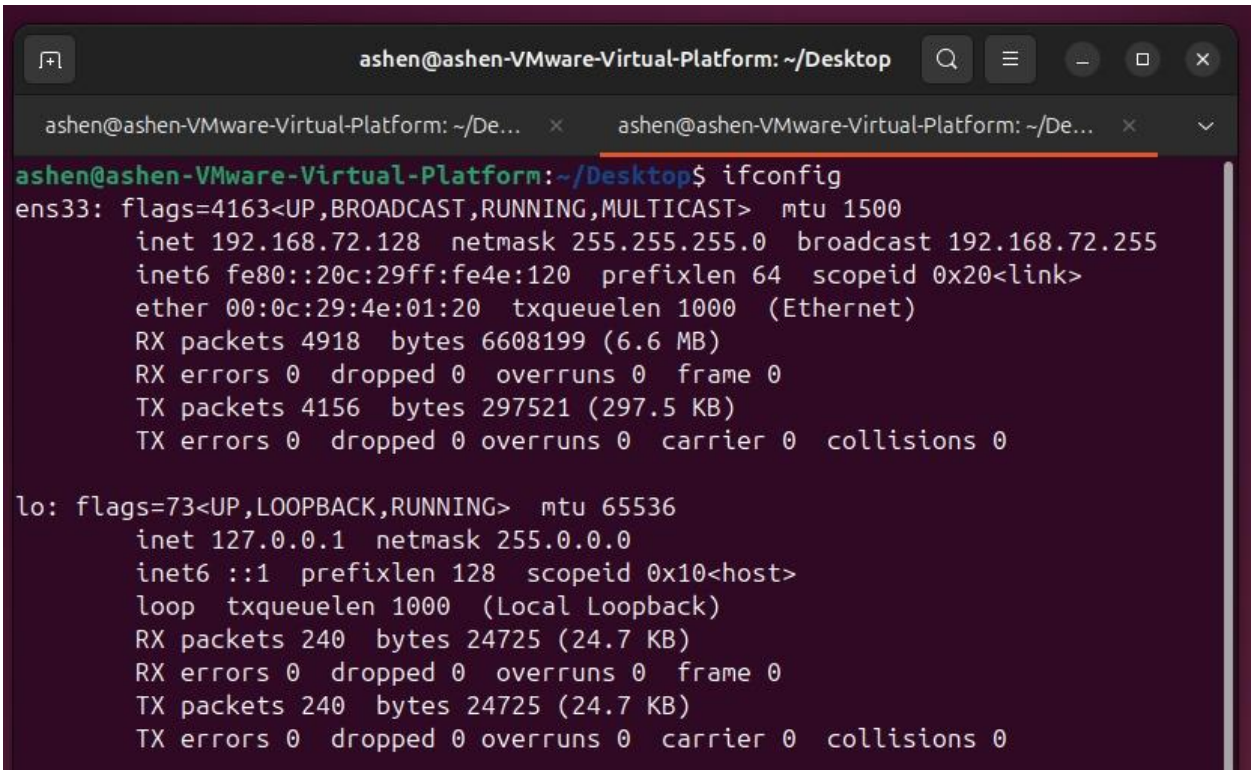


2. Install the **isc-dhcp-server** package

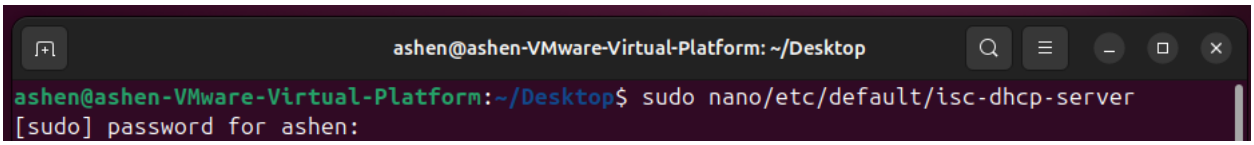


3. Use **ifconfig** command to view the network interfaces and record the NIC.

(if command not found install in with **sudo apt install net-tools**)

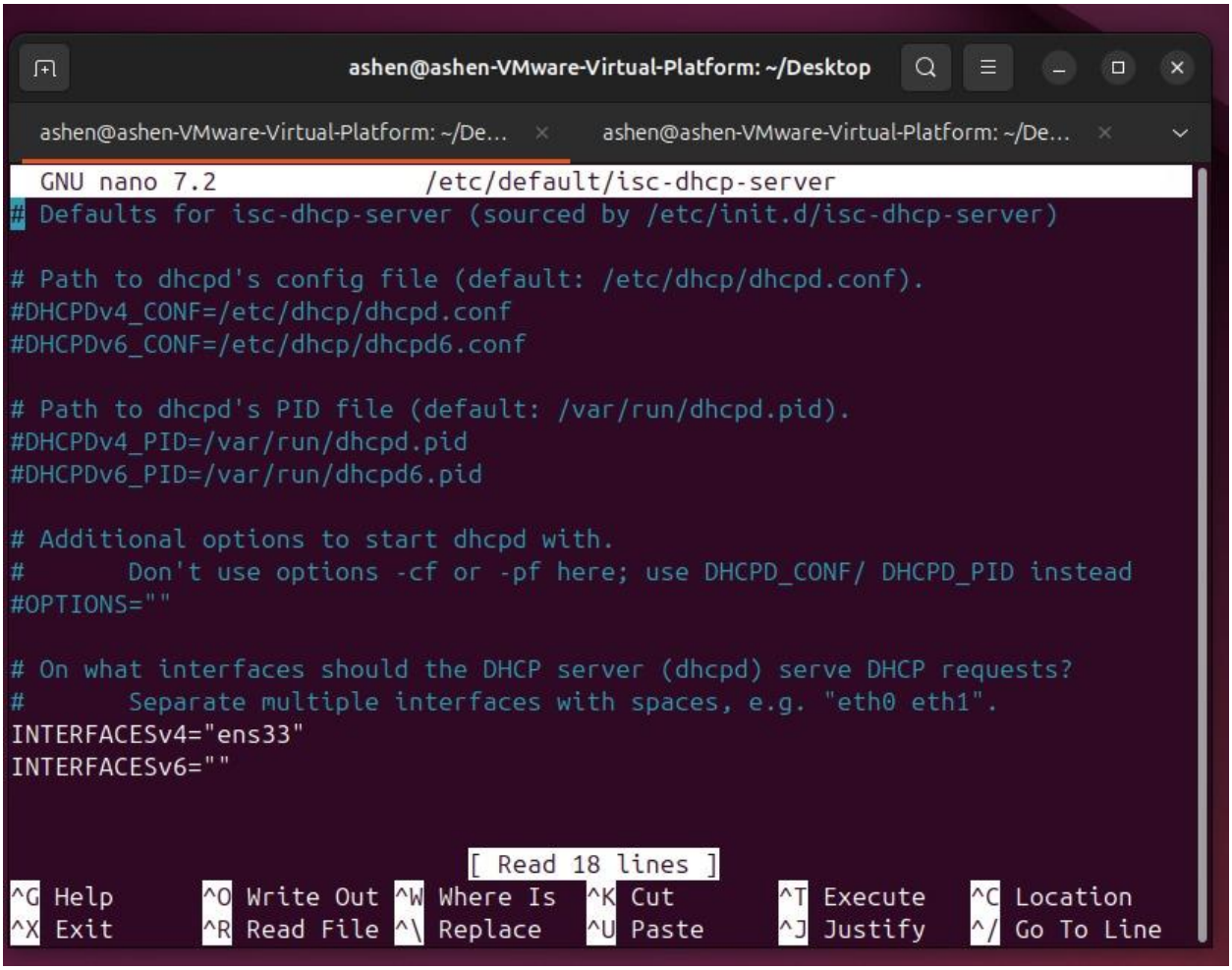


4. Open the configuration file for the ISC DHCP server using the nano text editor.

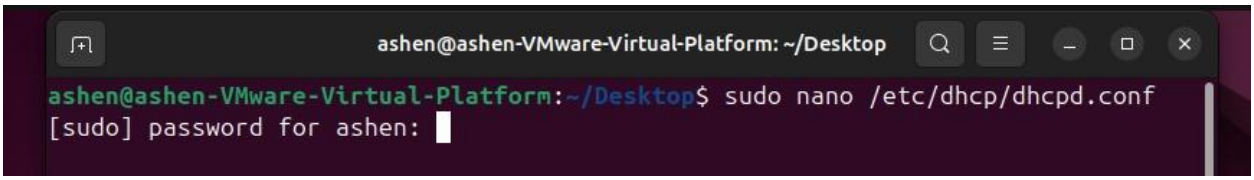


5. Modify the details for INTERFACESv4 inside the configuration file

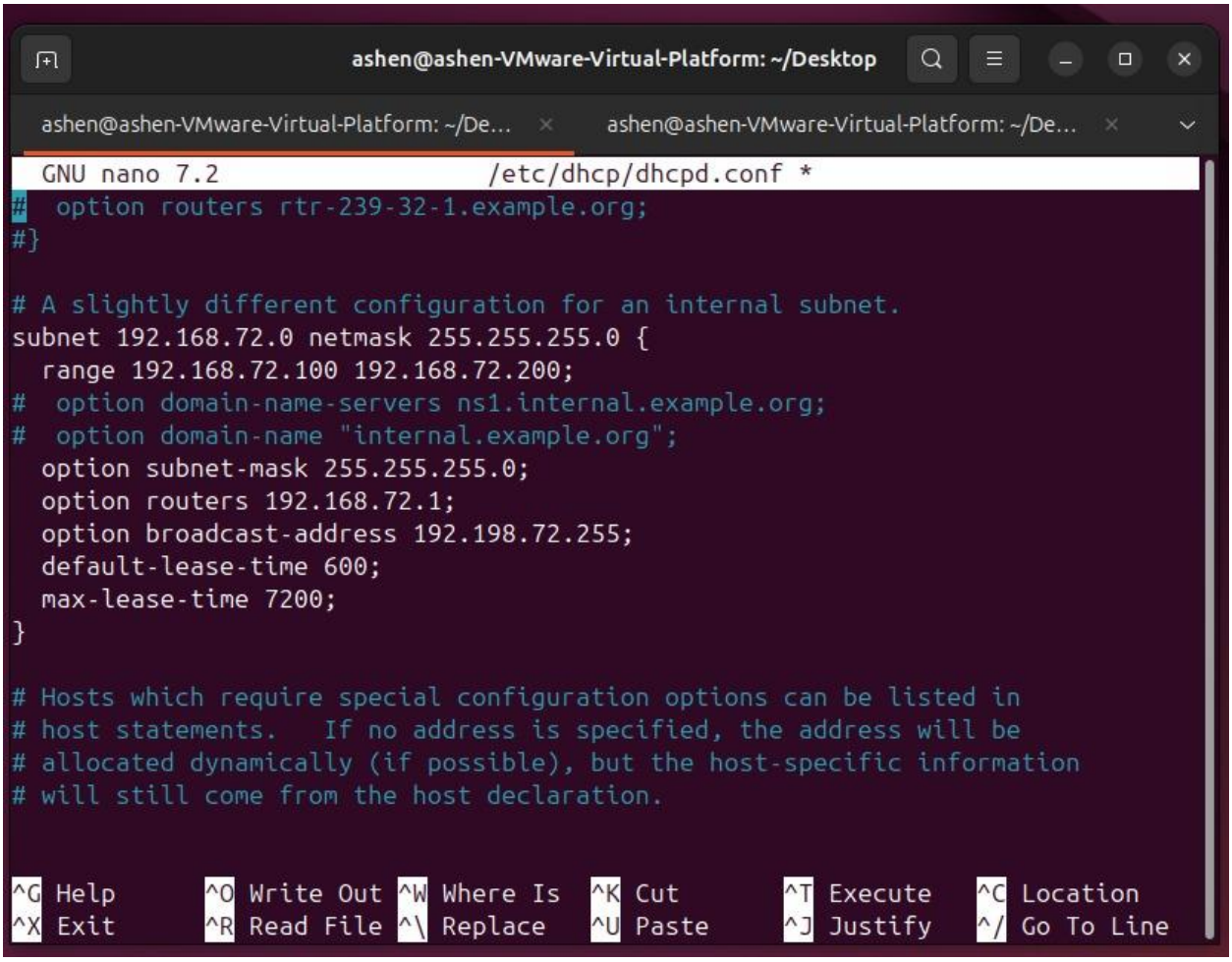
- Here we have to specify which network interfaces the DHCP server should listen on to provide IP addresses to clients.
- By setting INTERFACESv4="ens33"→ the DHCP server listens for DHCP requests on the ens33 network interface.
- By keeping INTERFACESv6=" " variable empty →this implies that the DHCP server will not handle IPv6 DHCP requests.



6. Now open the main configuration file of DHCP.

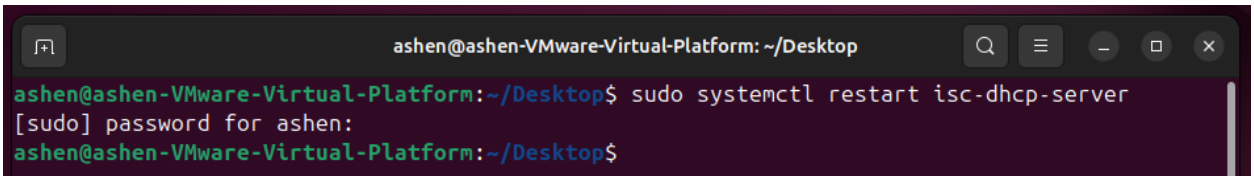


7. Modify and save the details inside the main configuration file according to our NIC to complete the DHCP configuration.

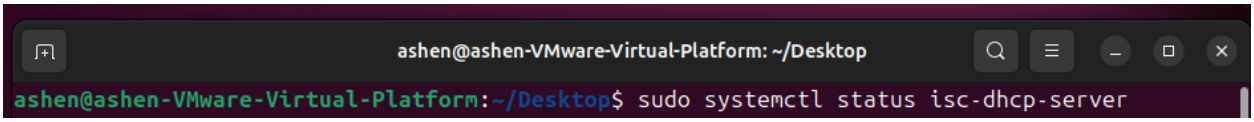


8. Restart the DHCP server.

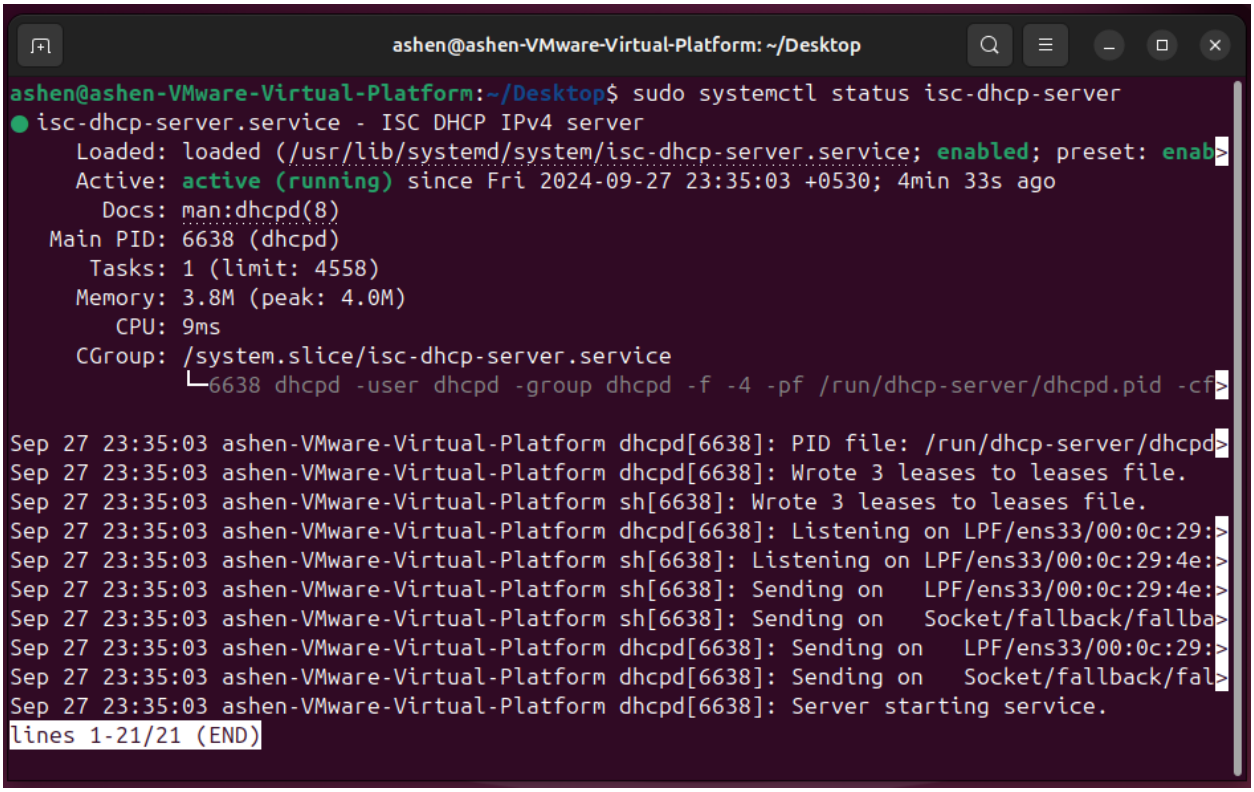
- sudo systemctl restart isc-dhcp-server



9. To verify the configuration check the status of the DHCP server.

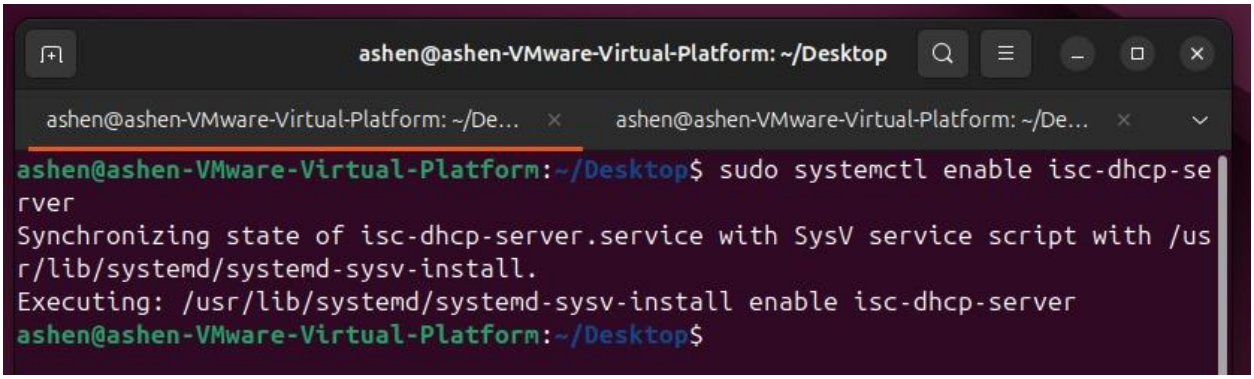


10. If the configuration is successful, it is indicated with a status: active



11. Enable the service for the boot time (ensure that the DHCP server (isc-dhcp-server) starts automatically whenever the system is booted)

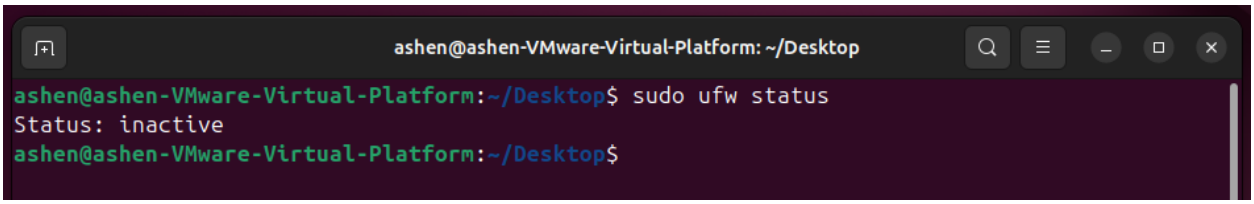
- sudo systemctl enable isc-dhcp-server



12. Check the current status of the firewall.

- sudo ufw status

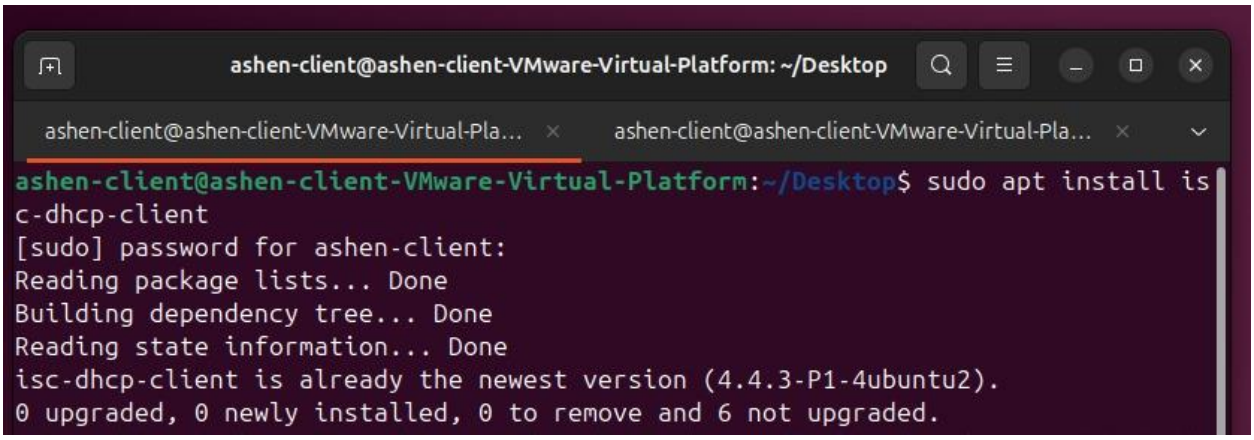
If the firewall status is inactive, the server will be able to communicate with clients without any restrictions.



➢ *DHCP Client Configuration*

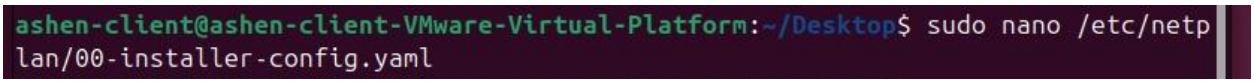
1. Install the DHCP client on another separate virtual machine (Linux distribution)

- sudo apt install isc-dhcp-client

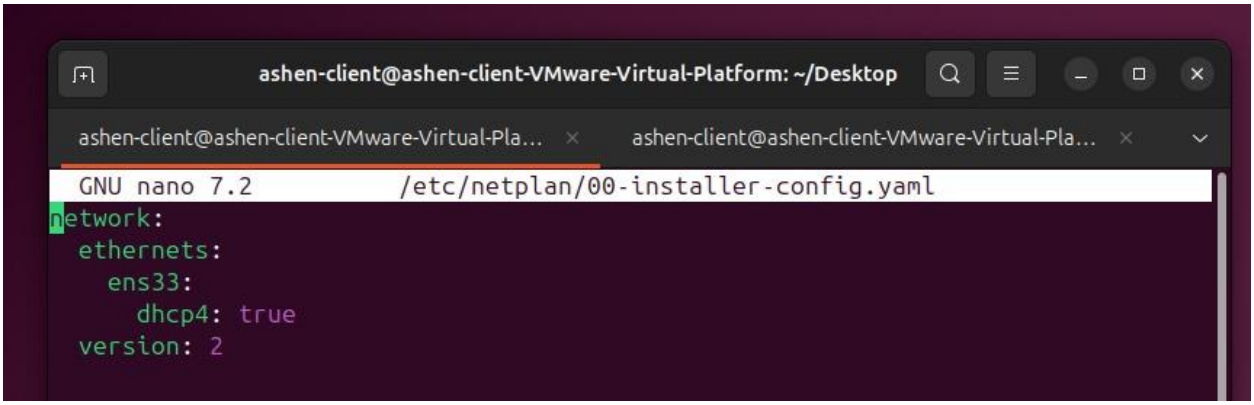


2. Configure the client to obtain an IP Address via DHCP

- Open the configuration file on the client.

