Sri Lanka Institute of Information Technology

Module: IE2012

Year 2, Semester 1

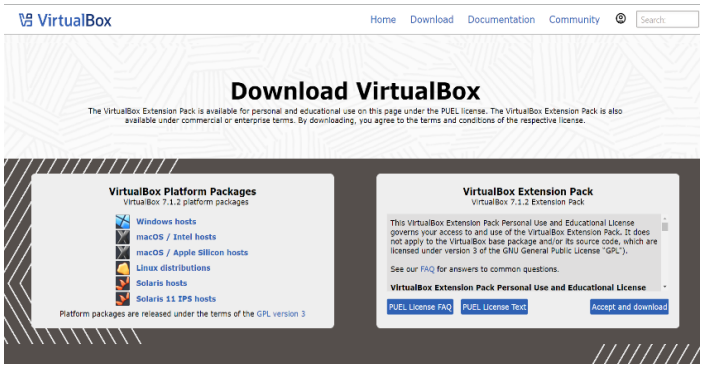
Systems and Networking Programming

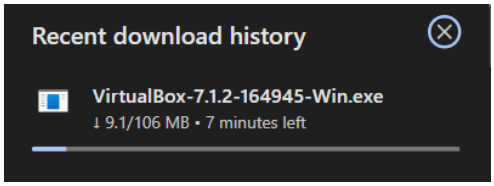
S M F Hana IT23255142

01.Basic Of Linux

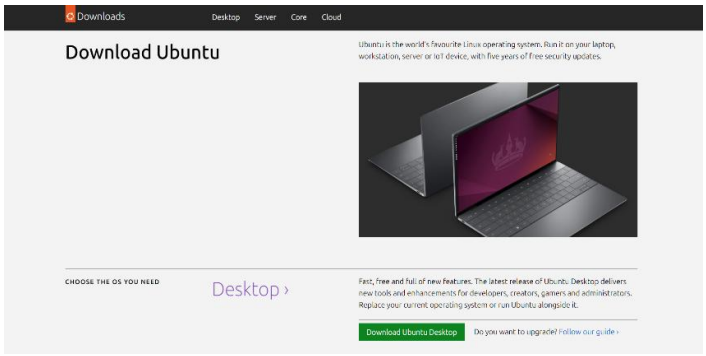
01.1Virtual Machine

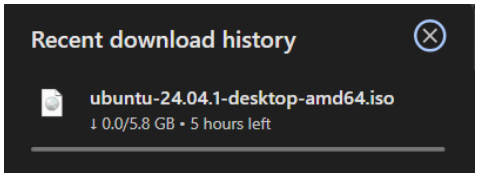
Downloaded the virtual machine after choosing the windows hosts version of the virtual platform package from the official VirtualBox website, https://www.virtualbox.org/ .



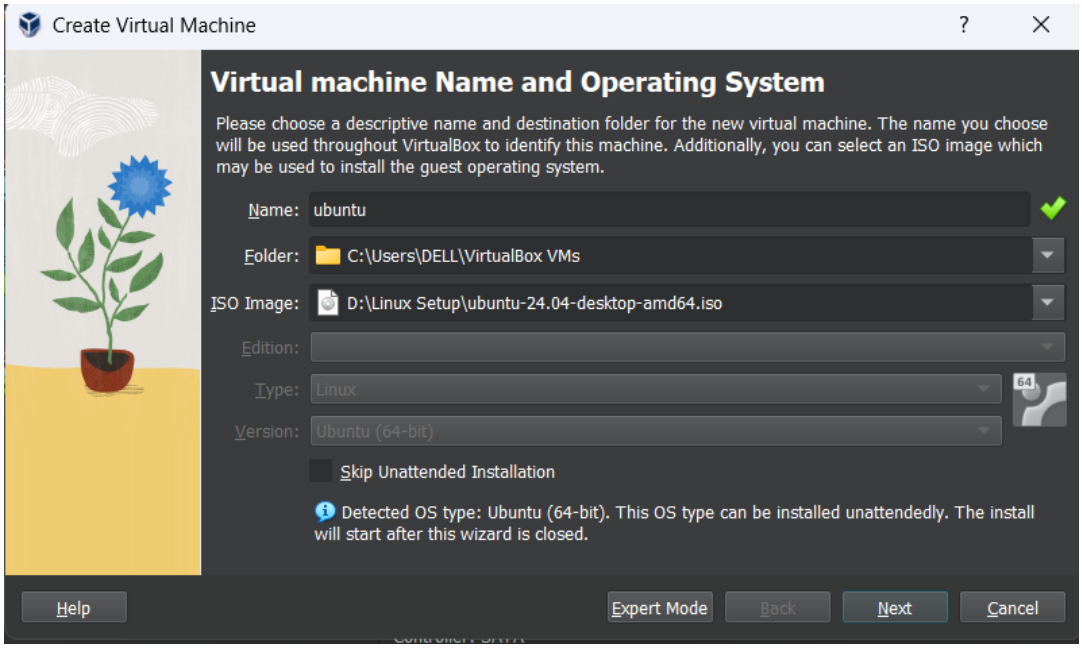


Downloaded the Linux-based system, Ubuntu desktop from https://ubuntu.com/download .Selected the version 24.04.1 LTS

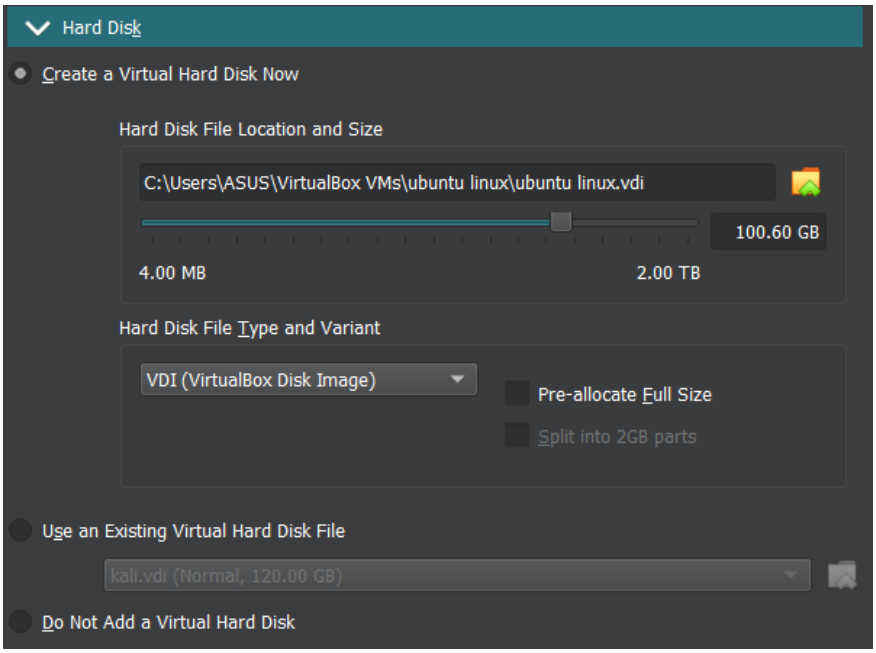




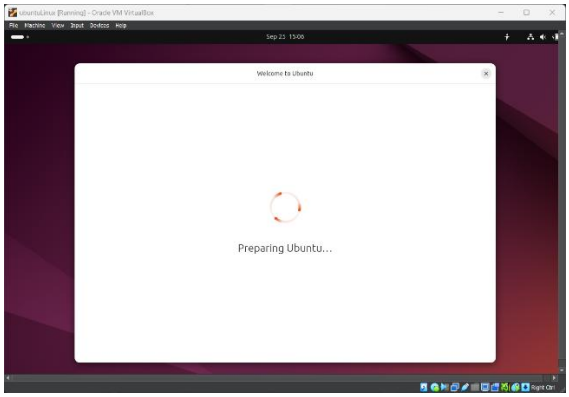
After the installation of virtual machine, it was opened and the **new** button was clicked. Then typed the name field as **ubuntu** as the VM name and assigned the path to store the VM files. For installation ISO image was selected by giving *other* and navigating to the previously mentioned ubuntu setup file

