IE2012 - System and Network Programming

Assignment - 2025 (Feb/June)

H.M. Buddhima Herath
Y2S1

1. Basics of Linux Environment.

1.1. Virtual Machine Setup:

The following steps will successfully be implemented virtual machine software and Linux Distribution.

Step 01: Select one software to run virtual machine.

- Download VMware Workstation Player from official website: https://www.vmware.com/products/workstation-player/workstation-player evaluation.html
- Download VirtualBox from the official website: https:/www.virtualbox.org/wiki/Downloads

Step 02: Download an .iso file from the Ubuntu or CentOS website.

- Ubuntu: (https://ubuntu.com/download/desktop)
- CentOS: (https://www.centos.org/download/)

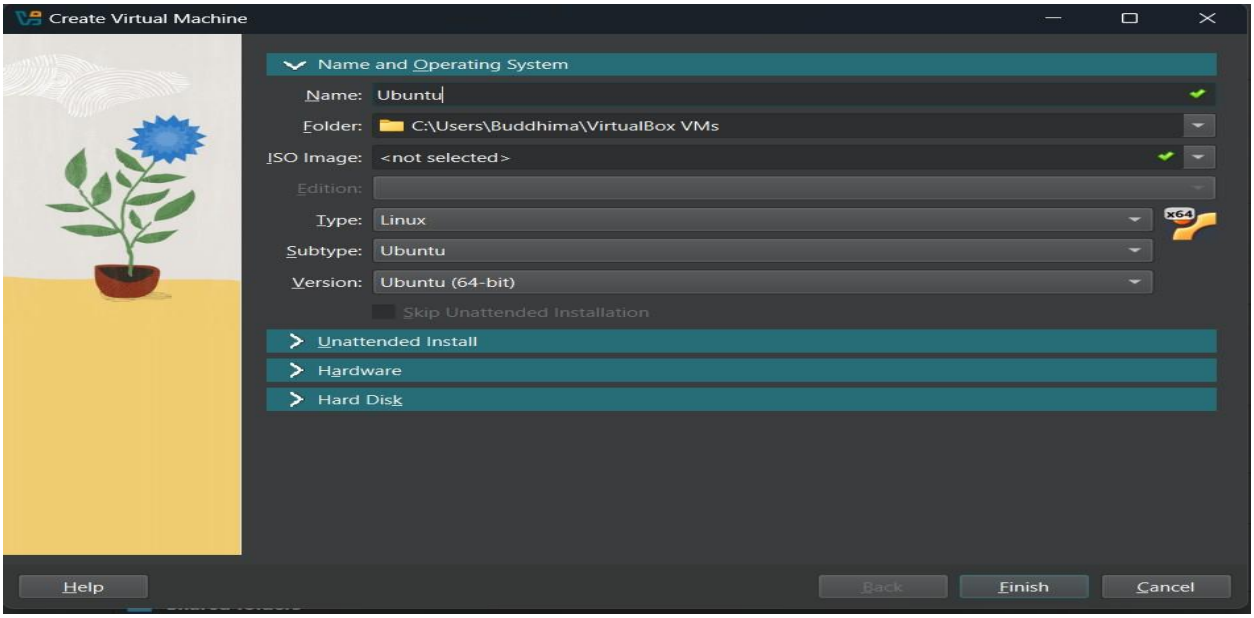
Step 03:

Open the Virtual Machine software and follow the guidelines below, In

VirtualBox:

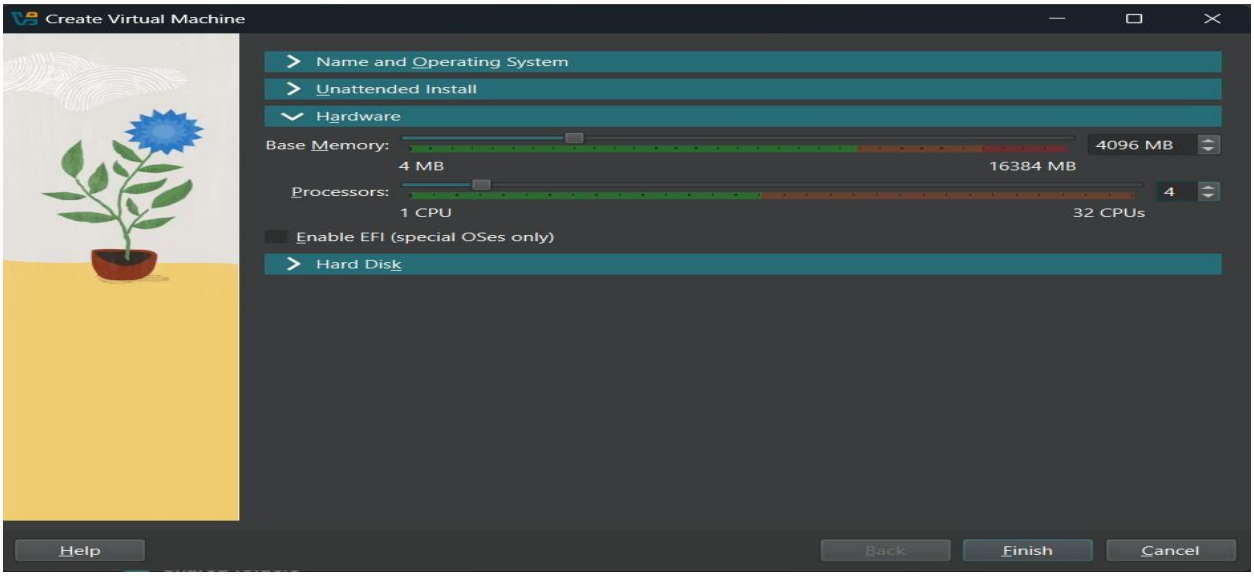
Open the VirtualBox software to create a new virtual machine.

1. Create a new virtual machine click on NEW icon.
2. Name: Ubuntu
3. Set the type and version as Linux and Ubuntu (64-bit)

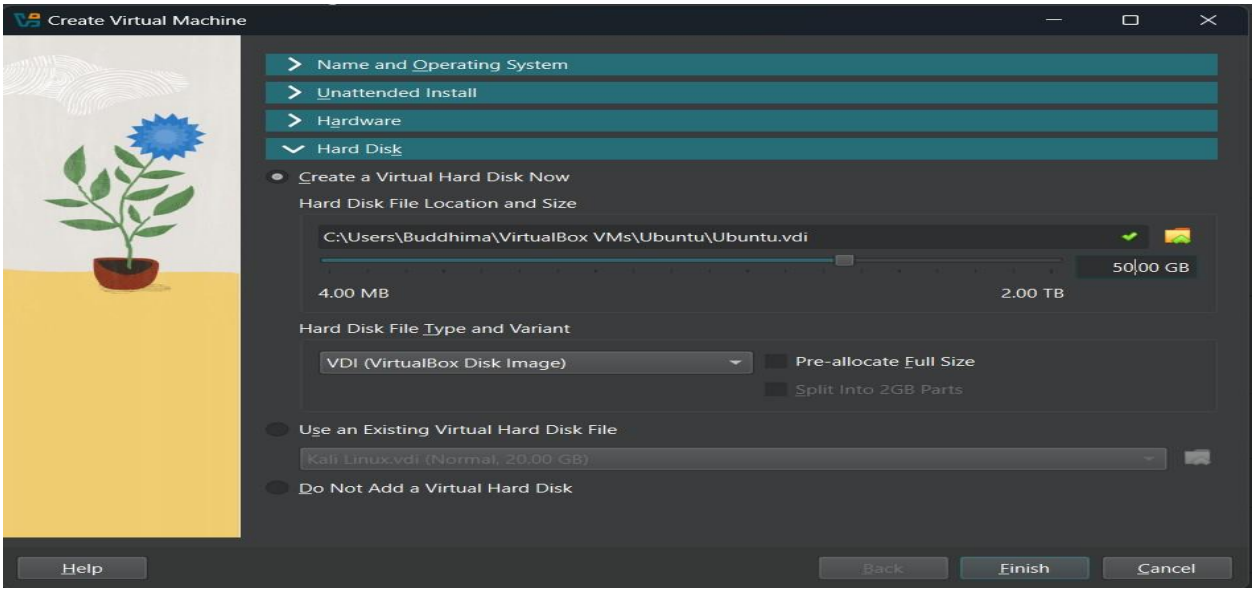


4. Click on Hardware

    - Base Memory (RAM): 4096 MB
    - Processor: 4

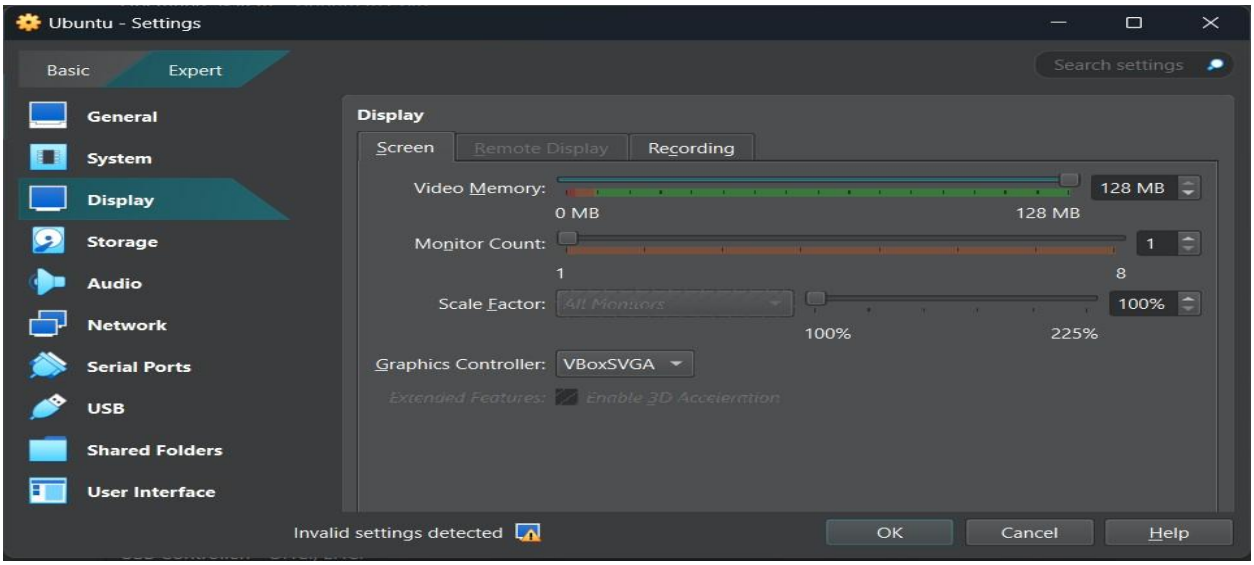


5. Click on Hard Disk
    - Change the size of the Hard Disk File Location into 50 GB

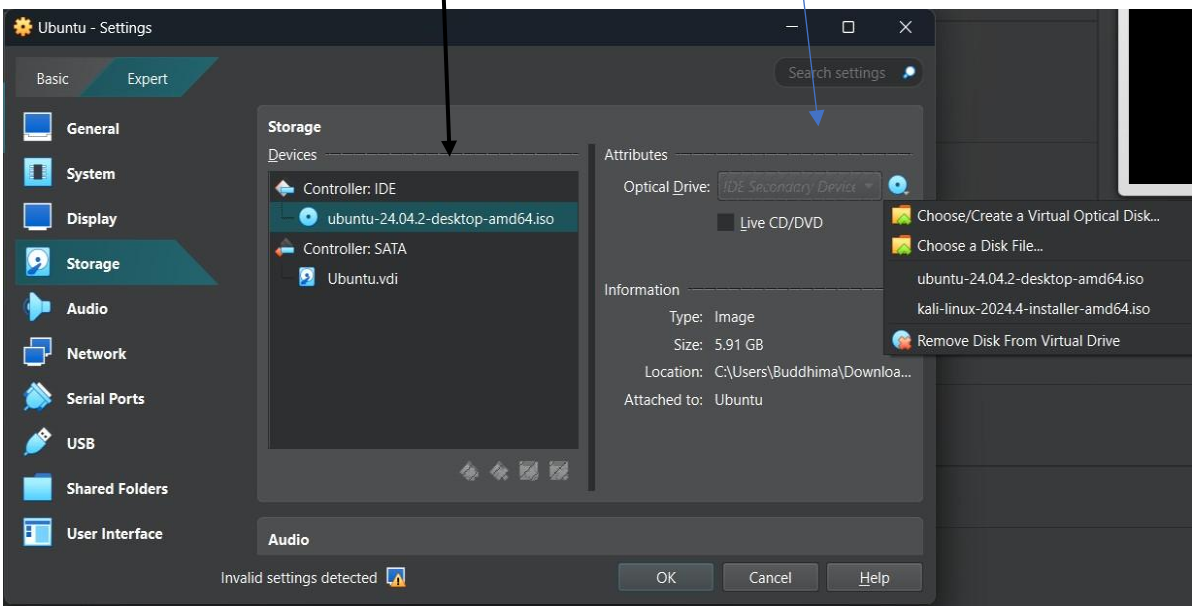


6. Open Ubuntu settings,
    - Change the display video memory size into 128 MB.

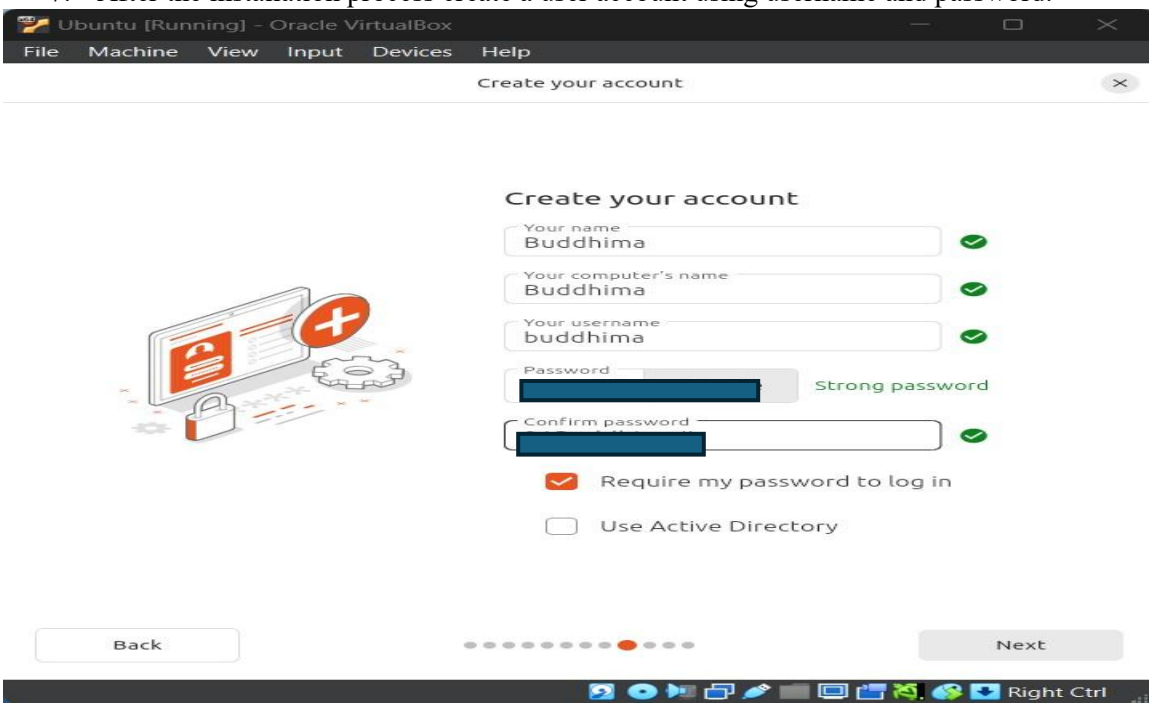
- Change the display Graphics Controller to VBoxSVGA.



- Change the Optical Drive as IDE Secondary Device 0
- Attach the downloaded ISO file to the empty controller IDE (ubuntu-24.04.2-desktop-amd64.iso)



7. After the installation process create a user account using username and password.

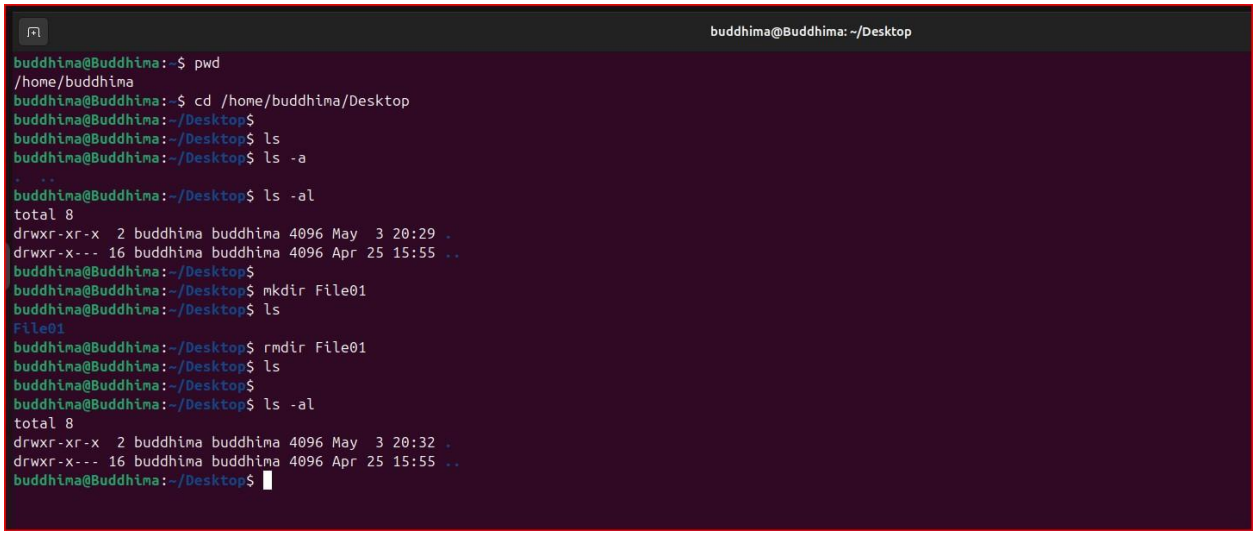


8. Login with the user credentials after rebooting the Virtual Machine.

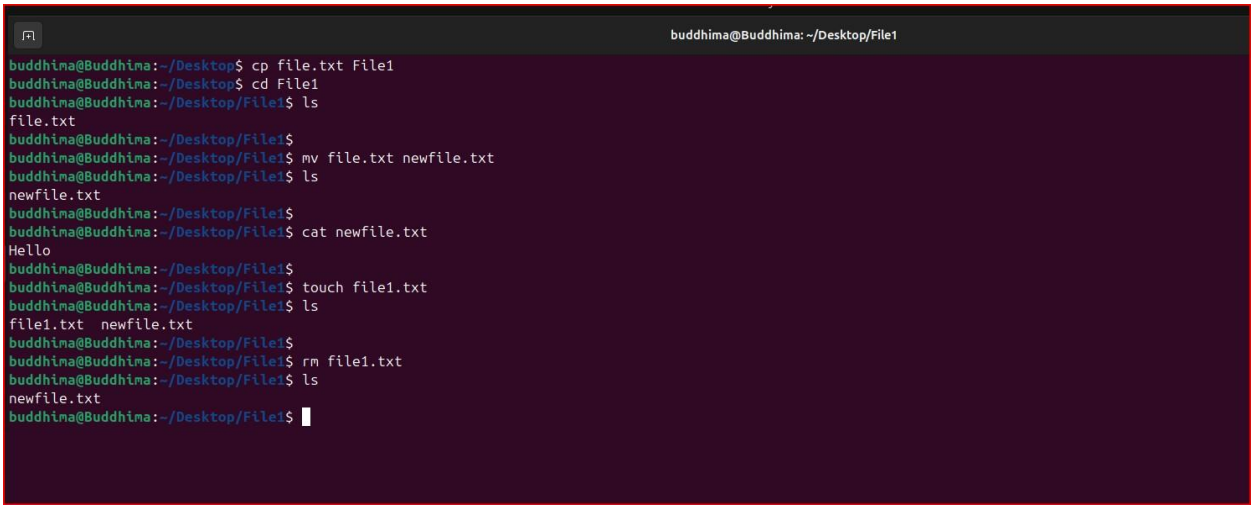
1.2. Command Line Introduction.

The following commands will be interacted with the Linux system to navigate the Linux environment effectively.

Navigation Commands:	
1.**pwd**	• This command will print the current working directory.
2.**ls ls** **-a ls** **-al**	• • This command will list files and directories. • This command will list files and directories with hidden files. This command will list all files and directories with size, permission, date.
3.**mkdir**	• This command will create a directory.
4.**rmdir**	• This command removes directory with all contents.



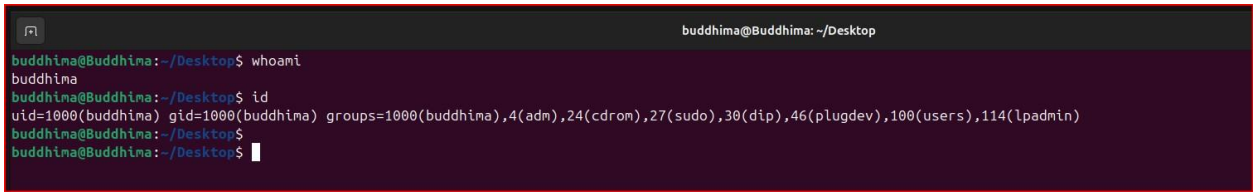
File Manipulation Commands.	
5.**cd –** **cd .. –**	• This command will change the working directory. This • command will move one directory up.
6.**cp –**	• This command will copy file.
7.**mv –**	• This command will move or rename file.
8.**touch –**	• This command will create an empty file.
9.**rm –**	• This command will remove file.



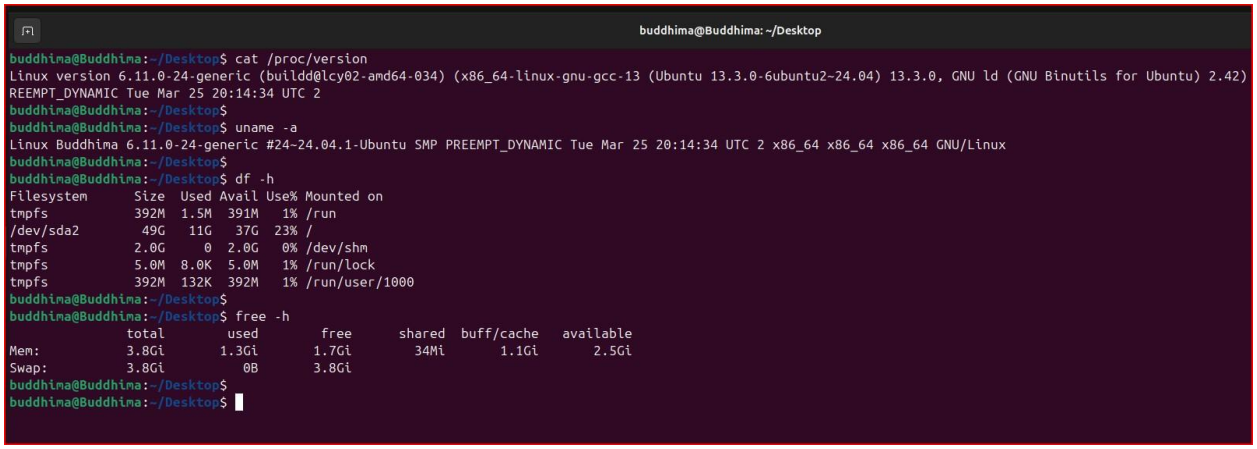
1.3. System Information and User Management.