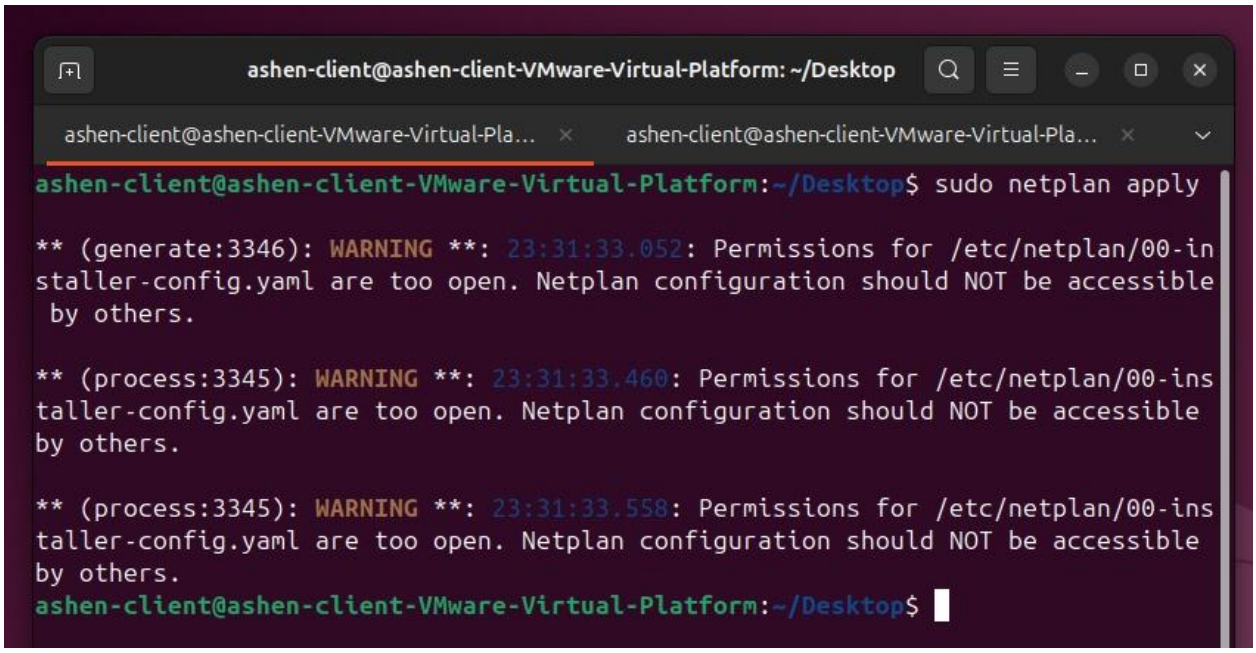


- Modify the configuration file.



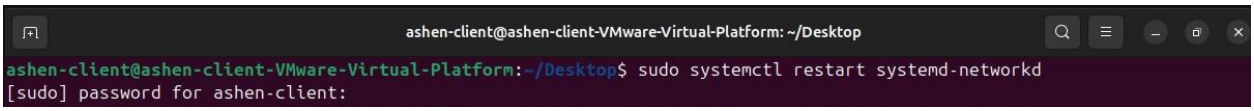
3. Apply the changes .

- sudo netplan apply



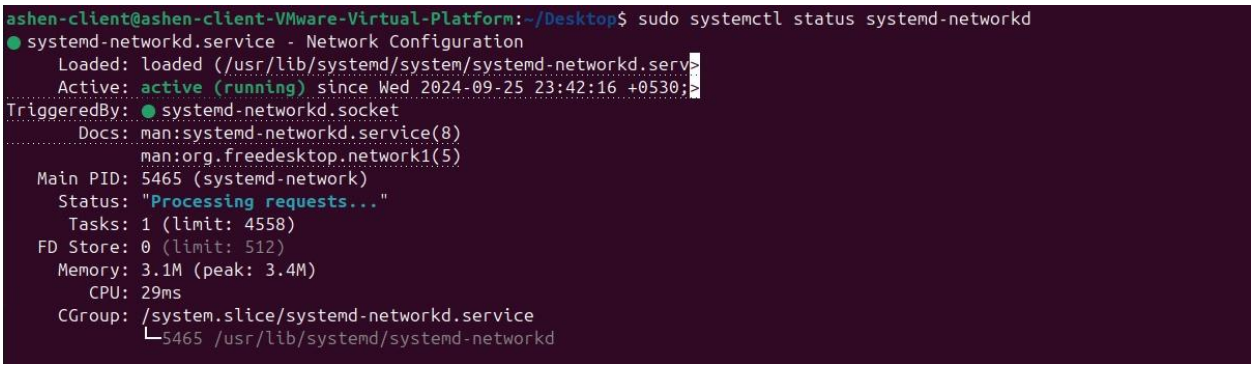
4. Restart the service to apply the changes.

- sudo systemctl restart systemd-networkd



5. To verify the configuration check the status.

- status : active

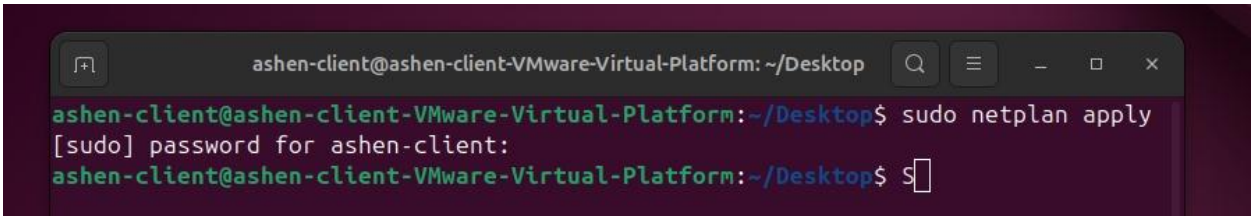


6. Enable the service for the boot time.

- sudo systemctl enable system-networkd

```
ashen-client@ashen-client-VMware-Virtual-Platform:~/Desktop$ sudo systemctl enable systemd-networkd
Created symlink /etc/systemd/system/dbus-org.freedesktop.network1.service → /usr/lib/systemd/system/systemd-networkd.ser
vice.
Created symlink /etc/systemd/system/multi-user.target.wants/systemd-networkd.service → /usr/lib/systemd/system/systemd-n
etworkd.service.
Created symlink /etc/systemd/system/sockets.target.wants/systemd-networkd.socket → /usr/lib/systemd/system/systemd-netwo
rkd.socket.
Created symlink /etc/systemd/system/sysinit.target.wants/systemd-network-generator.service → /usr/lib/systemd/system/sys
temd-network-generator.service.
Created symlink /etc/systemd/system/network-online.target.wants/systemd-networkd-wait-online.service → /usr/lib/systemd/
system/systemd-networkd-wait-online.service.
```

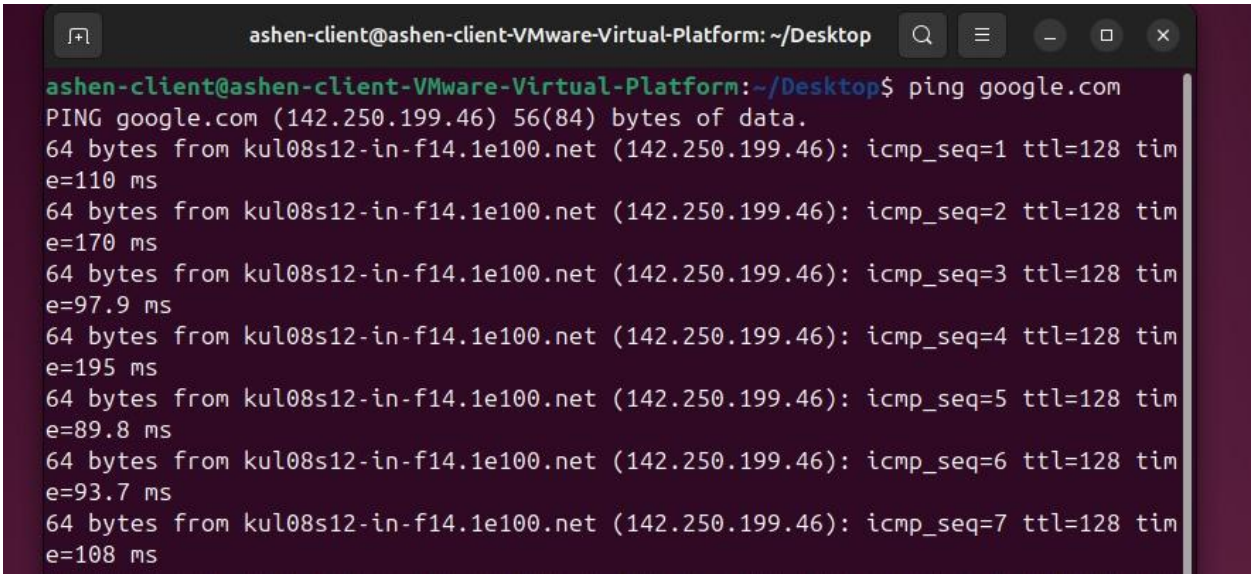
7. Reapply the netplan.



Verification

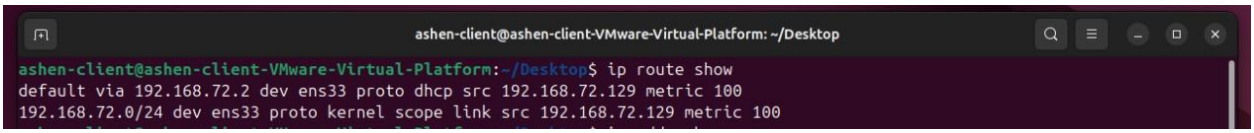
Check Network Connectivity

Try to check your internet or network connectivity by pinging an external website or the gateway provided by the DHCP server.

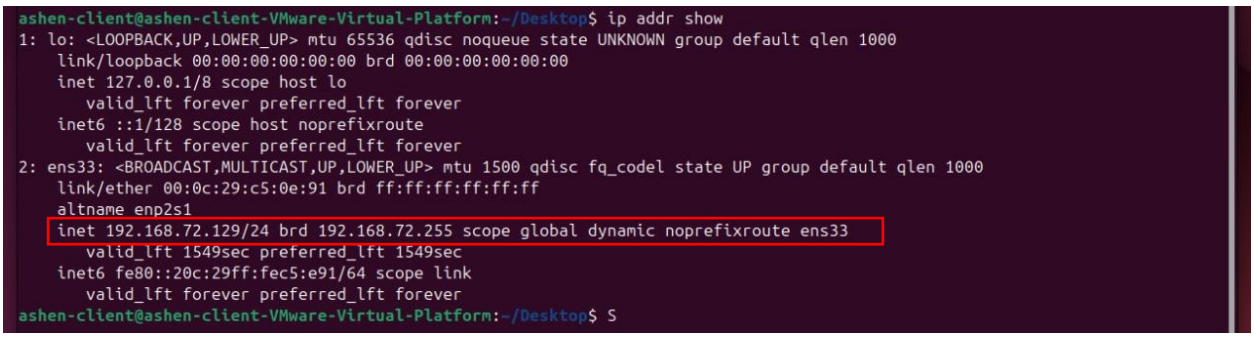


(The ping google.com command's output verifies that your Ubuntu client, configured as a DHCP client has effectively acquired network connectivity and can thus interact with outside networks).

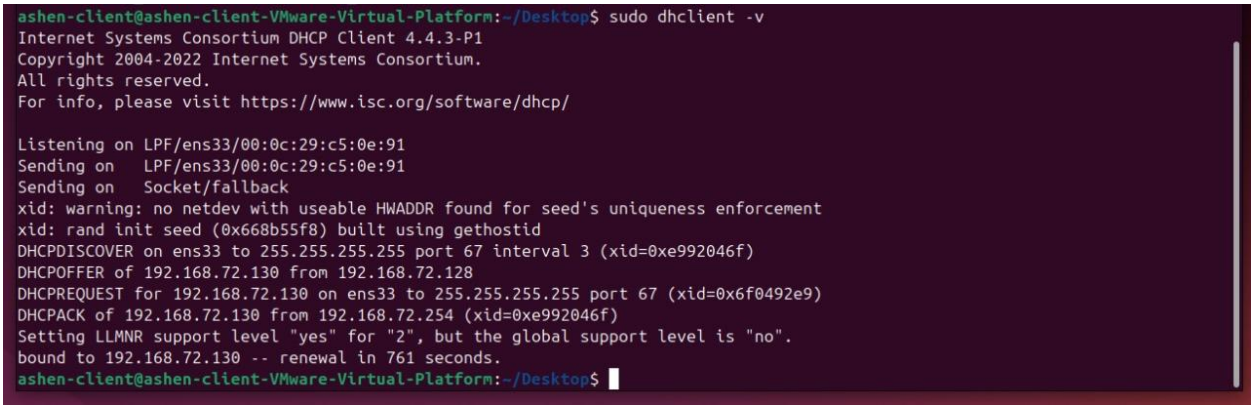
To check the default gateway



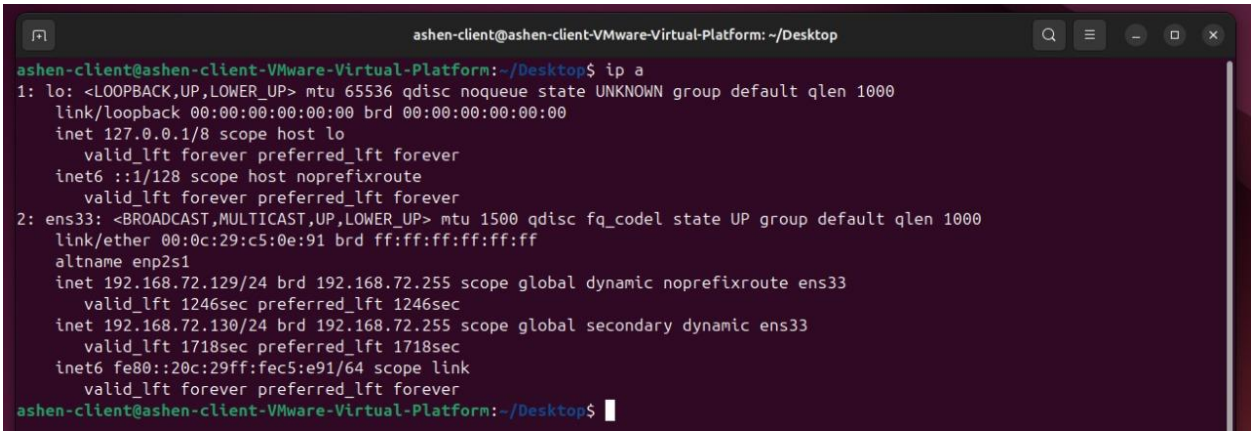
Check the status of your network interfaces and whether they've obtained IP addresses



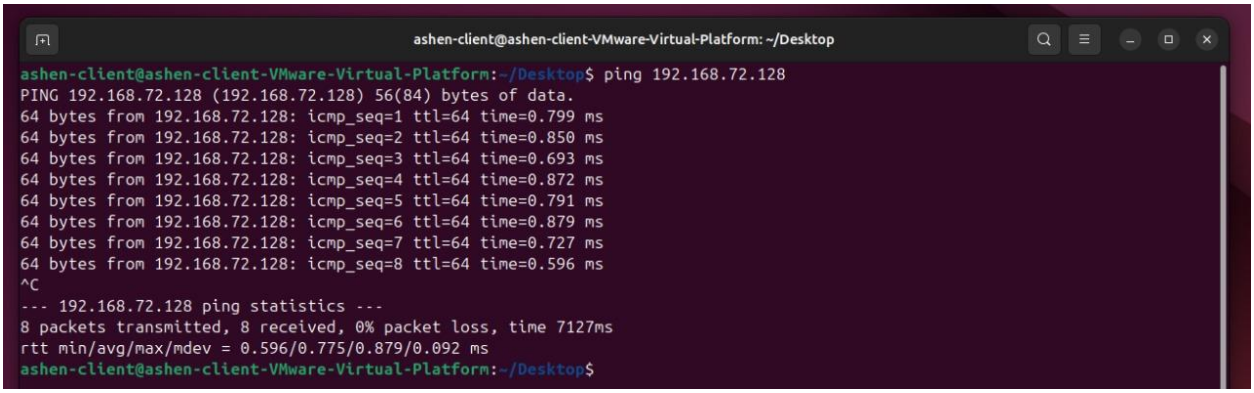
Check if the DHCP Client Obtained an IP Address.



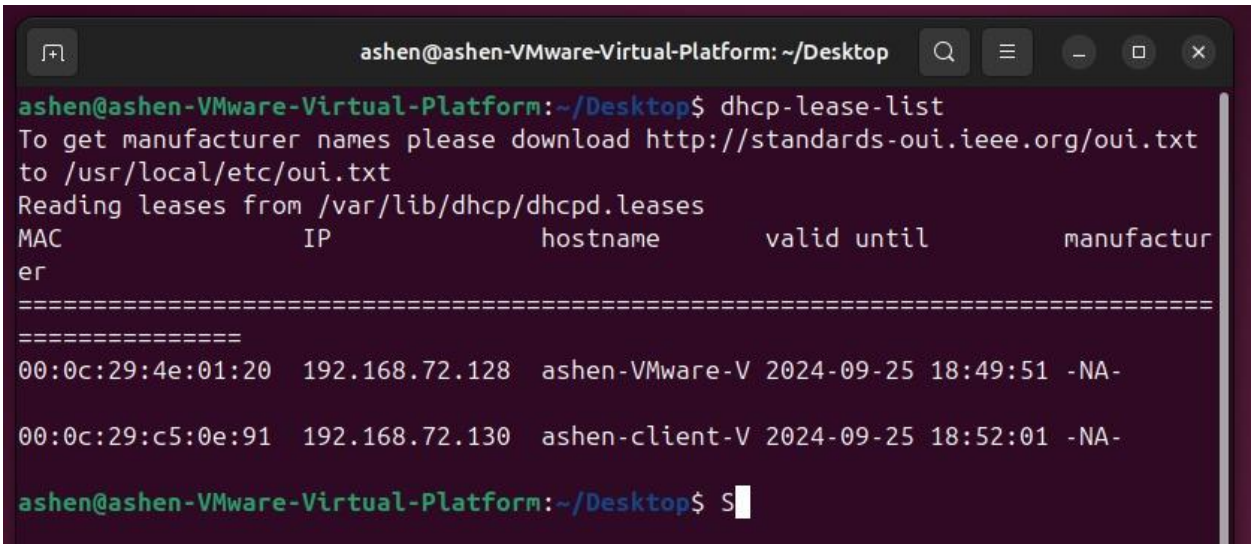
Verify the DHCP client setup



Get the dhcp server ip address and ping it in the dhcp client



Finally, in the dhcp server/ubuntu server get a list of the clients that are connected to the server.



DNS(Domain Name System Protocol) Installation and Configuration

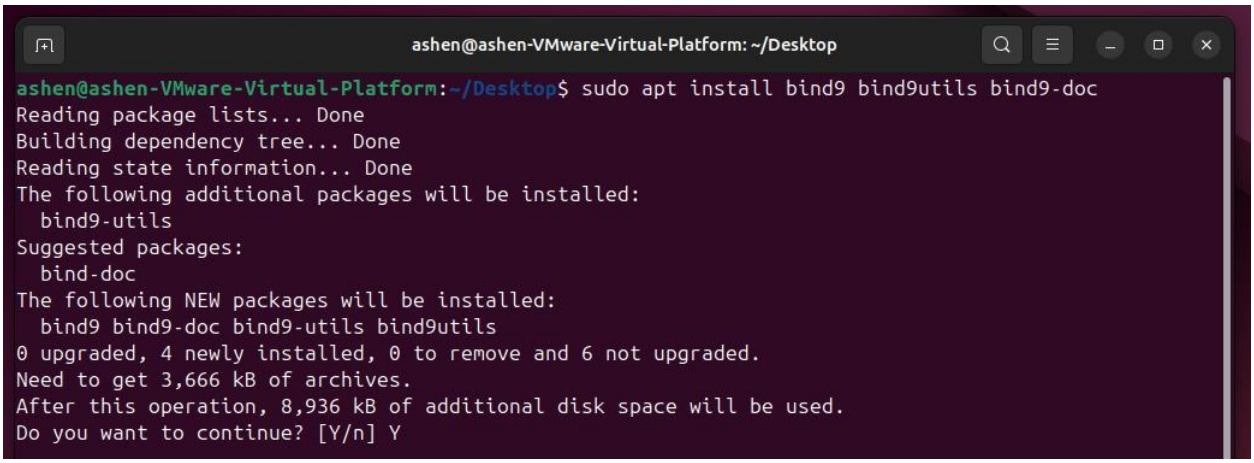
➢ DNS Sever Configuration

1. Update the package list

- sudo apt update

2. Install BIND9 (BIND9 DNS server and related utilities on a Debian-based system like Ubuntu)

- sudo apt install bind9 bind9utils bind9-doc



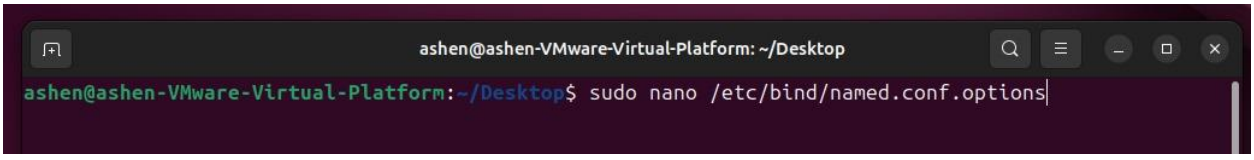
3. Configure DNS forwarders

- When your DNS server receives a query it cannot resolve locally, it will forward the query to another DNS server.

- If your DNS server also needs to resolve external domain names DNS forwarders is necessary.

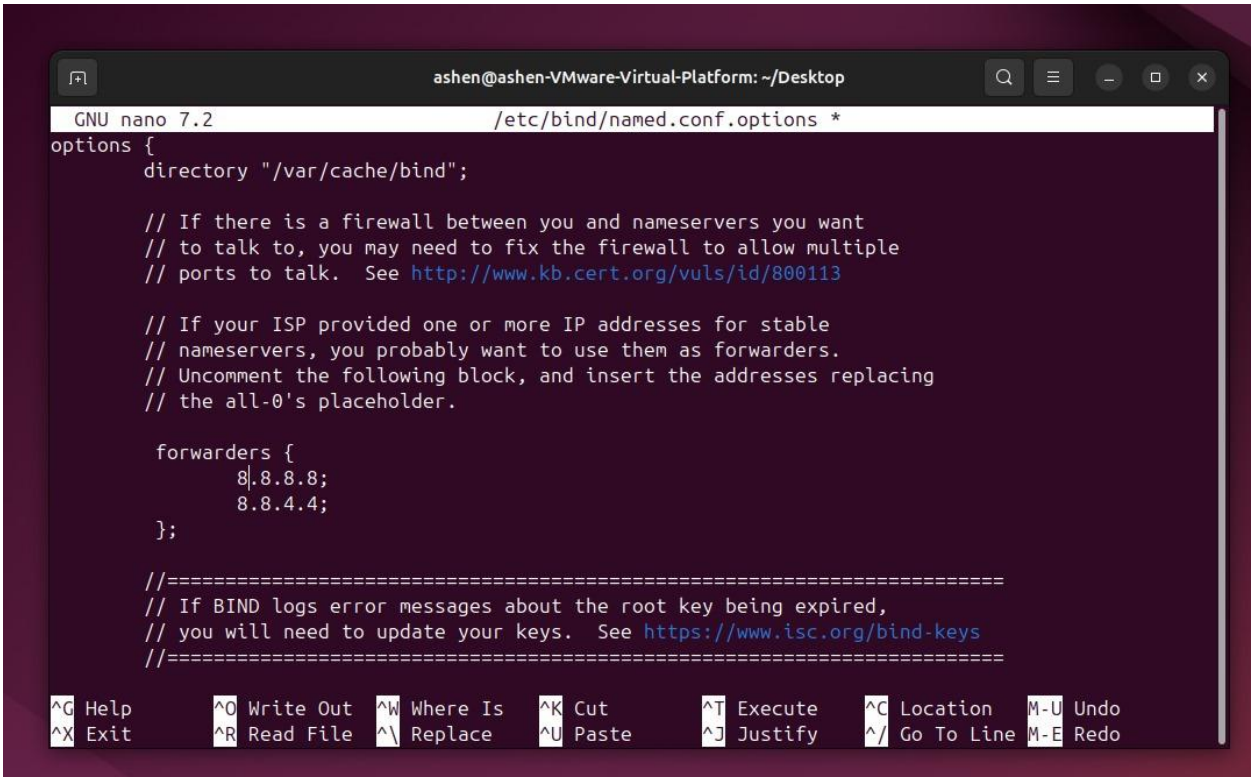
a) Open the configuration file.

- sudo nano /etc/bind/named.conf.options



b) Modify the forwarders section

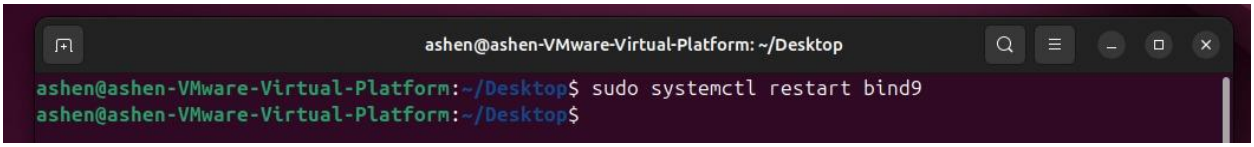
- we can add Google's public DNS servers or Cloudflare's public DNS servers.
- here I have added Google's public and Secondary DNS servers



c) Save and Exit.

d) Restart the service to apply changes.