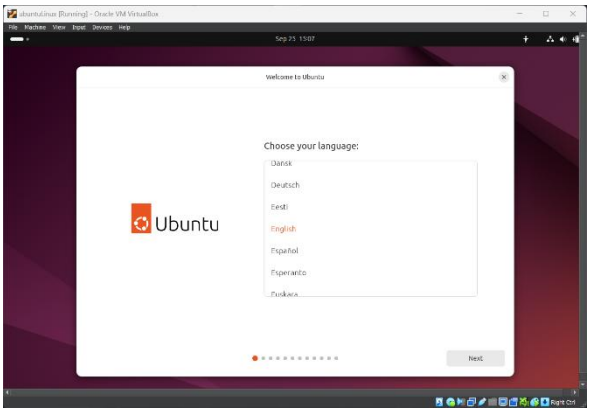


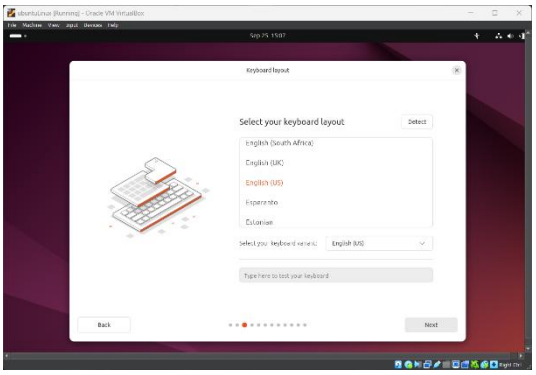
After clicking **Next, Base Memory** was set to 4000 MB and **Processors** was set to 7. Set the size as follows and clicked finish

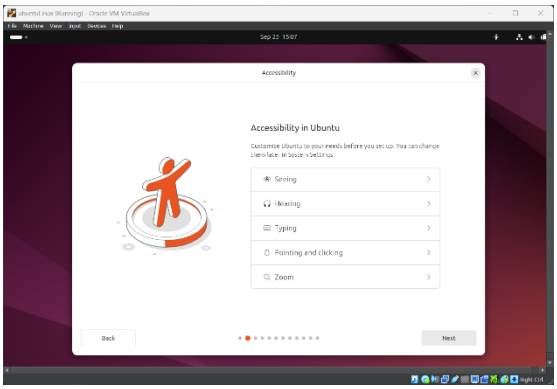


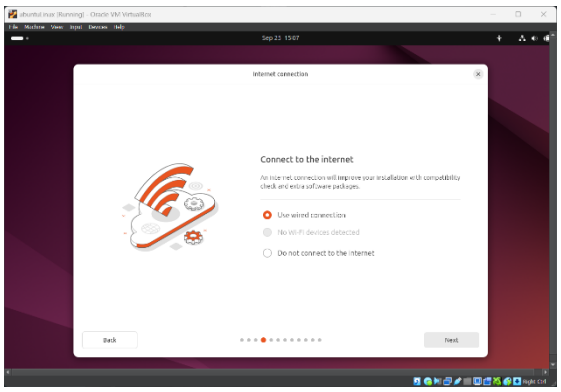
Thereafter, followed the following steps