Can retrieve the system information of computer's hardware, software and performance. Explore the user management commands, checking system status, and troubleshooting.

User Management Commands:	
10.**whoami** –	• This command will Shows the current username.
11.**id –**	• This command will show user ID and group ID.



System Information Commands:	
12.**cat /proc/version** – **cat –**	• This command will show Linux version information by reading the /proc/version file. • This command will show the content of a file.
13.**uname -a –**	• This command will confirm the version of Linux running. (print the kernel name, version, machine type and OS)
14.**df -h –**	• This command will show the disk space usage of all mounted filesystem. (In a human-readable format – GB/MB)
15.**free -h –**	• This command will show memory usage and available memory. (In a human-readable format.)



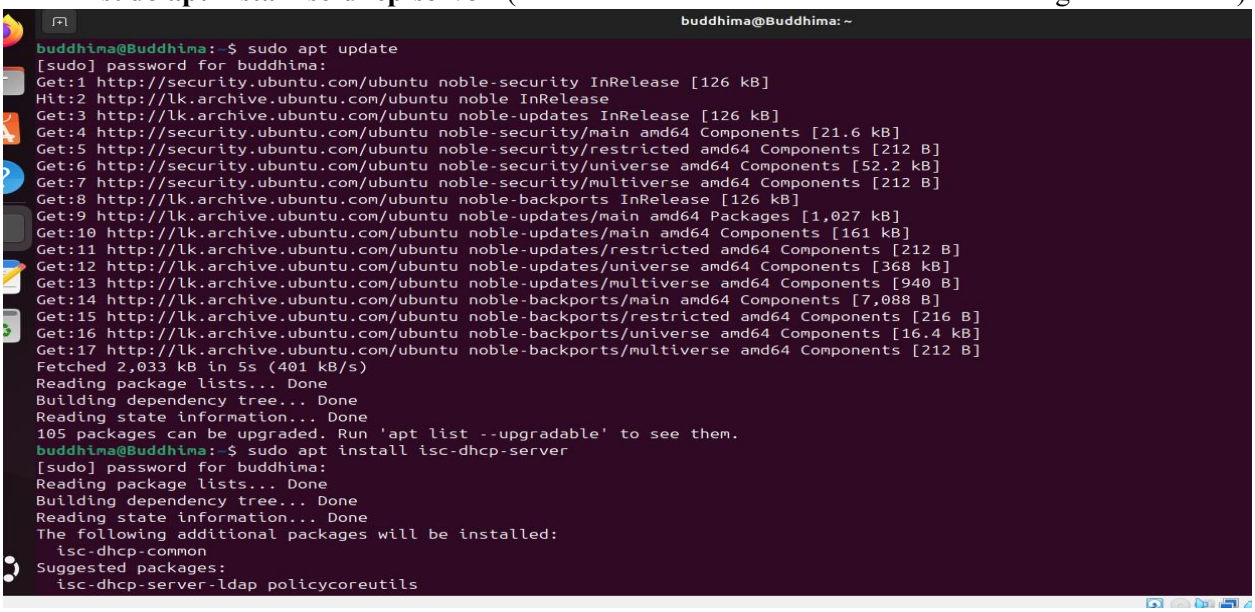
2. DHCP, DNS and NTP Services.

2.1. DHCP (Dynamic Host Configuration Protocol) Network Service:

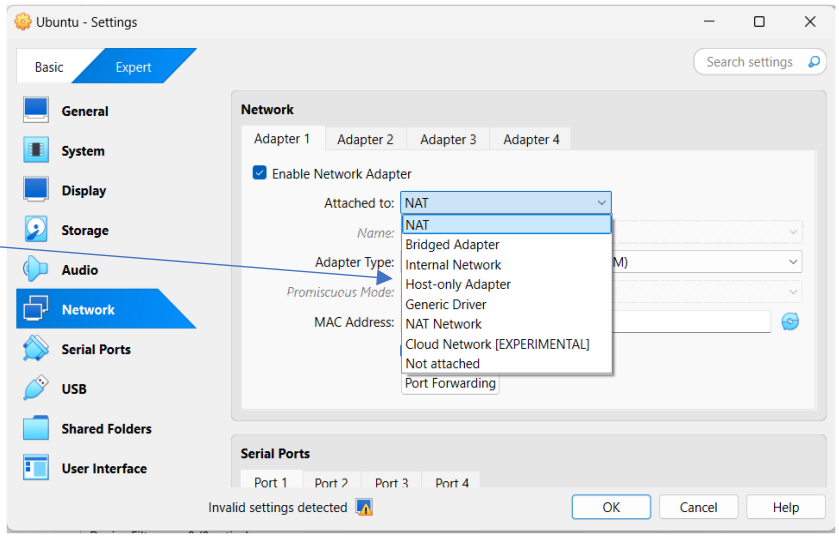
DHCP Network Service assigns automatically IP addresses, subnet masks gateway and configures other networks like DNS servers to devices in a network. It helps to manage IP addresses without manually assigning them.

Step 01: Install the DHCP server using the commands below.

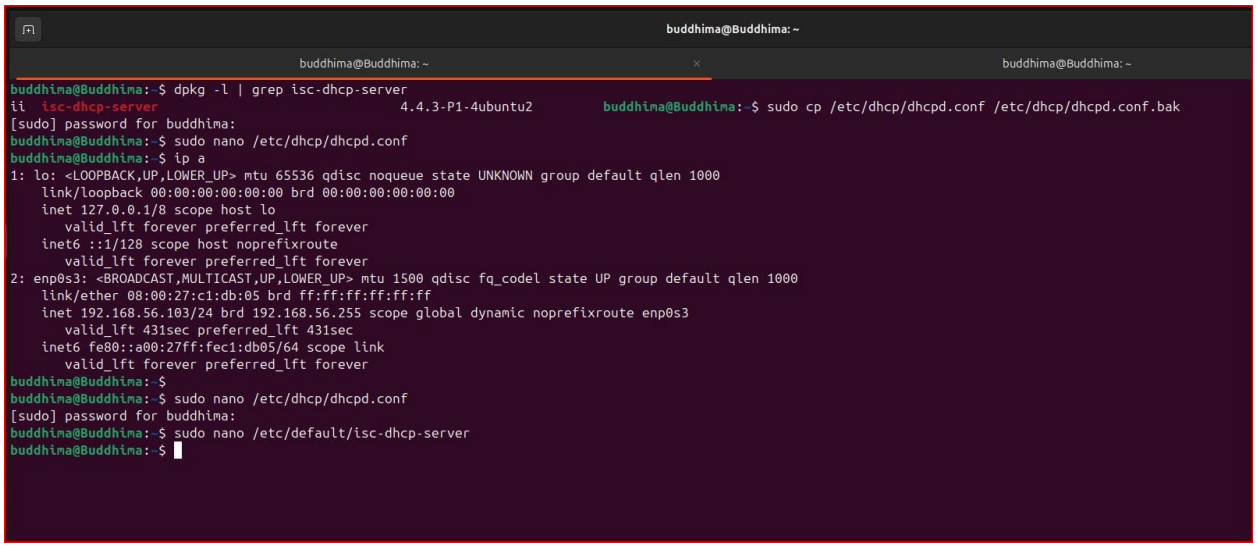
- **sudo apt update** (update the list of packages that are available and the version)
- **sudo apt install isc-dhcp-server** (install the ISC DHCP server and it allows assign IP addresses)



After the installation of DHCP should change the network setting from NAT to Host-Only, cause NAT is used for internet access and Host-Only used to create private network between Local host and the VMs.

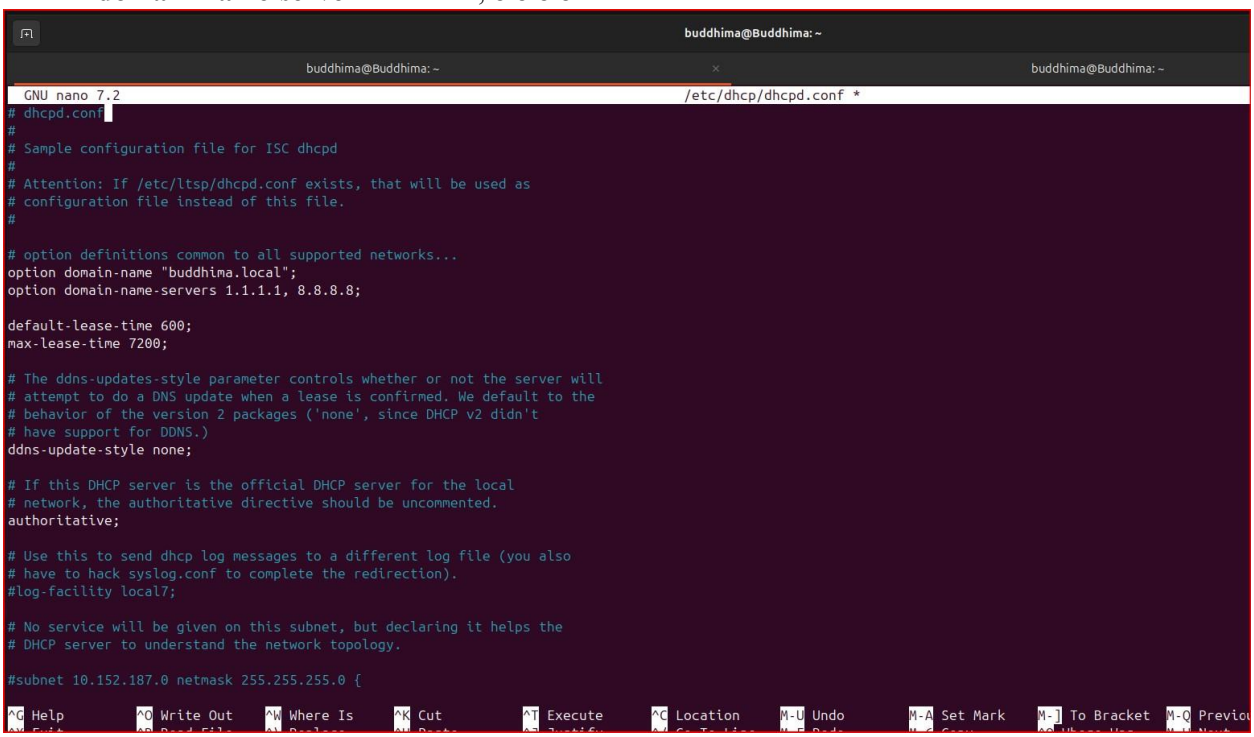


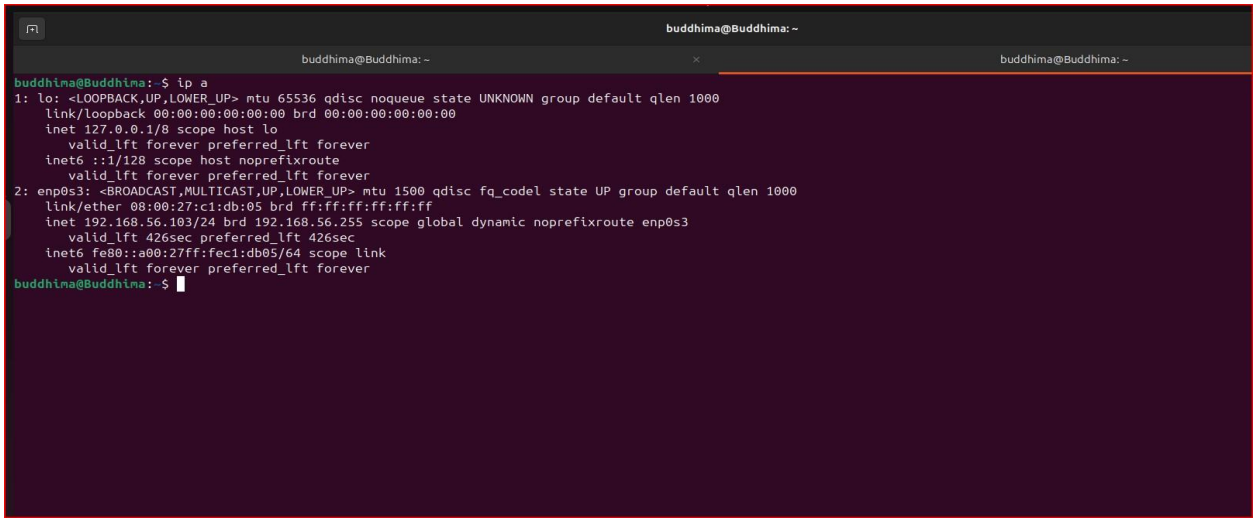
- dpkg -l | grep isc-dhcp-server – check DHCP is installed successfully.



Step 02: Configure DHCP

- sudo nano /etc/dhcp/dhcpd.conf - Using nano editor open dhcpd.conf file.
- Inside the dhcpd.conf file- change **option domain-name** as “buddhima.local” and **optional domain-name-server** as 1.1.1.1, 8.8.8.8.

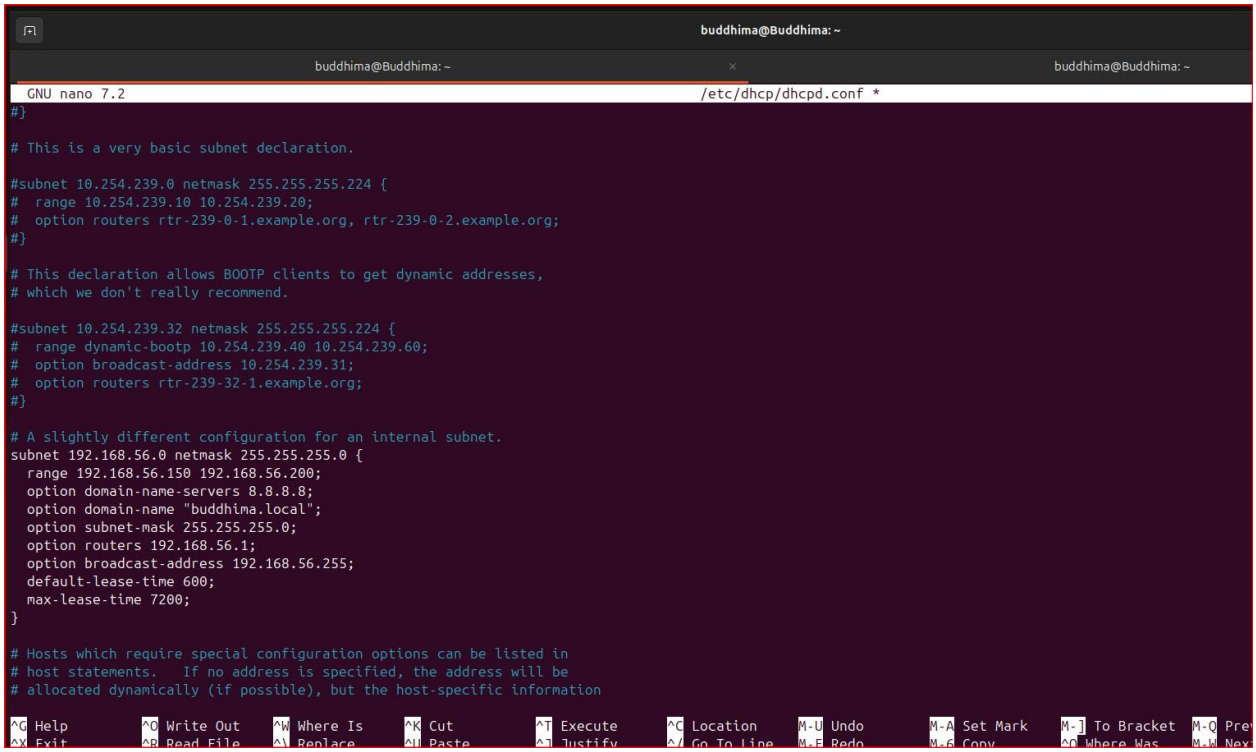




- Add below command to the configuration file, subnet 192.168.56.0 netmask 255.255.255.0 { range 192.168.56.100 192.168.56.200; option routers 192.168.56.1; option domain-name-servers 8.8.8.8; default-lease-time 600; max-lease-time 7200; }

Above mentioned commands define the network range, subnet mask, domain name server and max lease time.

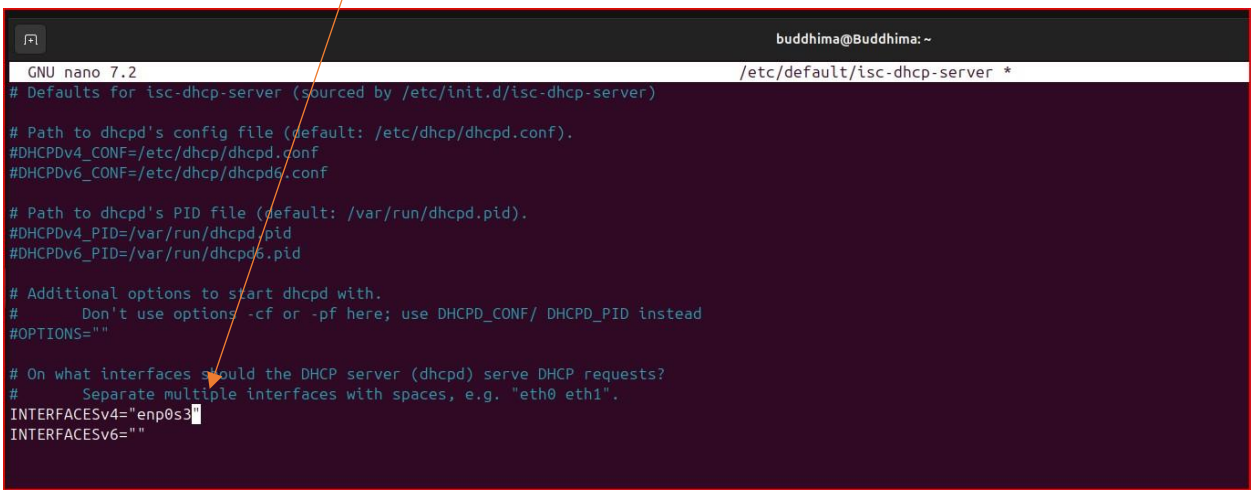
- **Ip a** – using IP a command shows all the interface of network, and the IP address details also.



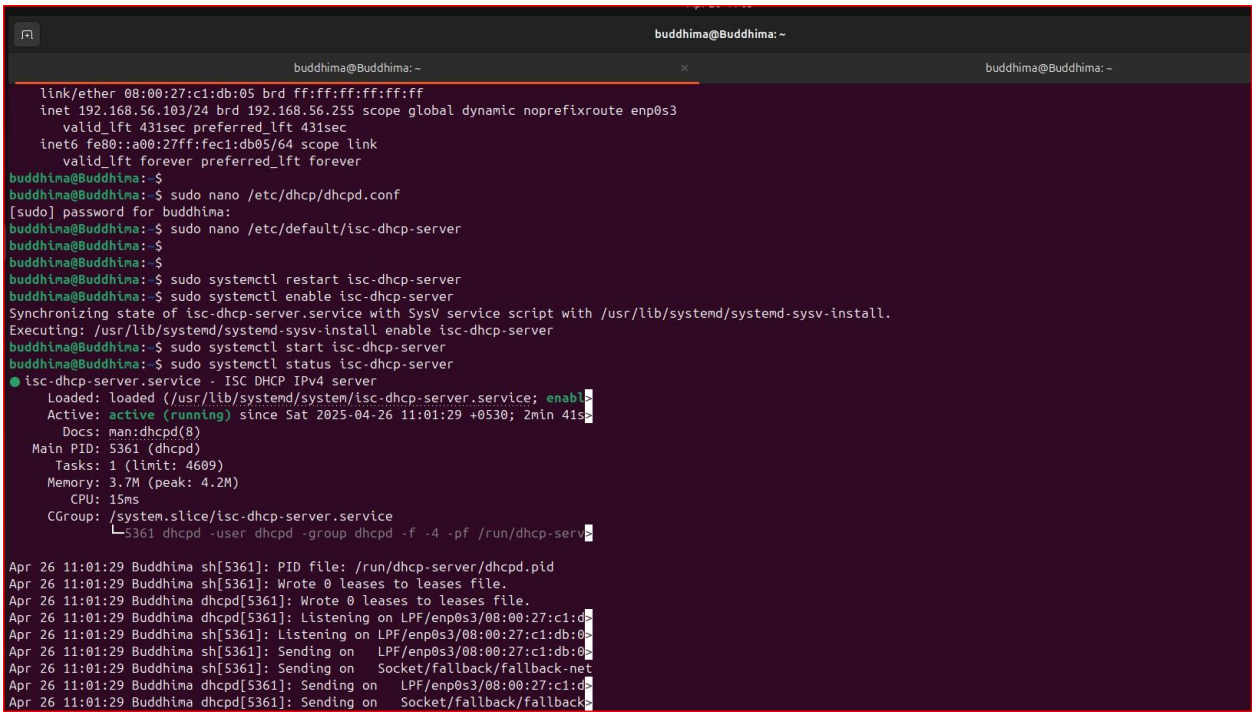
Step 03: Set the interface of DHCP Server

- **sudo nano /etc/default/isc-dhcp-server**- Set the network interface that already find from ip a command.
- Set as **INTERFACESv4="enp0s3"**

Network interfaces tell the DHCP to operate on the enp0s3 interface (Host-Only mode in VirtualBox)



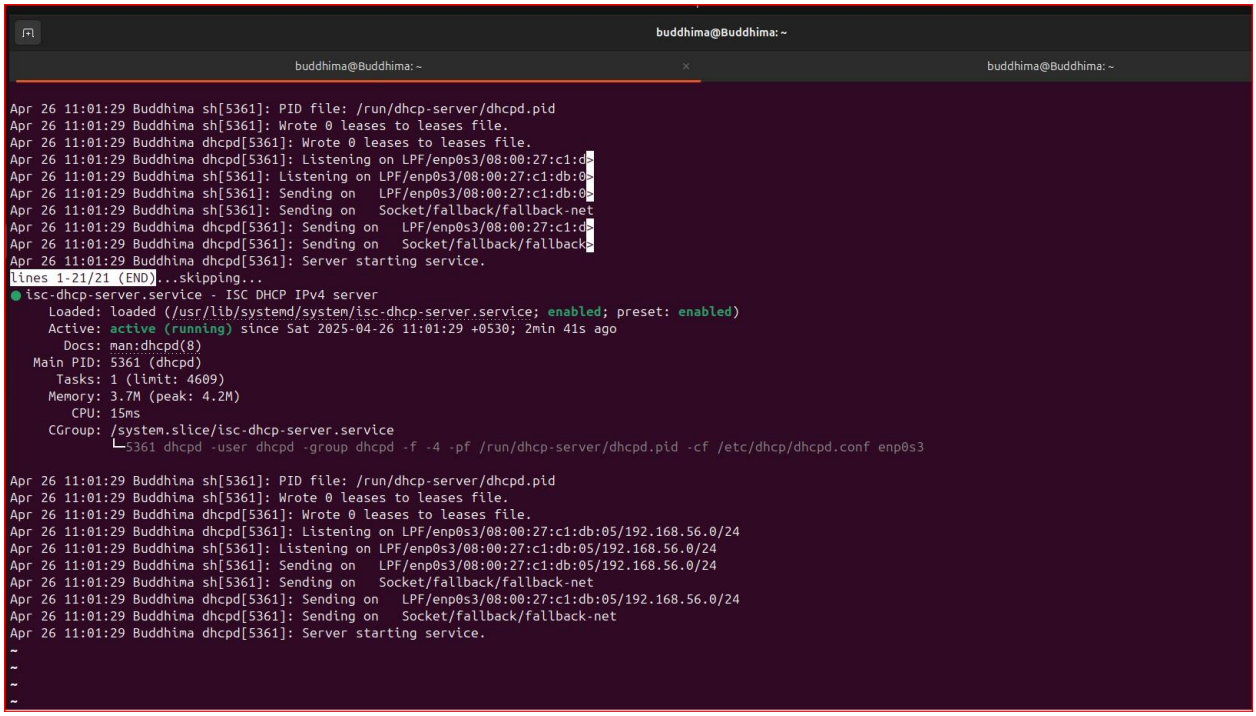
Step 04: Restart and Enable the DHCP server.