- sudo systemctl restart bind9

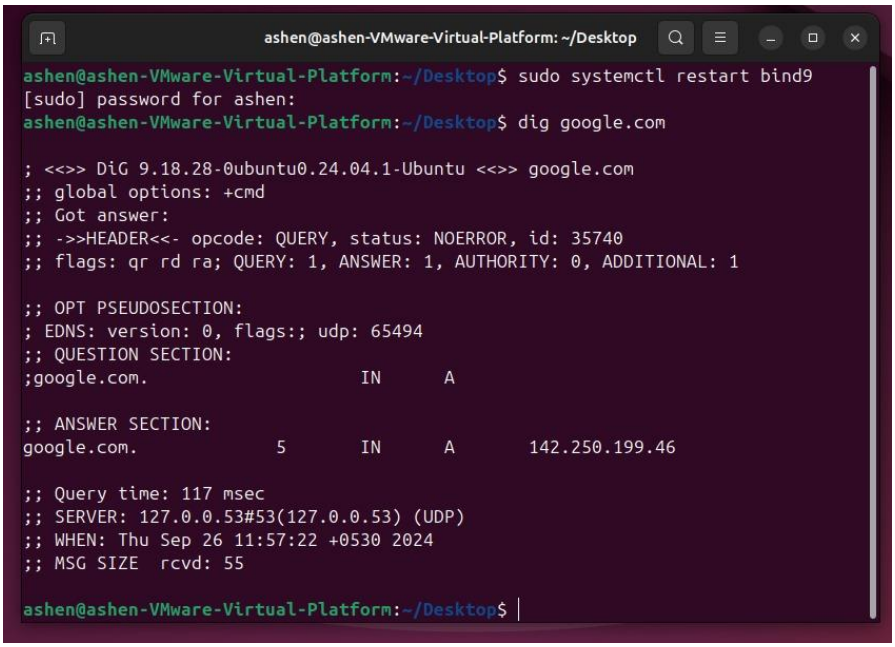


Verification of Forwarders

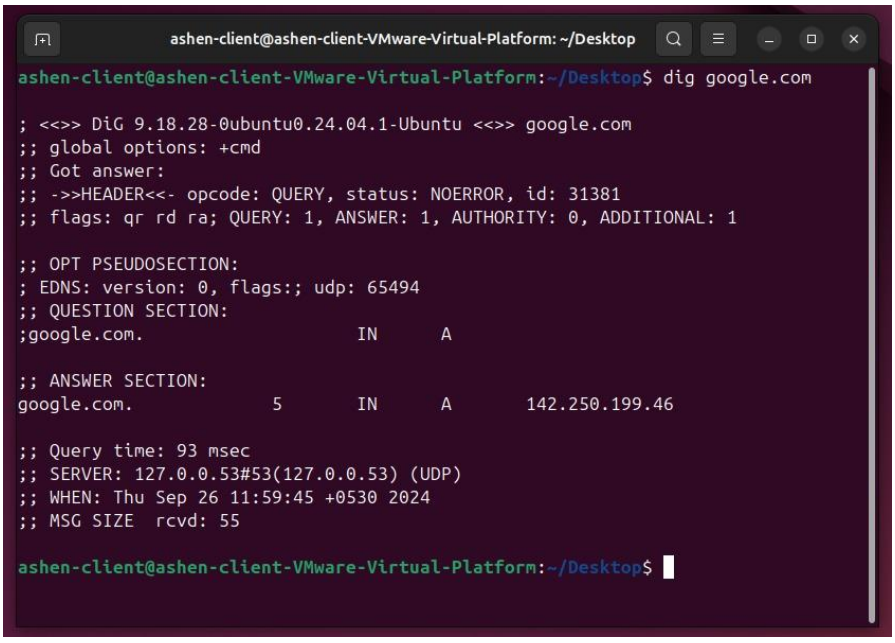
The **dig** tool is one of the most popular DNS troubleshooting tools. It allows you to query DNS servers and see how they respond.

- dig google.com

if this returns IP address of google.com , the forwarders are working correctly.



Server



Client

➢ DNS Client Configuration

1. Install the DNS Client

    1. Update the package list.

        - sudo apt update

    2. Install resolvconf if necessary.

        - sudo apt install resolvconf

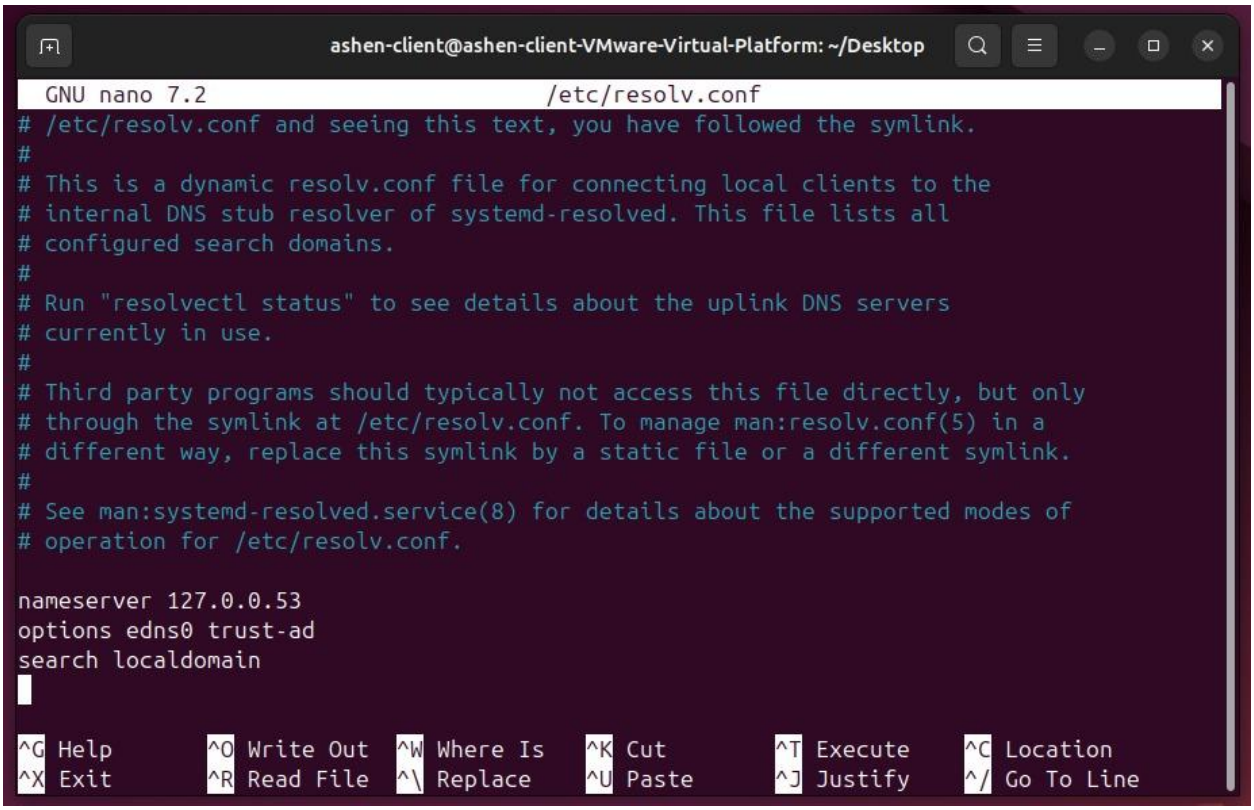
2.Configure the DNS client settings

    1. Open server VM conf file and note down the ip address of DNS server.

        - cat /etc/resolv.conf

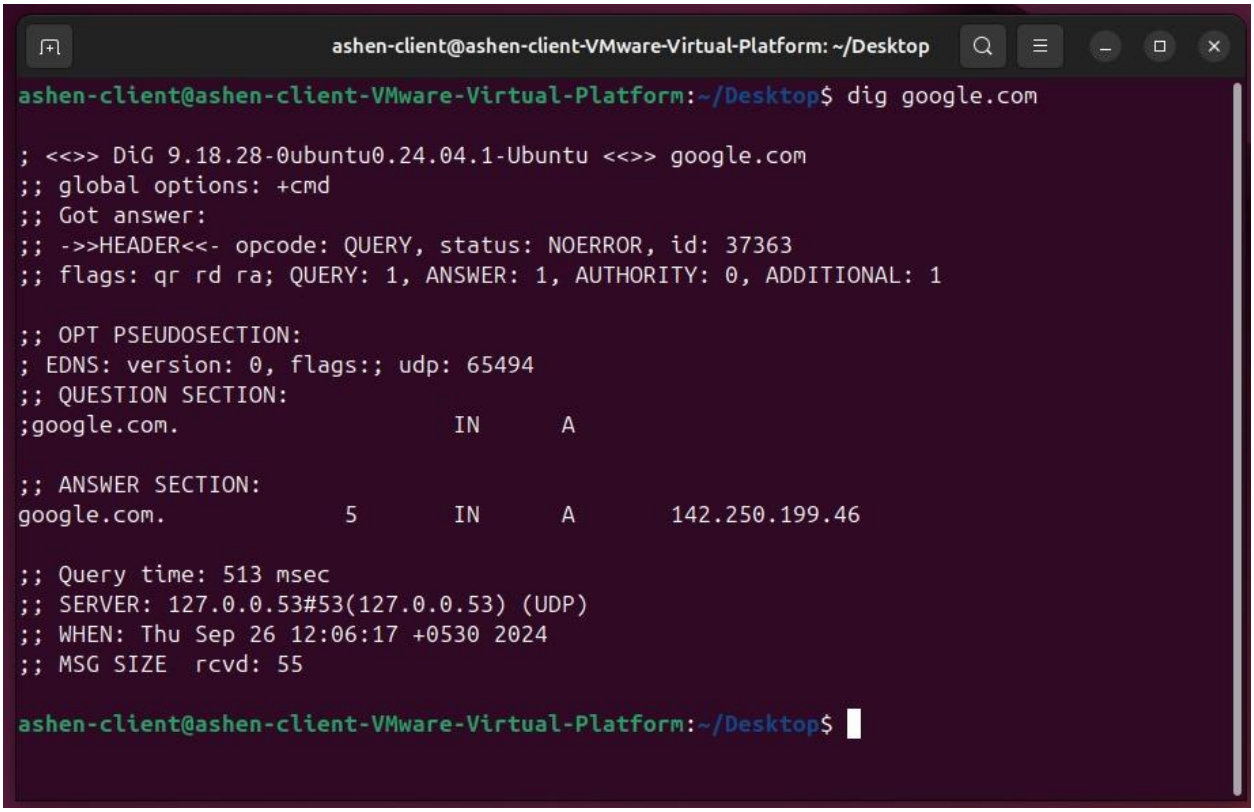
    2. Set the DNS server as the nameserver of the client VM .

        - sudo nano /etc/resolv.conf



3. Test DNS Resolution

- To test DNS resolution on your system, you can use tools like nslookup, dig, and ping.

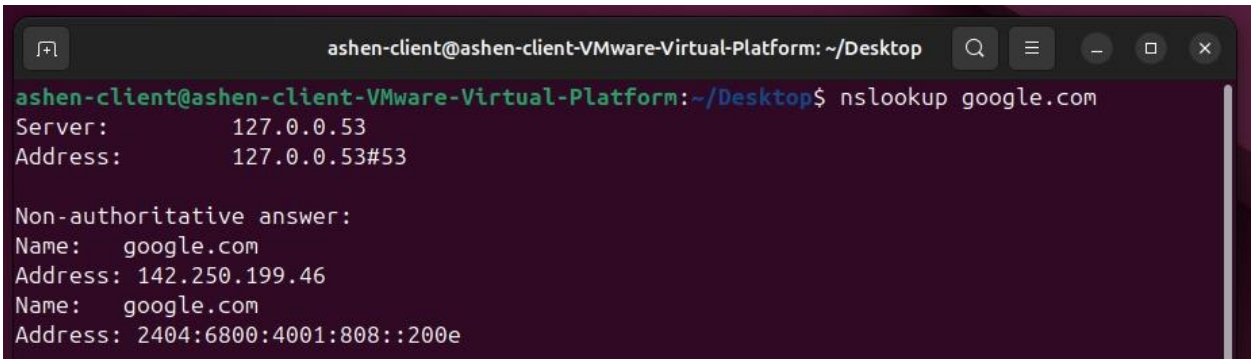


ANSWER SECTION : This shows the resolved IP address for the domain

SERVER : Indicates which DNS server was used for the query

- **nslookup**

- **Address :** This shows the resolved IP address for the domain
- **Server :** Indicates which DNS server was used for the query



NTP(Network Time Protocol) Installation and Configuration

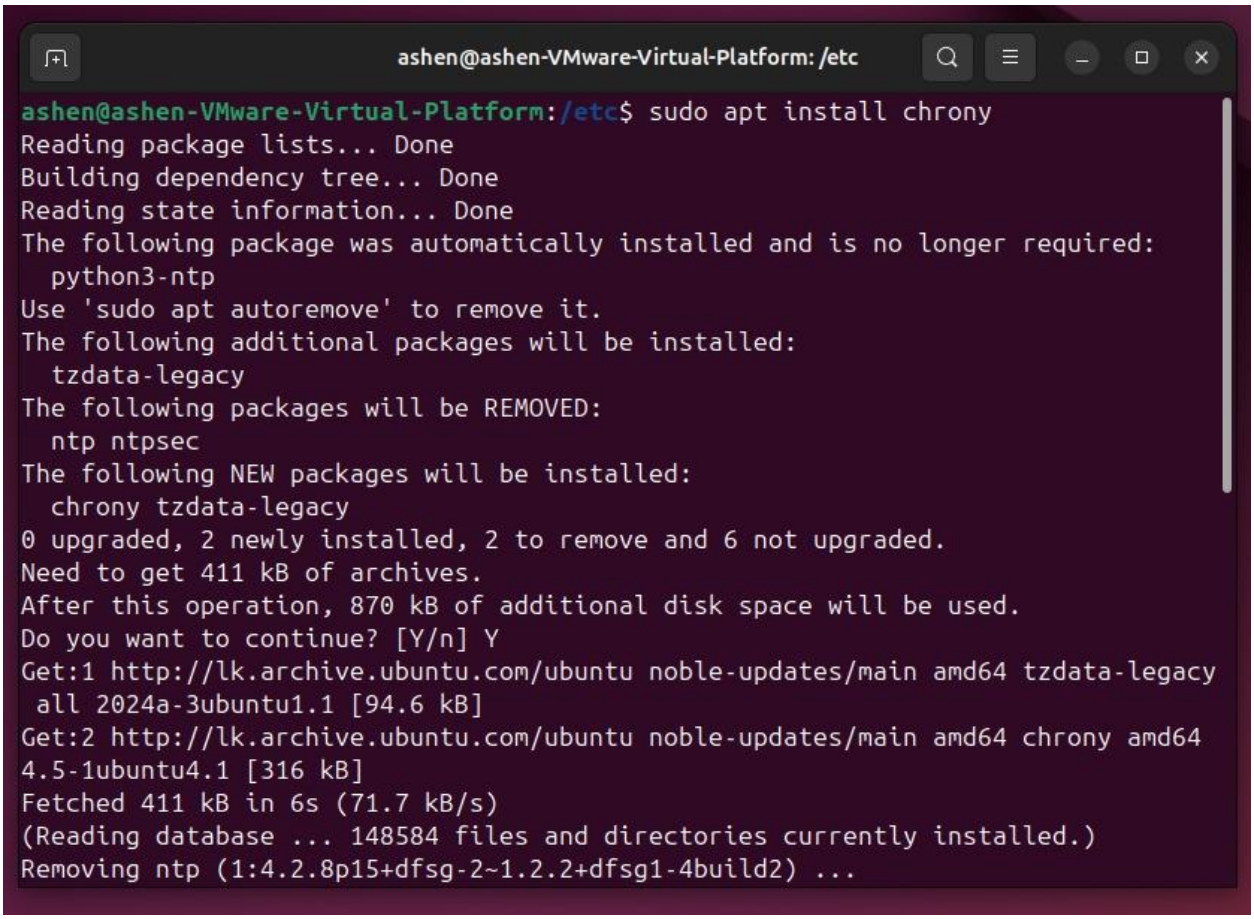
➢ NTP Server Configuration

1. update the package list

    - sudo apt update

2. install chrony in the Server machine (chrony - Network Time Protocol (NTP) used to synchronize the system clock with NTP servers.)

    - sudo apt install chrony



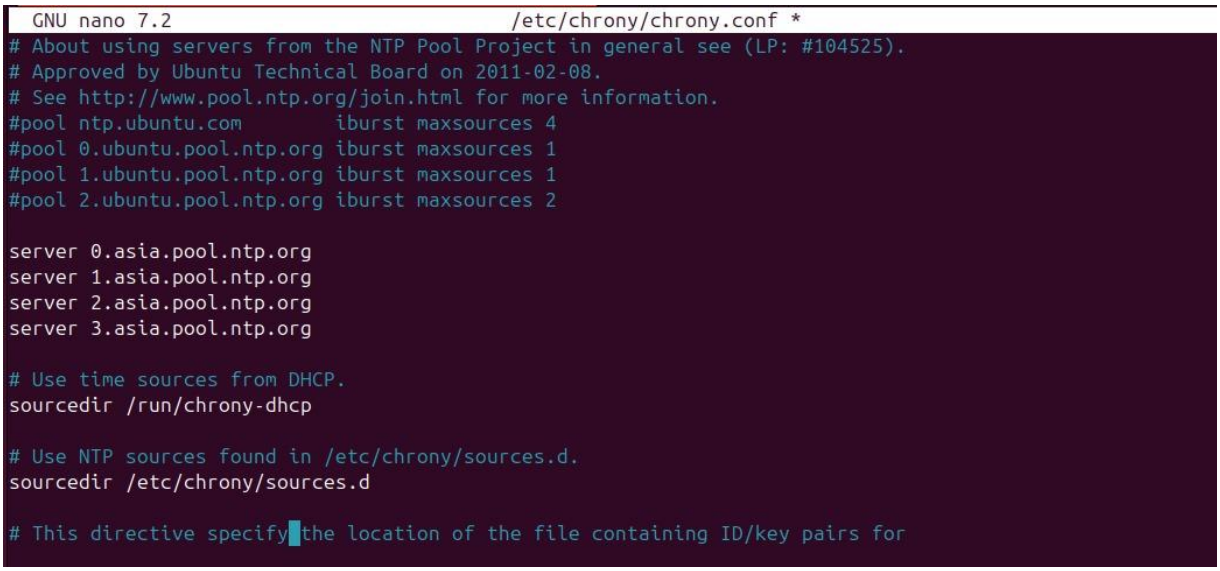
3. Open and modify the configuration file using nano/vi text editor.

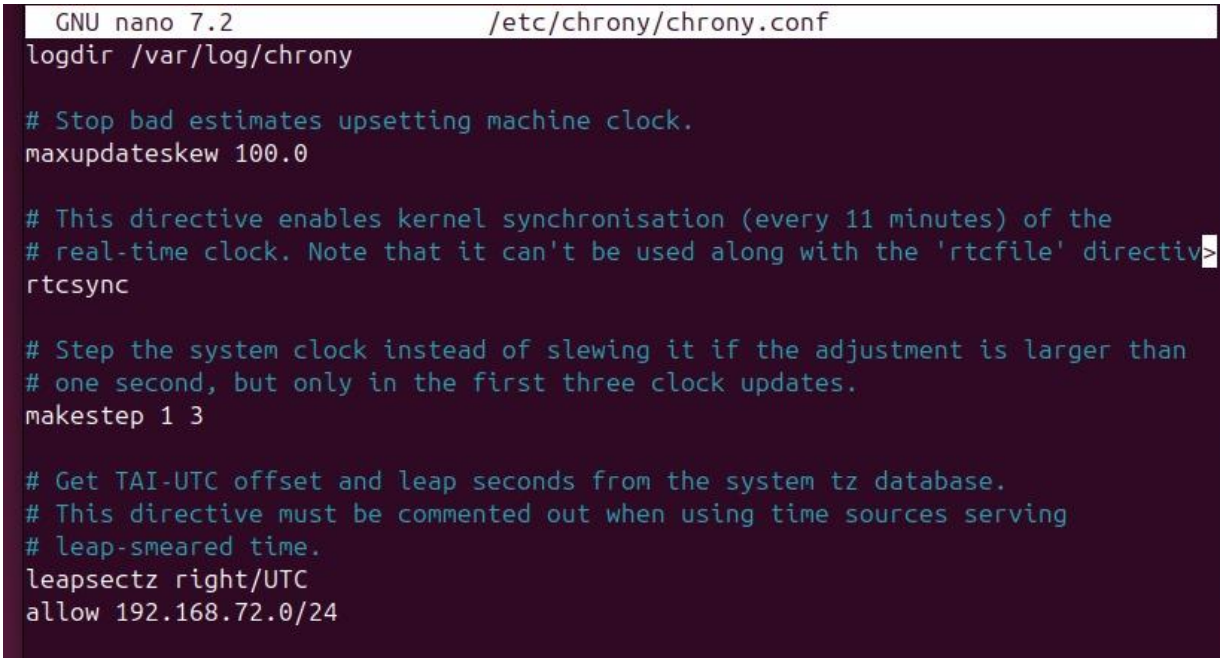
-sudo nano /etc/chrony/chrony.conf

-remove existing server entries by commenting out

-Add relevant external NTP servers

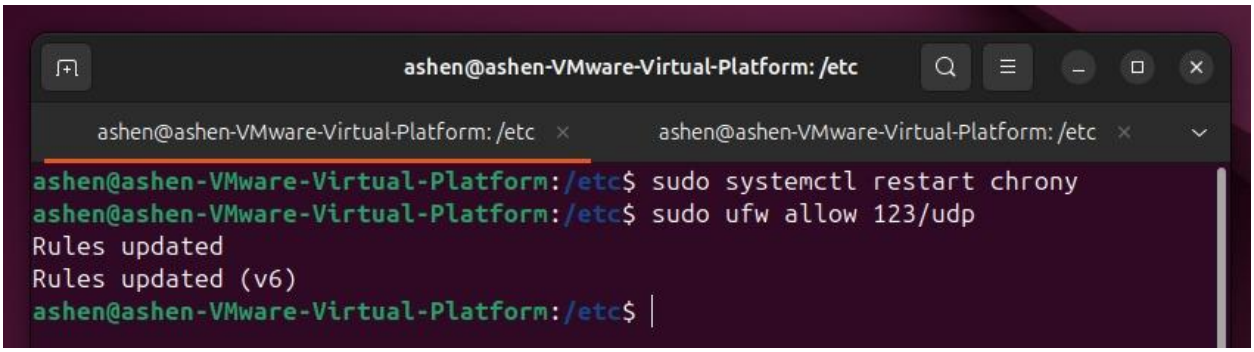
-Add the local network that is allows to use this server.(add the server ip address)





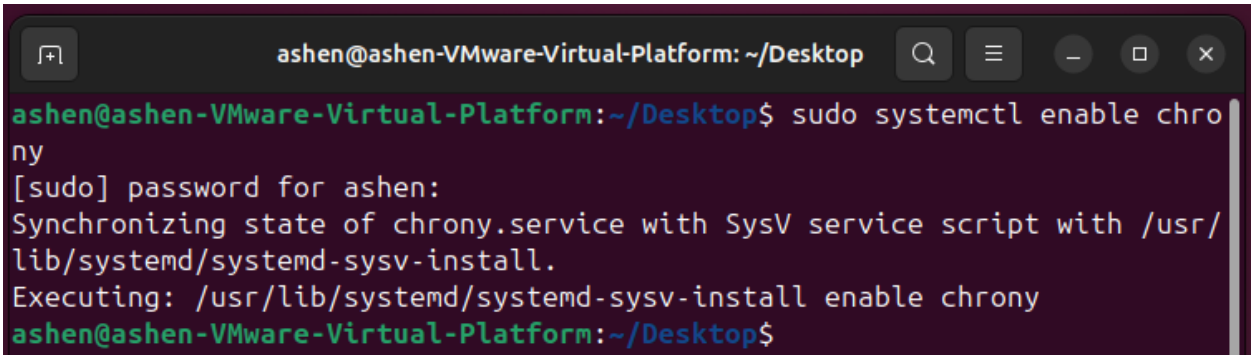
4. Restart the service to apply changes and allow NTP service through firewall (Allow incoming UDP traffic on port 123 which typically used by NTP, This is an essential step if you want your server to receive time synchronization requests from NTP clients.)