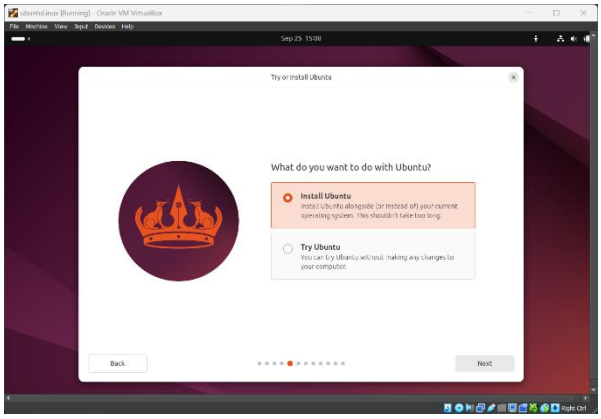


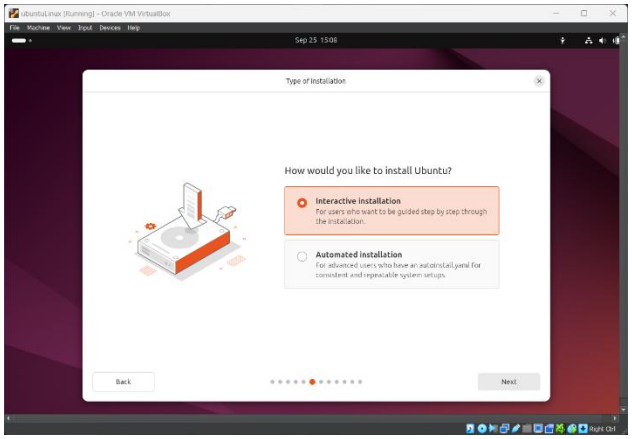


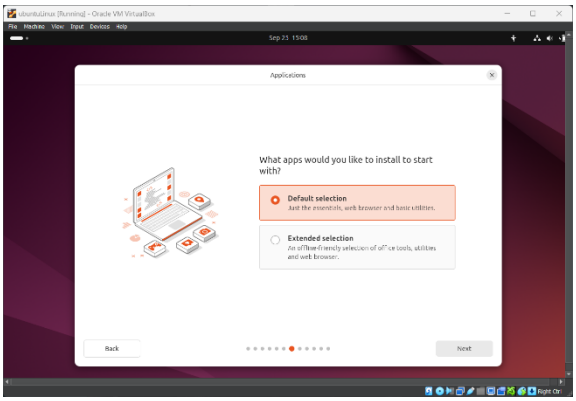


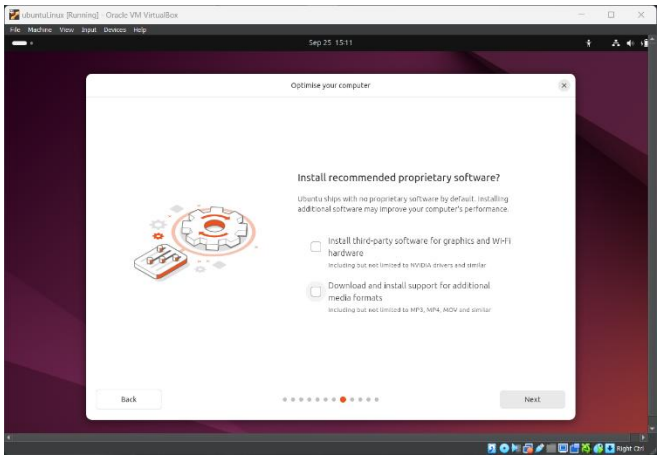


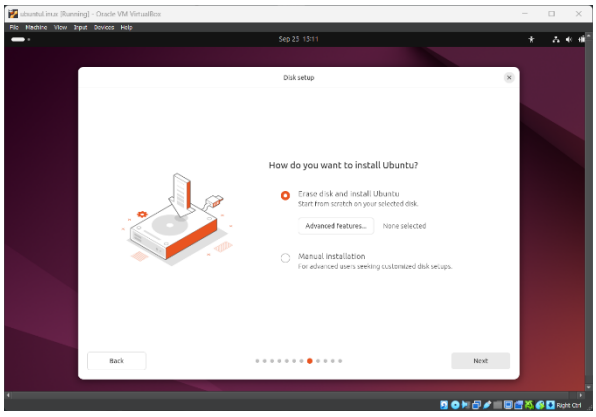


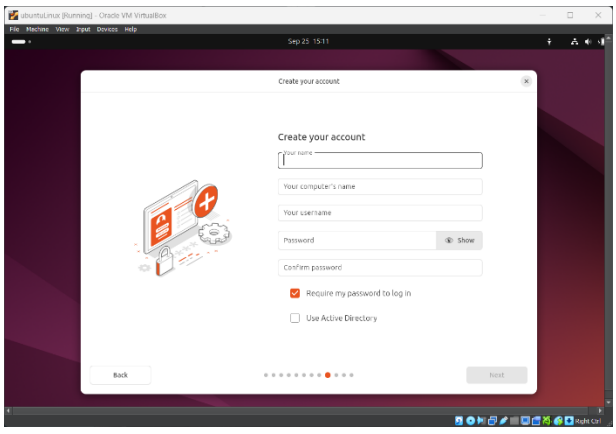


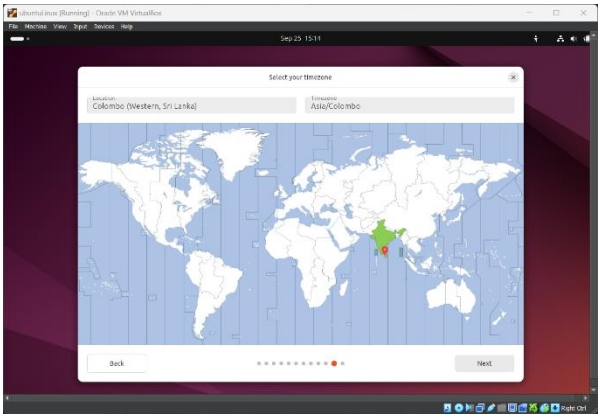

Type of Installation
How would you like to install Ubuntu?
Interactive installation
For users who want to be guided step by step through the installation.
Automated installation
For advanced users who have an autoinstall.yaml for consistent and repeatable system setups.
Back
Next

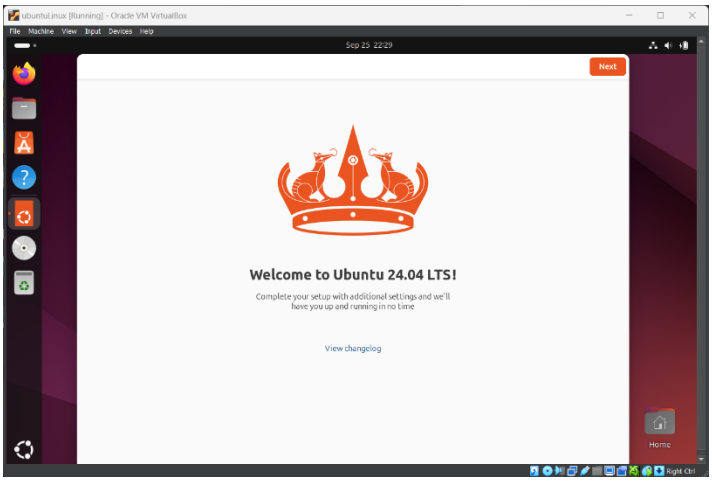

Applications
What apps would you like to install to start with?
Default selection
Just the essentials, web browser and basic utilities.
Extended selection
Back
Next

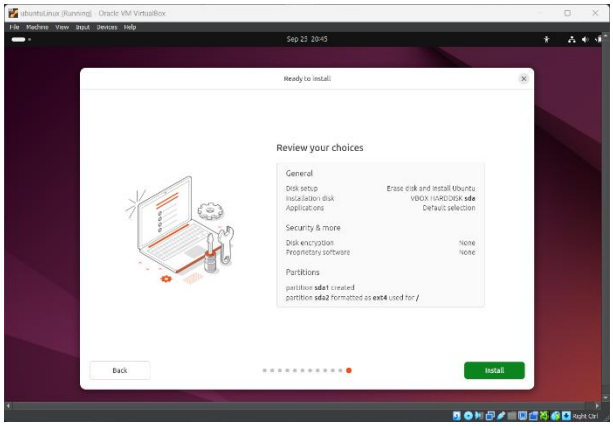

Optimise your computer
Install recommended proprietary software?
Ubuntu ships with no proprietary software by default. Installing additional software may improve your computer's performance.
Install third-party software for graphics and Wi-Fi hardware
Including but not limited to NVIDIA drivers and similar
Download and install support for additional media formats
Including but not limited to MP3, MP4, MOV and similar
Back
Next

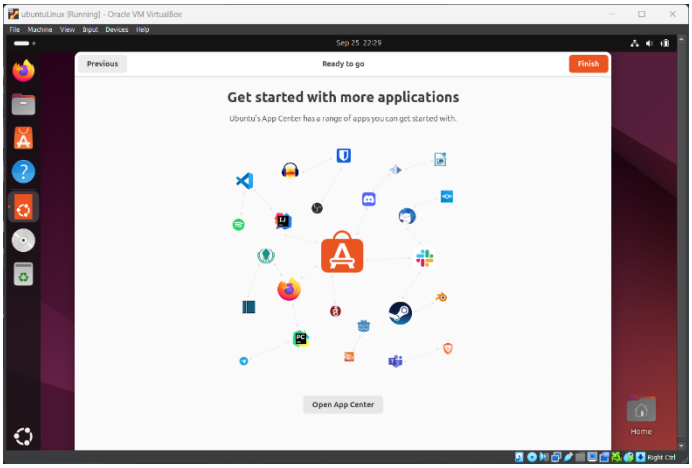

Disk setup
How do you want to install Ubuntu?
Erase disk and install Ubuntu
Start from scratch on your selected disk.
Advanced features...
None selected
Manual installation
For advanced users seeking customized disk setups.
Back
Next

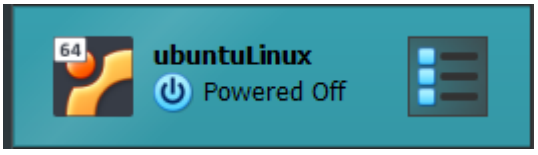

Create your account
Create your account
Your name
Your computer's name
Your username
Password
Show
Confirm password
Require my password to log in
Use Active Directory
Back
Next


Select your timezone
Location
Colombo (Western, Sri Lanka)
Timezone
Asia/Colombo
Back
Next


Next
Welcome to Ubuntu 24.04 LTS!
Complete your setup with additional settings and we'll have you up and running in no time
View changelog
Home


Review your choices
Back
Install


Previous
Ready to go
Finish
Get started with more applications
Ubuntu's App Center has a range of apps you can get started with.
Open App Center
Home