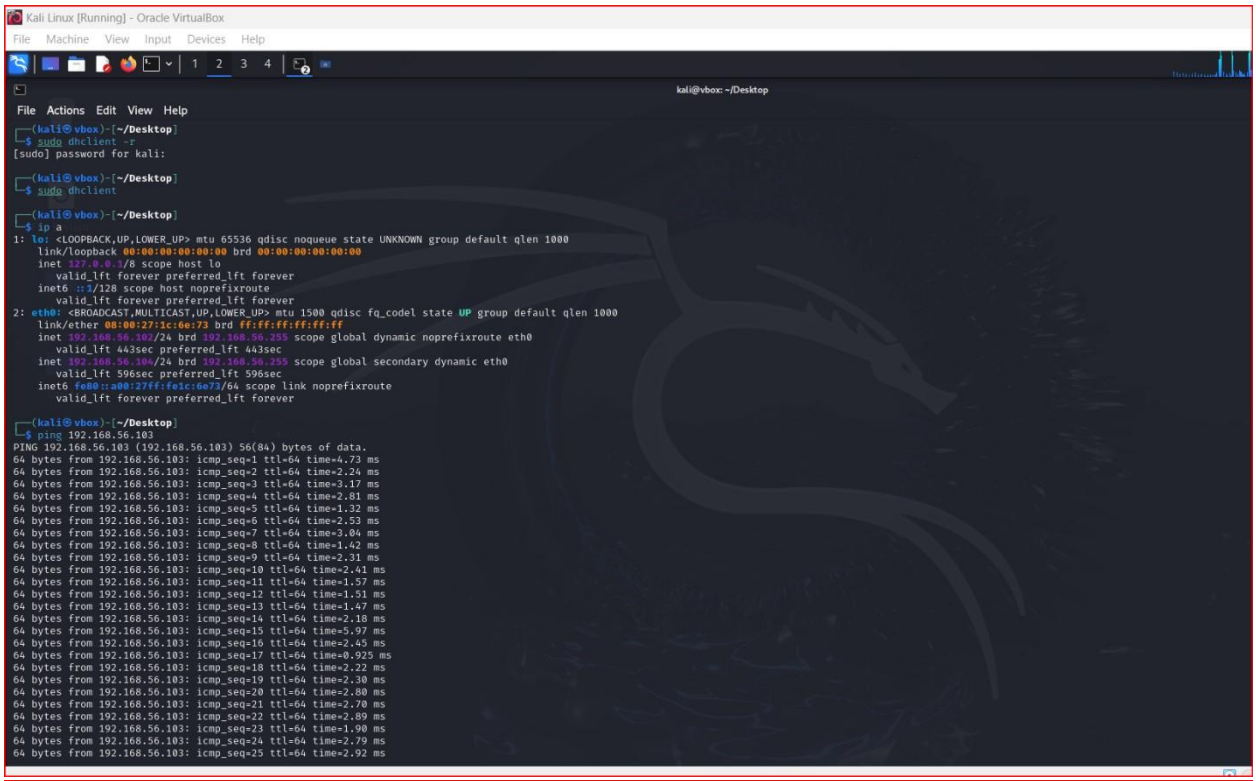
- **sudo systemctl restart isc-dhcp-server** – This command restarts the DHCP server applying any changes that made in conf file.
- **sudo systemctl enable isc-dhcp-server** - This command enables the DHCP server to start automatically.

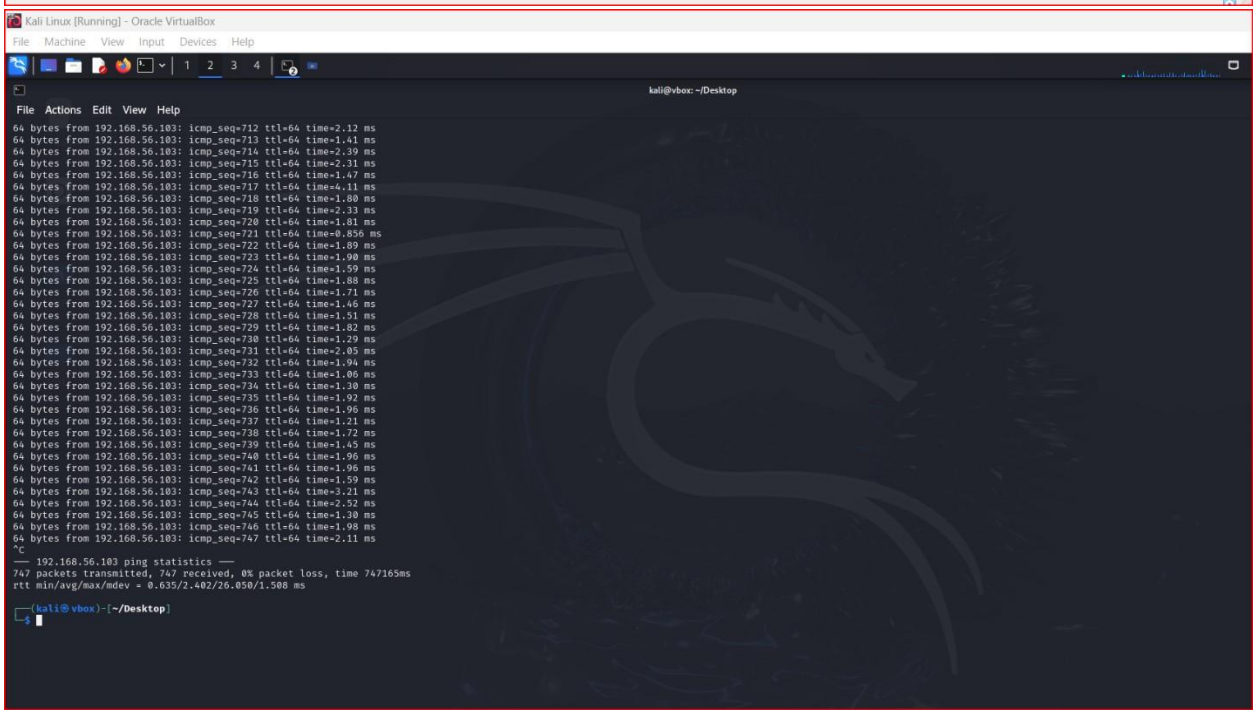
Step 05: Check the system status.

- **Sudo systemctl status isc-dhcp-server**- check the current status of ISC DHCP server whether it's active or fail.

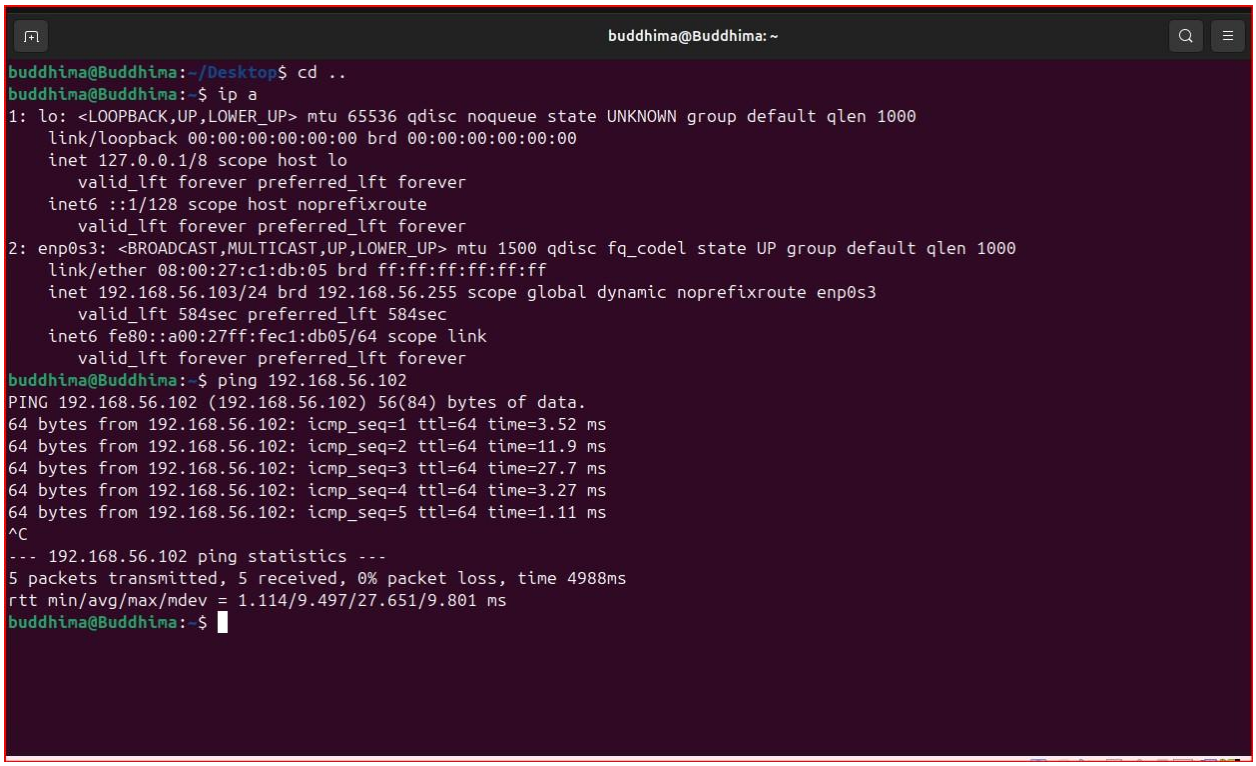

buddhima@Buddhima: ~
buddhima@Buddhima: ~
buddhima@Buddhima: ~
Apr 26 11:01:29 Buddhima sh[5361]: PID file: /run/dhcp-server/dhcpd.pid
Apr 26 11:01:29 Buddhima sh[5361]: Wrote 0 leases to leases file.
Apr 26 11:01:29 Buddhima dhcpd[5361]: Wrote 0 leases to leases file.
Apr 26 11:01:29 Buddhima dhcpd[5361]: Listening on LPF/enp0s3/08:00:27:c1:d
Apr 26 11:01:29 Buddhima sh[5361]: Listening on LPF/enp0s3/08:00:27:c1:db:0
Apr 26 11:01:29 Buddhima sh[5361]: Sending on LPF/enp0s3/08:00:27:c1:db:0
Apr 26 11:01:29 Buddhima sh[5361]: Sending on Socket/fallback/fallback-net
Apr 26 11:01:29 Buddhima dhcpd[5361]: Sending on LPF/enp0s3/08:00:27:c1:d
Apr 26 11:01:29 Buddhima dhcpd[5361]: Sending on Socket/fallback/fallback
Apr 26 11:01:29 Buddhima dhcpd[5361]: Server starting service.
lines 1-21/21 (END)...skipping...
isc-dhcp-server.service - ISC DHCP IPv4 server
Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled; preset: enabled)
Active: active (running) since Sat 2025-04-26 11:01:29 +0530; 2min 41s ago
Docs: man:dhcpd(8)
Main PID: 5361 (dhcpd)
Tasks: 1 (limit: 4609)
Memory: 3.7M (peak: 4.2M)
CPU: 15ms
CGroup: /system.slice/isc-dhcp-server.service
5361 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf enp0s3
Apr 26 11:01:29 Buddhima sh[5361]: PID file: /run/dhcp-server/dhcpd.pid
Apr 26 11:01:29 Buddhima sh[5361]: Wrote 0 leases to leases file.
Apr 26 11:01:29 Buddhima dhcpd[5361]: Wrote 0 leases to leases file.
Apr 26 11:01:29 Buddhima dhcpd[5361]: Listening on LPF/enp0s3/08:00:27:c1:db:05/192.168.56.0/24
Apr 26 11:01:29 Buddhima sh[5361]: Listening on LPF/enp0s3/08:00:27:c1:db:05/192.168.56.0/24
Apr 26 11:01:29 Buddhima sh[5361]: Sending on LPF/enp0s3/08:00:27:c1:db:05/192.168.56.0/24
Apr 26 11:01:29 Buddhima sh[5361]: Sending on Socket/fallback/fallback-net
Apr 26 11:01:29 Buddhima dhcpd[5361]: Sending on LPF/enp0s3/08:00:27:c1:db:05/192.168.56.0/24
Apr 26 11:01:29 Buddhima dhcpd[5361]: Sending on Socket/fallback/fallback-net
Apr 26 11:01:29 Buddhima dhcpd[5361]: Server starting service.

Ping <ubuntu ip address> - can check successfully data packets are transferring properly.





Ping <kali ip address> - can check successfully data packets are transferring.

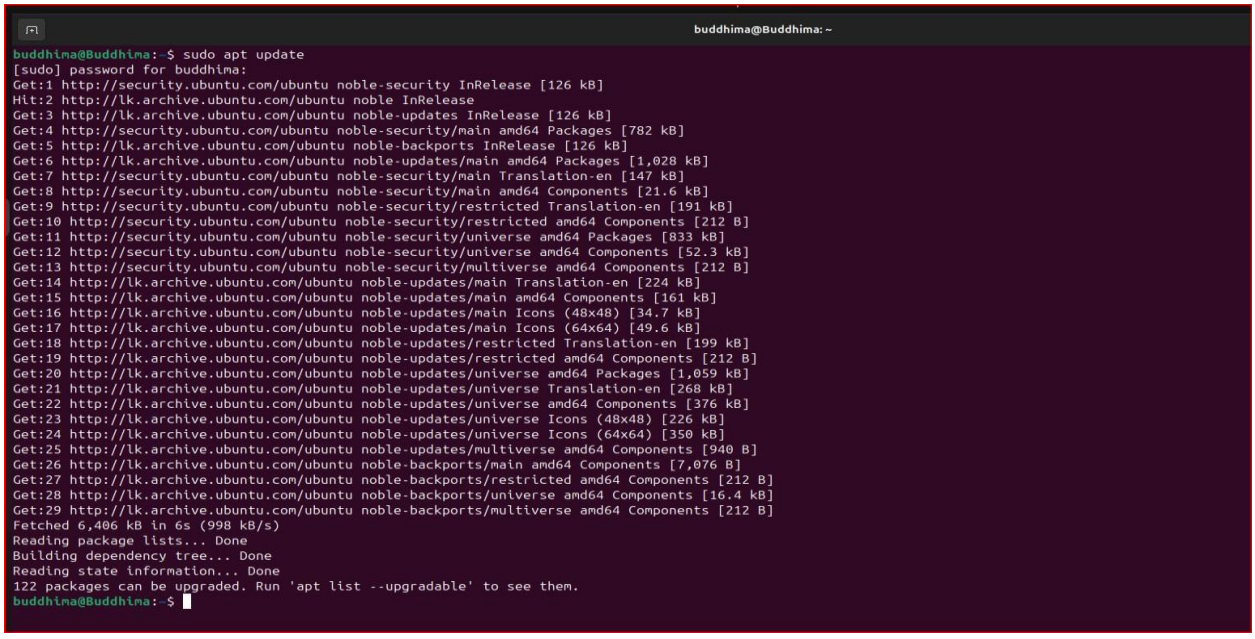


2.2. DNS (Domain Name System) Network Service:

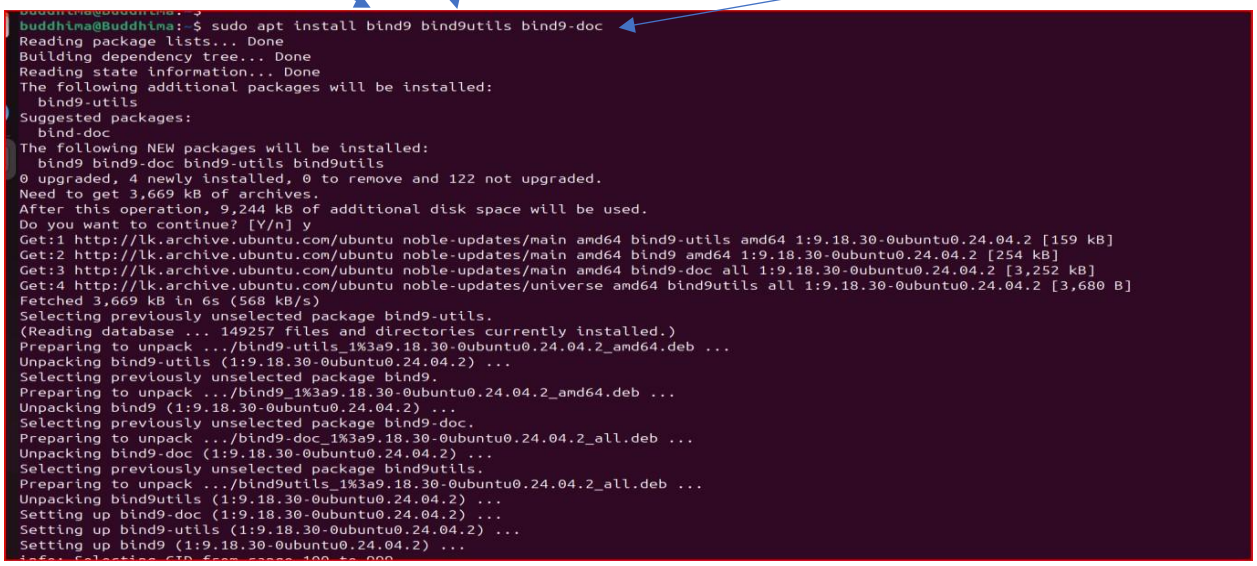
DNS can translate human readable domain names into IP addresses. Without DNS it will be very difficult to identify every website.

Step 01: Install DNS server (bind9)

- **Sudo apt update**
- **sudo apt install bind9 bind9utils bind9-doc -** manage the DNS server

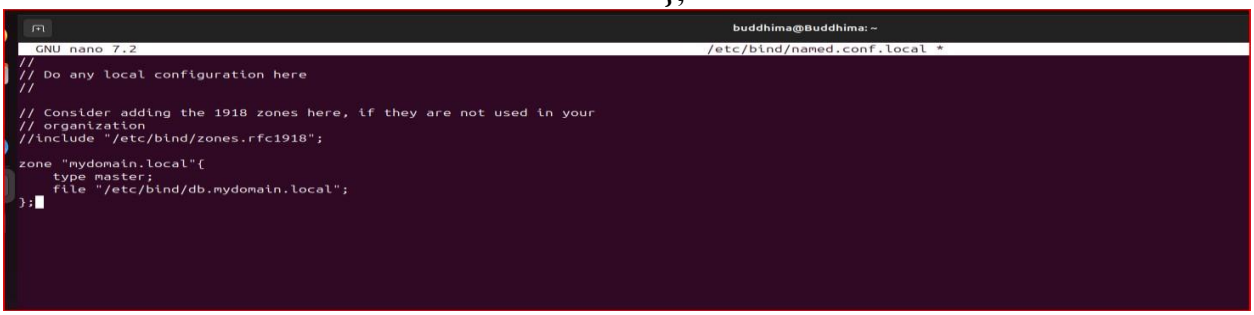


bind9- DNS server software. **Bind9utils**- help to manage and troubleshoot. **bind9-doc**- documentation files for bind9.



Step 03: Configure the DNS zone files .

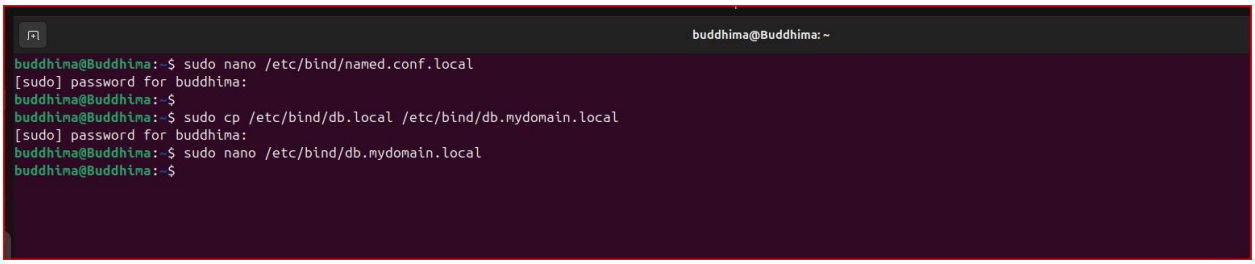
- **sudo nano /etc/bind/named.conf.local**- using nano editor view the DNS config file,
- Add this part into DNS zone file, **zone "mydomain.local" { type master;**
  file "/etc/bind/db.mydomain.local";
  };



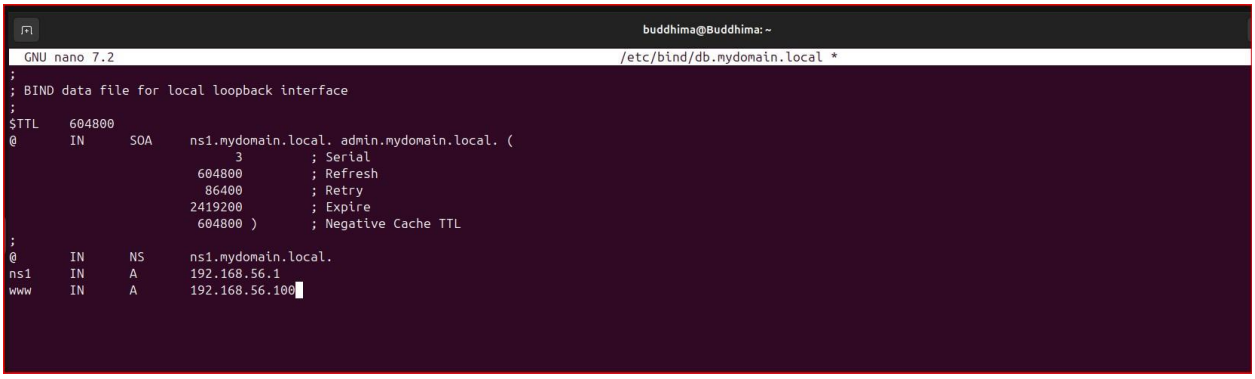
- zone “mydomain.local”- defines new DNS zone for the domain.
- Type master: define DNS is the primary server for this zone.
- file "/etc/bind/db.mydomain.local": points the DNSS zone file that stores the actual DN records..

Step 04: Create DNS zone file.

- **sudo cp /etc/bind/db.local /etc/bind/db.mydomain.local**- copy the default DNS zone file to db.mydomain.local file.