- sudo systemctl restart chrony

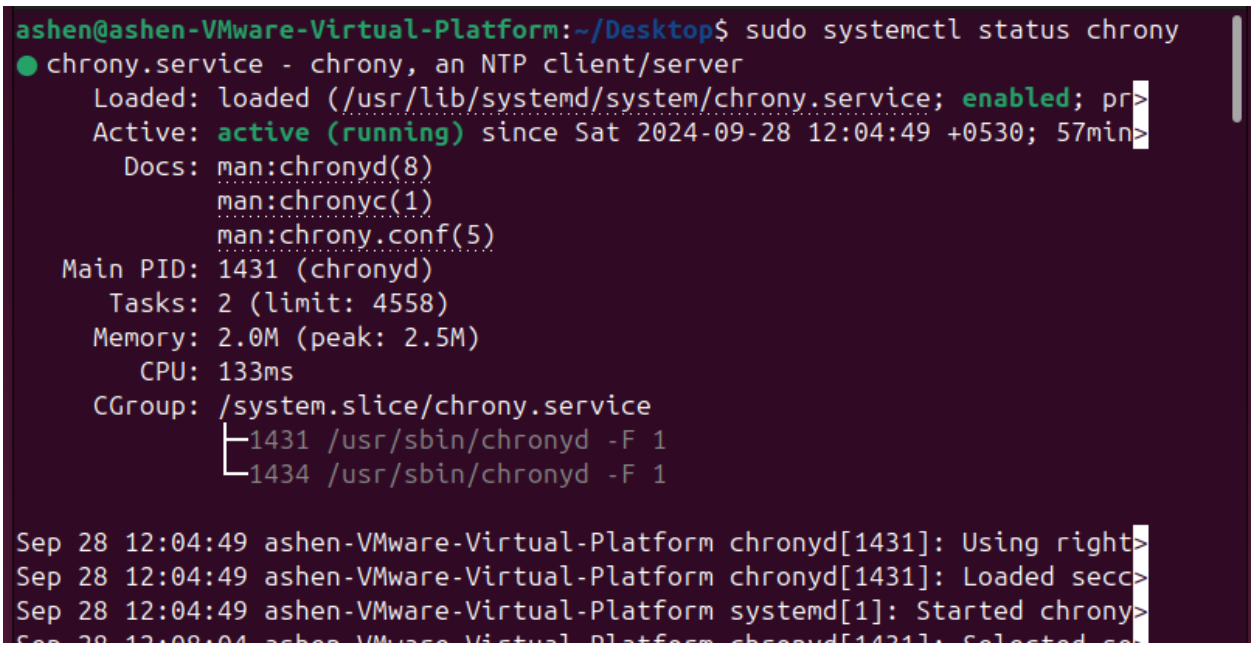


5. Enable chrony on boot time.

- sudo systemctl enable chrony



6. Check the status whether active or not



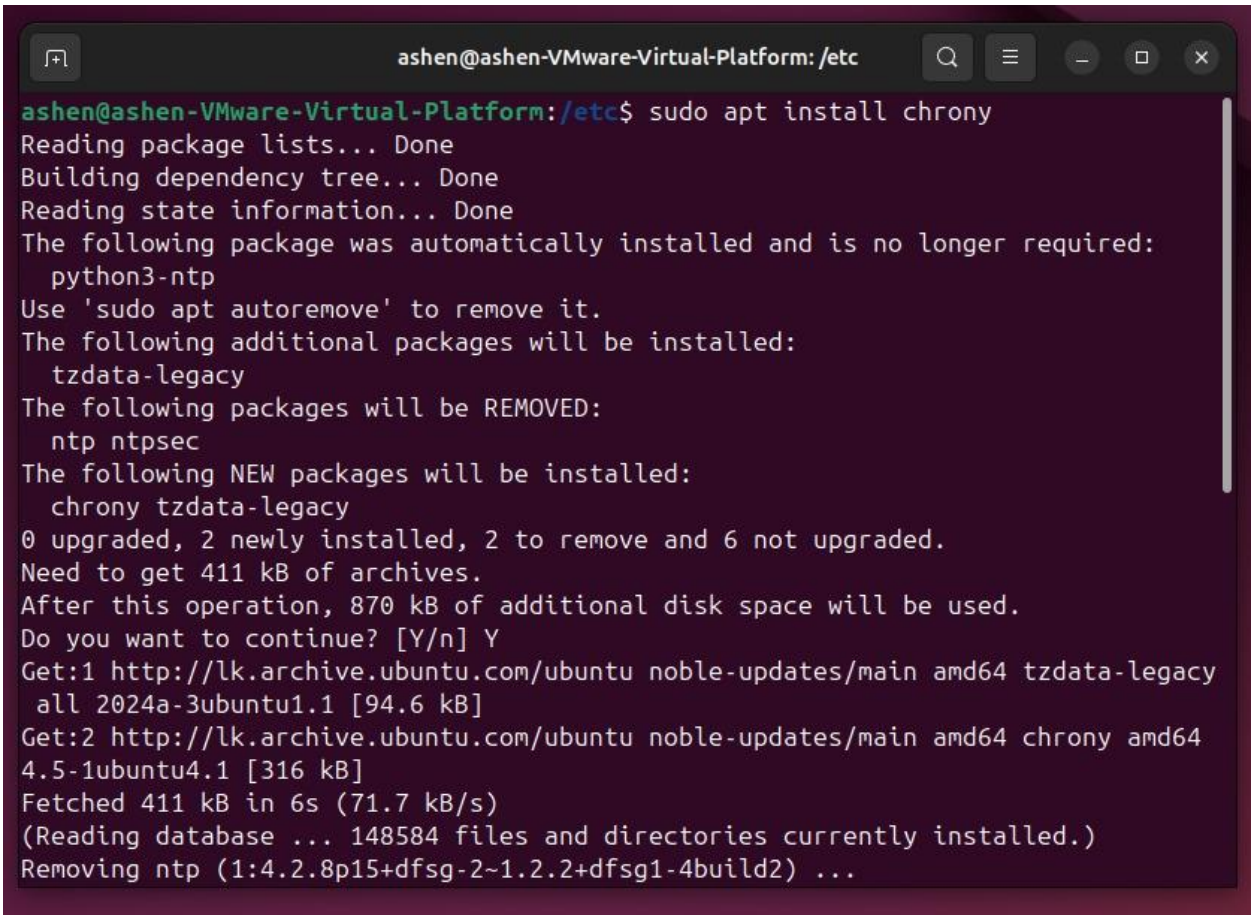
➢ NTP Client Configuration

1. update the package list

    - sudo apt update

2. Install chrony in the Client machine

    - sudo apt install chrony

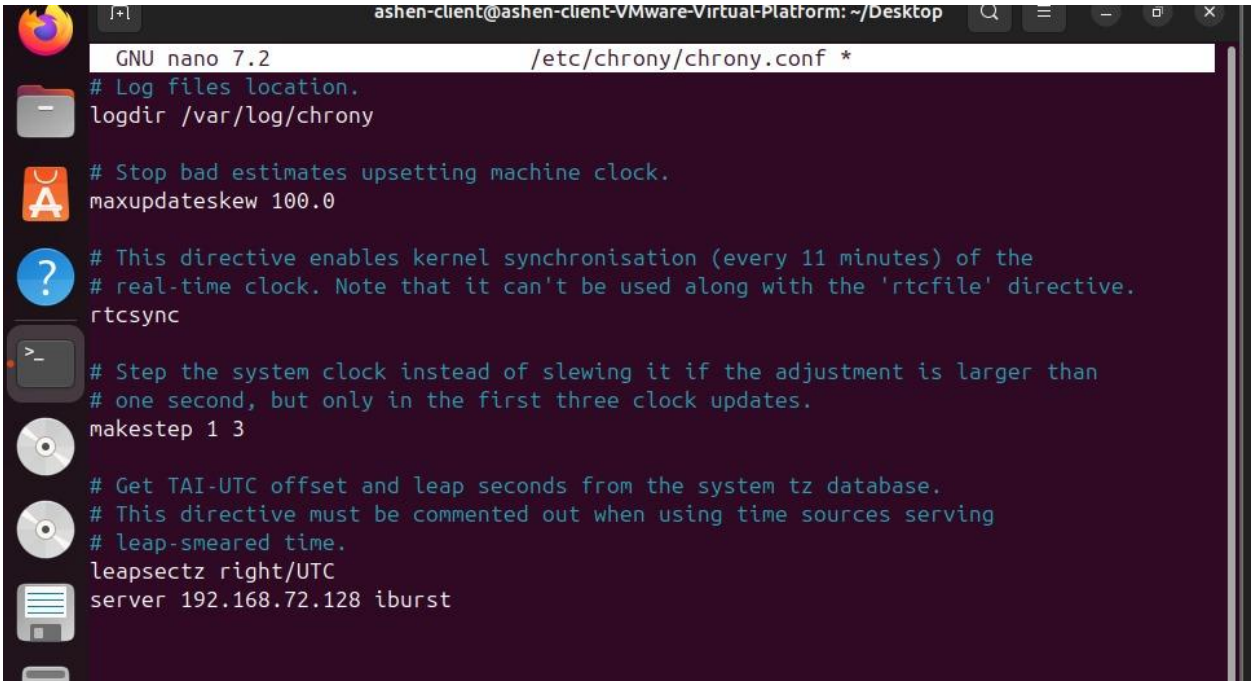


3. Open and modify the configuration file using nano/vi text editor.

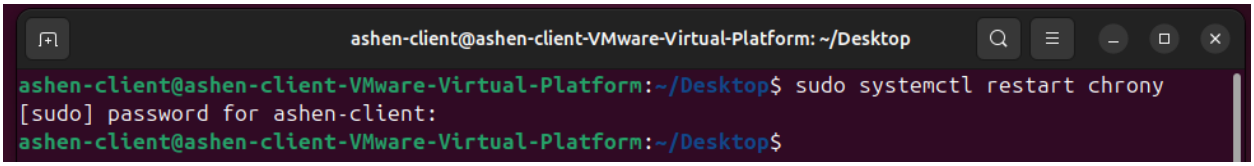
-sudo nano /etc/chrony/chrony.conf

-remove existing server entries by commenting out

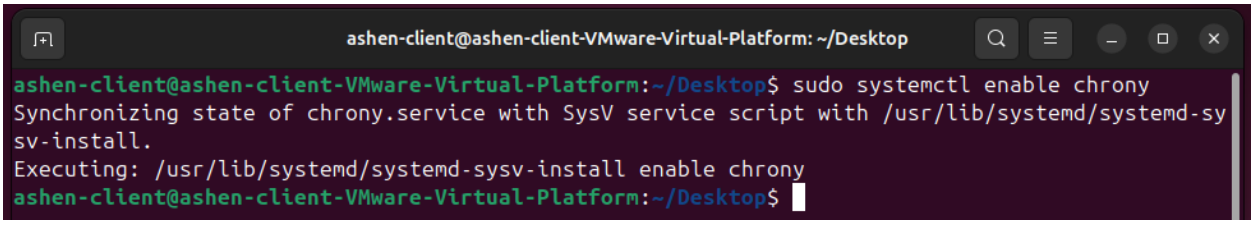
-Add the IP address of your NTP server



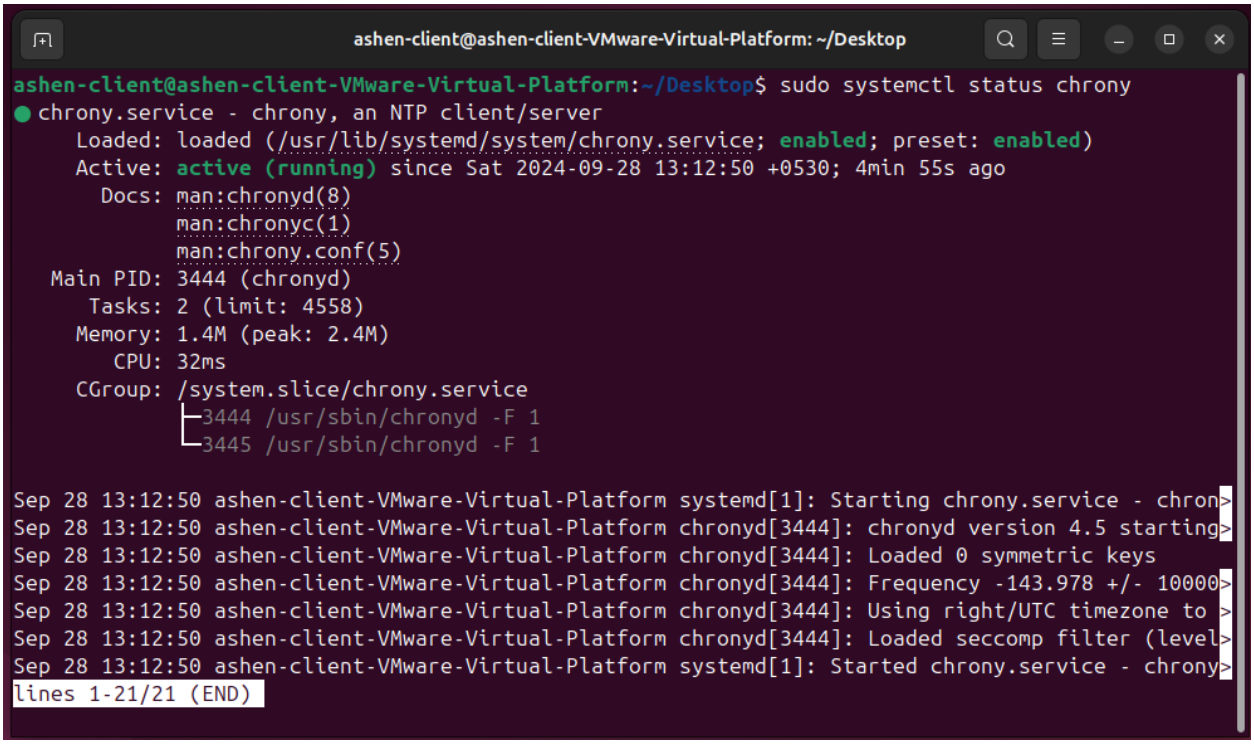
4. Restart chrony service



5. Enable chrony to start on boot



6. check the status of the chrony client.

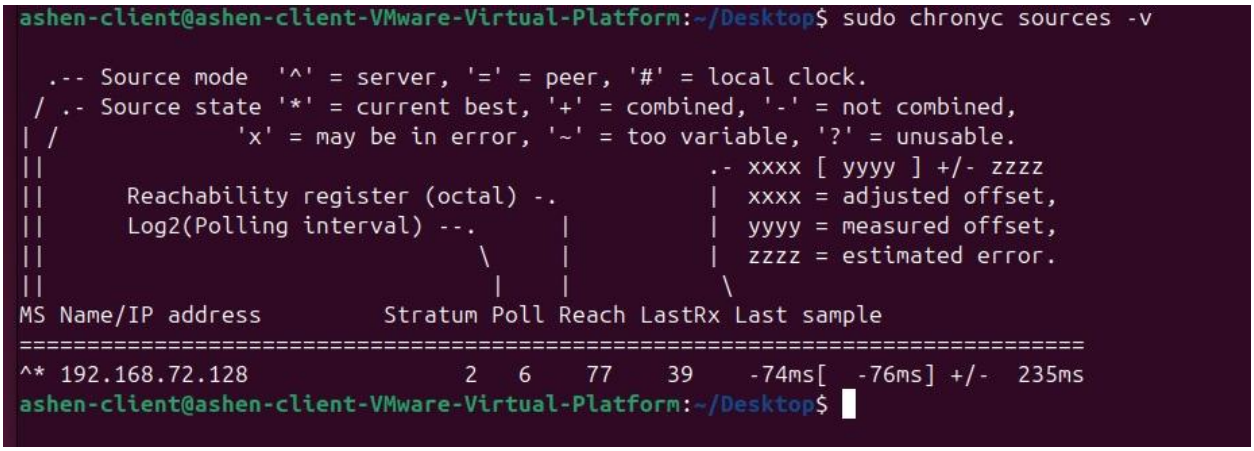


Verification

Client-Side verification

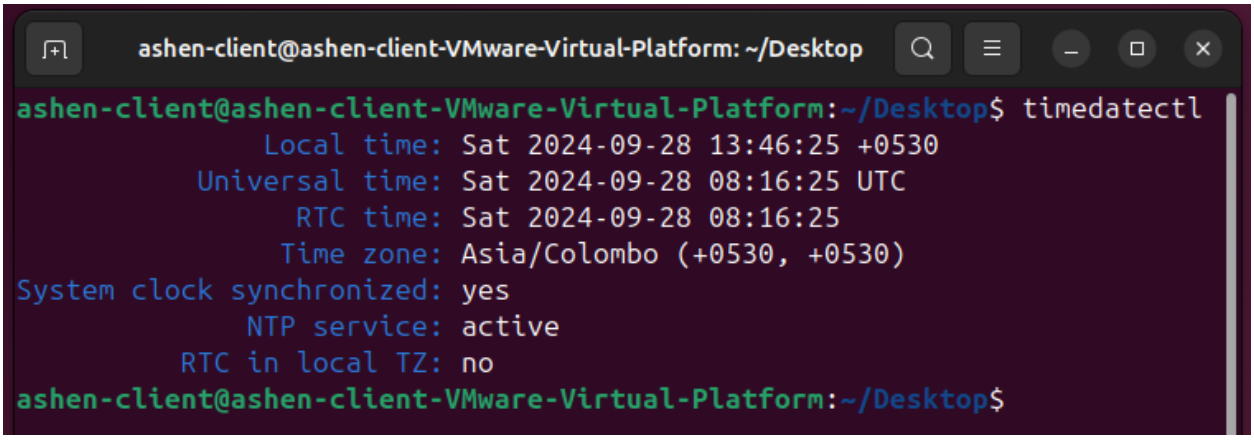
sudo chronyc sources -v

- Command is used to display detailed information about the NTP sources that the Chrony NTP client is using for synchronization.



timedatectl

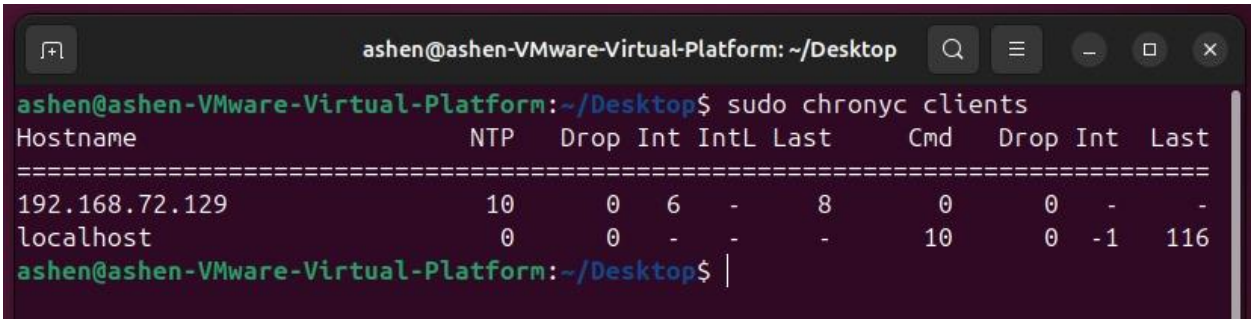
- Helps to verify the system's time is synchronized with the NTP server.



Server-Side verification

sudo chronyc clients

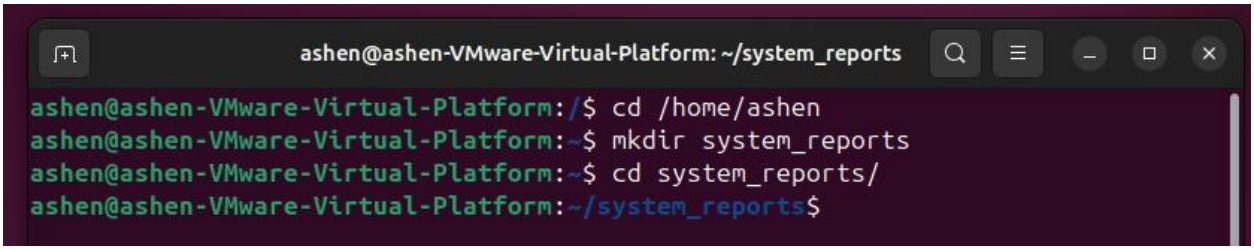
- Command is used to display information about the clients that are currently connected to the chrony NTP server.



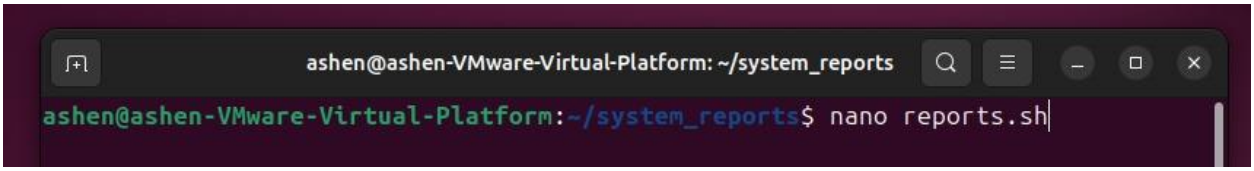
4. Shell Scripting and Security

Q1→ Write a script to automate a report that captures key system details every day.

1. Create a new directory in the home directory.



2. Create a new shell script in the home directory inside the directory created above.



3. Write the below shell script inside the report.sh file using the text editor.

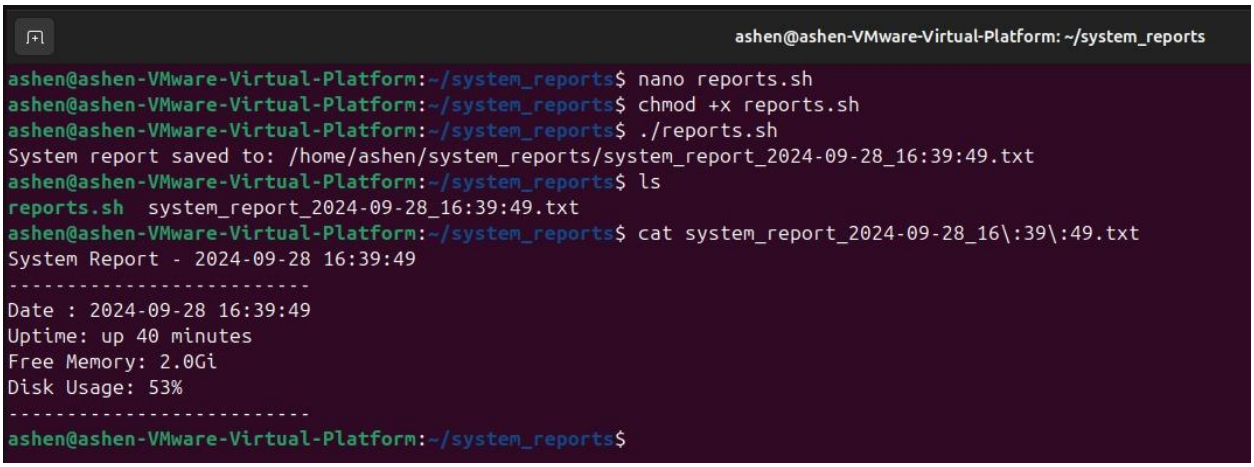
```
  GNU nano 7.2                                                  reports.sh *
#!/bin/bash

# Set the directory for storing reports
REPORT_DIR="/home/ashen/system_reports"

# Ensure the directory exists, creating it if necessary
mkdir -p "$REPORT_DIR"

# Capture the current date and time
DATE=$(date "+%Y-%m-%d %H:%M:%S")

# Gather system stats like uptime, available memory, and disk usage
UPTIME=$(uptime -p)  # Shows system running time
FREE_MEMORY=$(free -h | grep Mem | awk '{print $4}')  # Available RAM in human-readable format
DISK_USAGE=$(df -h / | grep / | awk '{print $5}')  # Root directory disk usage percentage

# Define the report file name with a timestamp
SYSTEM_REPORT="$REPORT_DIR/system_report_$(date "+%Y-%m-%d_%H:%M:%S").txt"

# Output system details to the report file
echo "System Report - $DATE" > "$SYSTEM_REPORT"
echo "-------------------------" >> "$SYSTEM_REPORT"
echo "Date : $DATE" >> "$SYSTEM_REPORT"
echo "Uptime: $UPTIME" >> "$SYSTEM_REPORT"
echo "Free Memory: $FREE_MEMORY" >> "$SYSTEM_REPORT"
echo "Disk Usage: $DISK_USAGE" >> "$SYSTEM_REPORT"
echo "-------------------------" >> "$SYSTEM_REPORT"

# Notify the user where the report was saved
echo "System report saved to: $SYSTEM_REPORT"
```

Script

```
#!/bin/bash

# Set the directory for storing reports

REPORT_DIR="/home/ashen/system_reports"

# Ensure the directory exists, creating it if necessary

mkdir -p "$REPORT_DIR"


# Capture the current date and time

DATE=$(date "+%Y-%m-%d %H:%M:%S")

# Gather system stats like uptime, available memory, and disk usage

UPTIME=$(uptime -p)  # Shows system running time

FREE_MEMORY=$(free -h | grep Mem | awk '{print $4}')  # Available RAM in human-readable format

DISK_USAGE=$(df -h / | grep / | awk '{print $5}')  # Root directory disk usage percentage

# Define the report file name with a timestamp

SYSTEM_REPORT="$REPORT_DIR/system_report_$(date "+%Y-%m-%d_%H:%M:%S").txt"


# Output system details to the report file

echo "System Report - $DATE" > "$SYSTEM_REPORT"

echo "--------------------------" >> "$SYSTEM_REPORT"

echo "Date : $DATE" >> "$SYSTEM_REPORT"

echo "Uptime: $UPTIME" >> "$SYSTEM_REPORT"

echo "Free Memory: $FREE_MEMORY" >> "$SYSTEM_REPORT"

echo "Disk Usage: $DISK_USAGE" >> "$SYSTEM_REPORT"

echo "--------------------------" >> "$SYSTEM_REPORT"

# Notify the user where the report was saved

echo "System report saved to: $SYSTEM_REPORT"
```