64
ubuntuLinux
Powered Off

02.2 Command Line Introduction

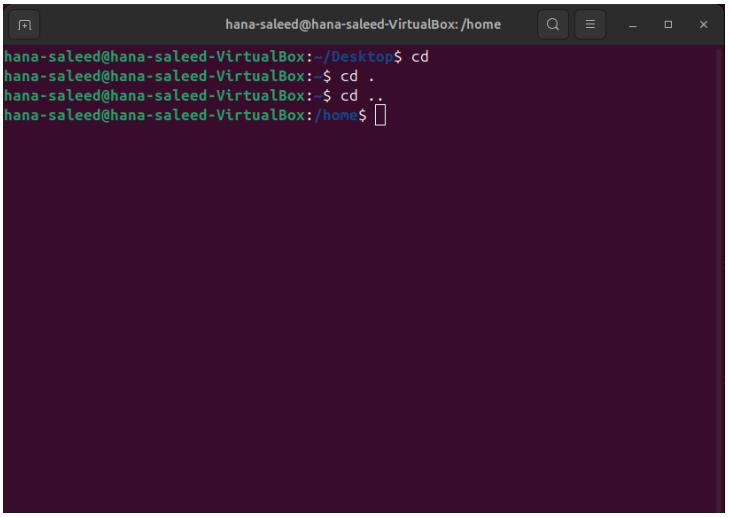
I. Basic Navigation Commands

a. cd (Change Directory) – Used to navigate between directories in the file system

cd : Navigates to home directory

cd . : Stays in the current directory

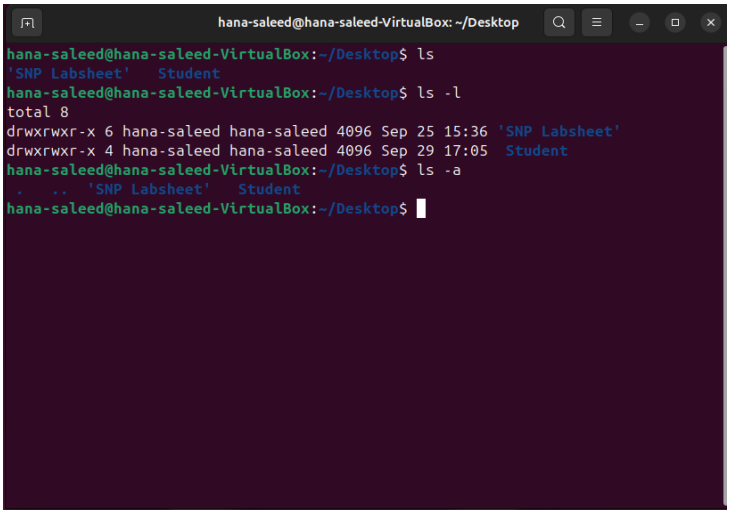
cd .. : Moves up one directory level



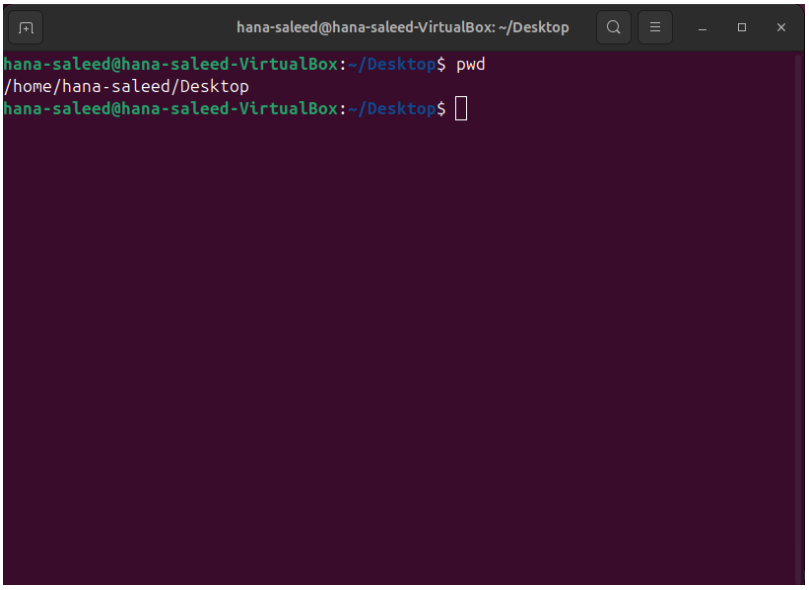
b. ls (list) – Lists the files and the directories in the current directory

i. ls -l : Display a detailed list

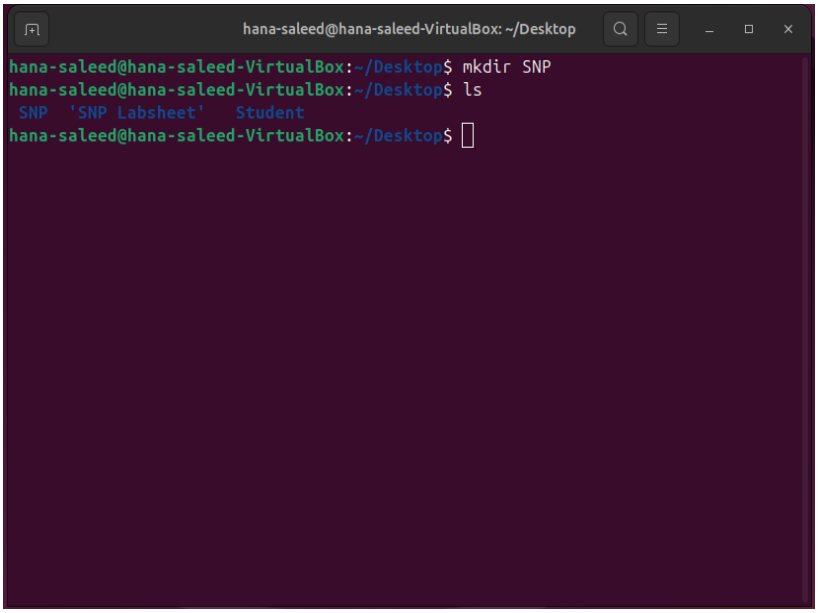
ii. ls -a : Display files including the hidden files



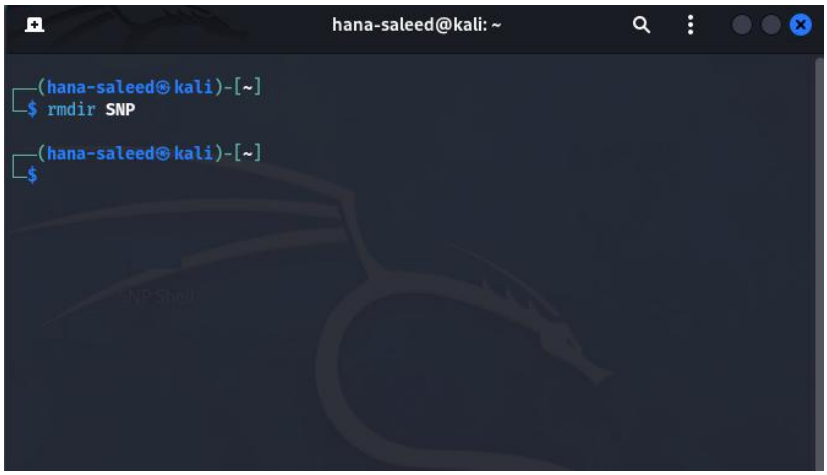
c. pwd (print work directory) – Displays the full path of the current directory