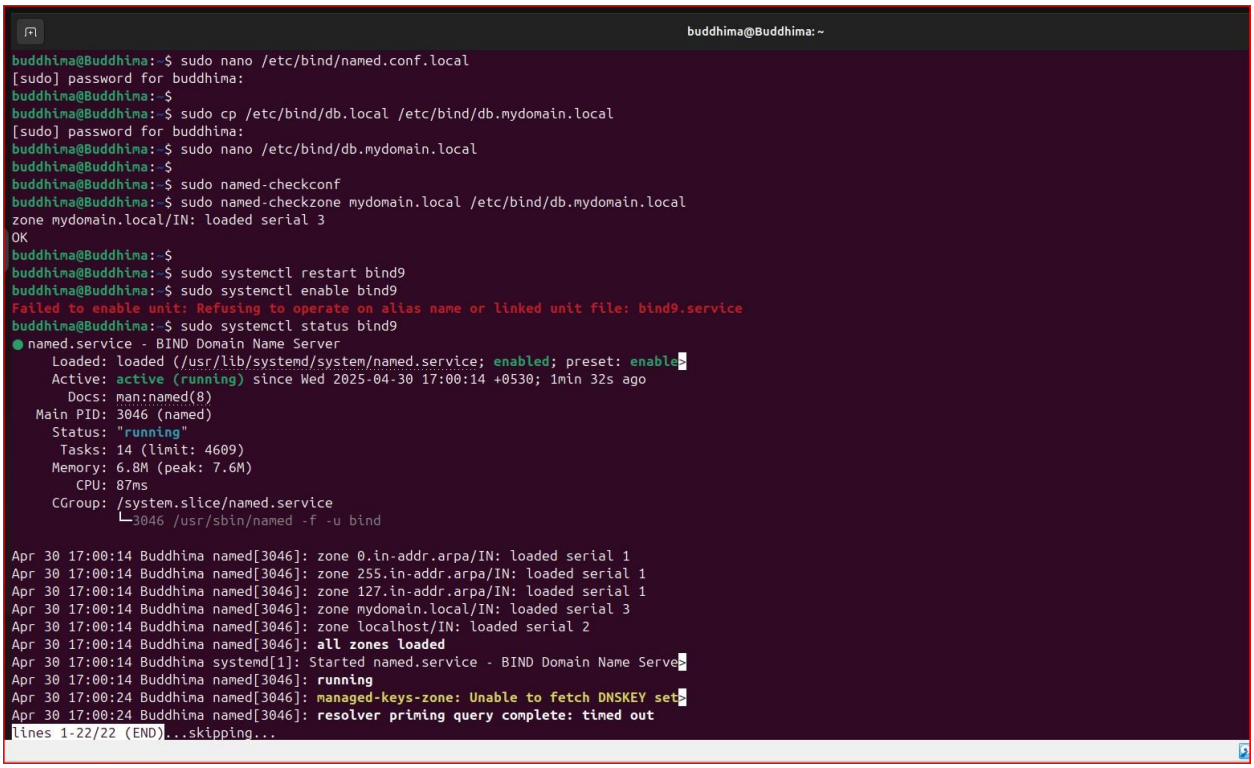


- **sudo nano /etc/bind/db.mydomain.local-** This command explain how DNS server resolve names in mydomain.local.



- SOA- Identify the primary DNS server.
- NS- Define the domain name server.
- A- Map IP address.

Step 05: Check the errors restart and enable the bind9.



- **Sudo named-checkconf**- check the syntax of bind9 DNS server and make sure there is no mistakes before the server restarts.
- **sudo systemctl restart bind9-** restart it applying any changes that are made in DNS zone file.

- **sudo systemctl enable bind9**- enable bind9 to start automatically.

Step 07: Resolve the hostname

- **nslookup db.mydomain.local-** used to resolve the hostname db.mydomain.local into an IP address using DNS

- If it properly configured zone file, bind9 server is running without errors like following output**.**

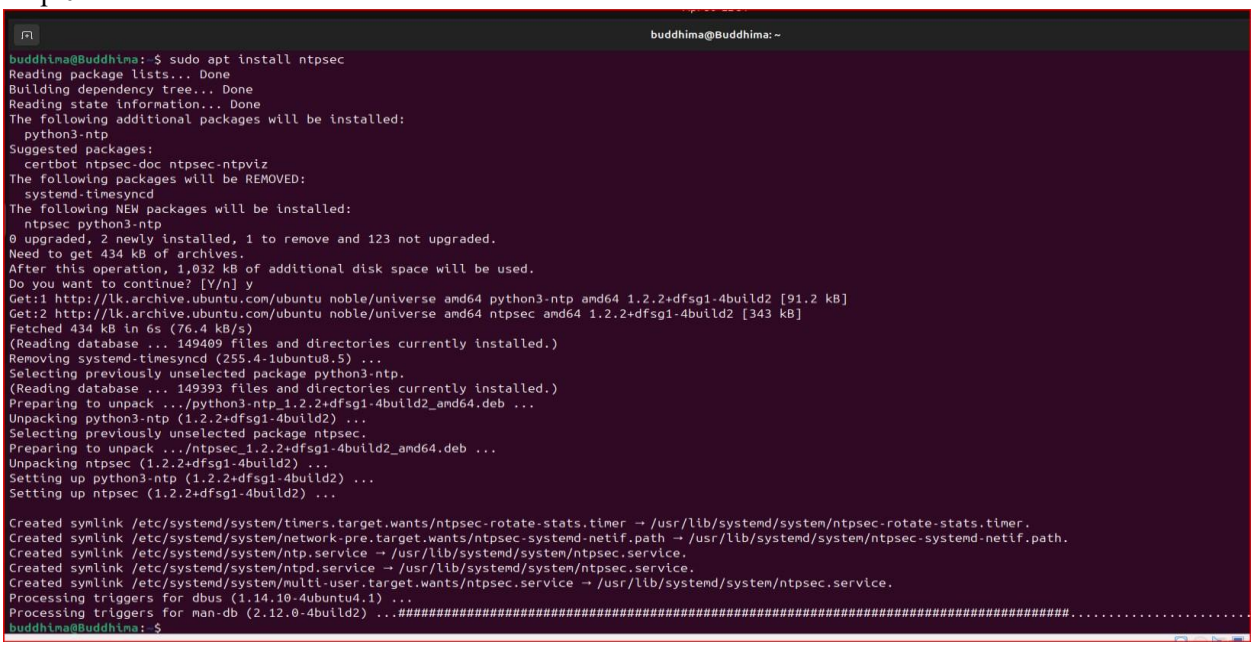
Server: 192.168.56.10
Address: 192.168.56.10#53

Name: db.mydomain.local
Address: 192.168.56.10

2.3. NTP (Network Time Protocol) Network Service:

Network Time Protocol (NTP) can synchronize the system clock of computer devices. It makes all devices set the same and correct constant time.

Step 01: Install the NTP server.



- **sudo apt install ntp-** download the NTP package that can automatically sync computer system with internet time severs.

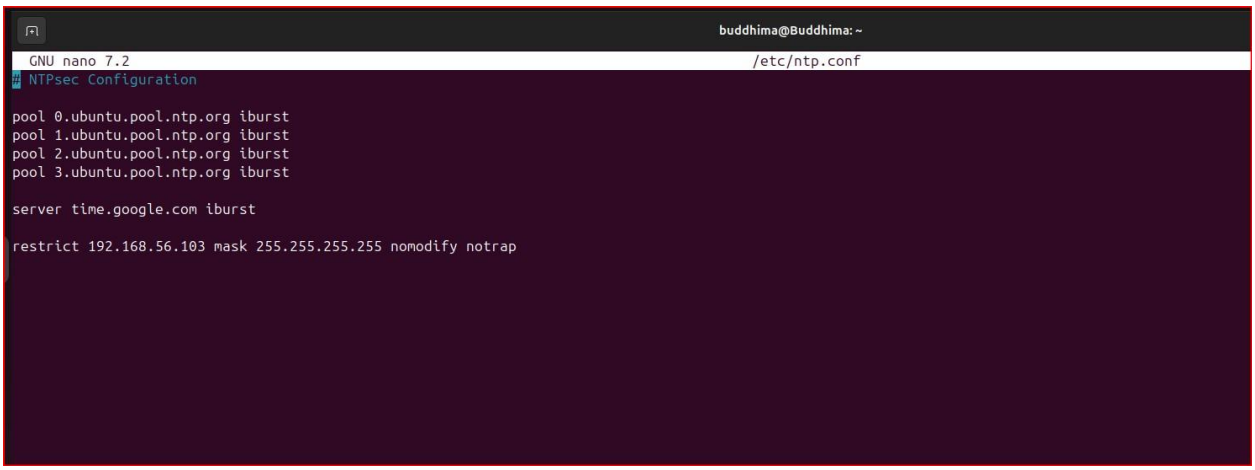
Step 02: Edit the NAT configuration file.

- **sudo nano /etc/ntp.conf-** using nano editor, view the configuration file.

- Set NTP servers,
  Server- add each line to, NTP from ubuntu time server pool.
  Iburst- increase the time speed of system sending a burst of packets.

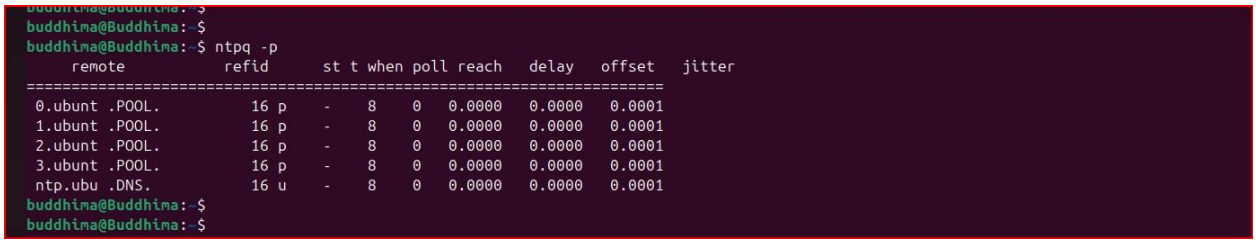
server 0.ubuntu.pool.ntp.org iburst server
1.ubuntu.pool.ntp.org iburst server
0.ubuntu.pool.ntp.org iburst server
1.ubuntu.pool.ntp.org iburst

- ***server time.google.com iburst-*** *Google public NTP used to set sync the system time.*



Step 03: Check the status of NTP server.

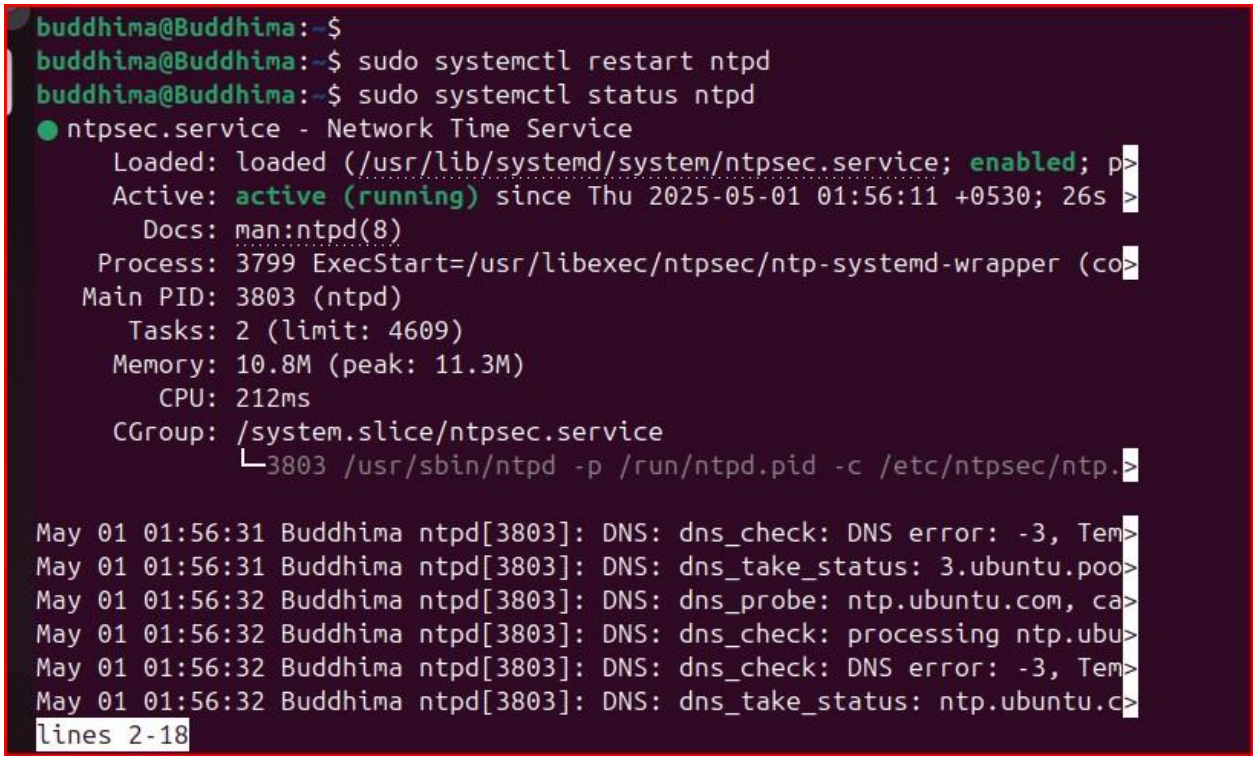
- **ntpq -p-** shows a list of sync time with their status**.**



- remote- Hostname or IP of NTP server.
- refid- Sync NTP server.
- st- shows how close the server is to the actual time source
- t- type of connection
- when- seconds since last poll
- poll-polling interval in seconds
- reach- shows success of recent contact attempts
- delay-Round-trip-time
- offset-time difference
- jitter-variability in delay

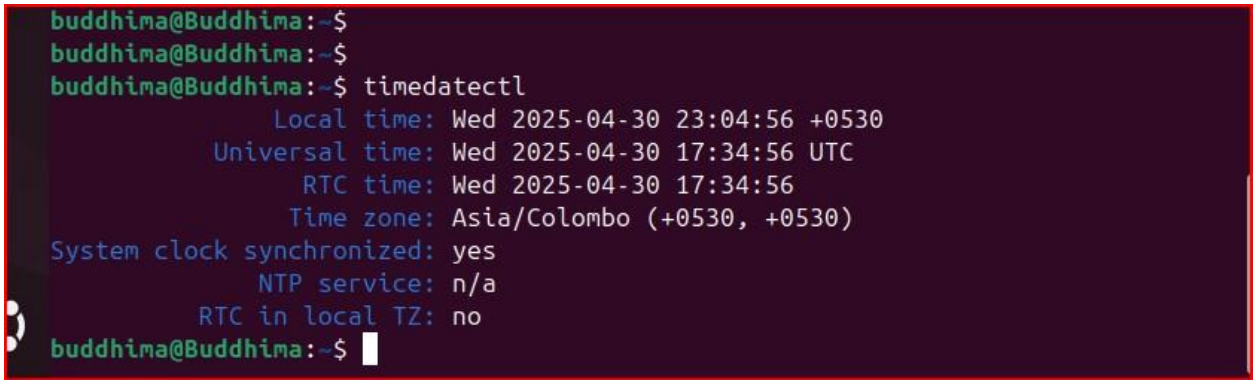
Step 04: Restart and check the status of NTP.

- **sudo systemctl restart ntpd-** restart the DNS server.
- **sudo systemctl status ntpd**- check the DNS server status is active or failed. If it is active NTP server is running correctly.



Step 05: check the current time.

- **timedatectl**- This command will show the current time status, date, whether its synchronized or not.



3.Security and Other Servers.

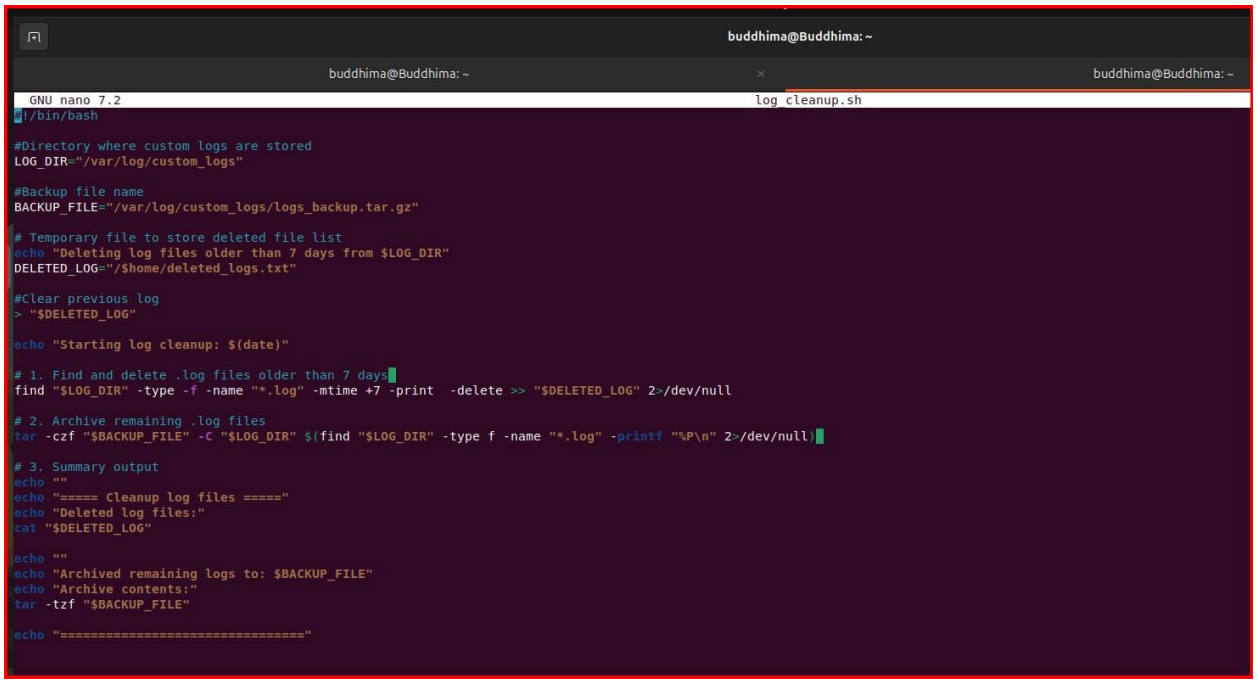
3.1. Shell Scripting:

In this section learn the basics of shell scripting syntax creating a shell script file.

i. Step 01: Create a script file (log_cleanup.sh) to write the bash script.

- **nano log_cleanup.sh** – create using nano text editor.

- Write the script as follows:



- This Bash script automatically deletes old log files, archives current logs, and prints a summary to manage disk space efficiently, and is scheduled to run weekly using cron.

LOG_DIR="/var/log/custom_logs" – Set LOG_DIR variable to store the custom logs files.

BACKUP_FILE="/var/log/custom_logs/logs_backup.tar.gz" – Set BACKUP_FILE variable to save compressed archived file (.tar.gz) in the same custom log directory.

DELETED_LOG="/$home/deleted_logs.txt" – Set TEMP_DELETED variable to store a list of deleted files.

- **find** "$LOG_DIR" **-type** f -name "*.log" -**mtime** +7 -**print** -delete >> **"$TEMP_DELETED"** **2>/dev/null**

find- search for files and directories.

"$LOG_DIR – find inside the directory stored in this variable.(path is /var/log/custom_logs) **type**

f- filter that used to find the file type, search to regular files only.

name "*.log" – match the extension that have .log -**mtime**

+7- find files that were created 7 days ago.

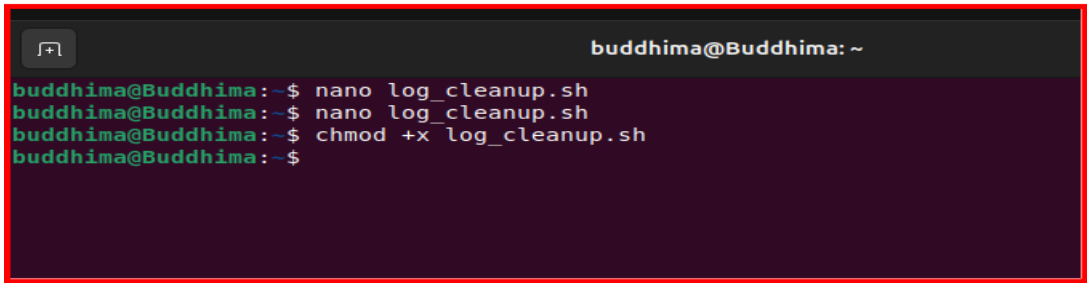
-**print –** print the full path of matches files**.**

-**delete**- delete the matched file after determining more than 7 days old files.

>> "$TEMP_DELETED- display in summary what was deleted previously**.**

2>/dev/null- hide or ignore the error messages**.**

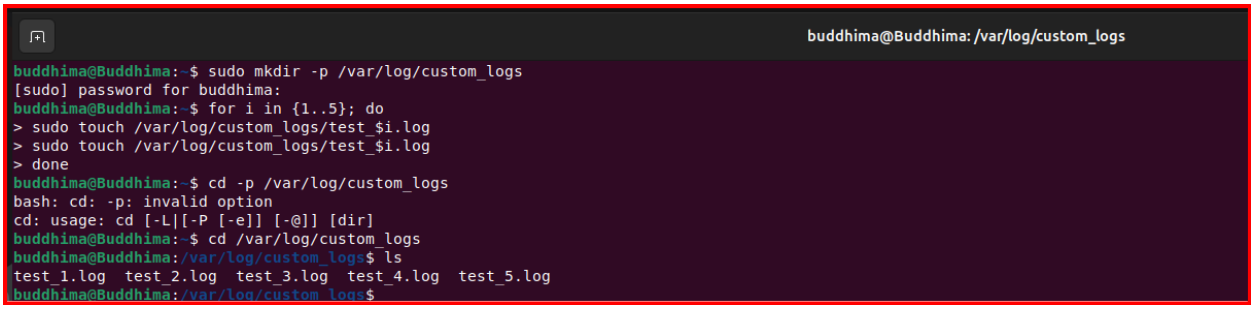
- **chmod +x log_cleanup.sh –** make the log_cleanup.sh file executable.



Step 02: Create several dummy files in custom_logs directory in **"/var/log/custom_logs**" .

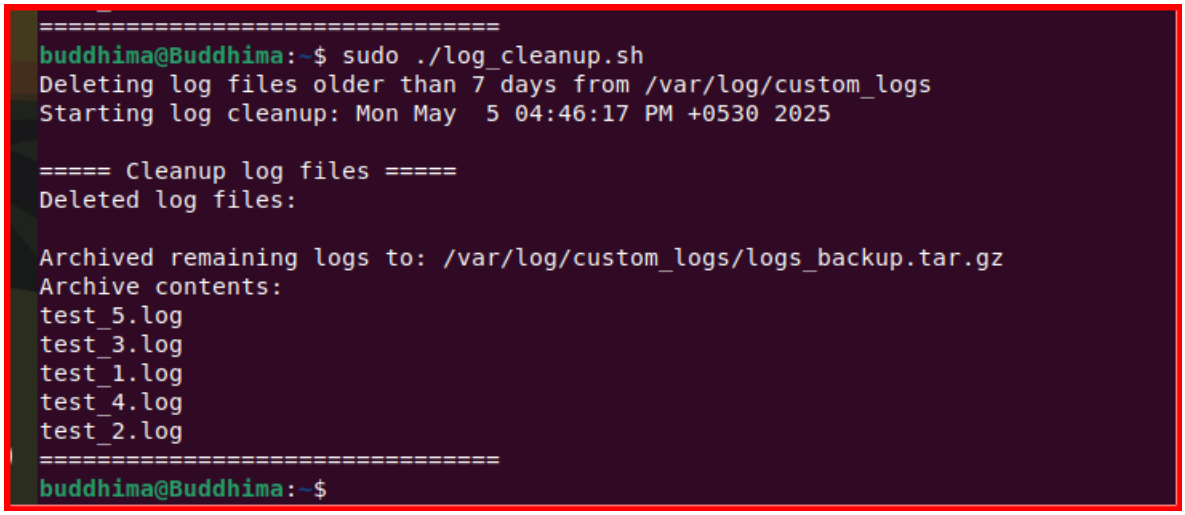
- **sudo mkdir -p /var/log/custom_logs –** create a directory using mkdir as **custom_logs**

- **for i in {1..5}; do sudo touch /var/log/custom_logs/test_$i.log**
  - create 5 dummy log iles using for loop done



- **cd -p /var/log/custom_logs - move into custom_logs directory.**
- **Ls – using ls , to check dummy files are created properly.**

Step 03: Test the log_cleanup.sh file manually.