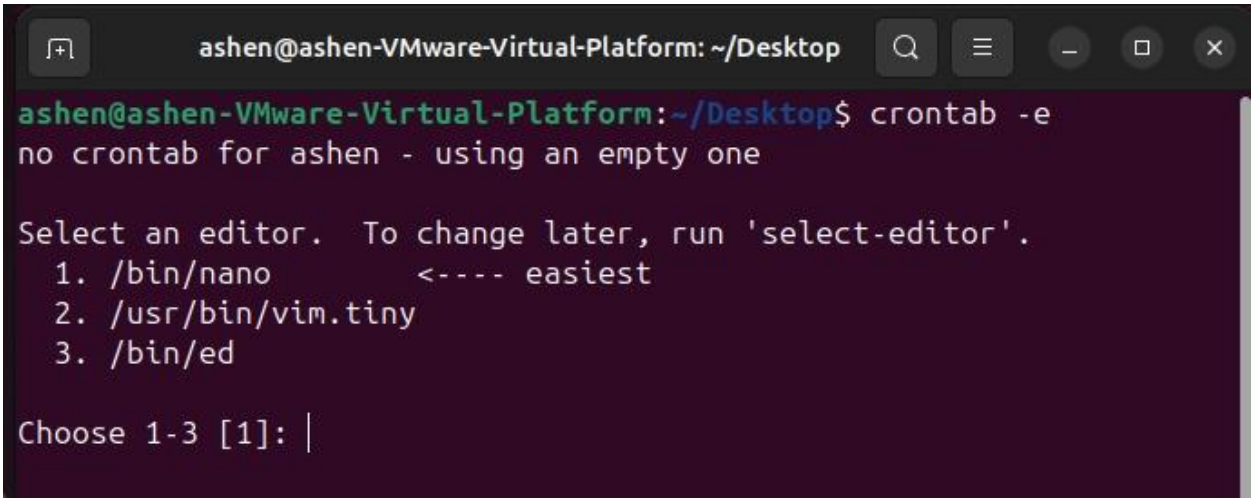
4. Change the permissions of the script file to make it executable.
    - chmod +x report.sh

5. Execute the script.
    - ./report.sh

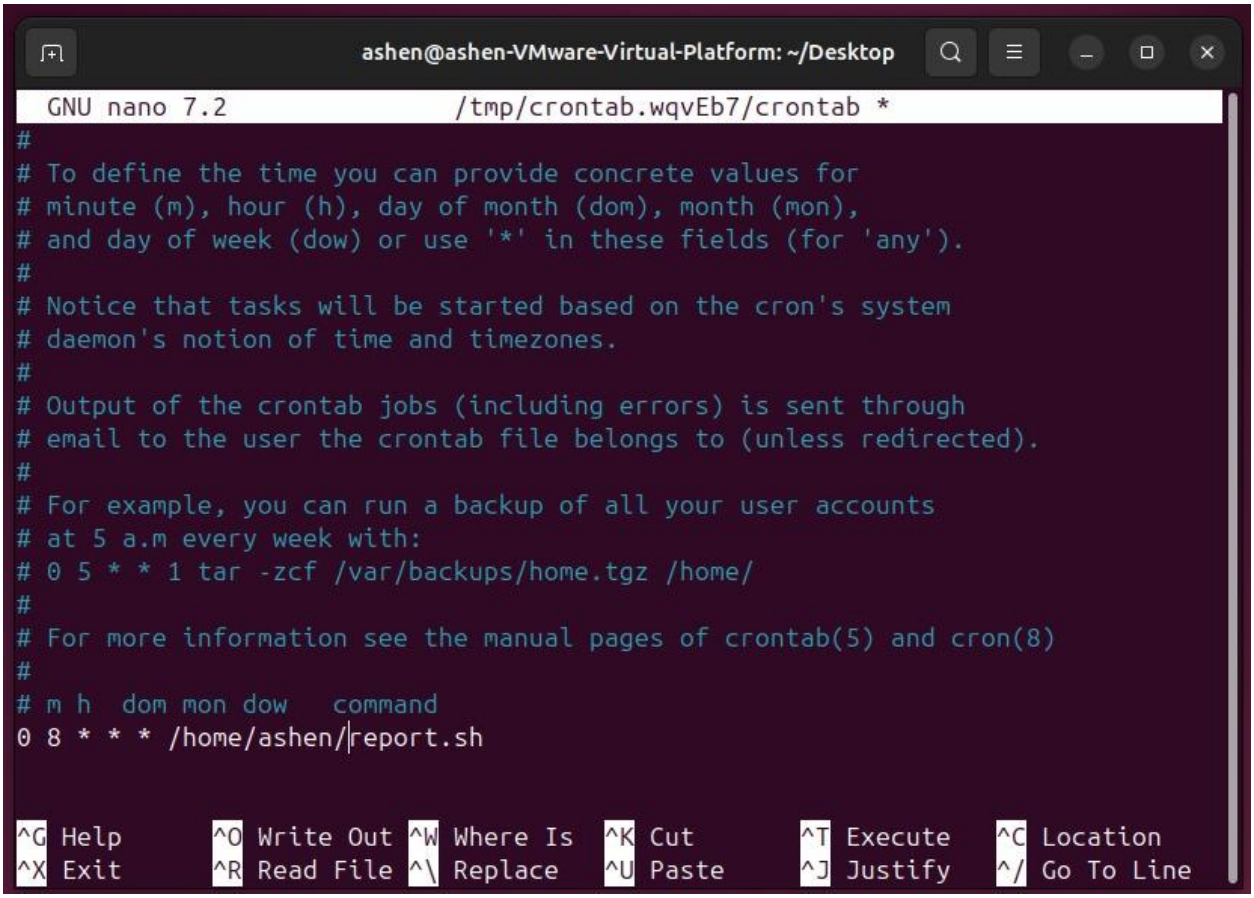


If a system report is created, it means that the script works.

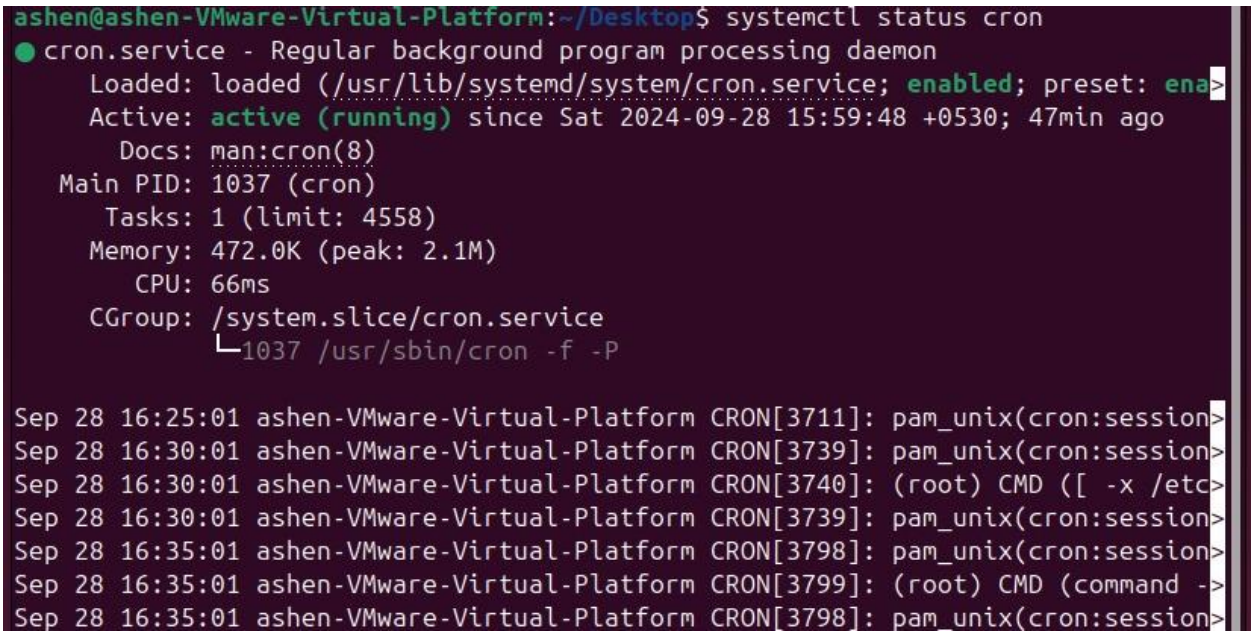
6. Schedule the script to run automatically using cron.
    - crontab -e
    - select the preferred text editor (select 1)



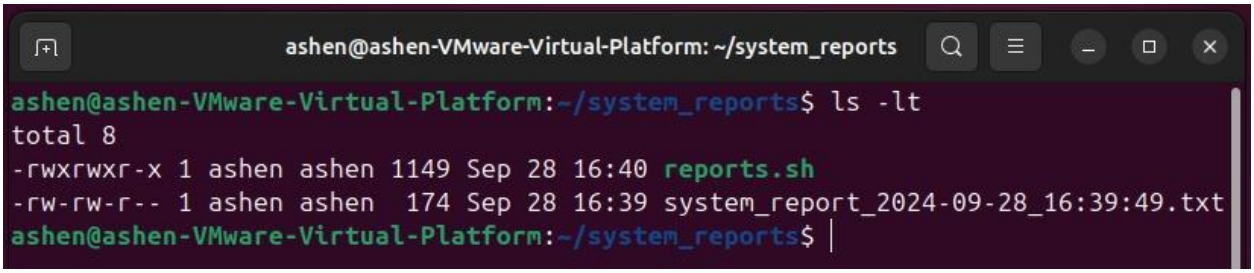
7. Add the below command to run the script everyday at 8 AM
    - 0 8 * * * /home/ashen/report.sh



8. Save and Exit .
9. Check the status of the cron service.

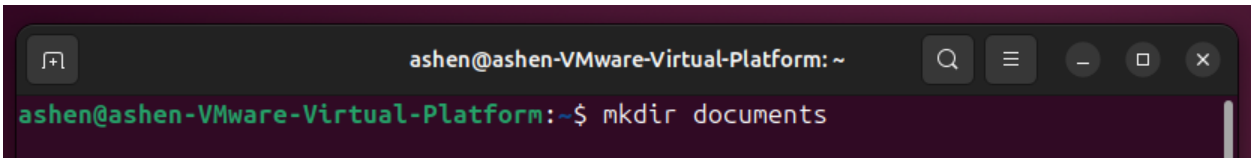


10. Verify the Generated Reports Files.
11. If our script creates report files in /home/ashen/system_reports, can check the directory to see if new files are being generated every day at 8:00 AM.



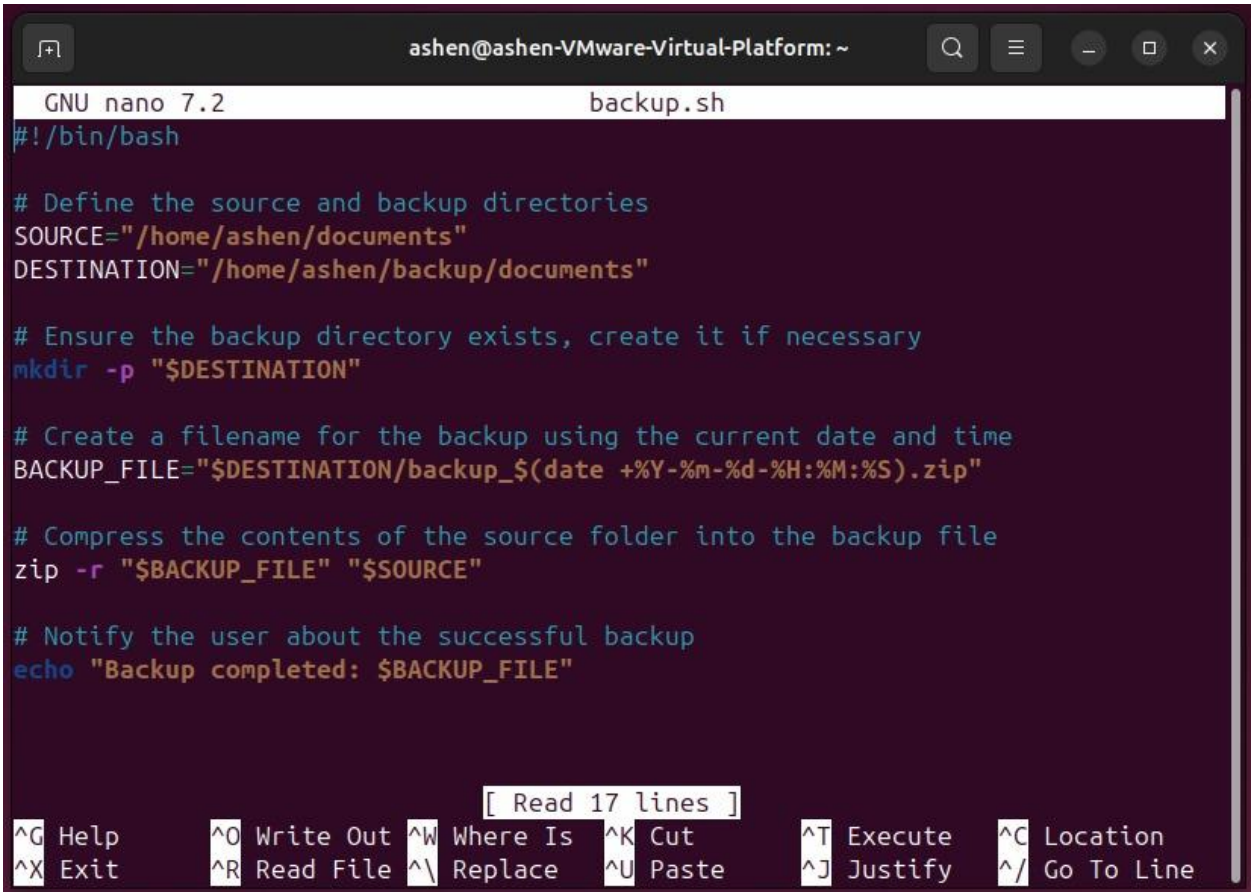
Q2→ Write a script to automate the backup of a critical directory containing important files.

1. Create a new Directory which needs to be backup.



2. Create a new script file .

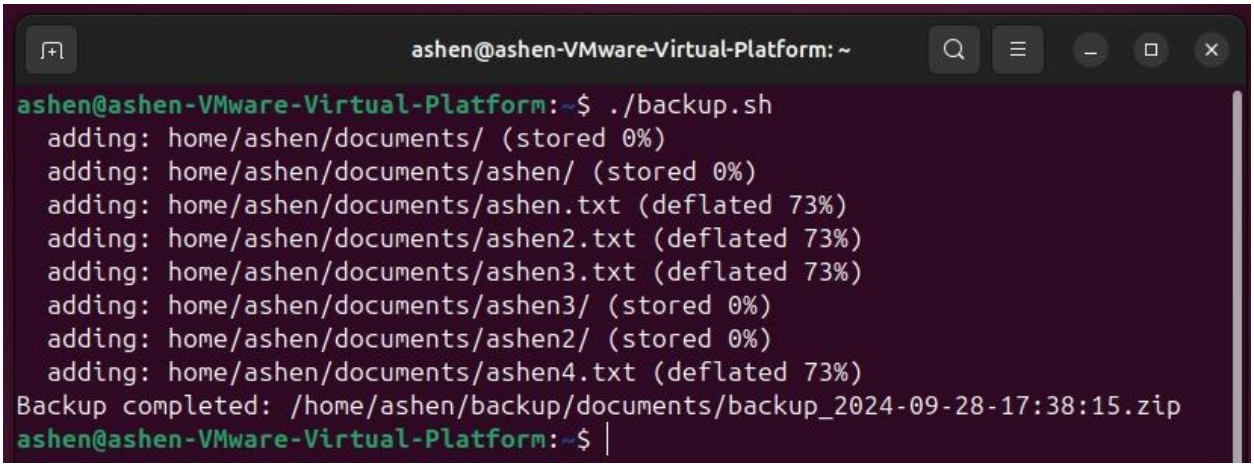
3.Write the below shell script inside the opened script file.



Script

```
#!/bin/bash

# Define the source and backup directories
SOURCE="/home/ashen/documents"
DESTINATION="/home/ashen/backup/documents"

# Ensure the backup directory exists, create it if necessary
mkdir -p "$DESTINATION"

# Create a filename for the backup using the current date and time
BACKUP_FILE="$DESTINATION/backup_$(date +%Y-%m-%d-%H:%M:%S).zip"

# Compress the contents of the source folder into the backup file
zip -r "$BACKUP_FILE" "$SOURCE"

# Notify the user about the successful backup
echo "Backup completed: $BACKUP_FILE"
```

4. Execute the script to check that it works.

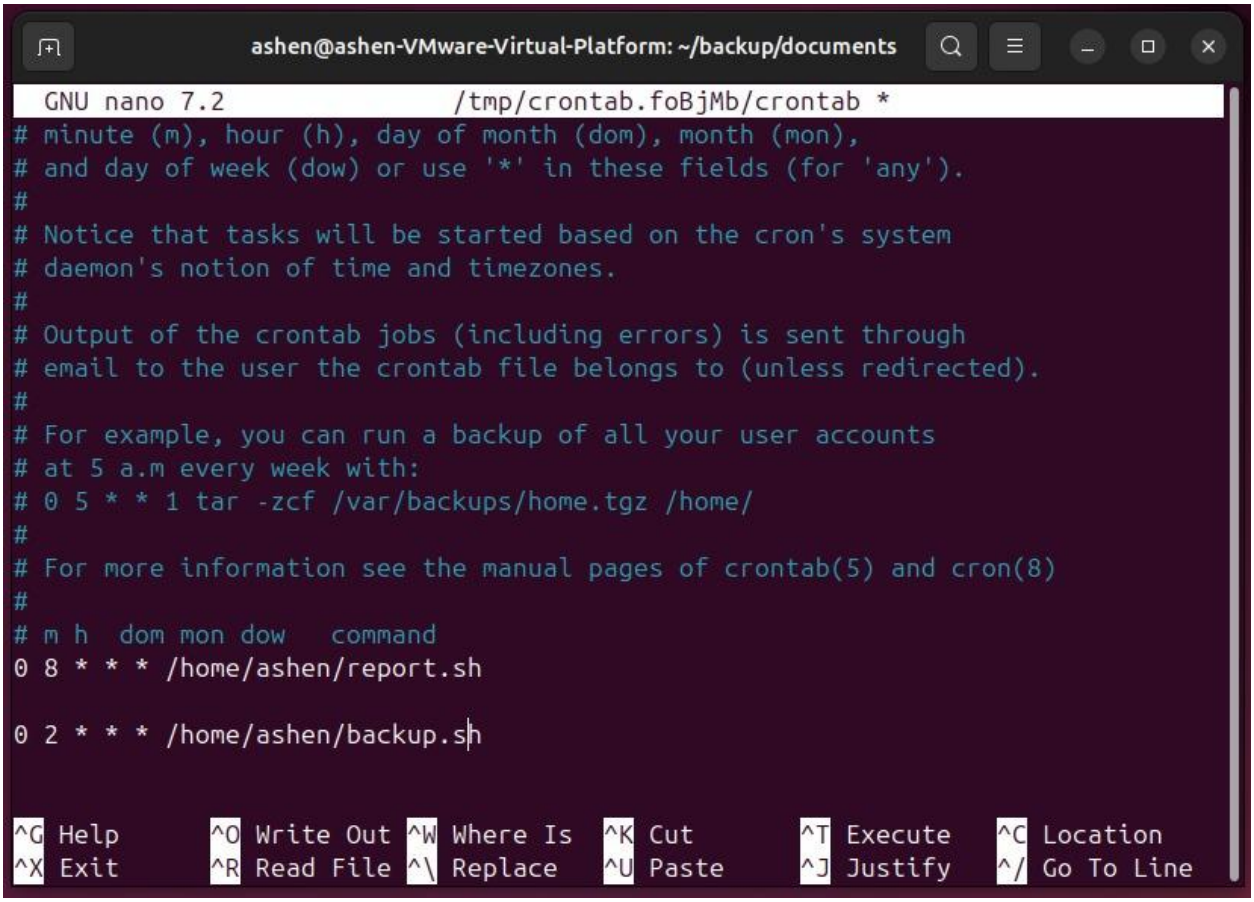


5. Verify that a new backup file with current date and time had been created in the given directory.

```
ashen@ashen-VMware-Virtual-Platform:~$ cd backup/documents/
ashen@ashen-VMware-Virtual-Platform:~/backup/documents$ ls
backup_2024-09-28-17:34:56.zip  backup_2024-09-28-17:38:15.zip
ashen@ashen-VMware-Virtual-Platform:~/backup/documents$
```

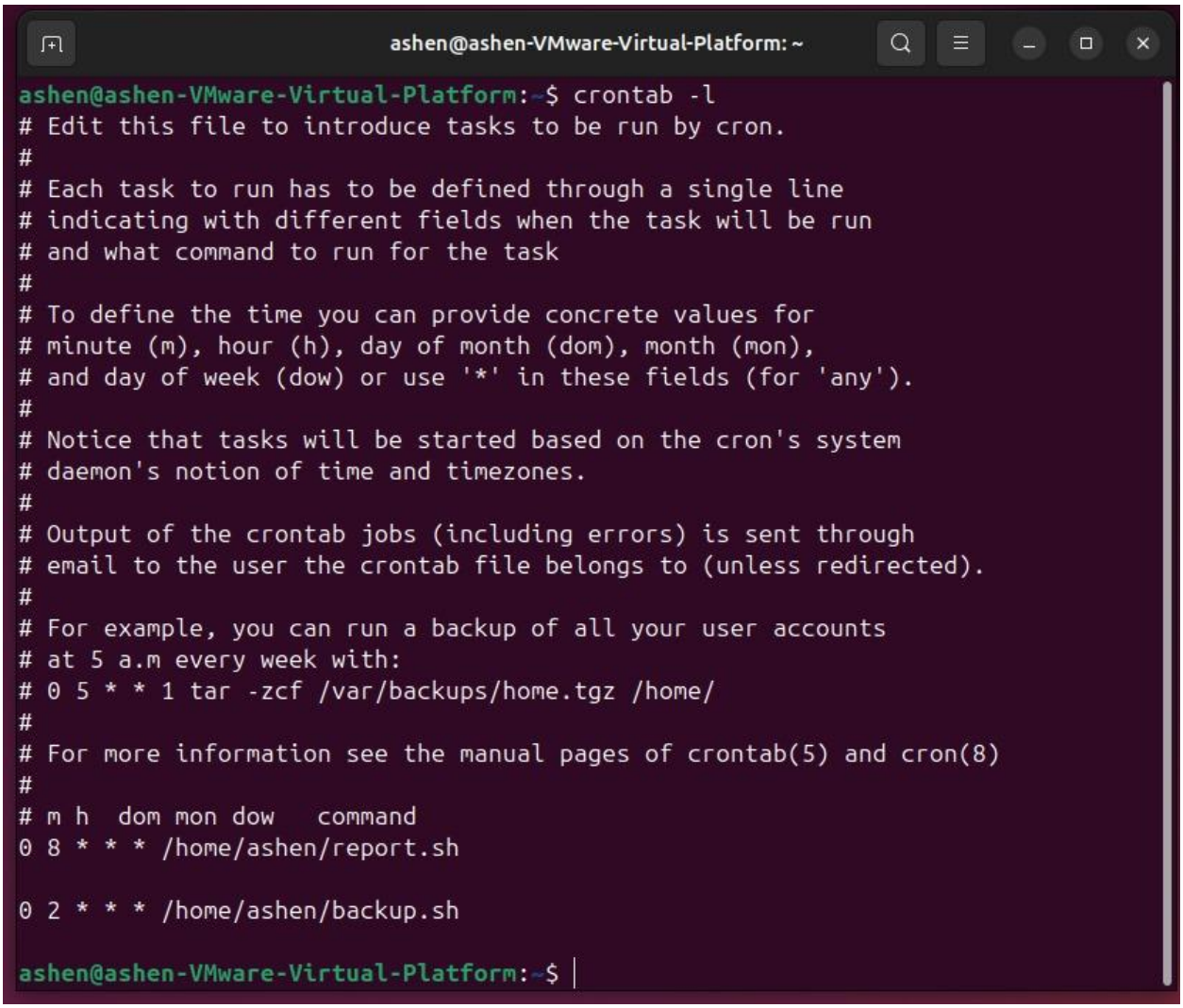
6. Schedule the script to run automatically using cron.

- Open the configuration file
  - crontab -e
  - add the below line to schedule the backup script to run daily at 2.00 am.
    - 0 2 * * * /home/ashen/backup.sh



Save and exit.