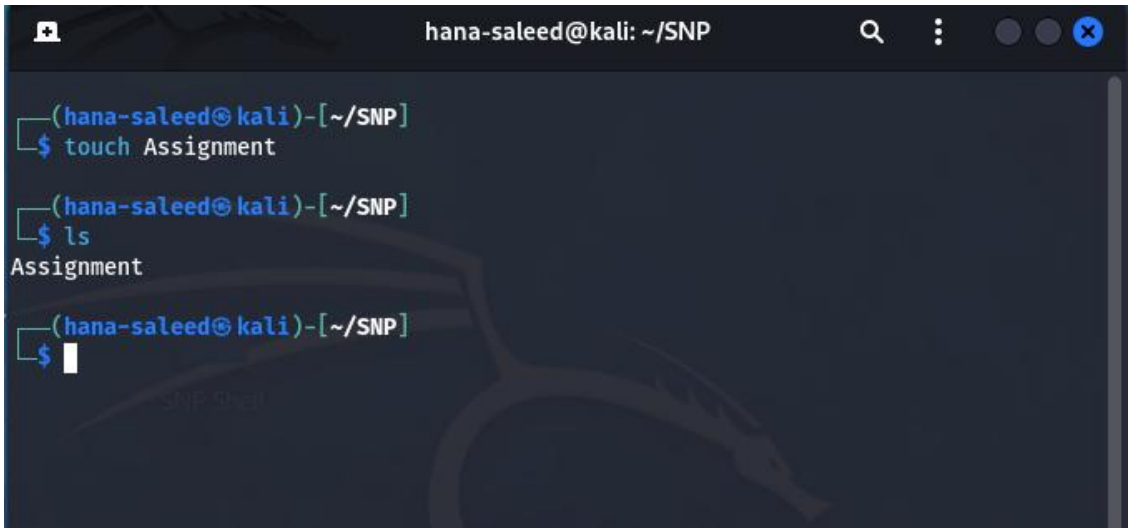
d. mkdir (make directory) – Creates a new directory in the current location. We can verify the creation of the directory with **mkdir** after it appears in the list when the **ls** command is run



d. rmdir (remover directory) - Removes or deletes an empty directory from the file system.

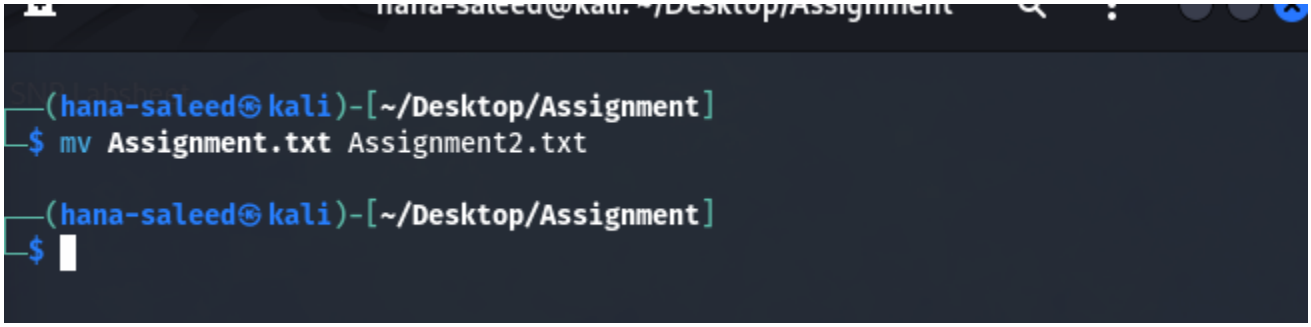


e. touch – Creates a new empty file or update timestamps. The filename has to be specified after the command. The syntax follows as **touch *filename***

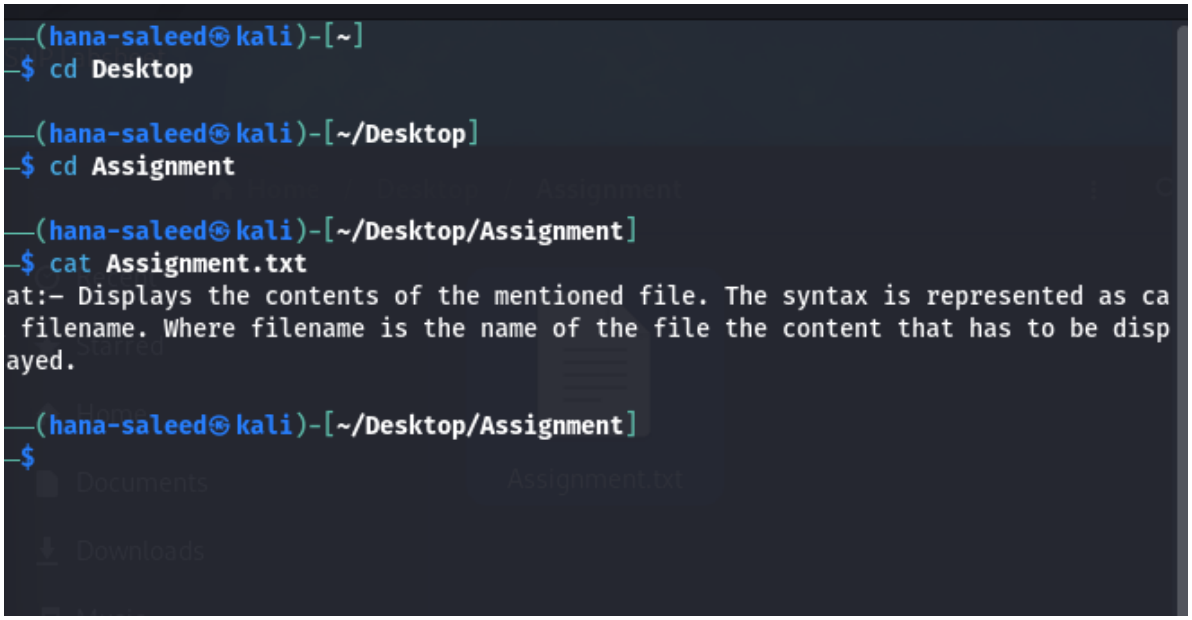


f. cp (copy) – Copies files or directories from one location to another. The syntax of the command is **cp *source destination***, where ***source*** is the file or directory that has to be copied and ***destination*** is the destination of the file or directory where the file needs to be copied.

g. mv (move) – Moves or renames files and directories. To move the **mv *source path_to_new_destination***, where ***source the*** is the file or directory that has to be moved and ***path_to_new_destination*** is the new location of the file. To rename the file **mv *filename new_filename***, where ***filename*** renames to ***new_filename.***



h.cat (concatenate) – Displays the contents of the mentioned file. The syntax is represented as **cat *filename.*** Where ***filename*** is the name of the file the content that has to be displayed.



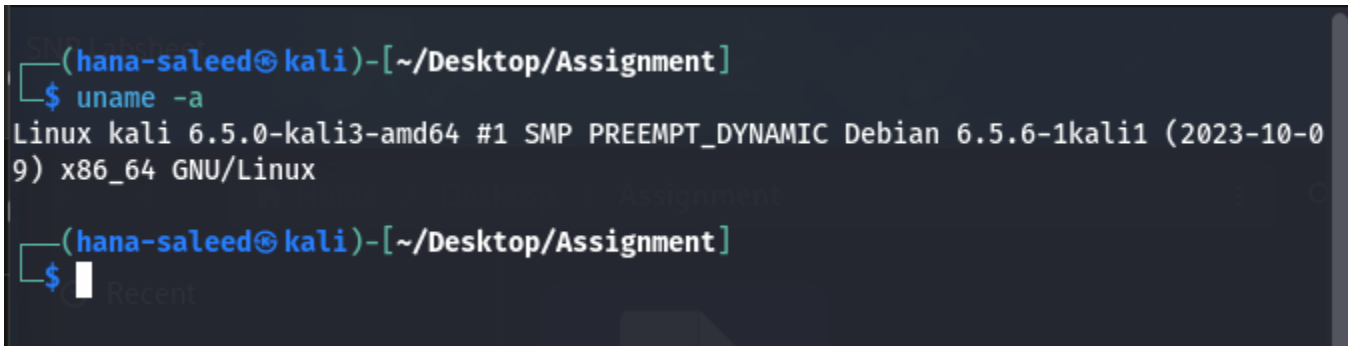
i. more – Displays the contents of a file one screen at a time