- **sudo ./log_cleanup.sh - using this command test it manually.**

Step 04: Schedule the script with corn job.

- sudo crontab –e –> open the corntab editor.

```
buddhima@Buddhima:~$
buddhima@Buddhima:~$ sudo crontab -e
[sudo] password for buddhima:
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /bin/ed

Choose 1-3 [1]: 1
crontab: installing new crontab
```

- **0 0 * * 0 /path/to/log_cleanup.sh** – add this line as mentioned below.

```
buddhima@Buddhima: ~
buddhima@Buddhima: ~                                    buddhima@Buddhima:
  GNU nano 7.2                          /tmp/crontab.E4TCR4/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

0 0 * * 0 /var/log/log_cleanup.sh

# Minute 0, Hour 0, Anby day, any month, Day 0 (sunday) is what is implemented
```

- **0 0 * * 0** - This part schedule corn runs at 12 A.M every Sunday.
- **/var/log/log_cleanup.sh** – This part replace with the path of log_cleanup.sh

```
buddhima@Buddhima: /var/log/custom_logs
buddhima@Buddhima:~/Desktop$ cd ..
buddhima@Buddhima:~$ cd /var/log/custom_logs
buddhima@Buddhima:/var/log/custom_logs$ ls -l
total 4
-rw-r--r-- 1 root root 150 May  5 16:46 logs_backup.tar.gz
-rw-r--r-- 1 root root   0 May  5 15:03 test_1.log
-rw-r--r-- 1 root root   0 May  5 15:03 test_2.log
-rw-r--r-- 1 root root   0 May  5 15:03 test_3.log
-rw-r--r-- 1 root root   0 May  5 15:03 test_4.log
-rw-r--r-- 1 root root   0 May  5 15:03 test_5.log
buddhima@Buddhima:/var/log/custom_logs$
```

3.2. SSH (Secure Shell):

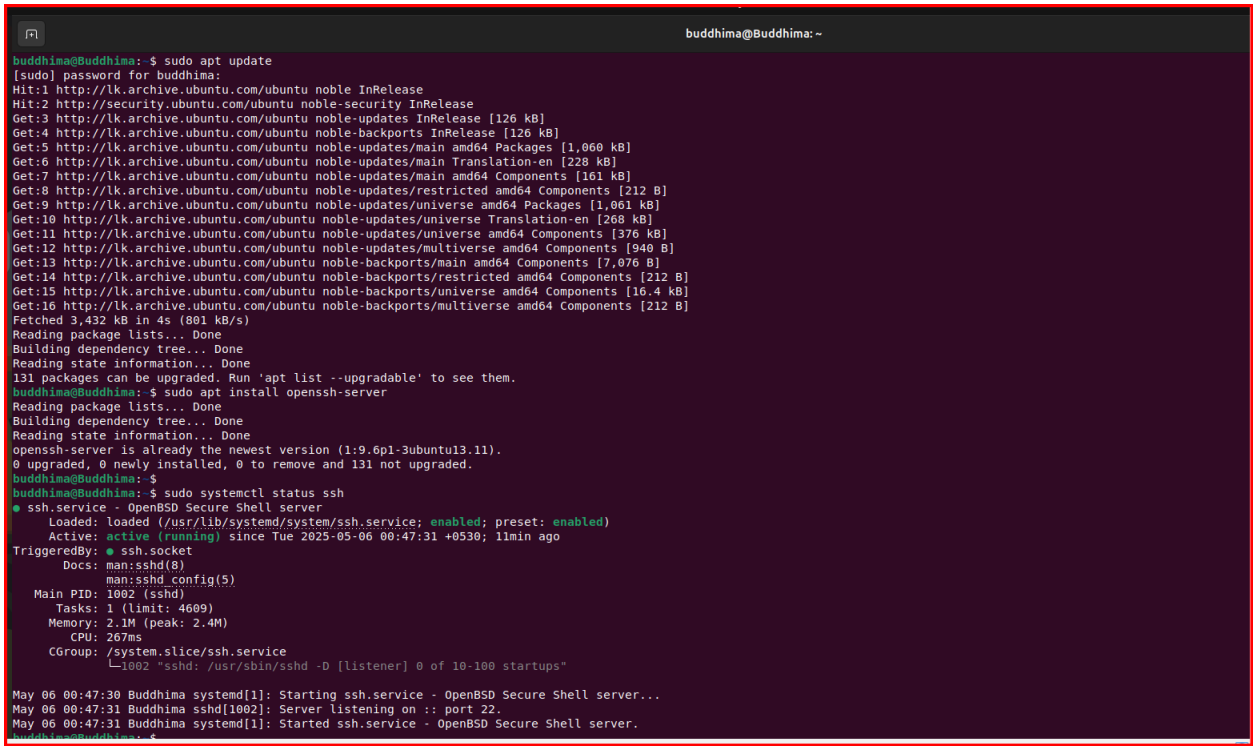
To configure an SSH server on a Ubuntu for secure remote login using tools like openssh and connect to client machine (kali) remotely.

Step 01: Install the openssh server.

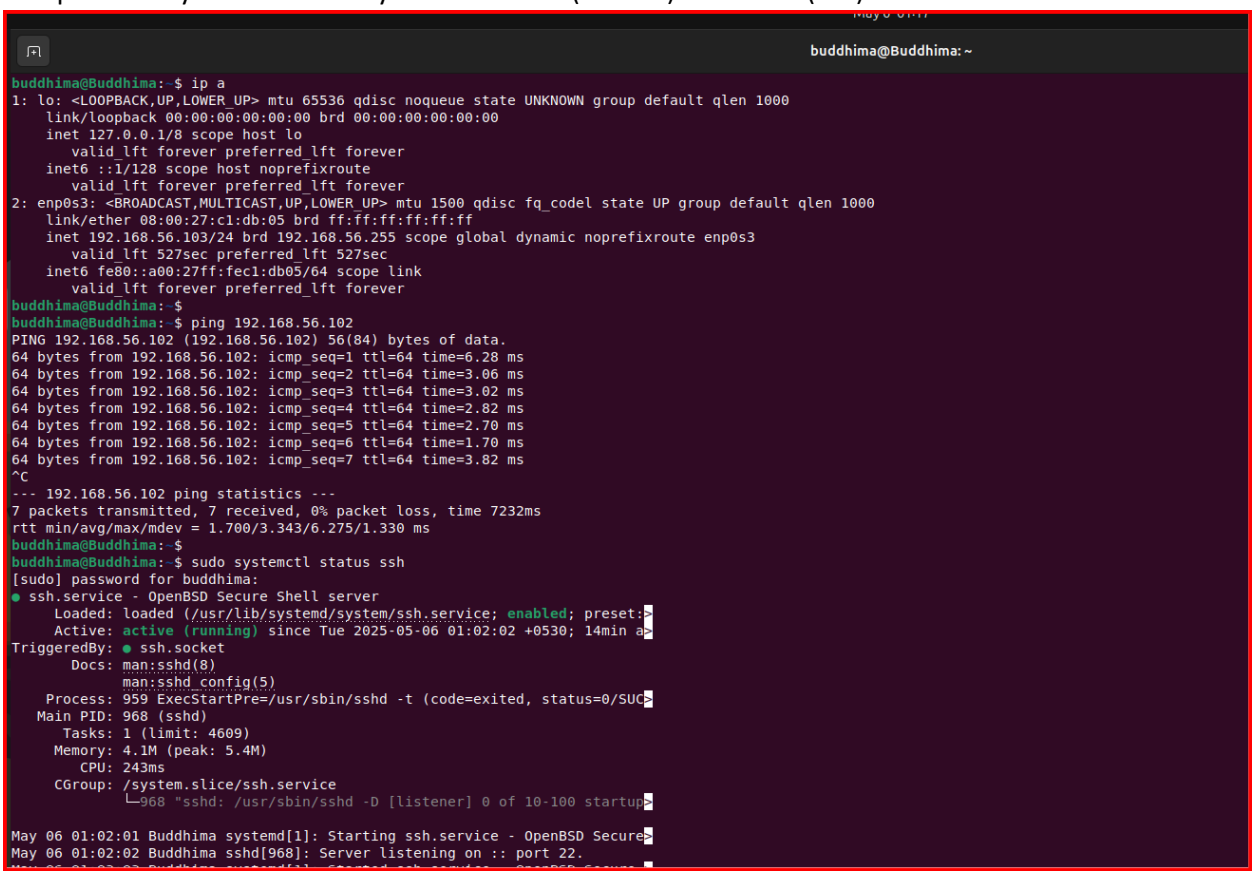
- **sudo apt update**
- **sudo apt install openssh-server -y**

Step 02: Check the status of SSH service.

- **sudo systemctl status ssh – check the status of SSH service is active or fail.**

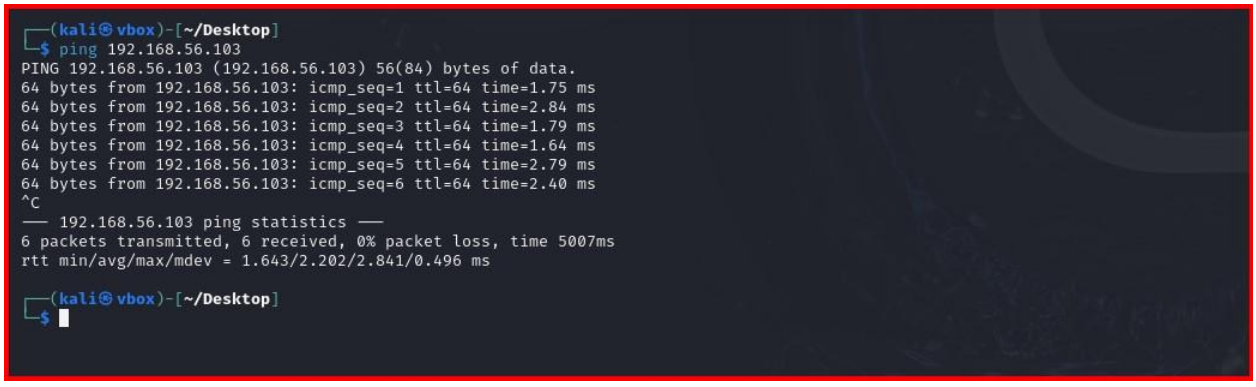


Step 03: Verify the connectivity of both server (ubuntu) and client (kali).



- ip a – find the IP address. (192.168.56.103)

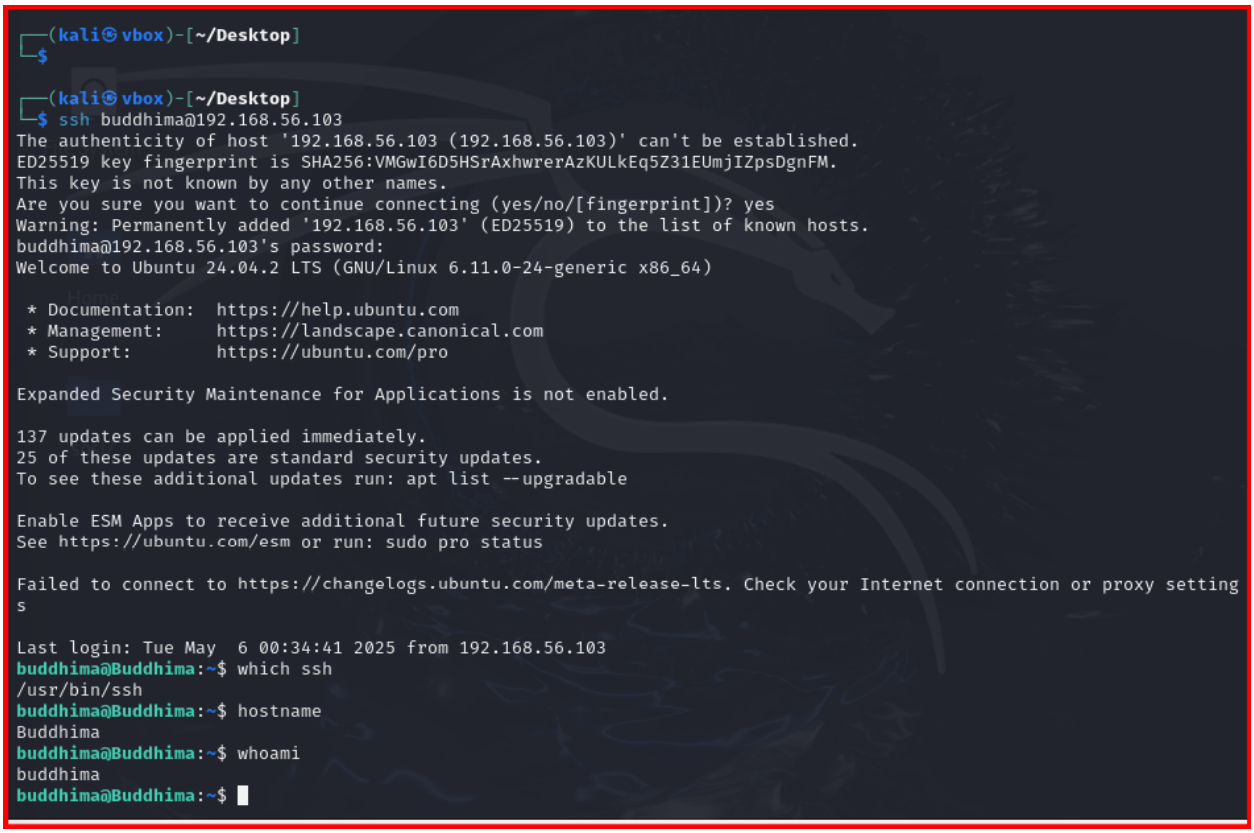
- ping 192.168.56.102 – If successfully transfer and response confirm bidirectional commiunication.



- ping 192.168.56.103 – If successful response confirms client can access the server.

Step 04:Check the remote connection using SSH client.

- ssh buddhima@192.168.56.103 – Initiate a SSH connection to the machine 192.168.56.103



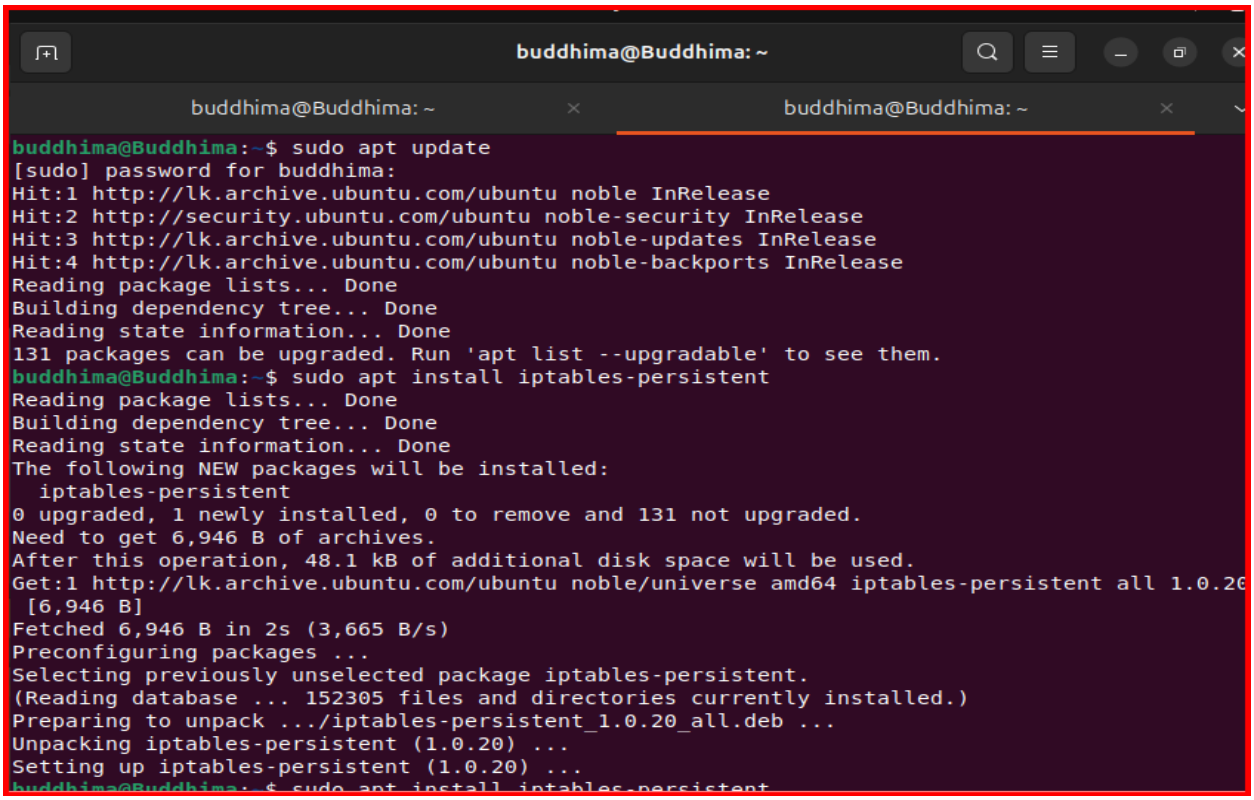
- which ssh- shows the full path of the ssh
- hostname- name of the machine that logged into (ubuntu hostname)
- whoami – user name that currently login , and it confirms the identity of remote machine(ubuntu).

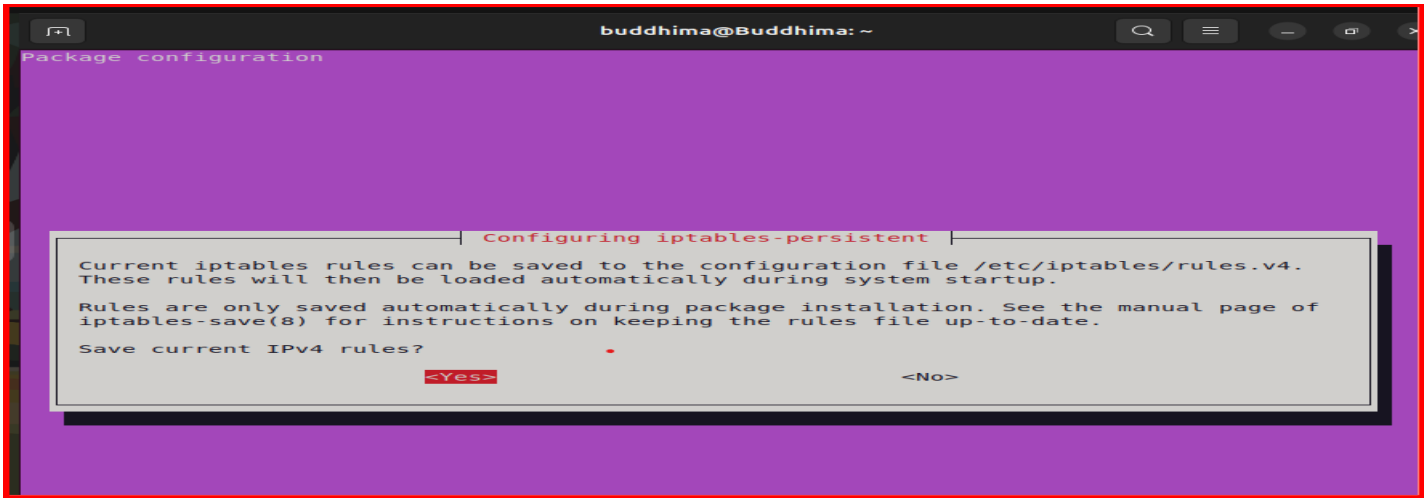
3.3. Iptables and ACLs:

Using firewalls and iptables rules, managing the network traffic. In this section block social media websites like facebook, Instagram and twitter. And block HTTP (port 80) traffic and allow only (port 443) HTTPS and ensure the security of network by defining iptables rules to implement these security services.

Step 01: Install and update

- Sudo apt update
- Sudo apt install iptables-persistent - save iptables rules





Step 02: Create iptables rules to block HTTP (port 80) and allow HTTPS (443) .

- **sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT**
- **sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT**

```
buddhima@Buddhima:~$
buddhima@Buddhima:~$
buddhima@Buddhima:~$ sudo iptables -A OUTPUT -p tcp --dport 80 -j REJECT
buddhima@Buddhima:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
buddhima@Buddhima:~$
```

Step 03: Create iptables rules to block (facebook, Instagram, twitter) specific IP addresses.

- **sudo iptables -A OUTPUT -p tcp -d 157.240.22.35 --dport 443 -j REJECT** # Facebook
- **sudo iptables -A OUTPUT -p tcp -d 31.13.71.36 --dport 443 -j REJECT** # Instagram
- **sudo iptables -A OUTPUT -p tcp -d 104.244.42.129 --dport 443 -j REJECT** # Twitter

```
buddhima@Buddhima:~$
buddhima@Buddhima:~$ sudo iptables -A OUTPUT -d 157.240.0.0/16 -j DROP
buddhima@Buddhima:~$ sudo iptables -A OUTPUT -d 185.60.216.0/22 -j DROP
buddhima@Buddhima:~$ sudo iptables -A OUTPUT -d 104.244.42.0/24 -j DROP
buddhima@Buddhima:~$
```

- **sudo iptables-save > /etc/iptables/rules.v4 – Ensure iptables rules are persist after using this command.**

- **Sudo iptables-save | sudo tee /etc/iptables/rules.v4 >/dev/null –** current iptables rules pipe into tee /etc/iptables/rules.v4

Steps 04: List iptables rules.

- sudo iptables -L -v -n
  -L – List all current rules.
  -v – Shows detailed output.
  -n – Prevent DNS lookups for IP addresses and ports.

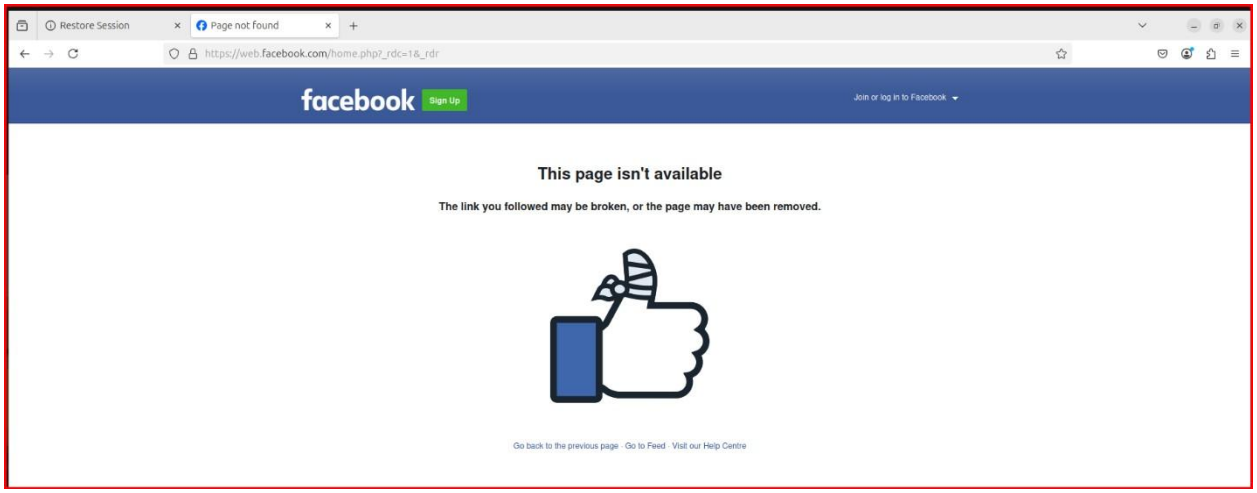
```
buddhima@Buddhima: ~$
buddhima@Buddhima: ~$ sudo iptables-save | sudo tee /etc/iptables/rules.v4 > /dev/null
buddhima@Buddhima: ~$
buddhima@Buddhima: ~$ sudo iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

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

Chain OUTPUT (policy ACCEPT 5492 packets, 422K bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       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
    0     0 DROP       0    --  *      *       0.0.0.0/0            157.240.0.0/16
    0     0 DROP       0    --  *      *       0.0.0.0/0            185.60.216.0/22
    0     0 DROP       0    --  *      *       0.0.0.0/0            104.244.42.0/24
buddhima@Buddhima: ~$
```

Step 05: Check the IP address are blocked successfully in the server.