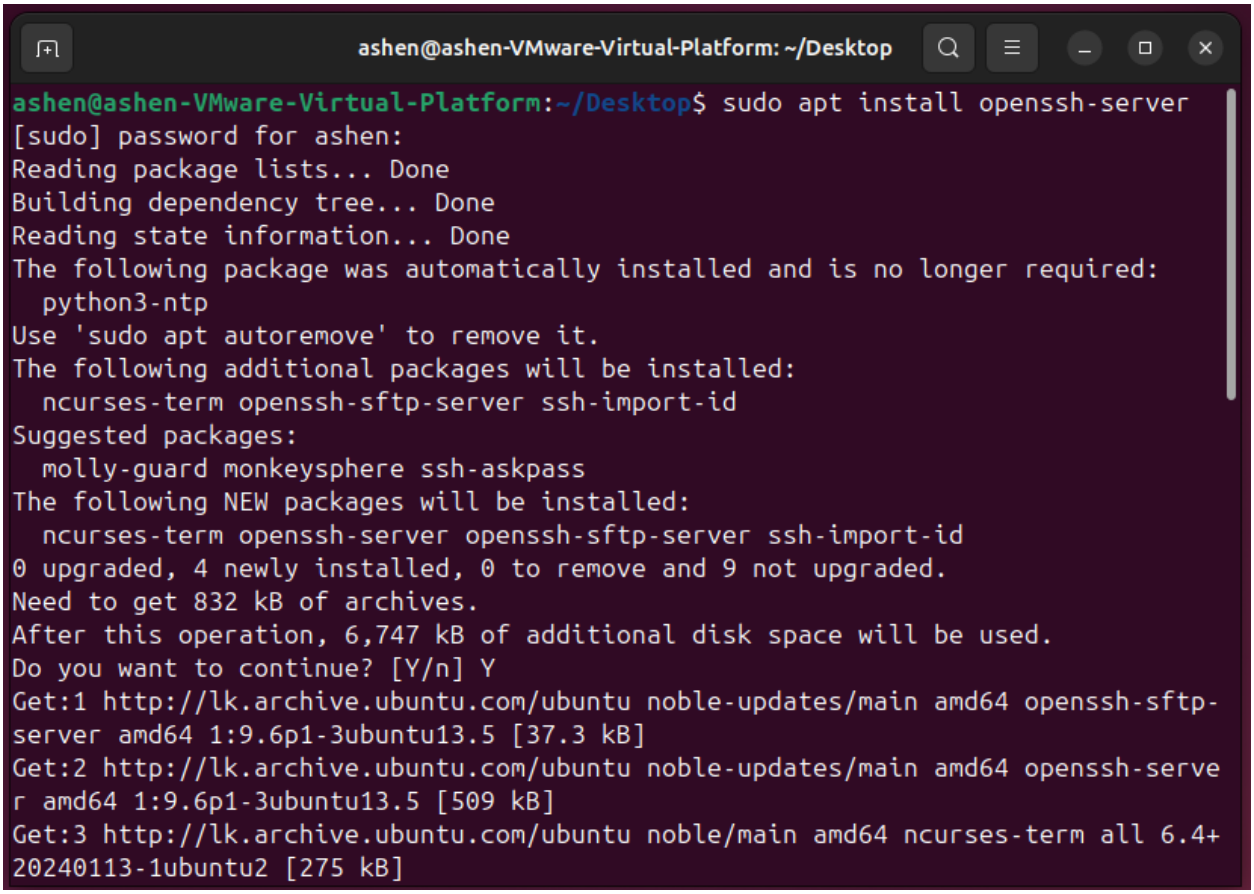
7. To ensure the cron job is set up correctly, list the cron jobs for the current user. The output should display the scheduled command.



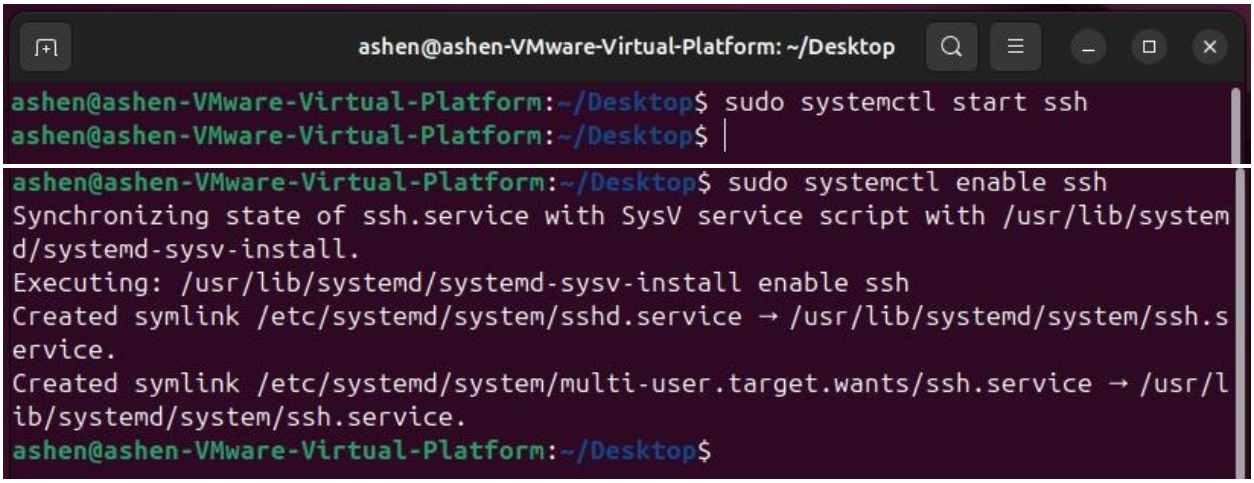
After all, above configurations the script will run automatically everyday 2.00am and generate a backup zip file in /home/user/backup/documents.

SSH (Secure Shell)

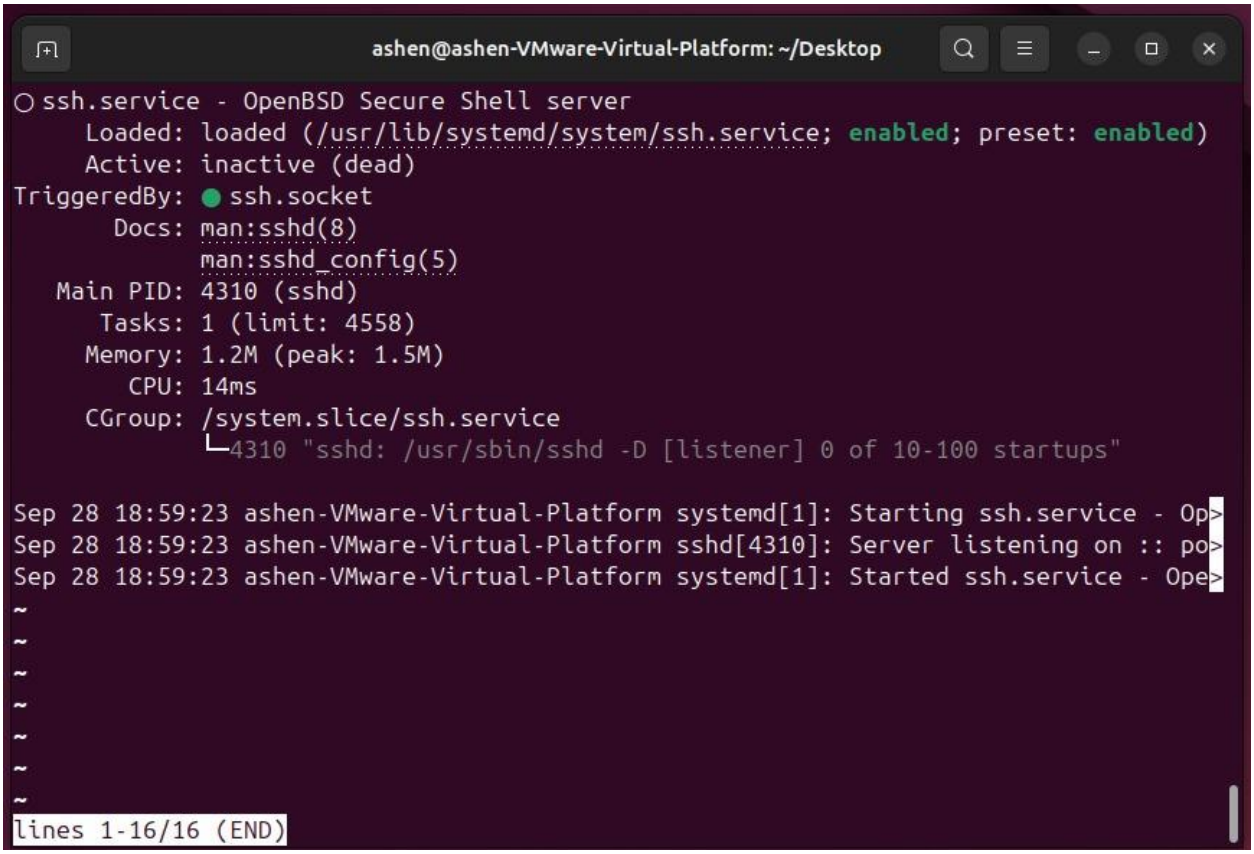
1. Install the OpenSSH Server.



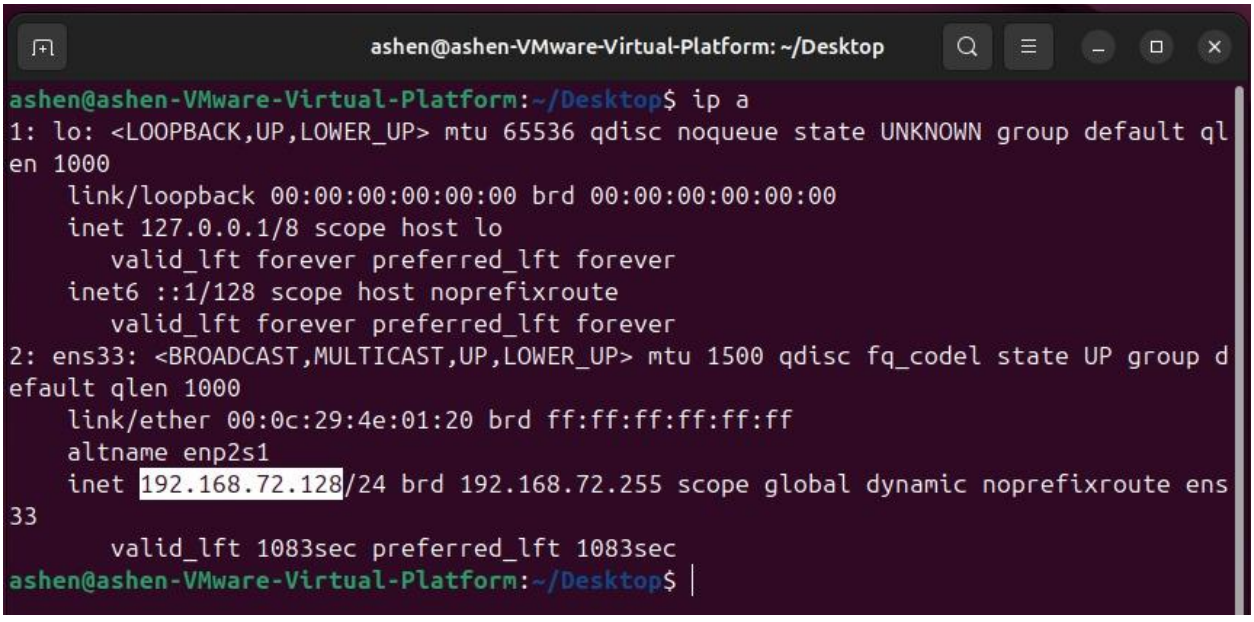
2. Start and enable the ssh service.



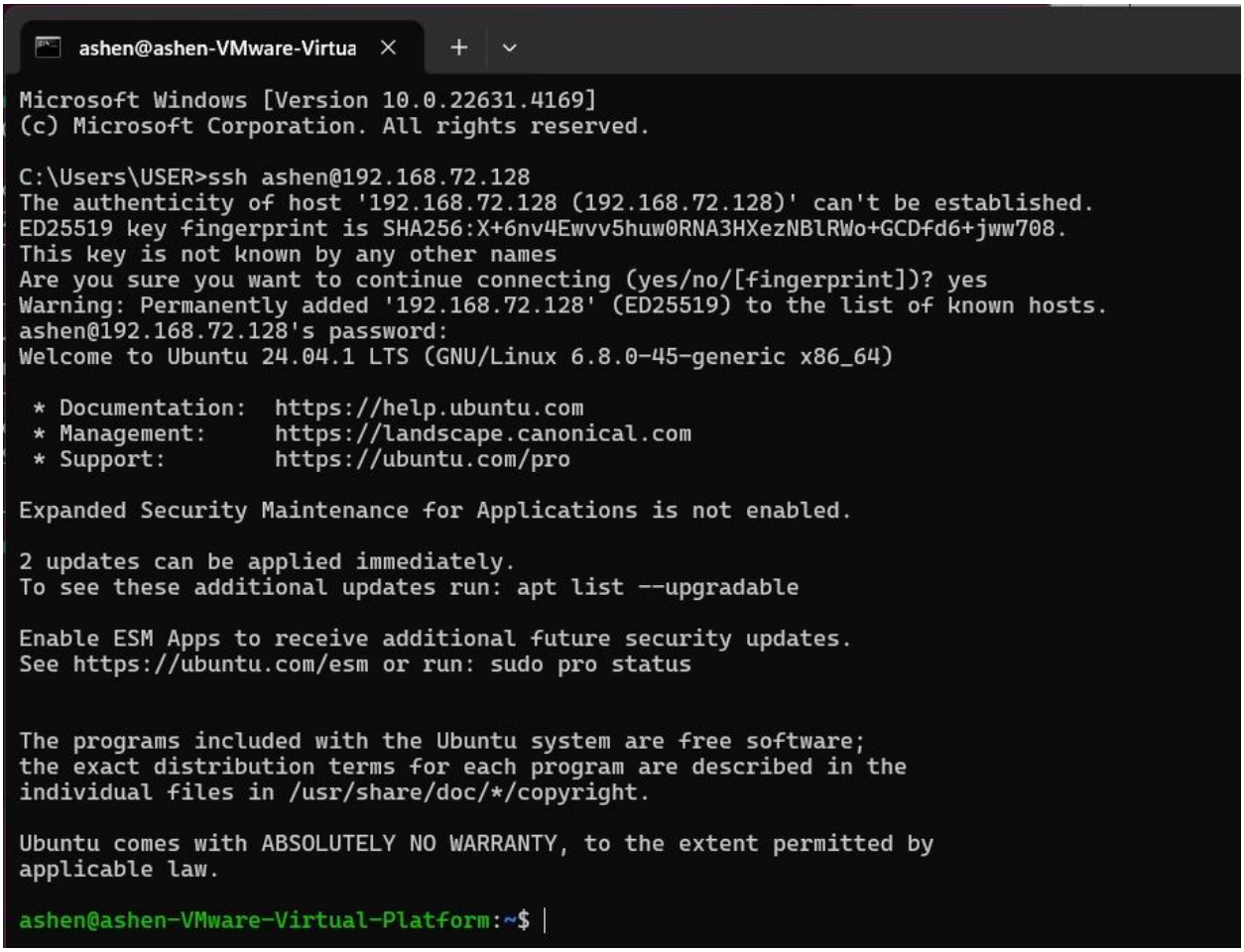
3. Verify the SSH Server is Running.



4. Check and note down the IP address of the ubuntu server.



5. Establish an SSH Connection.
   - To connect to the remote server, use the ssh command. Be sure to substitute username with the actual user account name on the server and server-ip with the server's IP address.
   - **ssh username@server-ip**
   - **Confirm the Connection**: If you're connecting for the first time, you'll see a prompt asking you to verify the server's fingerprint. Type yes to proceed.
   - **Input Credentials**: After confirming, you'll be asked to provide the password for the specified user account on the server. Enter the password to successfully log in.



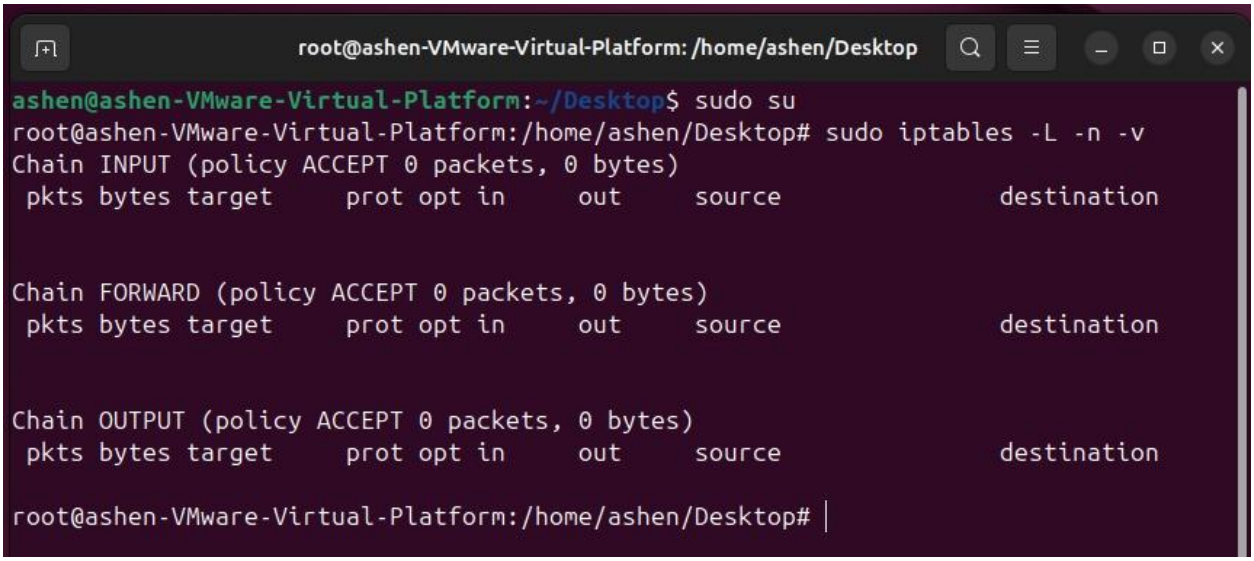
4. iptables and ACLs

i. Web Server Security: Allow incoming traffic only on port 80 (HTTP) and port 443 (HTTPS) for your web server. Block all other incoming traffic by default.

Step 01

Check any existing iptables rules.

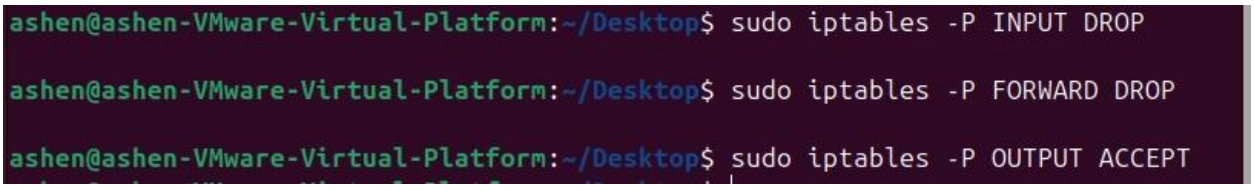
- sudo iptables -L -n -v
- (Optional) Flush existing iptables rules
    - sudo iptables -F



Step 02

Block all incoming traffic by default and allow all outgoing traffic.

- sudo iptables -P INPUT DROP
- sudo iptables -P FORWARD DROP
- sudo iptables -P OUTPUT ACCEPT



Step 03

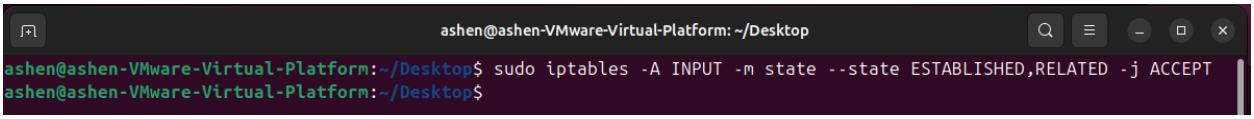
Allow incoming traffic only on ports 80 (HTTP) and 443 (HTTPS).

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

Step 04

Allow established and related Connections.

- sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT



Step 05

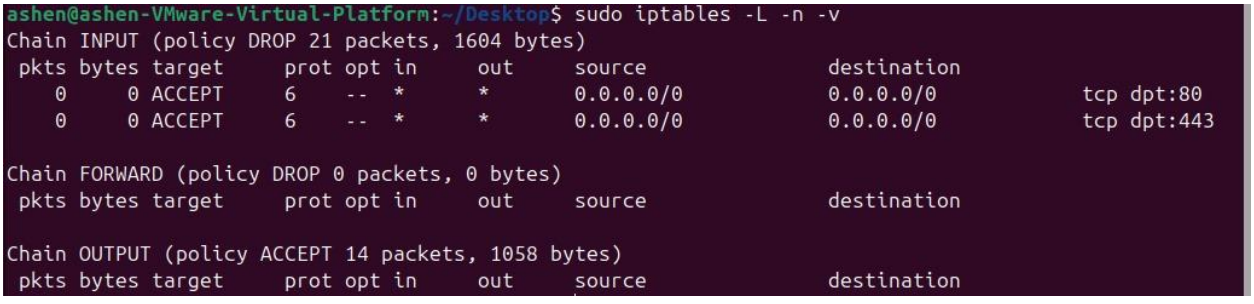
Save the rules.

- To save the rules first you need super user privileges.
  - sudo su
- Save the rules.
  - sudo iptables-save > /etc/iptables/rules.v4

Step 06

Verify the iptables configurations

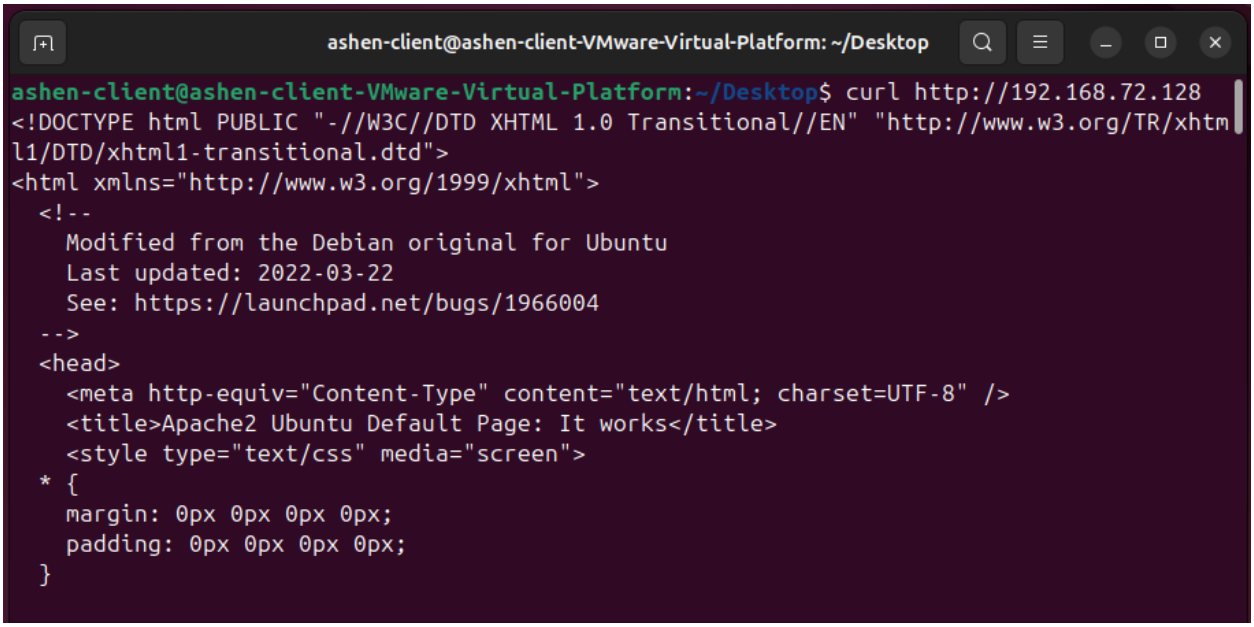
- sudo iptables -L -n -v



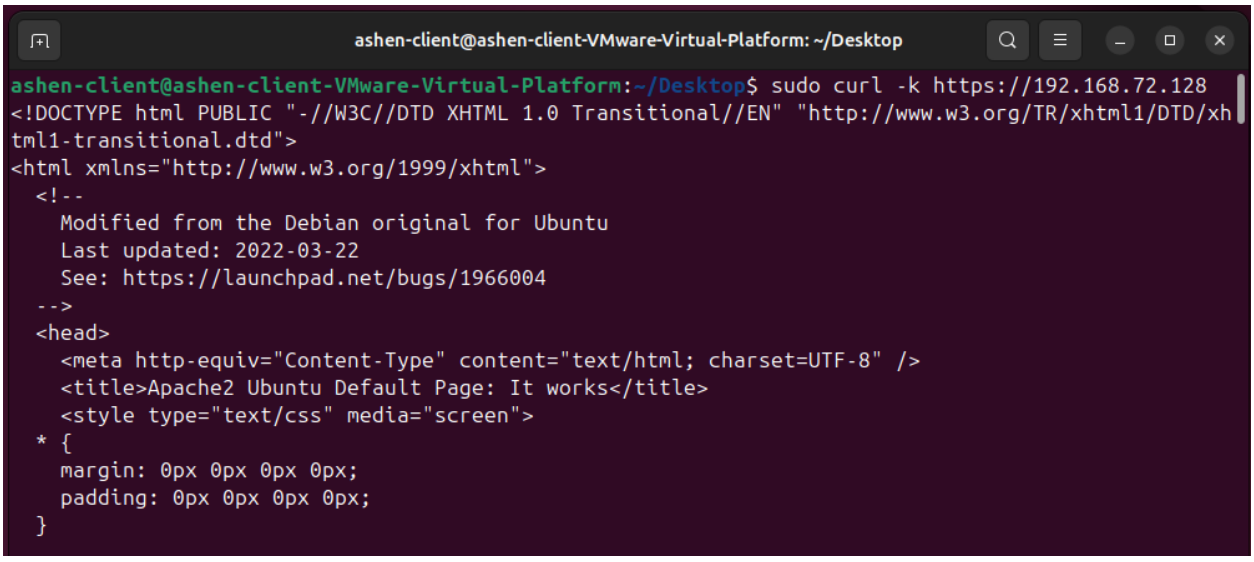
Step 07

Test the configurations (allowed ports).