j. head – Displays the first few lines in a file

k. tail – Displays the last few lines in a file

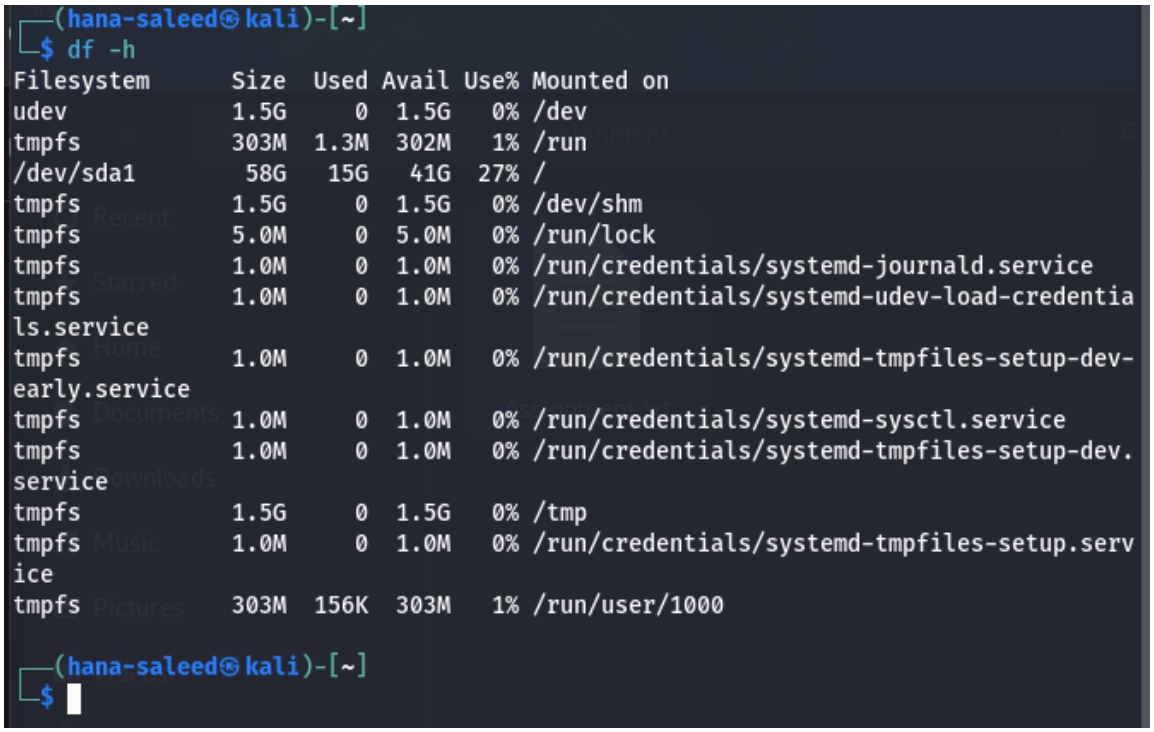
02.3 System Information and User Management

l. uname -a – Displays details of the system and kernel. The **-a** stands for all which provides a summary of the system's configuration and environment.

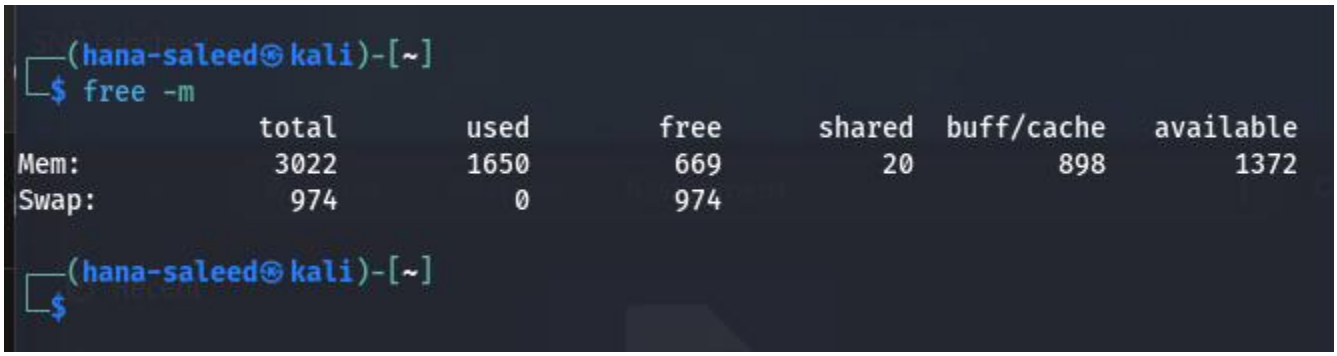


m. cat /proc/version - Displays the information about the currently running kernel and the system. **/proc/version** file contains details about the Linux kernel version.

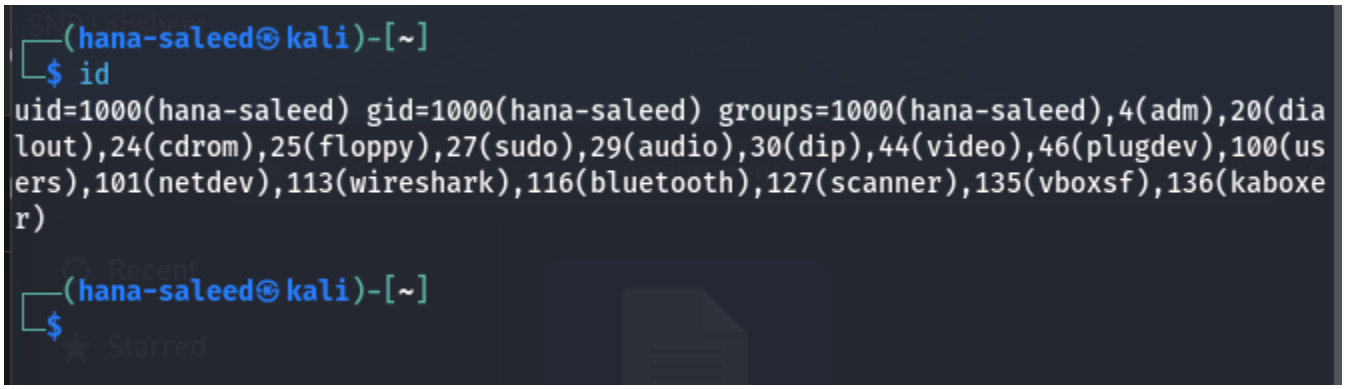
n. df -h (disk filesystem) – Displays the disk space usage for all mounted filesystems in human readable format. **df** reports the amount of disk space used and available in the filesystem while **-h** stands for human-readable which helps to display the output in a more understandable way.



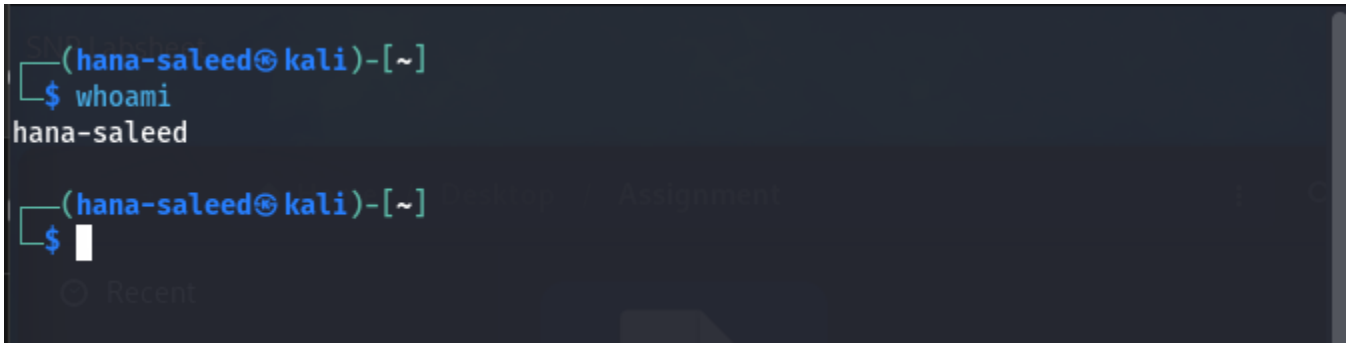
o. free -m – Displays information about total, used and available space of physical memory and swap memory. By default, **free** command will print the memory details in kilo bytes. **-m** stands for megabytes. This will print the details in megabytes.