- Surfing the browser and try to login www.facebook.com



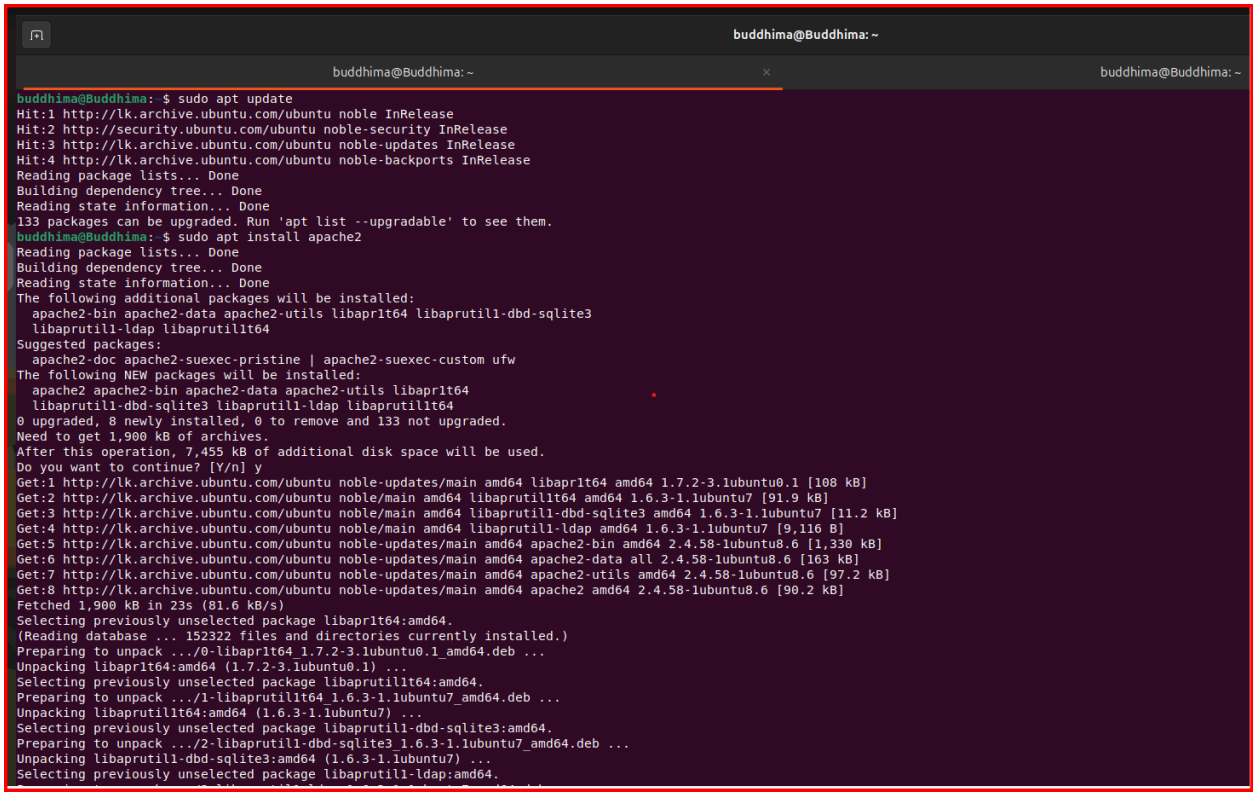
- Can check it in terminal also.

3.4. WEB Server (Apache):

WEB Server (Apache) is used to deliver web content to clients through the internet. It can handle HTTP/HTTPS requests from clients and respond to it. In this section create a simple web page and configure the web server to serve it in VM using Apache software.

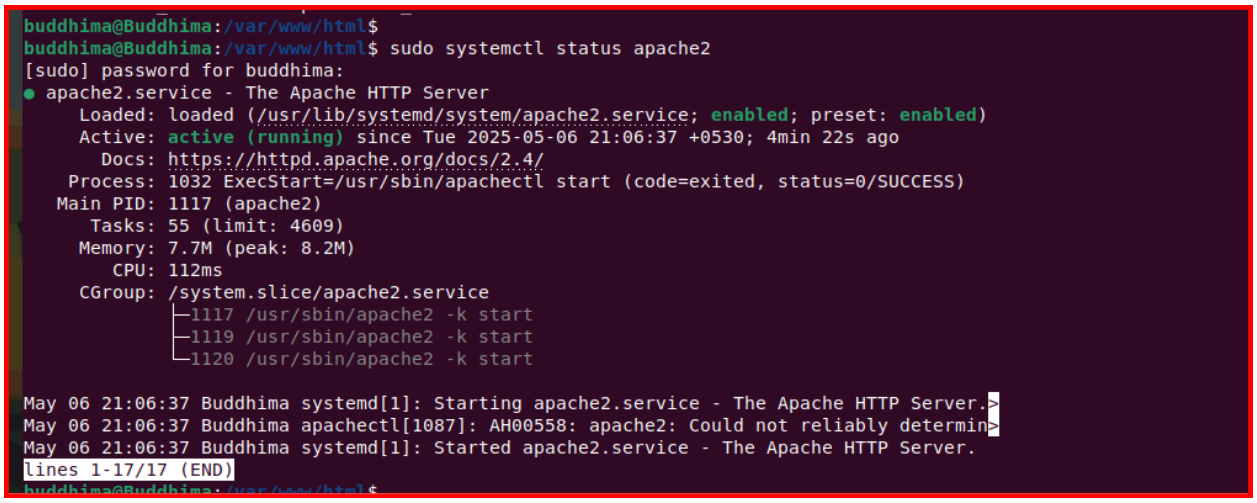
Step 01: Install apache2 and update.

- **sudo apt update**
- **sudo apt install apache2** – Install apache2 sever.



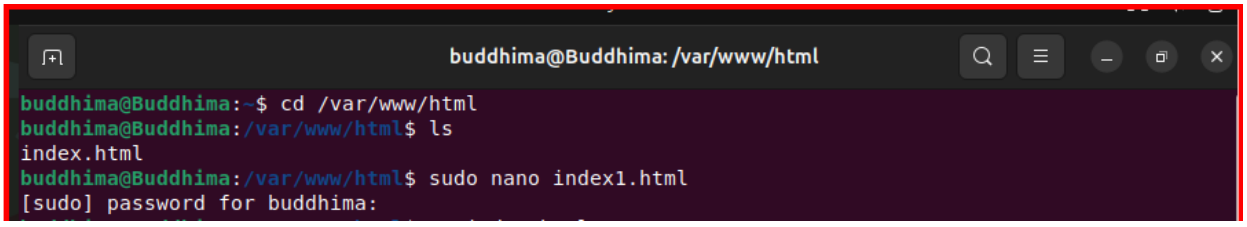
Step 02: Check the Apache server status

- **sudo systemctl status apache2 – check the Apache server status is successfully activated or fail.**



Step 03: Create a simple web page.

- **cd /var/www/html/ -** navigate to root directory of Apache document**.**
- **ls –** using ls can list the directory and find index.html file.
- **sudo nano index.html – create index1.html using nano text editor.**



- Add the html coding part to create a simple web page.

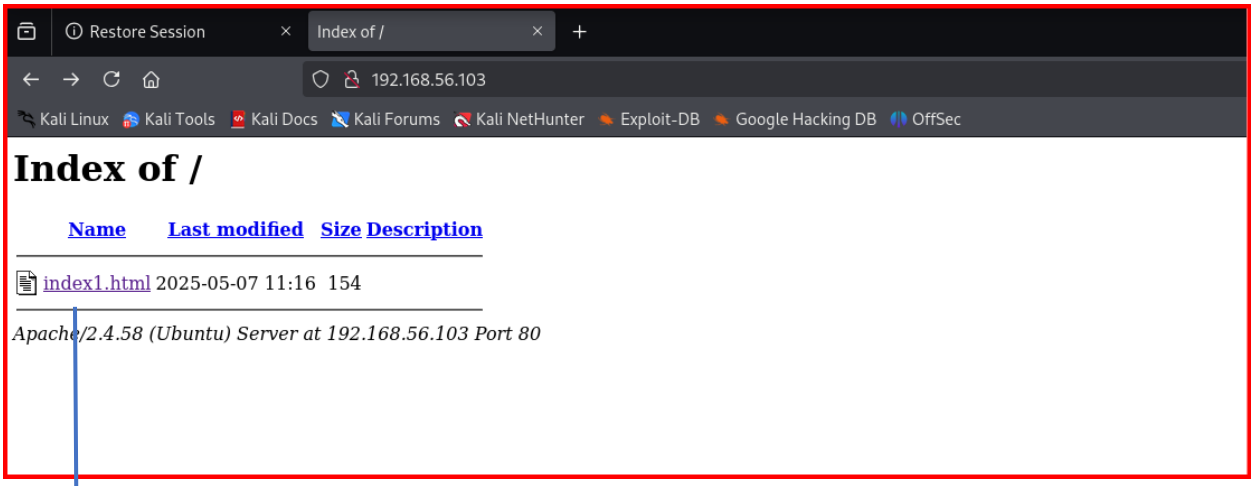
Press CTRL + O to save.
Press Enter to confirm.
Press CTRL + X to exit.

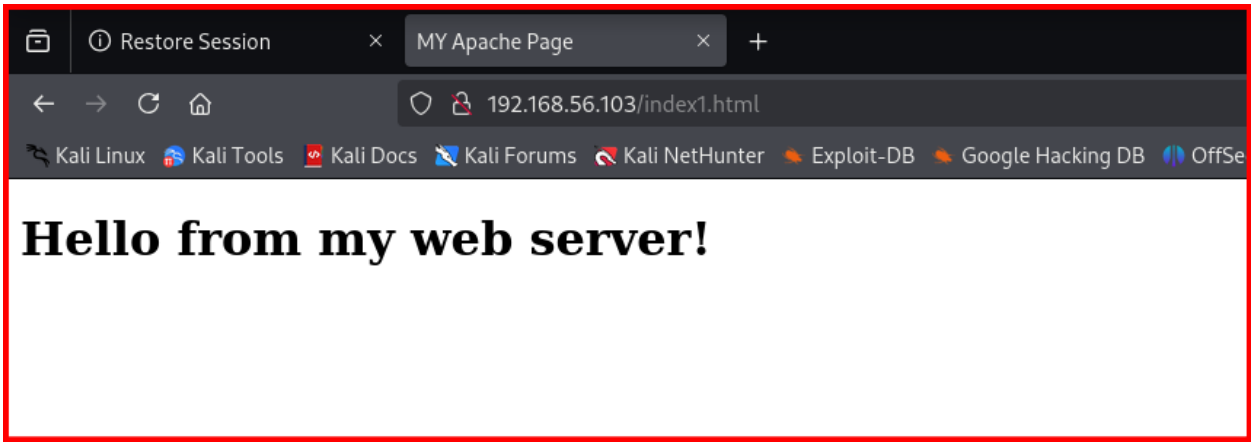
Step 04: Check the IP address of server (ubuntu IP address)

- ip a- select the IP address under enp0s3

Step 05:Test in the browser.

- Using http://192.168.56.103 IP address to test in browser.



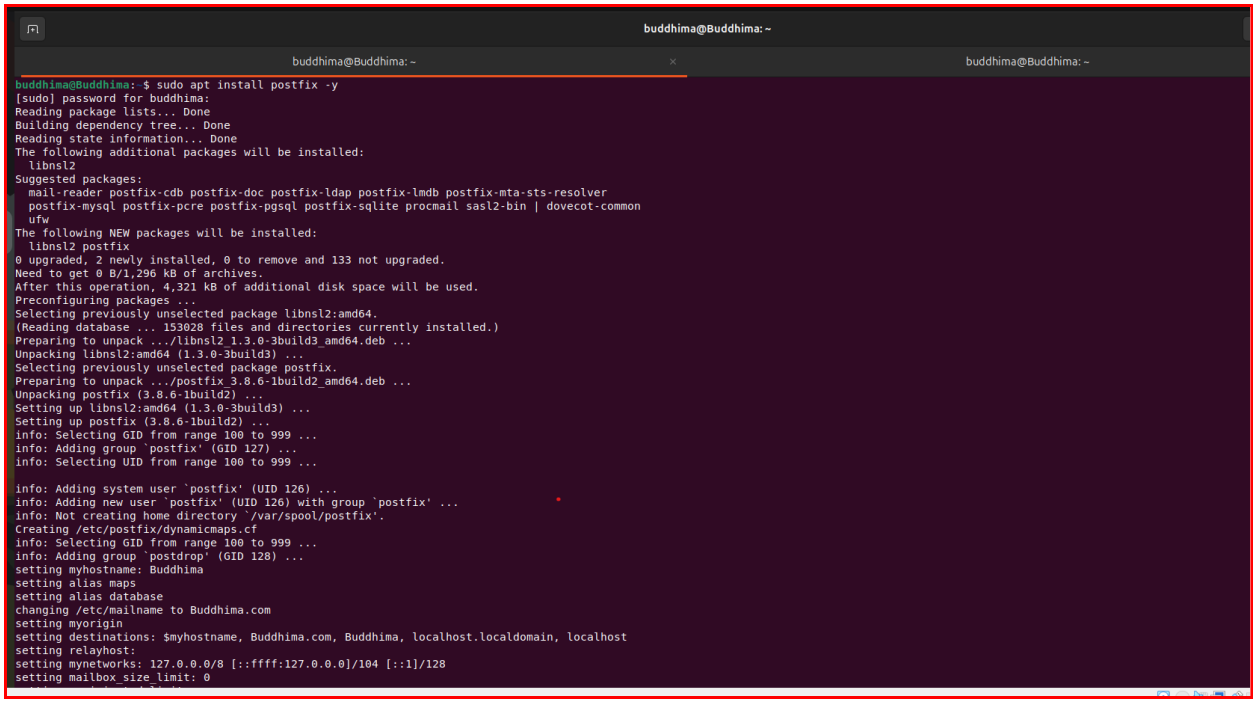


3.5. Email Server (postfix):

Email Server is a system where messages can be sent, receives, stores between users. In this section handle send and receive emails.

Step 01: Install postfix server.

- **sudo apt install postfix -y** – Install postfix with its required packages.



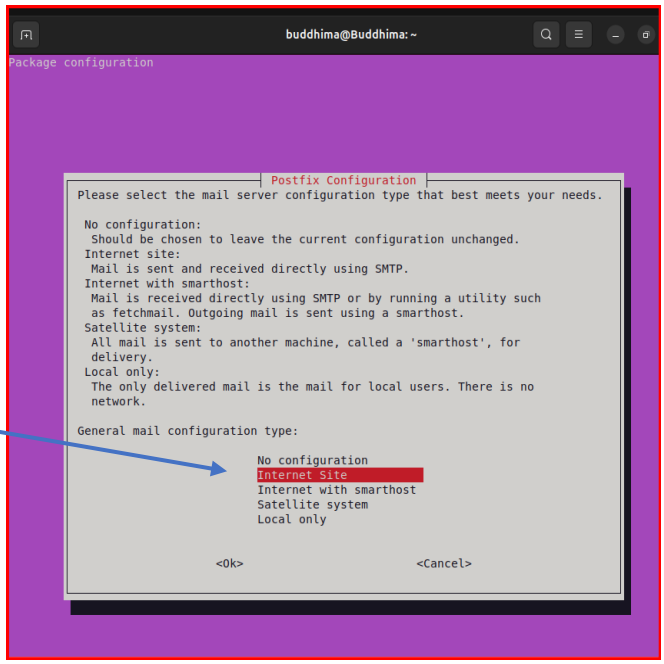
- During the installation,

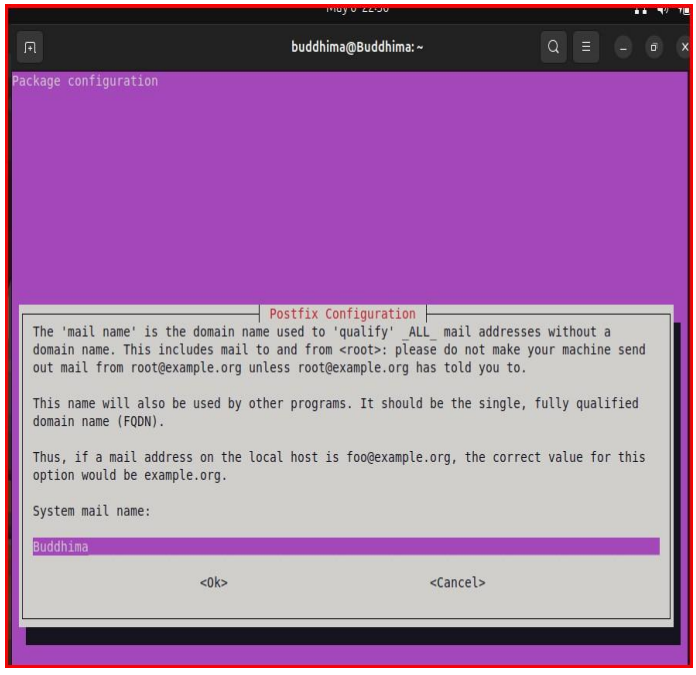
postfix configurations ask to select the **type of Mail configuration**,

- Select - Internet Site

Afterwards there is another selector asking to enter the **System mail name**,

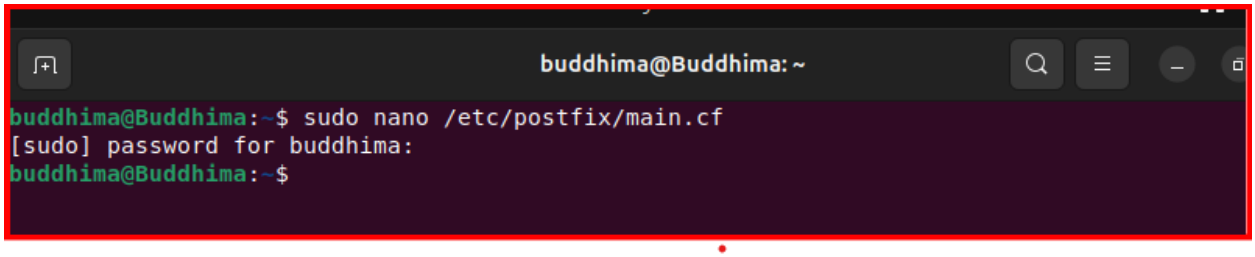
- Enter: Buddhima.com





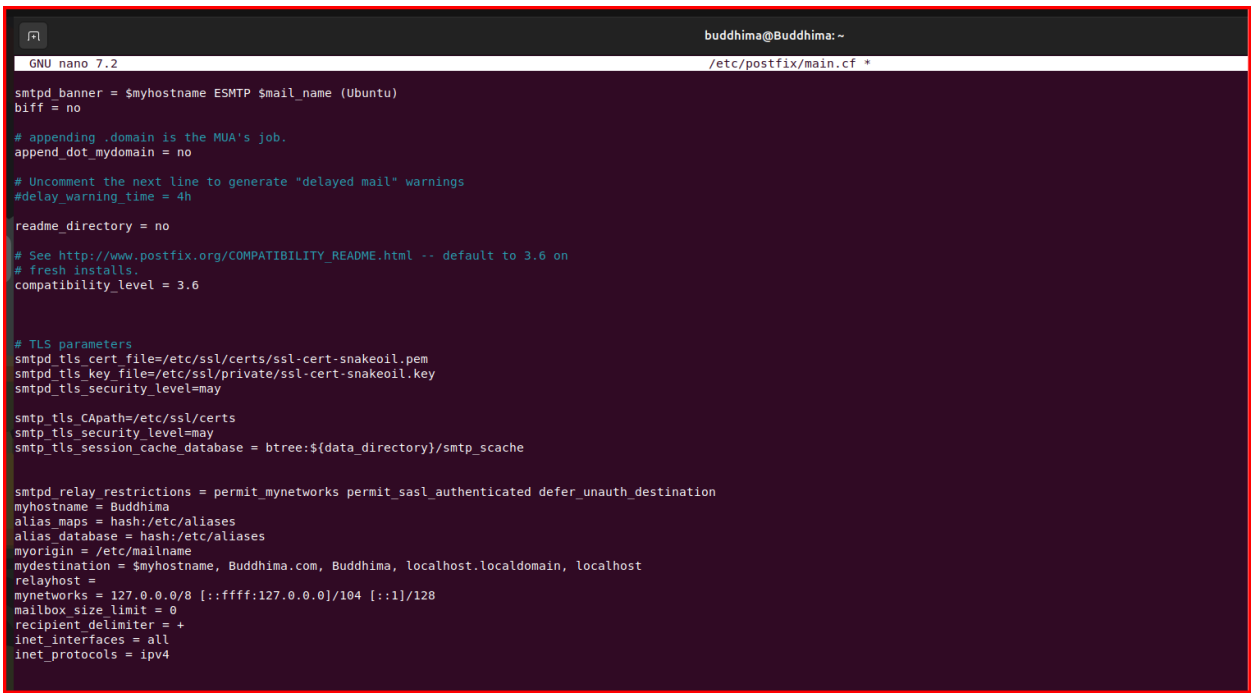
Step 02: Verify the configuration of postfix.

- **sudo nano /etc/postfix/main.cf –** open postfix configuration file using nano text editor**.**



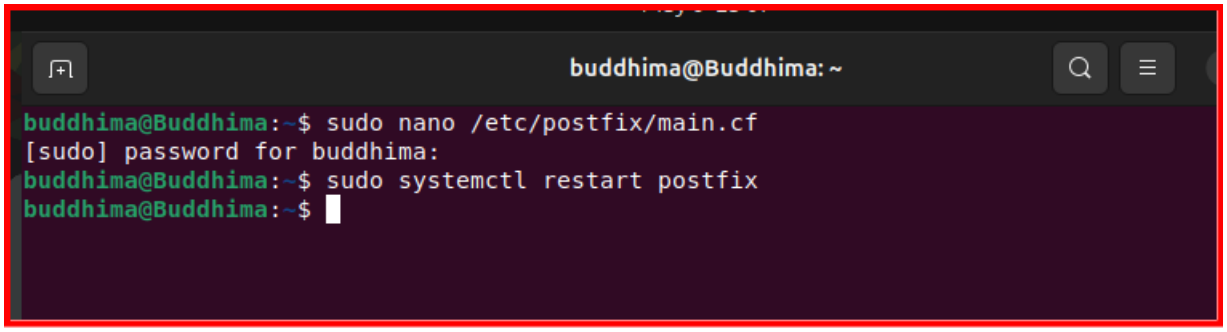
- Check following lines are modified during the configuration**.**

  myhostname = Buddhima.com myorigin
  = /etc/mailname
  mydestination = $myhostname, localhost.$mydomain, localhost inet_interfaces
  = all
  inet_protocols = ipv4

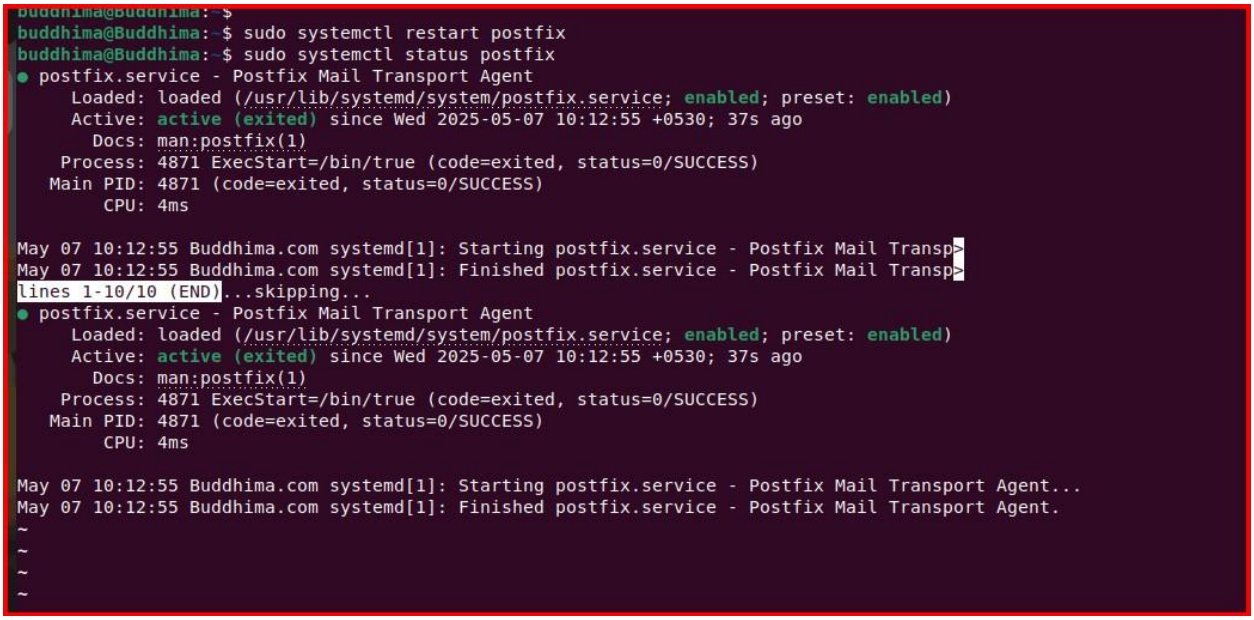


Step 03: Restart and status check of postfix.