- Test HTTP .
    - curl http://192.168.72.128



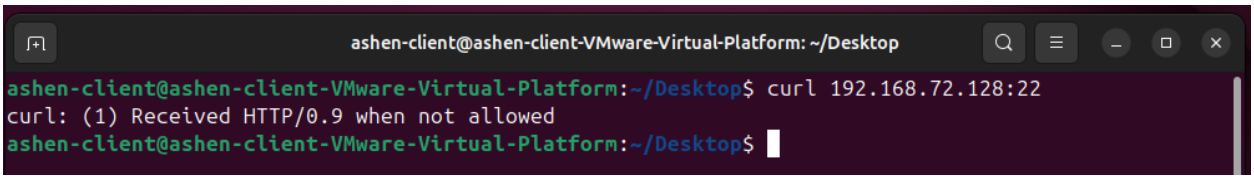
- Test HTTPS
    - sudo curl -k https://192.168.72.128
    - curl -k :- ignores SSL certificate validation



As above we could get the html content of the server if the configuration is success.

Test the configurations (blocked ports).

- Test port 22
    - curl 192.168.72.128:22



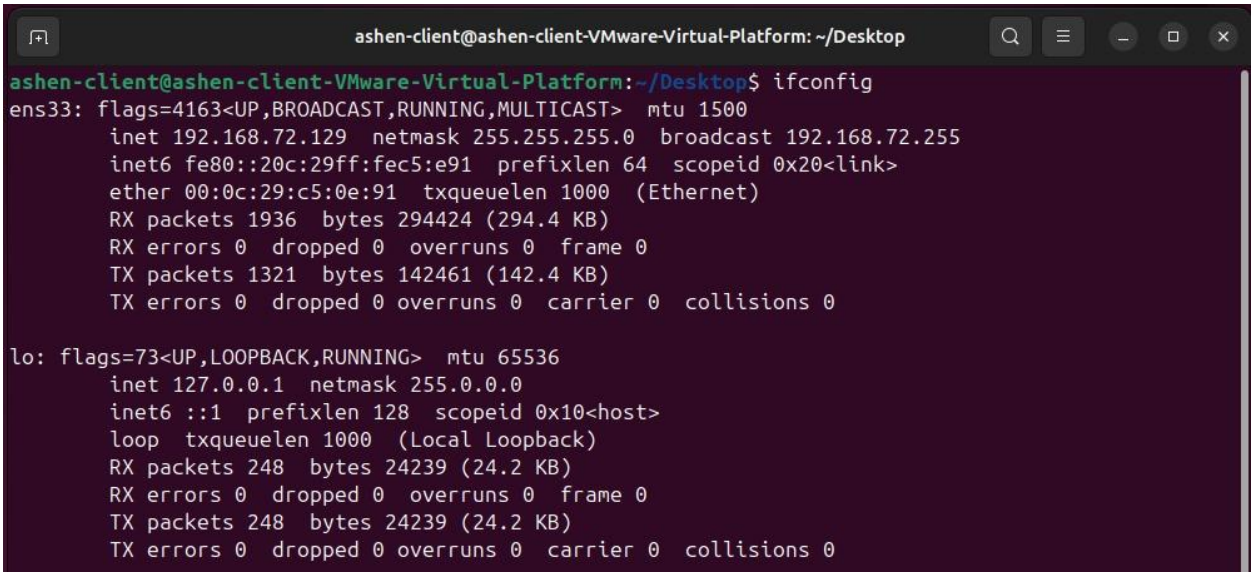
You should receive an error message as above if the ports are blocked.

ii. Remote Administration Access: Allow SSH access (port 22) only from specific IP addresses of your trusted machines used for administration. This restricts remote access attempts to authorized source.

Step 01

Identify the trusted IP addresses.

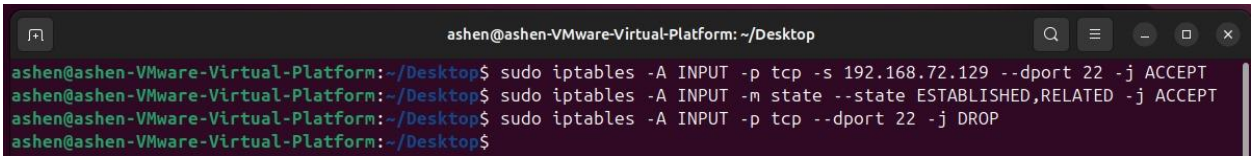
- I have chosen my ubuntu client VM.



Step 02

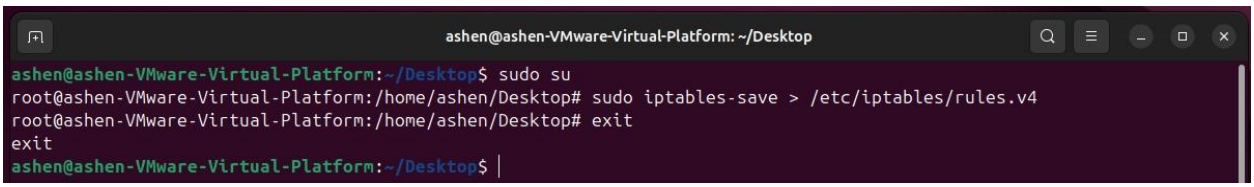
Allow SSH from Trusted Ips and drop all other incoming SSH connections.

- sudo iptables -A INPUT -p tcp -s 192.168.72.129 --dport 22 -j ACCEPT
- sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
- sudo iptables -A INPUT -p tcp --dport 22 -j DROP



Step 03

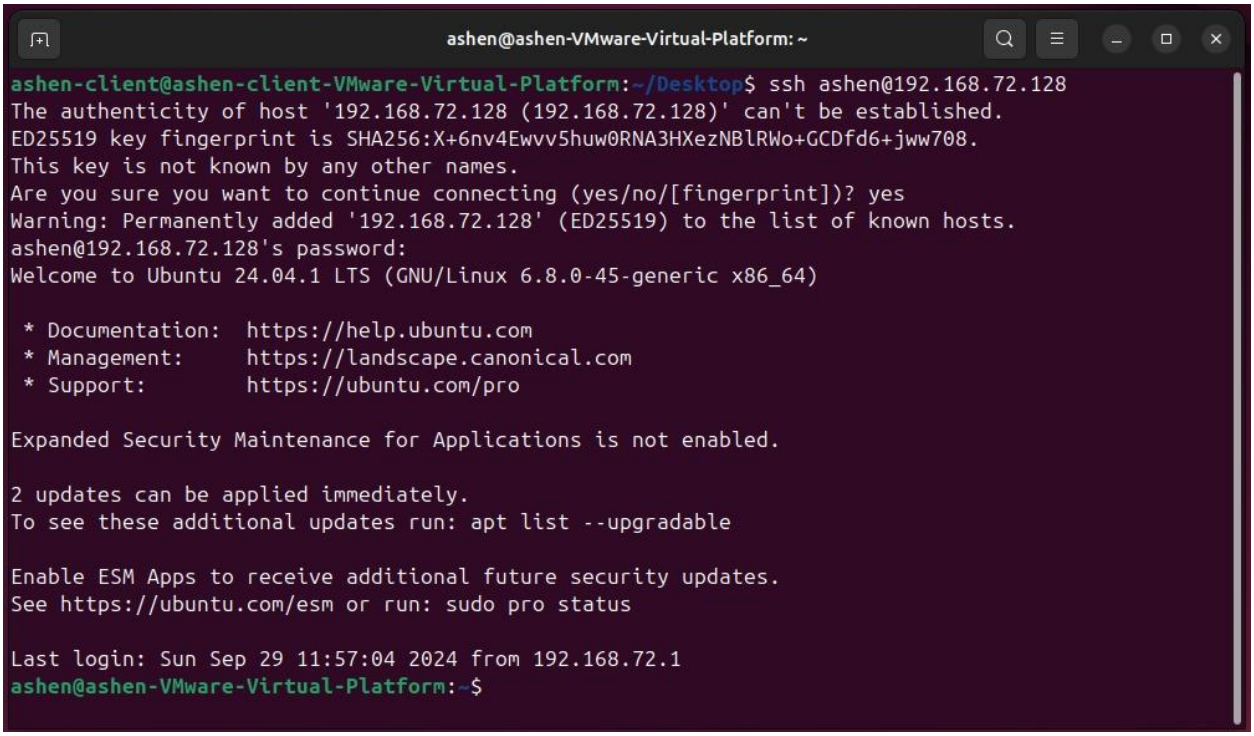
Save the rules.



Step 04

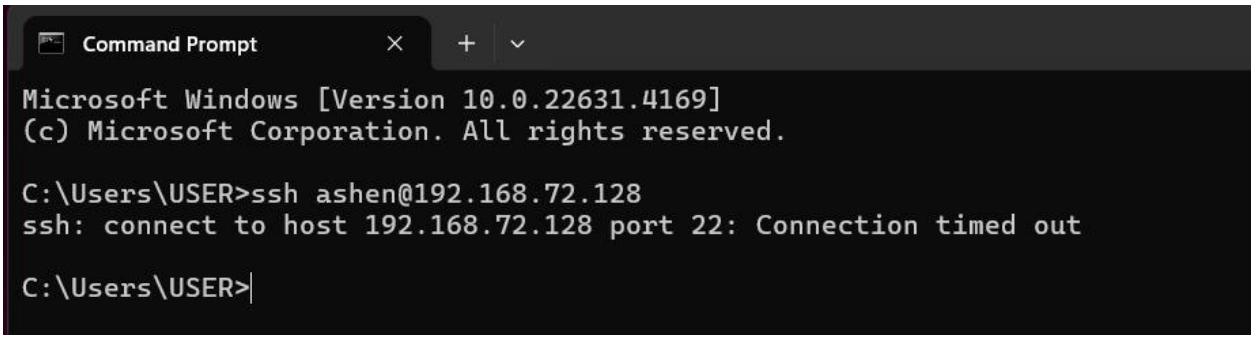
Testing the SSH access.

(From the allowed IP)



(From a blocked IP)

You should get a error message as below.



iii. Allow Specific Applications: If you know the port numbers used by specific applications you want to allow (like a video conferencing app using port 443), you can create an ACL rule to permit traffic only for those ports

Step 01

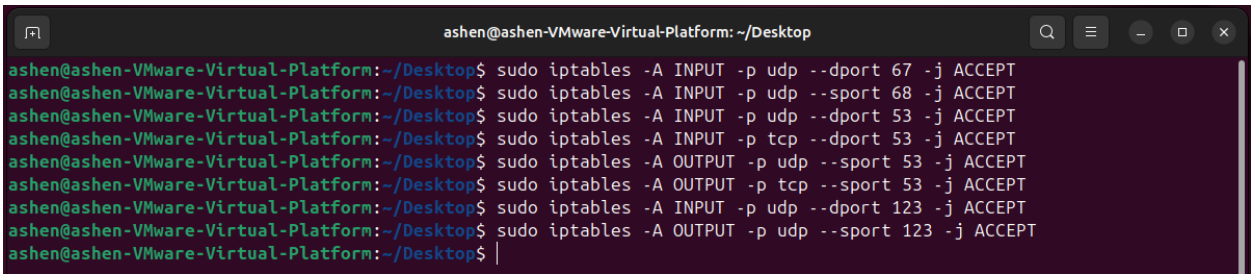
Identify the ports and application services want to allow.

a. DHCP
    a. port 67 - requests to the server
    b. port 68 – responses from the server
b. DNS
    a. port 53
c. NTP
    a. port 123

Step 02

Allow the identified services.

- sudo iptables -A INPUT -p udp --dport 67 -j ACCEPT (Allow DHCP requests to the server)
- sudo iptables -A INPUT -p udp --sport 68 -j ACCEPT (Allow DHCP responses from the server)
- sudo iptables -A INPUT -p udp --dport 53 -j ACCEPT (Allow incoming DNS requests)
- sudo iptables -A INPUT -p tcp --dport 53 -j ACCEPT (Allow DNS zone transfers or large queries)
- sudo iptables -A OUTPUT -p udp --sport 53 -j ACCEPT (Allow DNS responses)
- sudo iptables -A OUTPUT -p tcp --sport 53 -j ACCEPT
- sudo iptables -A INPUT -p udp --dport 123 -j ACCEPT (Allow incoming NTP requests)
- sudo iptables -A OUTPUT -p udp --sport 123 -j ACCEPT (Allow NTP responses)



Step 03

Save the rules.

```
root@ashen-VMware-Virtual-Platform: /home/ashen/Desktop
ashen@ashen-VMware-Virtual-Platform:~/Desktop$ sudo su
root@ashen-VMware-Virtual-Platform:/home/ashen/Desktop# sudo iptables-save > /etc/iptables/rules.v4
root@ashen-VMware-Virtual-Platform:/home/ashen/Desktop#
```

Step 04

Check the rules whether they are applied.

- sudo iptables -L -v -n

```
ashen@ashen-VMware-Virtual-Platform: ~/Desktop
ashen@ashen-VMware-Virtual-Platform:~/Desktop$ sudo iptables -L -v -n
Chain INPUT (policy DROP 37 packets, 4090 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
  221 25865 ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    1    60 ACCEPT     6    --  *      *       192.168.72.129       0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    5   260 DROP       6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     17   --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:67
    0     0 ACCEPT     17   --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:68
    0     0 ACCEPT     17   --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:53
    0     0 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:53
    2   152 ACCEPT     17   --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:123

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 238 packets, 24047 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     17   --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:53
    0     0 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:53
    2   152 ACCEPT     17   --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:123
ashen@ashen-VMware-Virtual-Platform:~/Desktop$
```

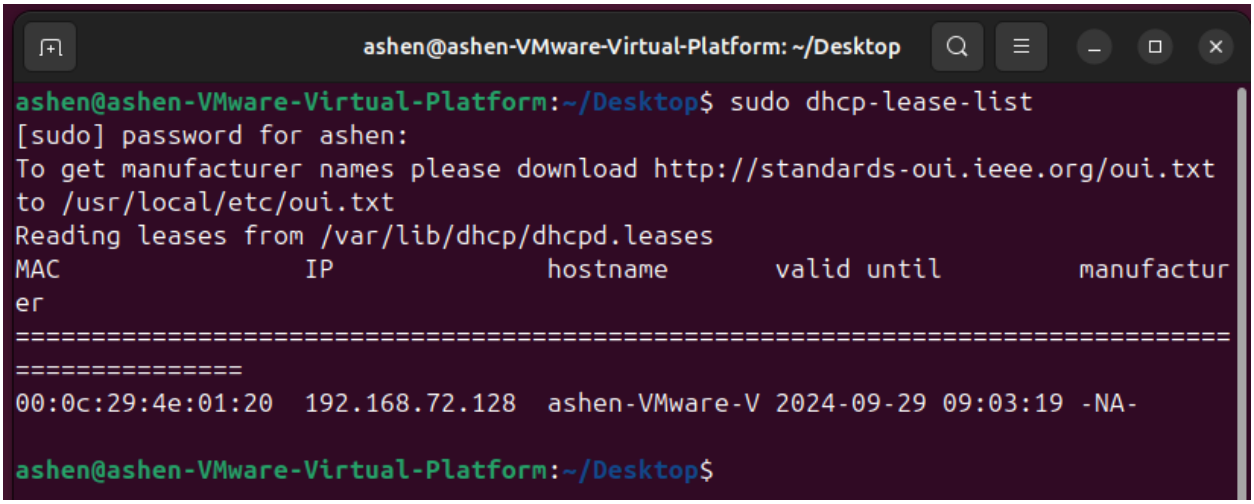
Step 05

Verification

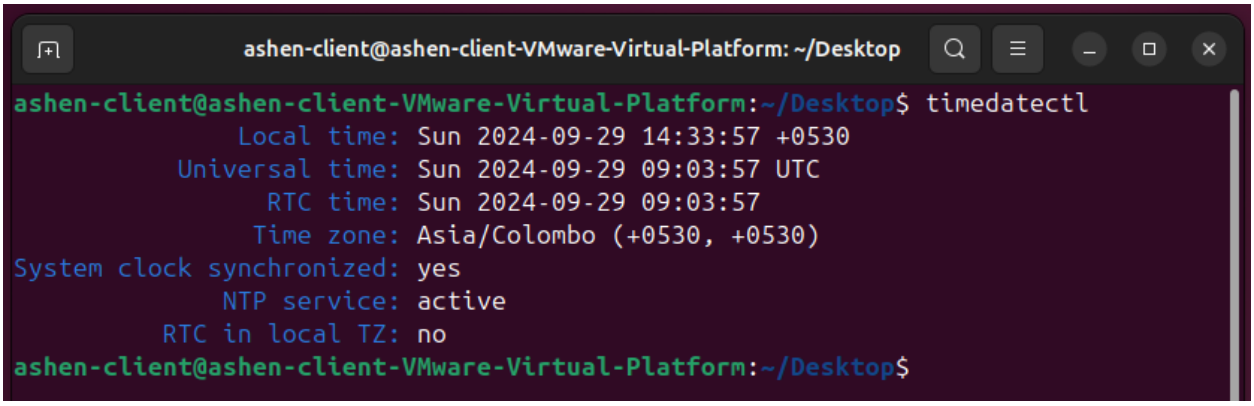
- Checking the DNS

```
ashen-client@ashen-client-VMware-Virtual-Platform: ~/Desktop
ashen-client@ashen-client-VMware-Virtual-Platform:~/Desktop$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.199.46
Name:   google.com
Address: 2404:6800:4001:808::200e

ashen-client@ashen-client-VMware-Virtual-Platform:~/Desktop$
```

- Checking DHCP



- Checking NTP



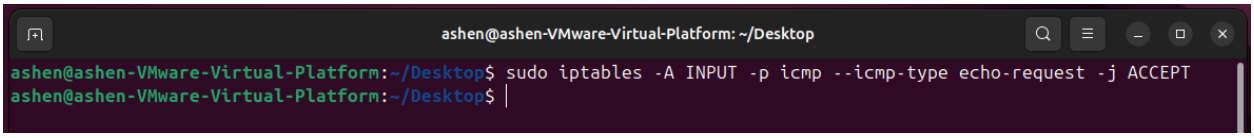
As you can see you will get the output as above if the configurations are successful.

iv. Allow Pings (ICMP Echo Request): This basic rule allows ping requests (ICMP Echo Request) to your machine, which can be helpful for troubleshooting network connectivity

Step 01

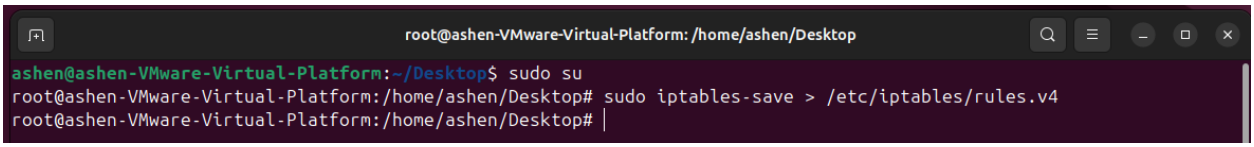
Add the rule to allow ICMP Echo Requests that used for pinging.

- sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT



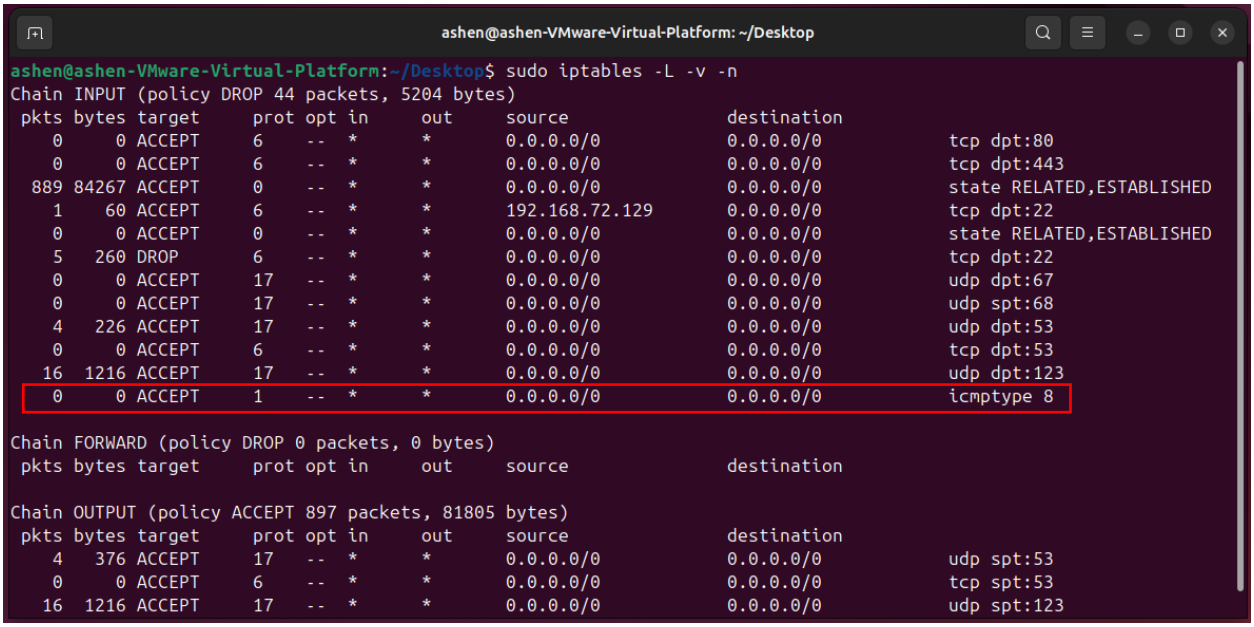
Step 02

Save the rules.



Step 03

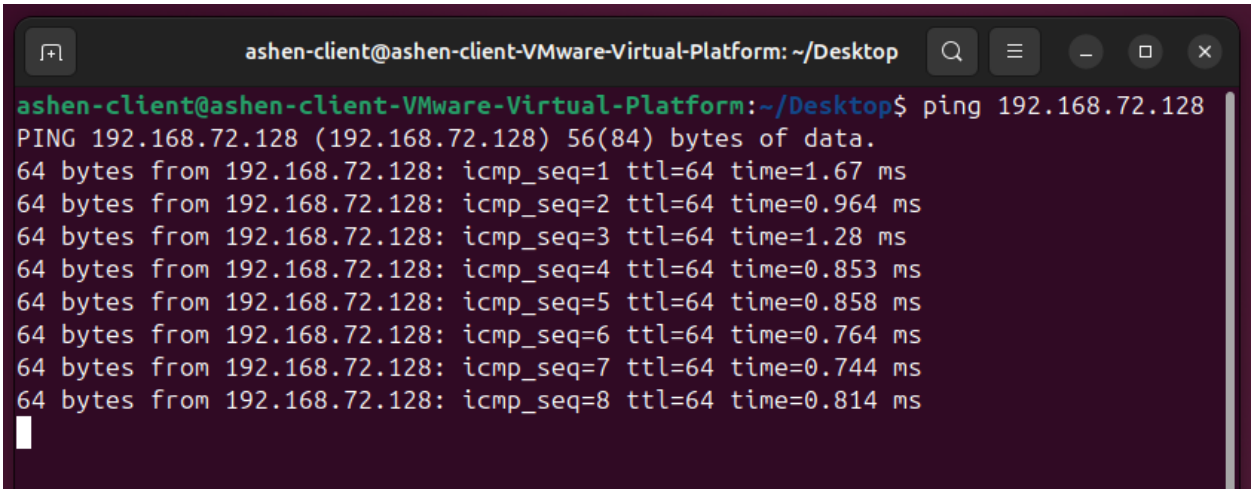
Verify the rules whether applied.