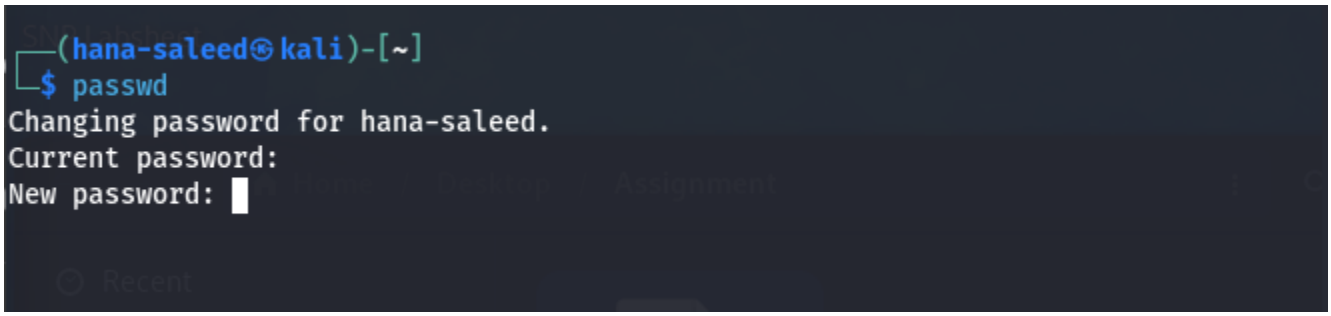
p. id – Displays the user and group details for mentioned user as **id *username.*** When username is not specified it displays the details of the current logged in user.



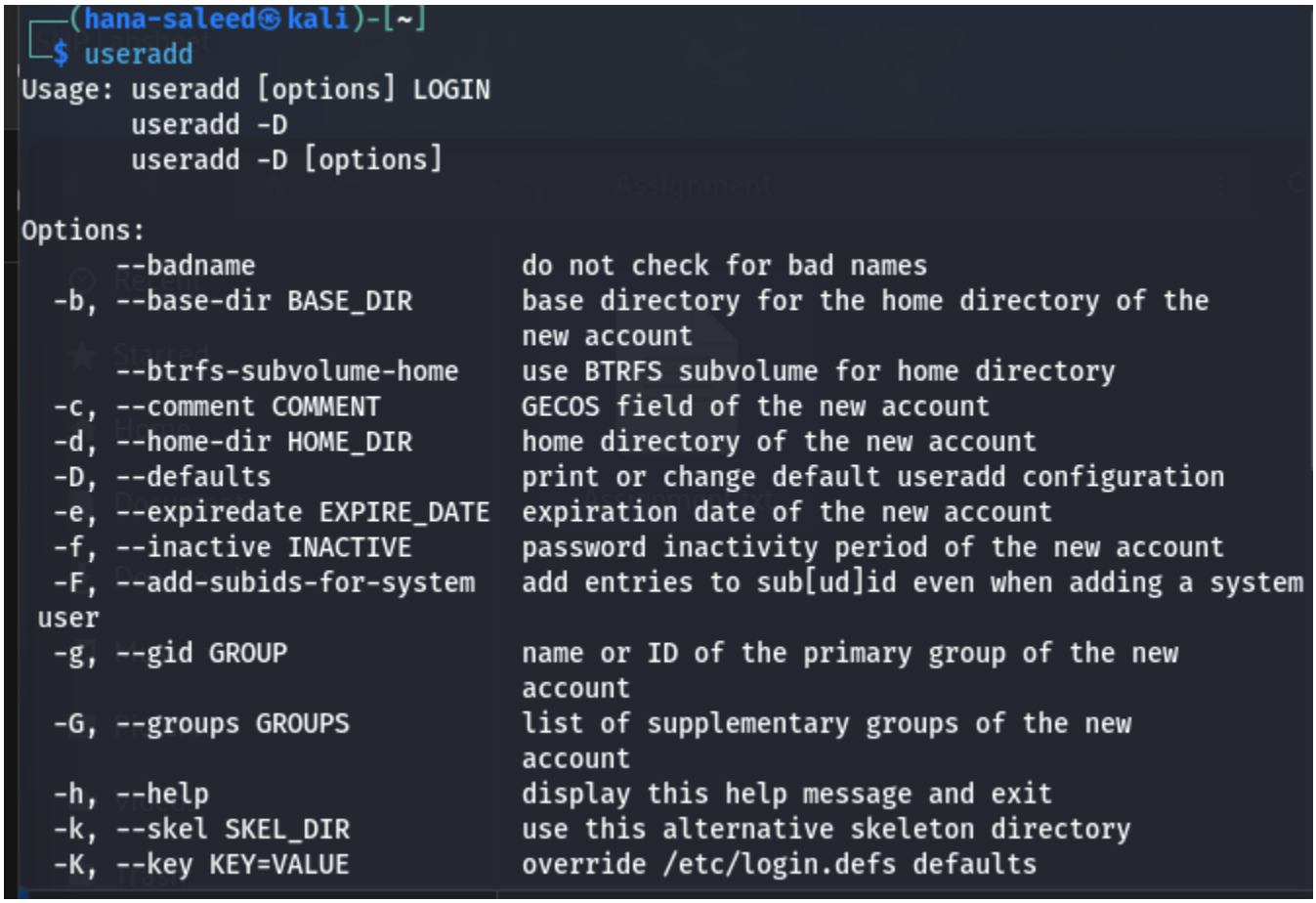
q. whoami – Displays the username of the user that has currently logged in.



r. passwd – This command is used to change the user password. By default, the system will prompt to enter the current password subsequently prompting to enter the new password