- **sudo systemctl restart postfix**



- **sudo systemctl status postfix –** check the status of postfix whether it is activated successfully or fail.



Step 04: Install mailutils.

- **sudo apt install mailutils -y –** Installing this server allows to test email from terminal.

```
buddhima@Buddhima:~$
buddhima@Buddhima:~$ sudo apt install mailutils -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  gsasl-common guile-3.0-libs libgc1 libgsasl18 libgssglue1 libmailutils9t64 libmysqlclient21
  libntlm0 libpq5 mailutils-common mysql-common
Suggested packages:
  mailutils-mh mailutils-doc
The following NEW packages will be installed:
  gsasl-common guile-3.0-libs libgc1 libgsasl18 libgssglue1 libmailutils9t64 libmysqlclient21
  libntlm0 libpq5 mailutils mailutils-common mysql-common
0 upgraded, 12 newly installed, 0 to remove and 133 not upgraded.
Need to get 10.3 MB of archives.
After this operation, 67.5 MB of additional disk space will be used.
Get:1 http://lk.archive.ubuntu.com/ubuntu noble/main amd64 gsasl-common all 2.2.1-1willsync1build2 [5,056 B]
Get:2 http://lk.archive.ubuntu.com/ubuntu noble/main amd64 libgc1 amd64 1:8.2.6-1build1 [90.3 kB]
Get:3 http://lk.archive.ubuntu.com/ubuntu noble/universe amd64 guile-3.0-libs amd64 3.0.9-1build2 [7,630 kB]
Get:4 http://lk.archive.ubuntu.com/ubuntu noble/main amd64 libgssglue1 amd64 0.9-1build1 [20.7 kB]
Get:5 http://lk.archive.ubuntu.com/ubuntu noble/main amd64 libntlm0 amd64 1.7-1build1 [19.4 kB]
Get:6 http://lk.archive.ubuntu.com/ubuntu noble/main amd64 libgsasl18 amd64 2.2.1-1willsync1build2 [72.8 kB]
Get:7 http://lk.archive.ubuntu.com/ubuntu noble/universe amd64 mailutils-common all 1:3.17-1.1build3 [389 kB]
Get:8 http://lk.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1 [6,746 B]
Get:9 http://lk.archive.ubuntu.com/ubuntu noble-updates/main amd64 libmysqlclient21 amd64 8.0.42-0ubuntu0.24.04.1 [1,254 kB]
Get:10 http://lk.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpq5 amd64 16.8-0ubuntu0.24.04.1 [142 kB]
Get:11 http://lk.archive.ubuntu.com/ubuntu noble/universe amd64 libmailutils9t64 amd64 1:3.17-1.1build3 [514 kB]
Get:12 http://lk.archive.ubuntu.com/ubuntu noble/universe amd64 mailutils amd64 1:3.17-1.1build3 [127 kB]
Fetched 10.3 MB in 9s (1,151 kB/s)
Selecting previously unselected package gsasl-common.
(Reading database ... 153231 files and directories currently installed.)
Preparing to unpack .../00-gsasl-common_2.2.1-1willsync1build2_all.deb ...
Unpacking gsasl-common (2.2.1-1willsync1build2) ...
Selecting previously unselected package libgc1:amd64.
Preparing to unpack .../01-libgc1_1%3a8.2.6-1build1_amd64.deb ...
Unpacking libgc1:amd64 (1:8.2.6-1build1) ...
Selecting previously unselected package guile-3.0-libs:amd64.
Preparing to unpack .../02-guile-3.0-libs_3.0.9-1build2_amd64.deb ...
```

Step 05: Add a new user.

- **sudo adduser testuser**

```
buddhima@Buddhima:~$
buddhima@Buddhima:~$ sudo adduser testuser
info: Adding user `testuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `testuser' (1001) ...
info: Adding new user `testuser' (1001) with group `testuser (1001)' ...
info: Creating home directory `/home/testuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `testuser' to supplemental / extra groups `users' ...
info: Adding user `testuser' to group `users' ...
buddhima@Buddhima:~$
```

Step 06: Send a test email.

- **echo "Hello from Postfix!" | mail -s "Test Email" testuser –** send a email to testuser
- **sudo su – testuser –** switch user (switch to testuser)

Step 07: Check the testuser's mailbox.

- **mail** - using this command can read the message.
- **q** – press q to exit
- **exit** – to logout from new user account.

4.Linux GDB.

This section focuses on **analyzing the behavior of an executable program using GNU Debugger (GDB)** and **monitoring its impact on the file system**. The goal is to understand how the program executes, what files it creates or modifies, and how it generates and processes output data.

Download the Executable file from here -
https://drive.google.com/drive/folders/19didjVba_W49b6RyrX0KHlEHmiAmGrk6?usp=drive_link

Step 01: Check your system architecture

First, the system architecture was identified using the uname -m command. Based on the detected architecture (such as x86_64, i686, or arm), the appropriate executable file was selected from the provided assignment files to ensure compatibility.

- **uname -m -** x86_64, choose the 64-bit executable.
- **Cd /home/Buddhima/Download – navigate to this directory and give ls command**
- **Cd Executable - navigate to Execute directory and ls**
- **Cd Executable - navigate to Executable directory and give ls command**
- **Chmod +x x86_64 – execute x86_64**
- **Sudo ./x86_64 – run this file**
- **Then it rename it usin IT236**

```
buddhima@Buddhima:~/Desktop$
buddhima@Buddhima:~/Desktop$ uname -m
x86_64
buddhima@Buddhima:~/Desktop$ cd /home/buddhima/Downloads
buddhima@Buddhima:~/Downloads$ ls
 Executables   'Executables(1).zip'   Executables.zip
buddhima@Buddhima:~/Downloads$ cd Executables
buddhima@Buddhima:~/Downloads/Executables$ ls
Executables  __MACOSX
buddhima@Buddhima:~/Downloads/Executables$ cd Executables
buddhima@Buddhima:~/Downloads/Executables/Executables$ ls
ARM  x86_64
buddhima@Buddhima:~/Downloads/Exe[redacted]ables$ chmod +x x86_64
buddhima@Buddhima:~/Downloads/Executables/Executables$ sudo ./x86_64
Enter the student IT number: IT23611788
buddhima@Buddhima:~/Downloads/Executables/Executables$ ls
ARM  IT23611788
buddhima@Buddhima:~/Downloads/Executables/Executables$ sudo ./IT23611788
buddhima@Buddhima:~/Downloads/Executables/Executables$ ls
ARM  data.txt  IT23611788
buddhima@Buddhima:~/Downloads/Executables/Executables$ cat data.txt
buddhima@Buddhima:~/Downloads/Executables/Executables$ cat data.txt
buddhima@Buddhima:~/Downloads/Executables/Executables$ ls
ARM  data.txt  IT23611788
buddhima@Buddhima:~/Downloads/Executables/Executables$ cat data.txt
WM[buddhima@Buddhima:~/Downloads/Executables/Executables$
buddhima@Buddhima:~/Downloads/Executables/Executables$
buddhima@Buddhima:~/Downloads/Executables/Executables$
buddhima@Buddhima:~/Downloads/Executables/Executables$
buddhima@Buddhima:~/Downloads/Executables/Executables$ xxd data.txt
00000000: 0d57 4d5b                                .WM[
buddhima@Buddhima:~/Downloads/Executables/Executables$
buddhima@Buddhima:~/Downloads/Executables/Executables$
buddhima@Buddhima:~/Downloads/Executables/Executables$ gdb ./IT23611788
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law
```

Step 02: Run the Selected Executable

- The selected executable was run with **root privileges**. During execution, the program prompted for an IT number.
  sudo ./IT2361[redacted]
- After providing the IT number, the program generated a **new executable file named after the entered IT number**.

Step 3: Analyze the Generated Executable

The newly created executable was executed. This execution resulted in the creation of a text file named **data.txt**.
The contents of this file were examined to identify patterns, encoded data, suspicious outputs, or hidden information.

Run the new executable:

- ./IT236[redacted]

Analyze data.txt:

```
buddhima@Buddhima:~/Downloads/Executables/Executables$
buddhima@Buddhima:~/Downloads/Executables/Executables$ cat data.txt
WM[buddhima@Buddhima:~/Downloads/Executables/Executables$
```

Step 04: Debug the New Executable Using GDB

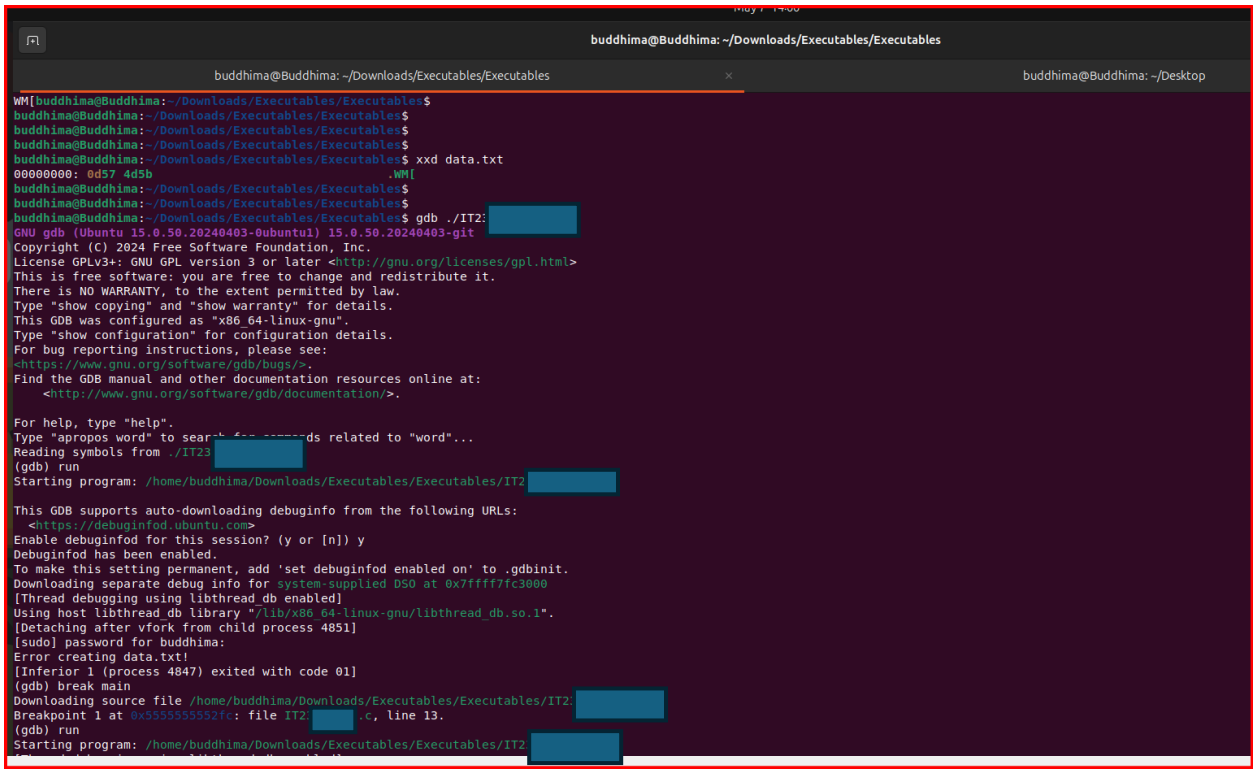
Start GDB:

gdb ./IT23[redacted]

Analyze the flow of execution:

(gdb) run- Start running the program within GDB:

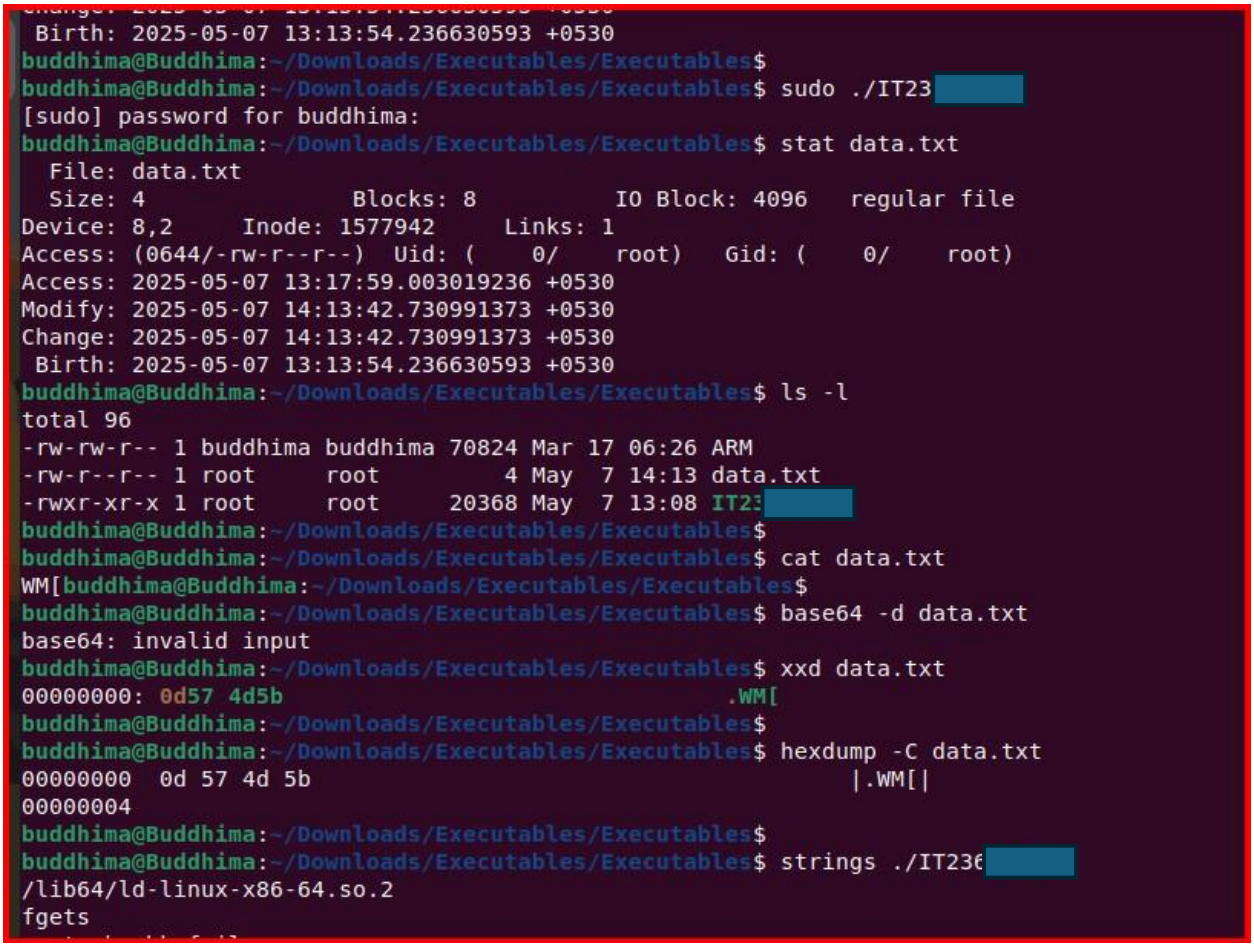
(gdb) break main- Set a breakpoint at the start:



(Gdb) disassemble

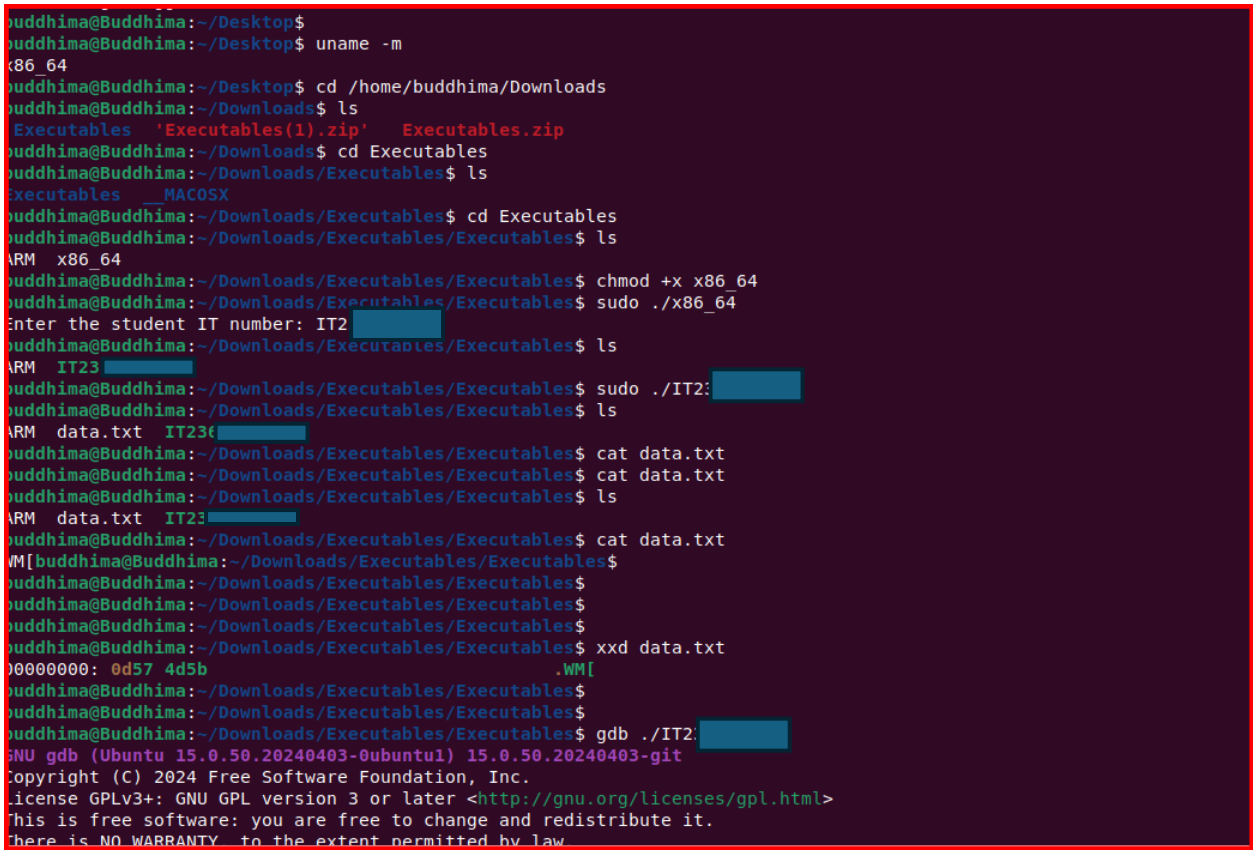
```
Quit
(gdb) disassemble main
Dump of assembler code for function main:
   0x00005555555552f0 <+0>:     endbr64
   0x00005555555552f4 <+4>:     push   %rbp
   0x00005555555552f5 <+5>:     mov    %rsp,%rbp
   0x00005555555552f8 <+8>:     sub    $0x60,%rsp
   0x00005555555552fc <+12>:    mov    %fs:0x28,%rax
   0x0000555555555305 <+21>:    mov    %rax,-0x8(%rbp)
   0x0000555555555309 <+25>:    xor    %eax,%eax
=> 0x000055555555530b <+27>:    lea    0xcf6(%rip),%rax        # 0x555555556008
   0x0000555555555312 <+34>:    mov    %rax,%rsi
   0x0000555555555315 <+37>:    lea    0xcf4(%rip),%rax        # 0x555555556010
   0x000055555555531c <+44>:    mov    %rax,%rdi
   0x000055555555531f <+47>:    call   0x555555555160 <popen@plt>
   0x0000555555555324 <+52>:    mov    %rax,-0x58(%rbp)
   0x0000555555555328 <+56>:    cmpq   $0x0,-0x58(%rbp)
   0x000055555555532d <+61>:    jne    0x555555555348 <main+88>
   0x000055555555532f <+63>:    lea    0xd02(%rip),%rax        # 0x555555556038
   0x0000555555555336 <+70>:    mov    %rax,%rdi
   0x0000555555555339 <+73>:    call   0x5555555550e0 <puts@plt>
   0x000055555555533e <+78>:    mov    $0x1,%eax
   0x0000555555555343 <+83>:    jmp    0x555555555400 <main+272>
   0x0000555555555348 <+88>:    mov    -0x58(%rbp),%rdx
   0x000055555555534c <+92>:    lea    -0x40(%rbp),%rax
   0x0000555555555350 <+96>:    mov    $0x32,%esi
   0x0000555555555355 <+101>:   mov    %rax,%rdi
   0x0000555555555358 <+104>:   call   0x555555555150 <fgets@plt>
   0x000055555555535d <+109>:   mov    -0x58(%rbp),%rax
   0x0000555555555361 <+113>:   mov    %rax,%rdi
   0x0000555555555364 <+116>:   call   0x555555555120 <pclose@plt>
   0x0000555555555369 <+121>:   lea    -0x40(%rbp),%rax
   0x000055555555536d <+125>:   lea    0xcda(%rip),%rdx        # 0x55555555604e
   0x0000555555555374 <+132>:   mov    %rdx,%rsi
   0x0000555555555377 <+135>:   mov    %rax,%rdi
   0x000055555555537a <+138>:   call   0x555555555140 <strcspn@plt>
   0x000055555555537f <+143>:   movb   $0x0,-0x40(%rbp,%rax,1)
   0x0000555555555384 <+148>:   lea    0xcc5(%rip),%rax        # 0x555555556050
   0x000055555555538b <+155>:   mov    %rax,-0x50(%rbp)
   0x000055555555538f <+159>:   mov    -0x50(%rbp),%rdx
   0x0000555555555393 <+163>:   lea    -0x40(%rbp),%rax
   0x0000555555555397 <+167>:   mov    %rdx,%rsi
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) exit
```

Step 05: Determine the Contents of data.txt



Step 07:

1.Execution Process:



2.**Debugging Process:**

System calls- **puts@plt -** Calls the puts() function (prints a null-terminated string with a newline to stdout).

fgets@plt -Calls fgets() (reads a line from a file or stdin into a buffer, up to a given size).

popen@plt -Calls popen() (opens a process by creating a pipe, used for running shell commands and reading their output).

strcspn@plt -Calls strcspn() (calculates the length of the initial segment of a string that does **not** contain any characters from another string).