Step 04

Testing the configurations

- ping 198.168.72.128

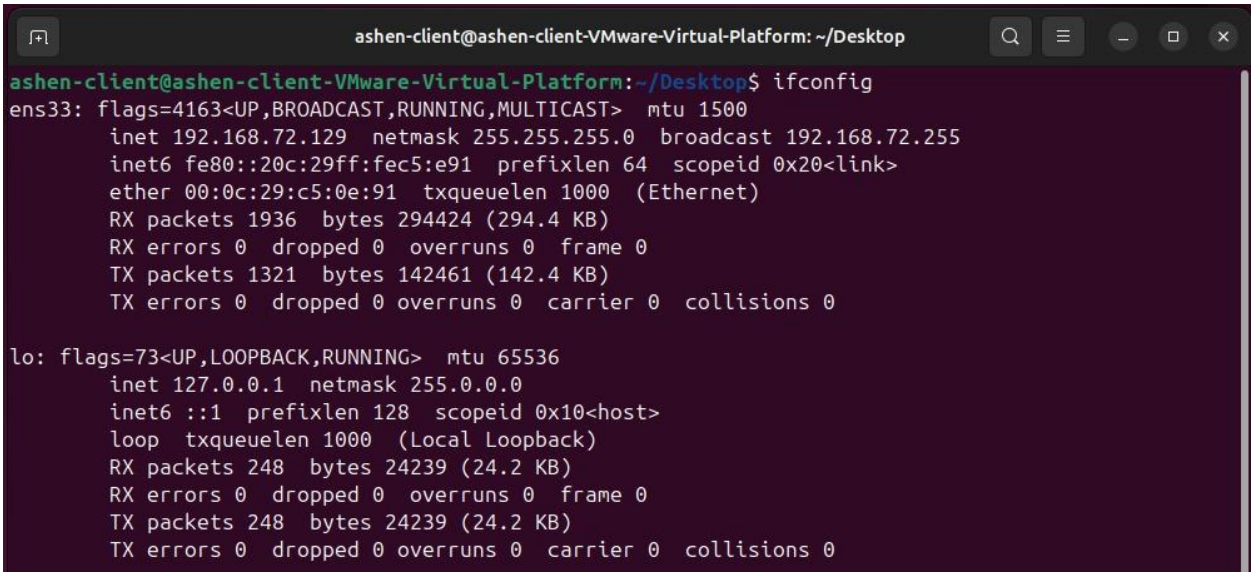


v. **Printer Server Access: For a printer server, allow printing traffic (port 9100) only from specific IP addresses within your local network. Block all external access to the printer server to prevent unauthorized printing.**

Step 01

Identify the trusted IP addresses.

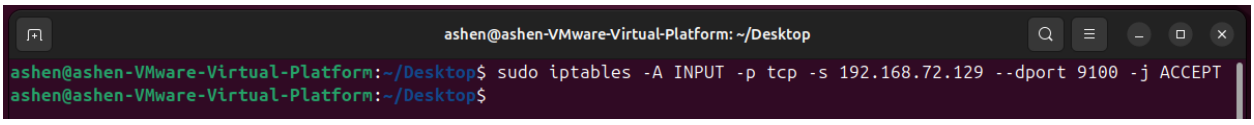
- I have chosen my ubuntu client VM.



Step 02

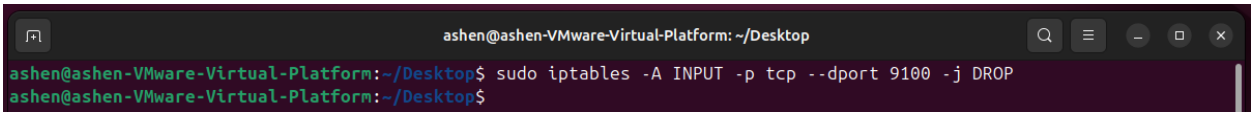
Allow printing traffic from trusted IPs

- sudo iptables -A INPUT -p tcp -s 192.168.72.129 --dport 9100 -j ACCEPT



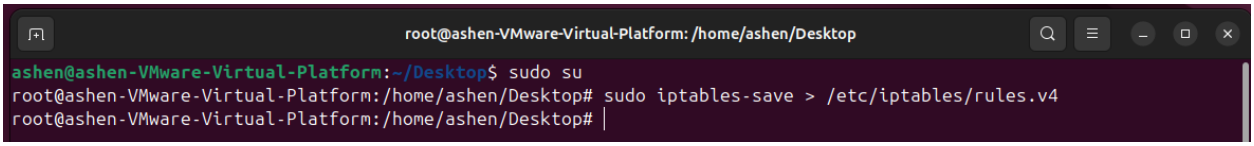
Step 03

Drop all the other incoming traffic from port 9100



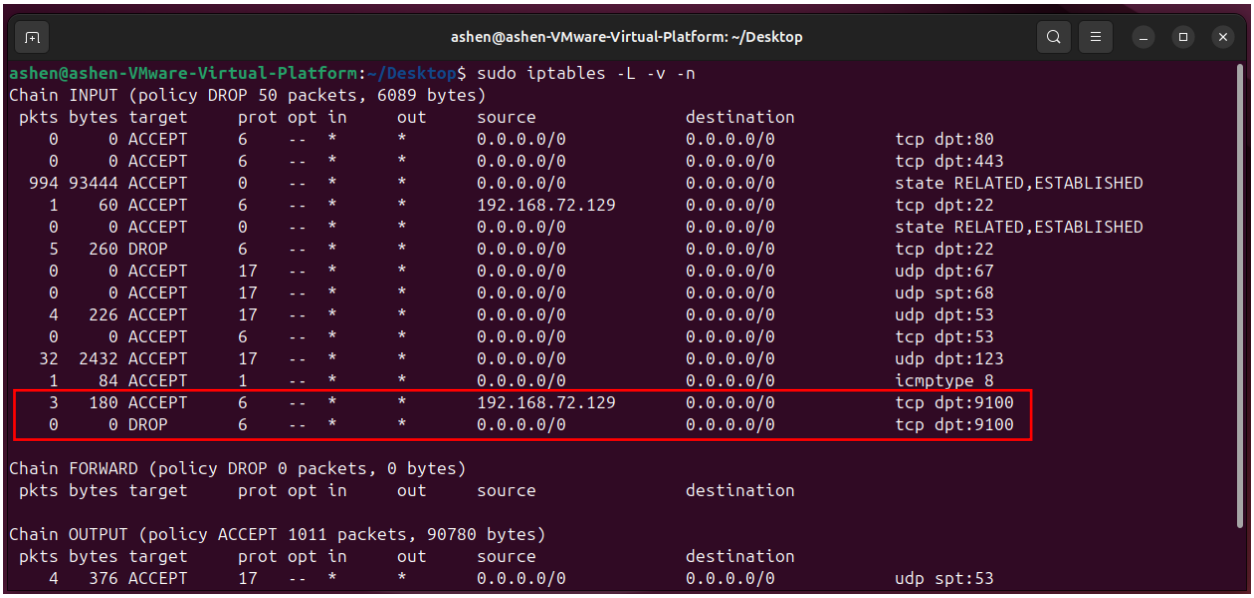
Step 04

Save the rules.



Step 05

Verify .



Step 06

Testing the configurations

- Since I'm working with two Ubuntu VMs and don't have a physical printer, we can simulate access to a "printer server" by testing connectivity on port 9100.

Run the command in server VM.

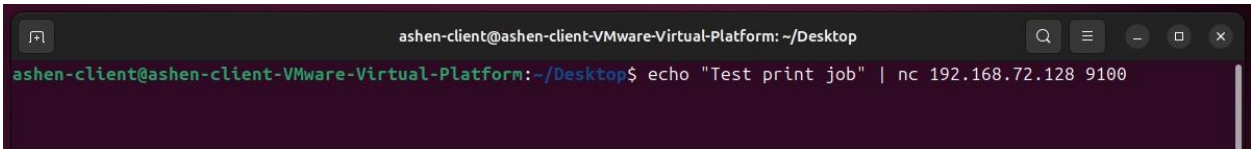
- sudo nc -l -p 9100
    - This command listens for any incoming connection on port 9100, which mimics a printer service.

Run the command in trusted machine.

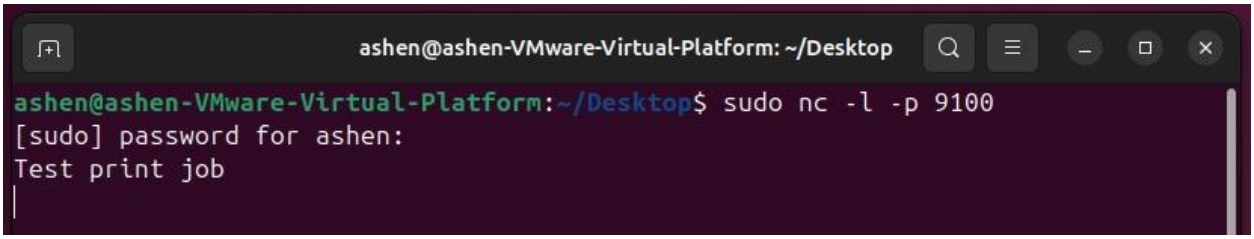
- echo "Test print job" | nc 192.168.72.128 9100

if the connection works we can see "Test print job" in server machine.

Client side



Server side



If you get the sent " Test print job" in the server side , it means that the service works properly.

5. Best Practices

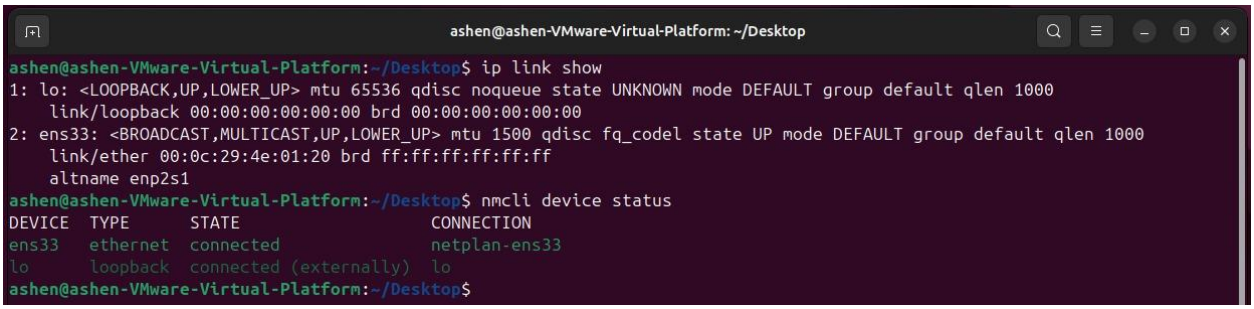
1. Disable Unused Network Interfaces.

- unused network interfaces can be potential attack vector, so disabling them reduce the attack surface.

Step 01

Check all network interfaces.

- ip link show
- nmcli device status



state up – active

state down – inactive or unused

Step 02

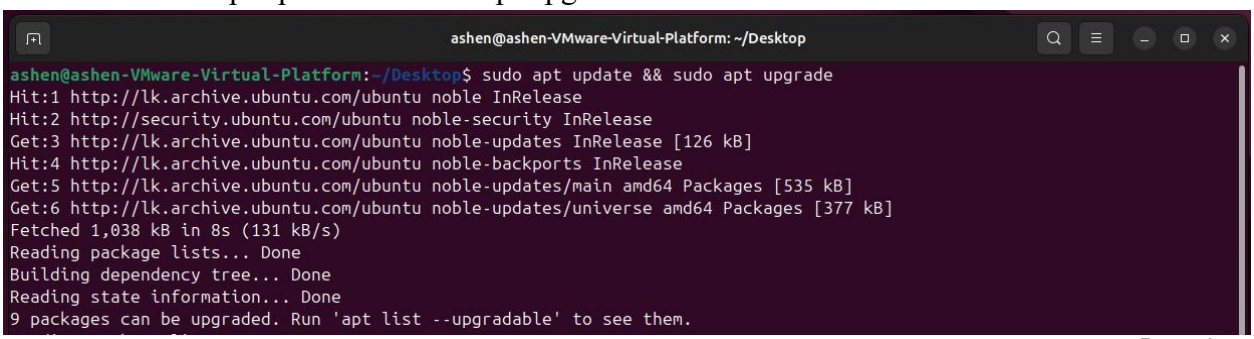
Check for any state down network interfaces and disable them.(In here I don't have any state down Network interfaces)

- sudo ip link set **<interface>** down
  - ex:- sudo ip link set ens33 down

2. Keep the system Updated

- Frequent updates help to ensure that patches are fixed, therefore lower the chance of exploitation.

- sudo apt update && sudo apt upgrade

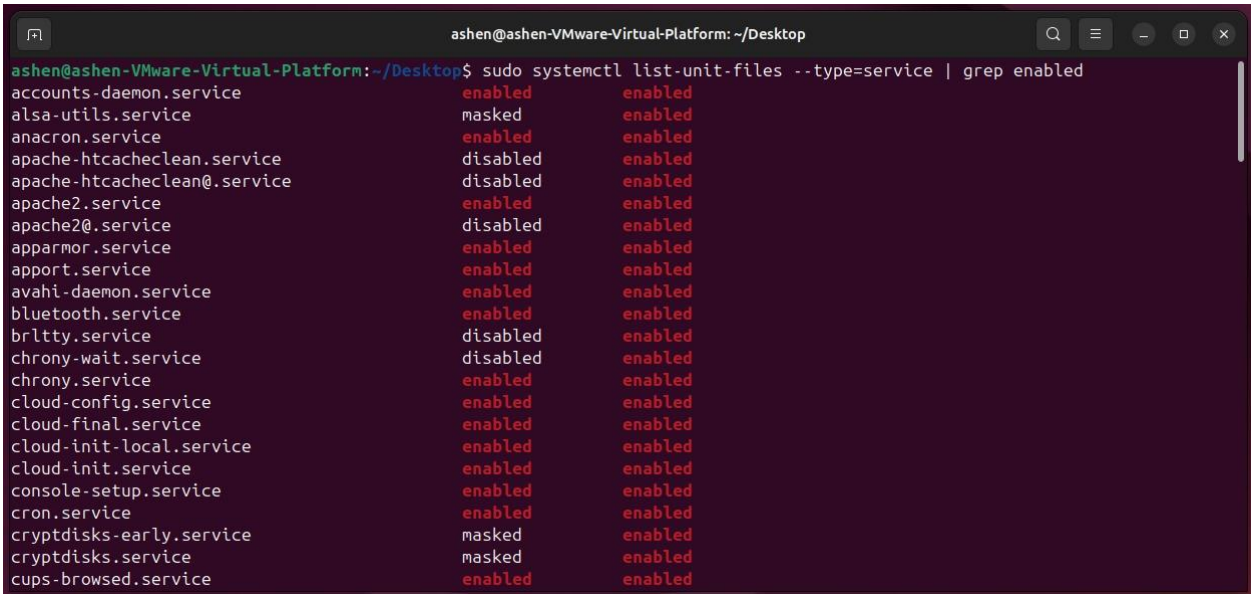


3. **Limit unused Network Services.**
    - Disable or remove any unnecessary network services that are not required, reducing the attack surface.

Step 01

List the enabled services.

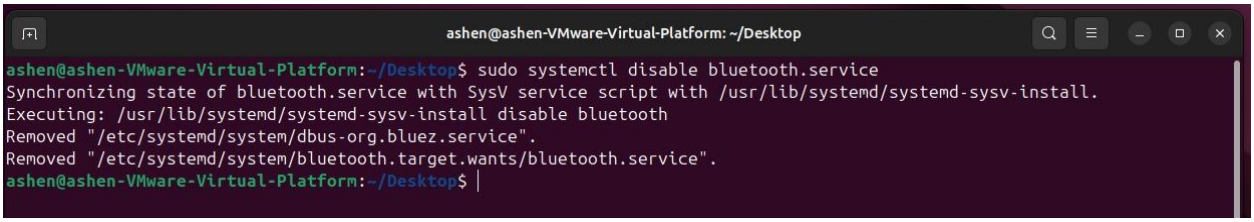
- sudo systemctl list-unit-files --type=service | grep enabled



Step 02

Disable unnecessary services.

- sudo systemctl disable bluetooth.service (Disable if your machine doesn't use Bluetooth, as it reduces the attack surface.)

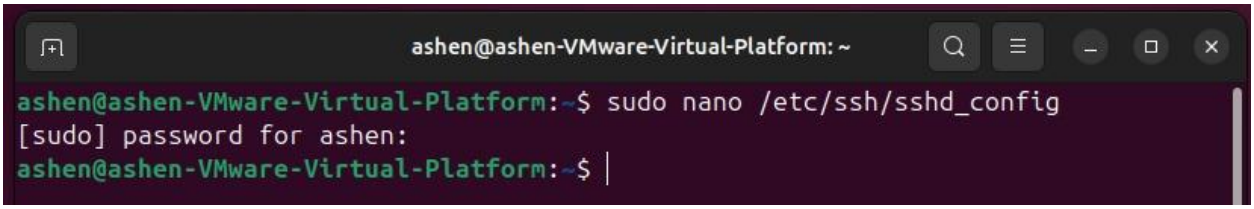


4. **Disable root login via ssh**
    - As we know the root access is highly privileged , the root account has complete control over the system , so it is important to limit root access to reduce the risk of exploitation.

Step 01

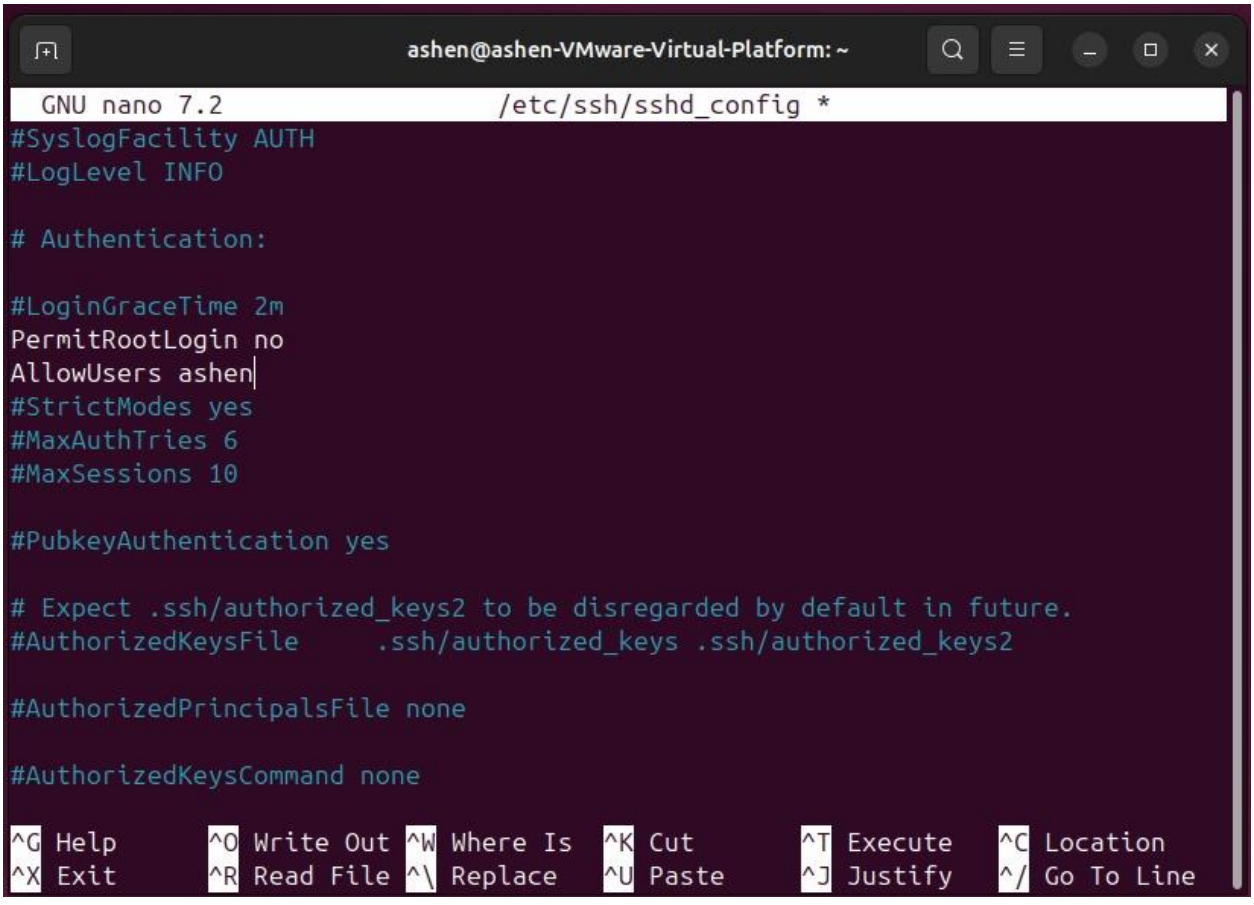
Open SSH configuration file.

- sudo nano /etc/ssh/sshd_config



Step 02

Update or Add PermitRoorLogin and the AllowUser Directive.

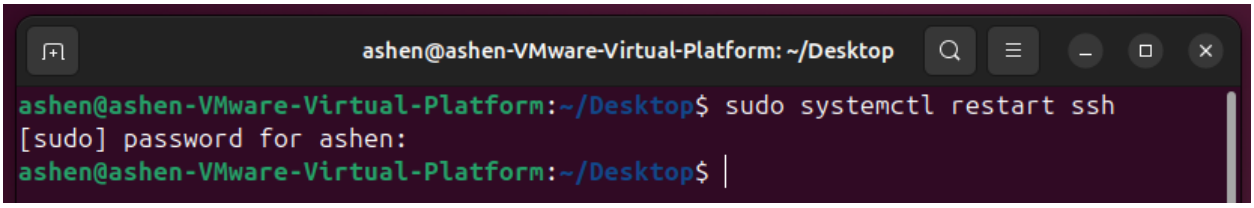


Save and Exit.

Step 03

Restart the ssh service to apply changes.

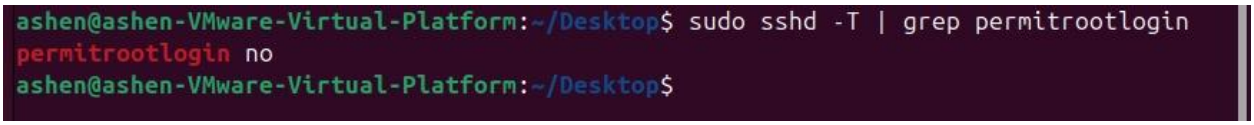
- sudo systemctl restart ssh



Step 04

Verify the configuration

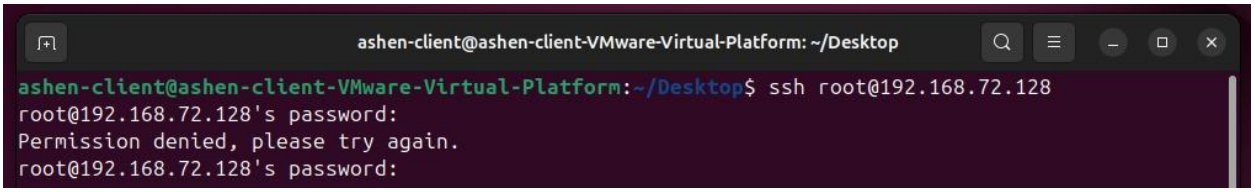
- sudo sshd -T | grep permitrootlogin



Step 05

Test root login using a client machine.

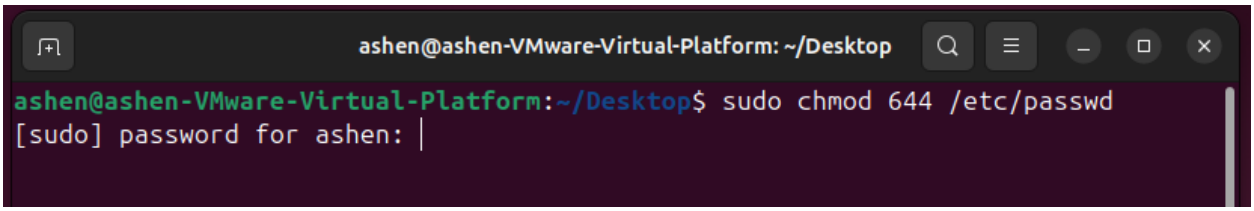
- ssh root@192.168.72.128

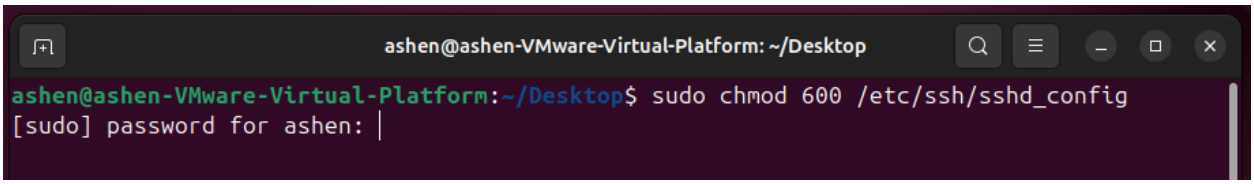


If the configuration is successful, you will get an error message as above.

5. **Use secure file permissions.**
   - Securing configuration files that contain sensitive information. Ensures that script and configuration files are owned by the root.
     - /etc/passwd
       - Contains user account information.
         - sudo chmod 644 /etc/passwd

         (Readable by all users, but only writable by the root user.)
       - /etc/ssh/sshd_config
         - ssh server configuration file
           - sudo chmod 600 /etc/ssh/sshd_config





Not only these, using iptables rules for only to allow trusted parties to use services is also a best practice considering security.