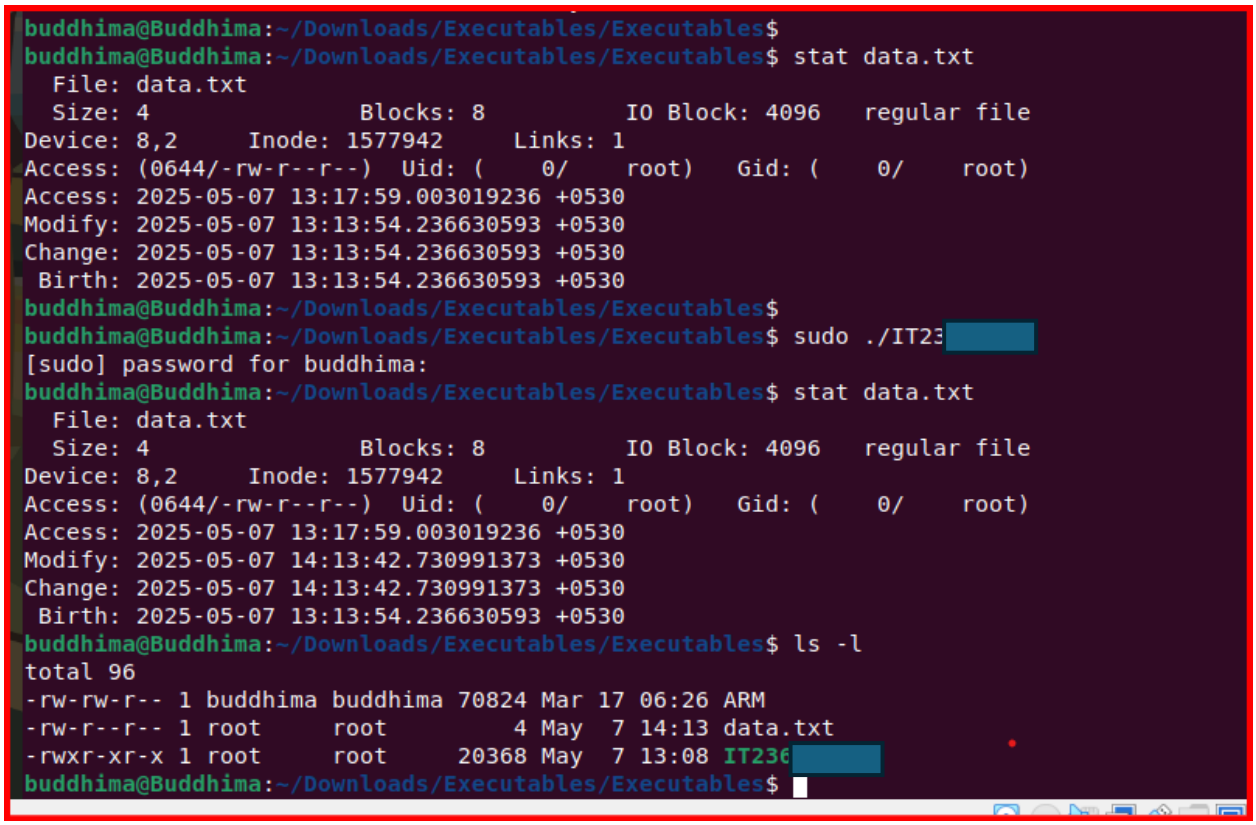
main+88 This refers to the instruction located 88 bytes (or so) into the main() function. This is used in debugging to refer to a specific location inside a function.

main+272 Similarly, this is the offset 272 bytes into main(). GDB might show breakpoints or execution state here.

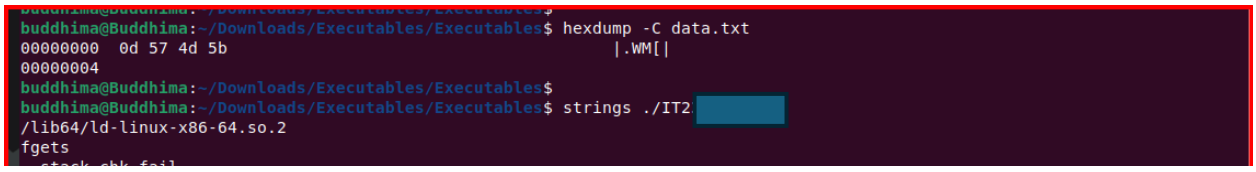
```
Quit
(gdb) disassemble main
Dump of assembler code for function main:
   0x00005555555552f0 <+0>:     endbr64
   0x00005555555552f4 <+4>:     push   %rbp
   0x00005555555552f5 <+5>:     mov    %rsp,%rbp
   0x00005555555552f8 <+8>:     sub    $0x60,%rsp
   0x00005555555552fc <+12>:    mov    %fs:0x28,%rax
   0x0000555555555305 <+21>:    mov    %rax,-0x8(%rbp)
   0x0000555555555309 <+25>:    xor    %eax,%eax
=> 0x000055555555530b <+27>:    lea    0xcf6(%rip),%rax        # 0x555555556008
   0x0000555555555312 <+34>:    mov    %rax,%rsi
   0x0000555555555315 <+37>:    lea    0xcf4(%rip),%rax        # 0x555555556010
   0x000055555555531c <+44>:    mov    %rax,%rdi
   0x000055555555531f <+47>:    call   0x555555555160 <popen@plt>
   0x0000555555555324 <+52>:    mov    %rax,-0x58(%rbp)
   0x0000555555555328 <+56>:    cmpq   $0x0,-0x58(%rbp)
   0x000055555555532d <+61>:    jne    0x555555555348 <main+88>
   0x000055555555532f <+63>:    lea    0xd02(%rip),%rax        # 0x555555556038
   0x0000555555555336 <+70>:    mov    %rax,%rdi
   0x0000555555555339 <+73>:    call   0x5555555550e0 <puts@plt>
   0x000055555555533e <+78>:    mov    $0x1,%eax
   0x0000555555555343 <+83>:    jmp    0x555555555400 <main+272>
   0x0000555555555348 <+88>:    mov    -0x58(%rbp),%rdx
   0x000055555555534c <+92>:    lea    -0x40(%rbp),%rax
   0x0000555555555350 <+96>:    mov    $0x32,%esi
   0x0000555555555355 <+101>:   mov    %rax,%rdi
   0x0000555555555358 <+104>:   call   0x555555555150 <fgets@plt>
   0x000055555555535d <+109>:   mov    -0x58(%rbp),%rax
   0x0000555555555361 <+113>:   mov    %rax,%rdi
   0x0000555555555364 <+116>:   call   0x555555555120 <pclose@plt>
   0x0000555555555369 <+121>:   lea    -0x40(%rbp),%rax
   0x000055555555536d <+125>:   lea    0xcda(%rip),%rdx        # 0x55555555604e
   0x0000555555555374 <+132>:   mov    %rdx,%rsi
   0x0000555555555377 <+135>:   mov    %rax,%rdi
   0x000055555555537a <+138>:   call   0x555555555140 <strcspn@plt>
   0x000055555555537f <+143>:   movb   $0x0,-0x40(%rbp,%rax,1)
   0x0000555555555384 <+148>:   lea    0xcc5(%rip),%rax        # 0x555555556050
   0x000055555555538b <+155>:   mov    %rax,-0x50(%rbp)
   0x000055555555538f <+159>:   mov    -0x50(%rbp),%rdx
   0x0000555555555393 <+163>:   lea    -0x40(%rbp),%rax
   0x0000555555555397 <+167>:   mov    %rdx,%rsi
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) exit
```

3.File System Analysis

- Stat data.txt – shows static data of data.txt file.

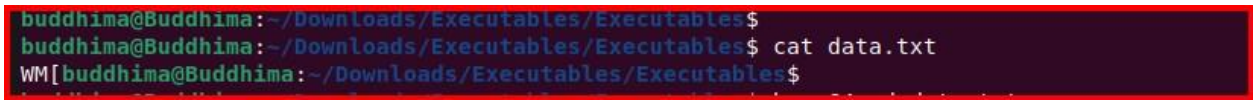


- Hexdump -C data.txt – convert into hexa value

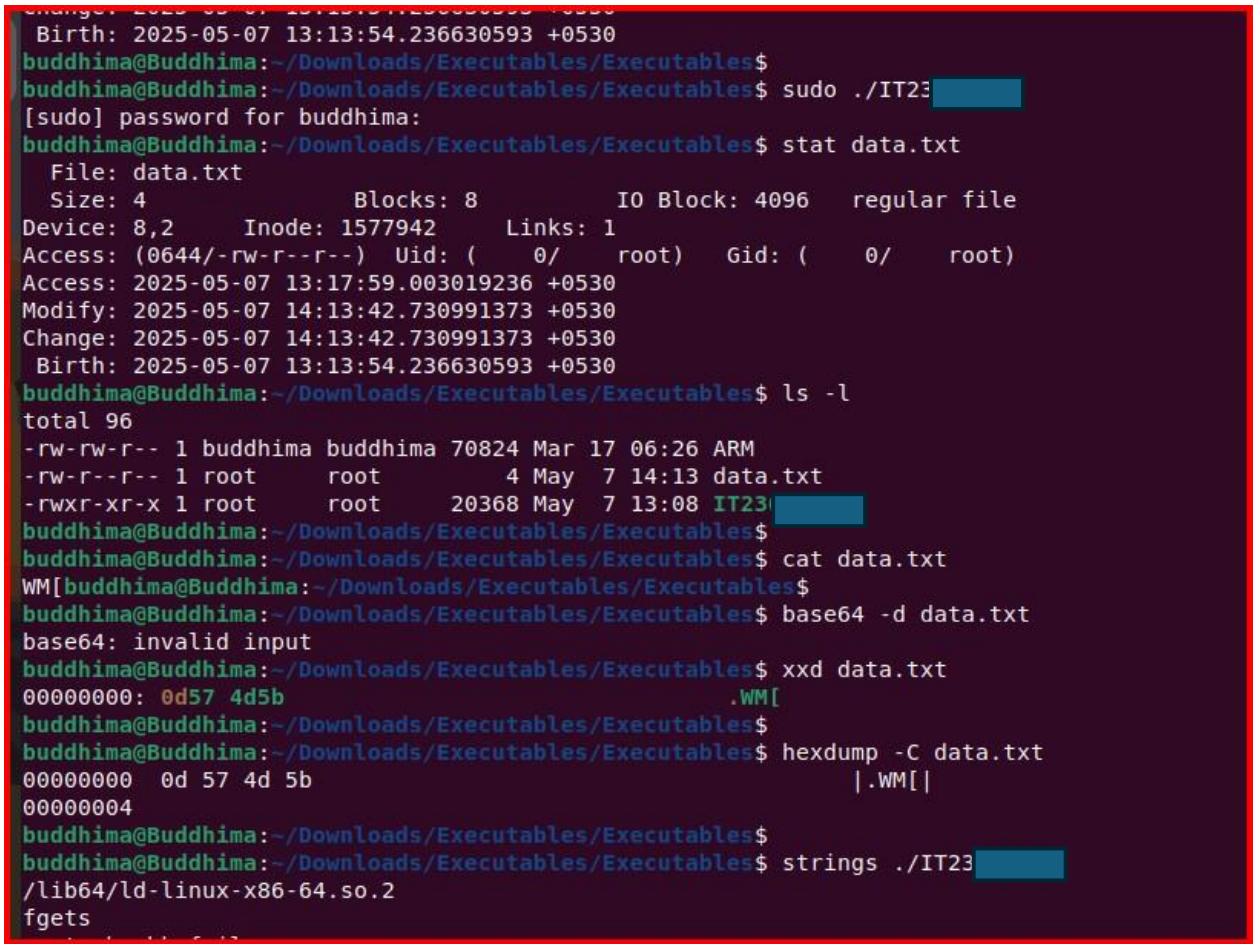


4.Analysis of "data.txt"

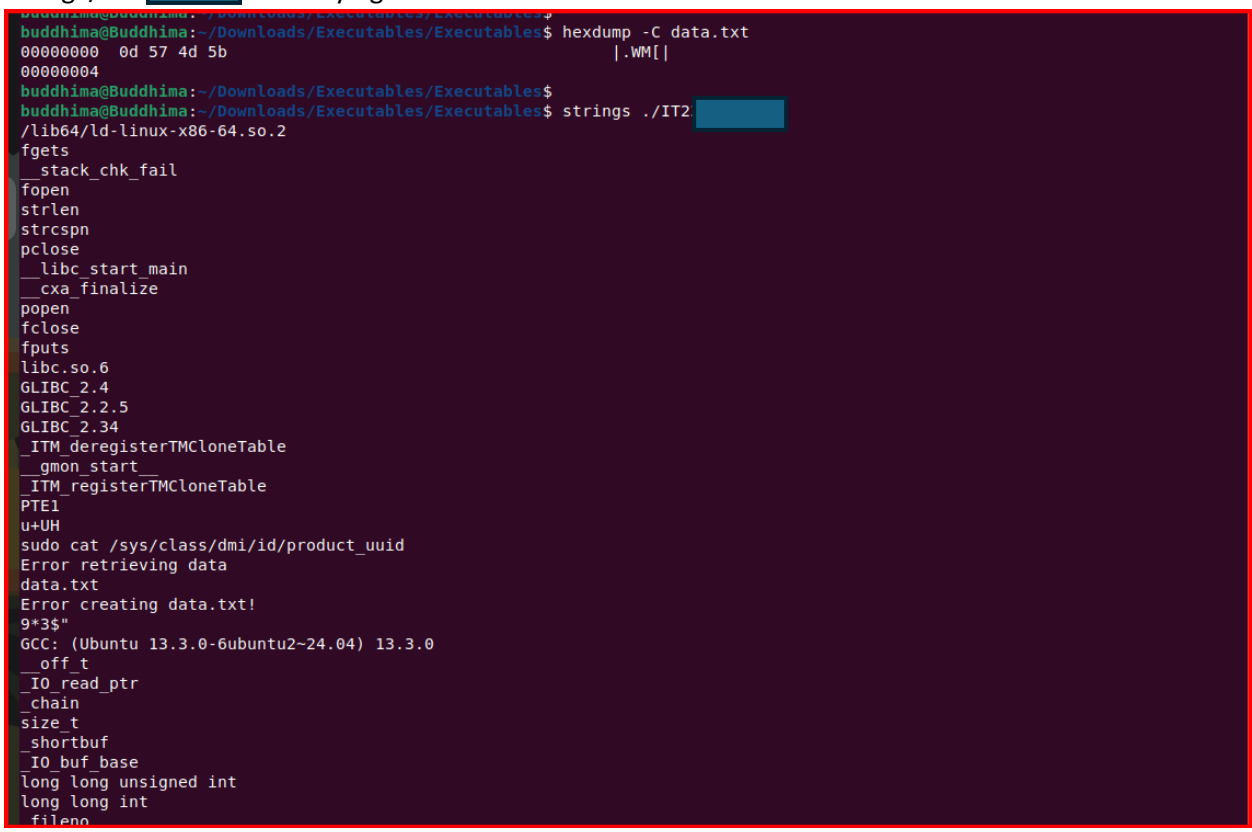
- Cat data.txt = display the encrypt message.



- Source of the contents of data.txt



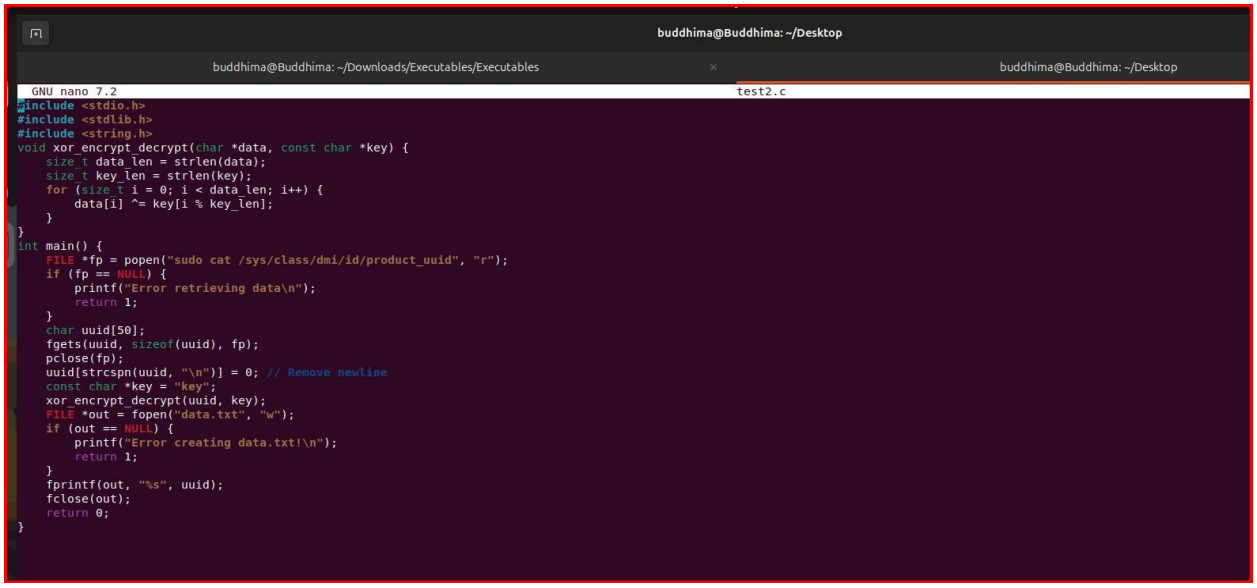
- Any decoded/processed information
  String ./IT23[redacted]- identifying encoded information inside executables.



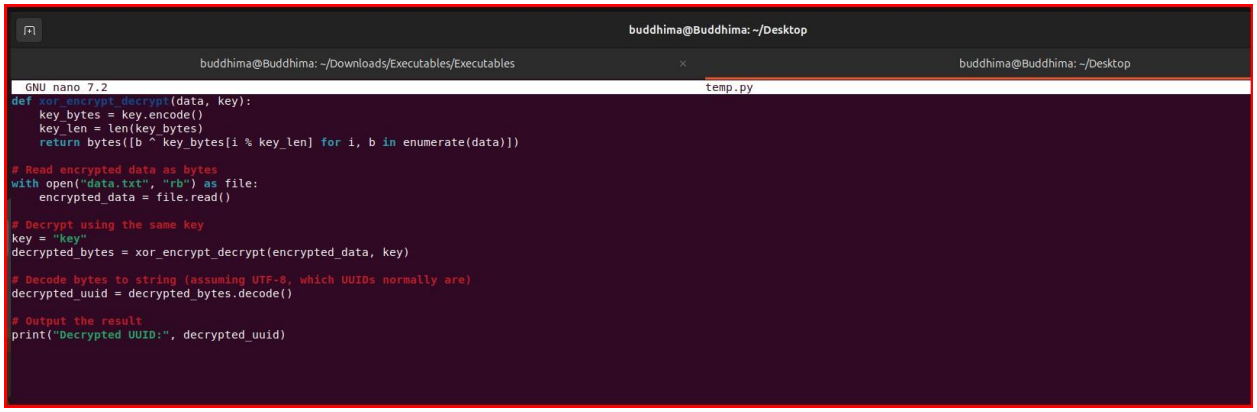
Nano test2.c -Check decrypted code

Copy the source code into test.c file

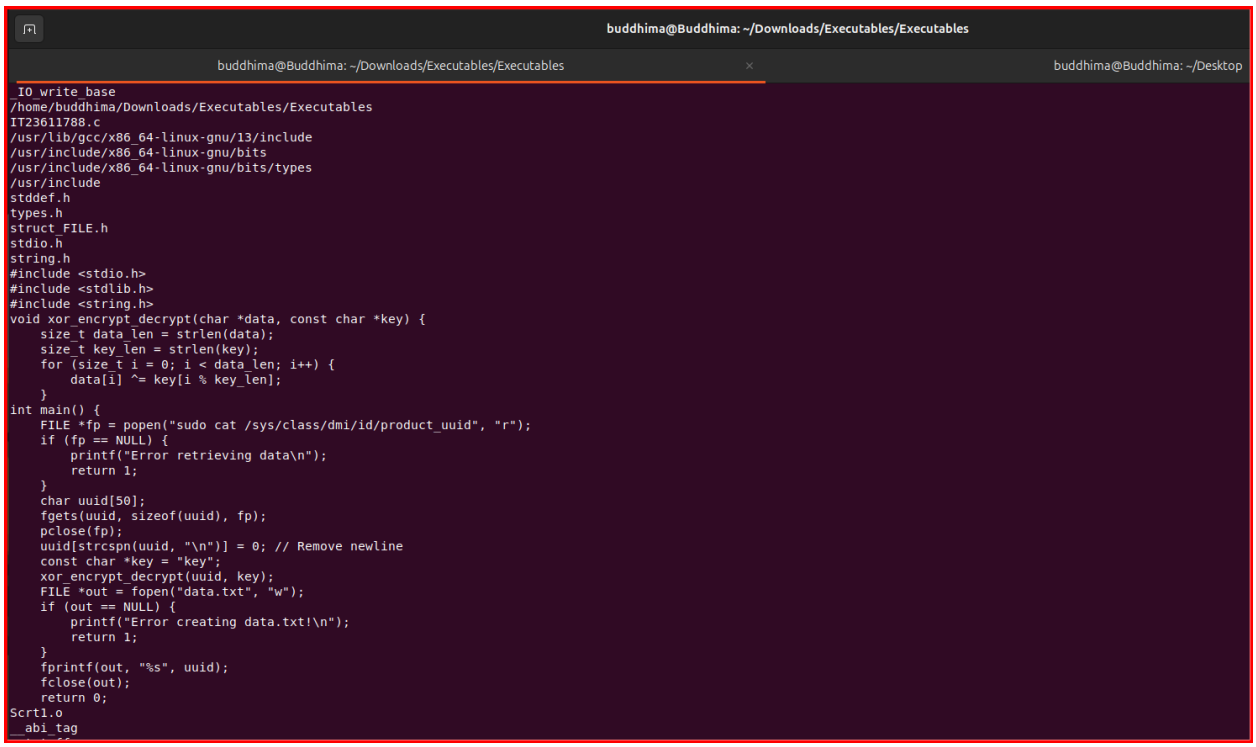
This program extracts the system's hardware UUID, encrypts it using XOR encryption, and stores the encrypted value in data.txt for later decryption.



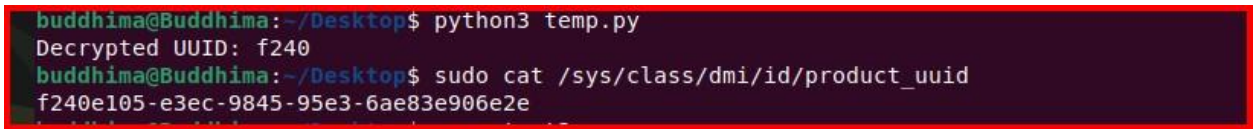
Nano temp.py – create a python code to execute



This Python script decrypts XOR-encrypted binary data from data.txt using a predefined key and reveals the original UUID stored in the file.



Decrypted messeg



Only a partial UUID (f240) was recovered because XOR encryption generated NULL bytes, causing the encrypted string to be truncated when written to data.txt using string-based file output