s. useradd – This command is used to create new user account

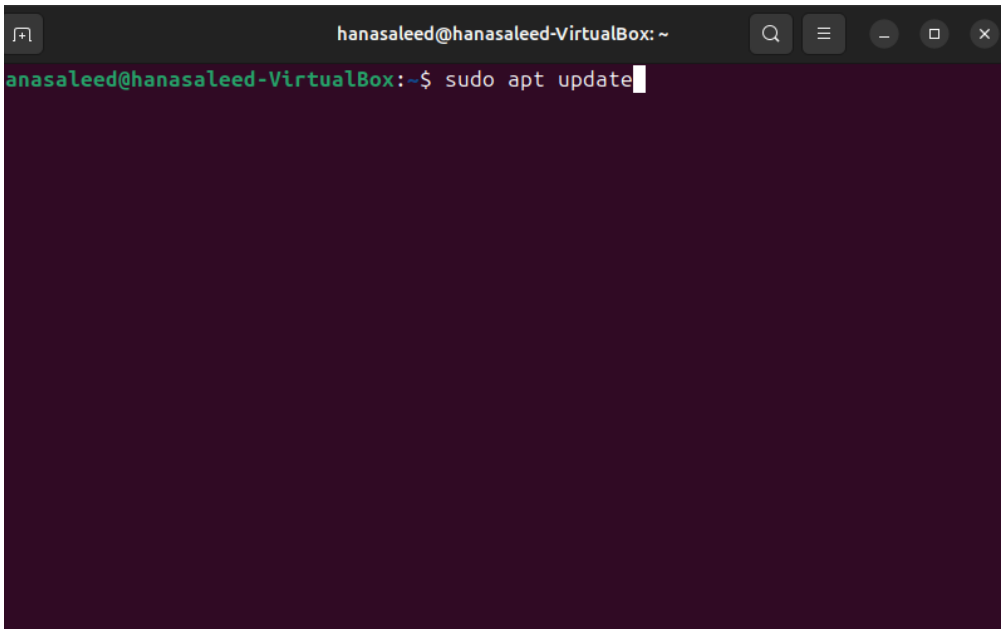


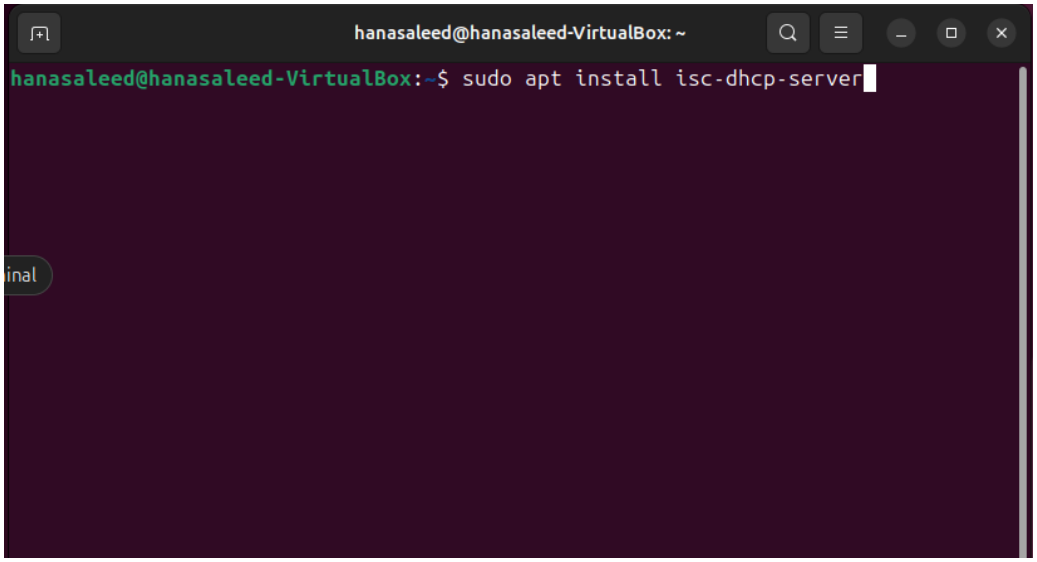
02. DHCP, DNS and NTP Services

02.1 DHCP (Dynamic Host Configuration Protocol)

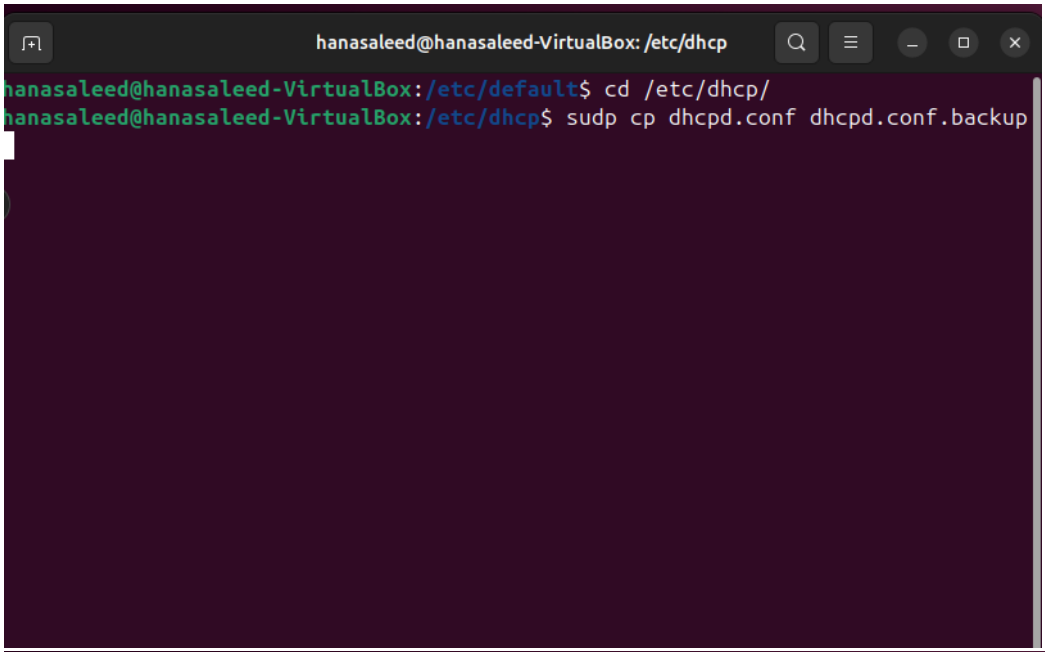
DHCP is a network protocol used to automatically assign IP addresses to devices on a network. This does not require a manual configuration. DHCP simplifies the process of managing IP address assignment, particularly in large networks helping to prevent IP address conflicts. First and foremost, to install DHCP the ubuntu has to be updated to patch vulnerabilities in the system, enhance or for better performances.

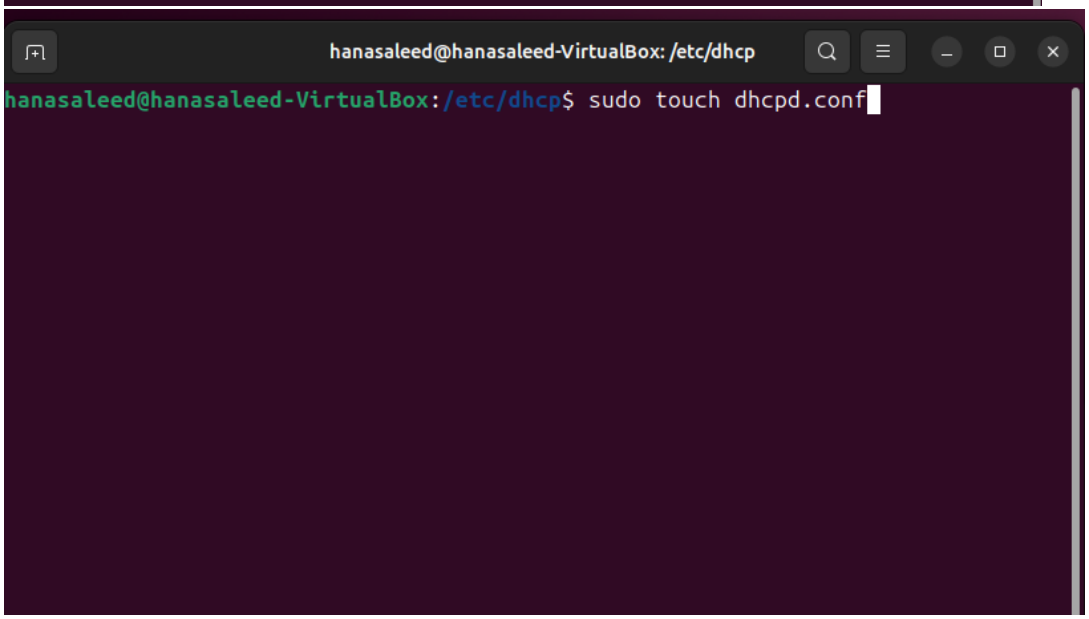
The DHCP server can be installed by the command **sudo apt install isc-dhcp-server**



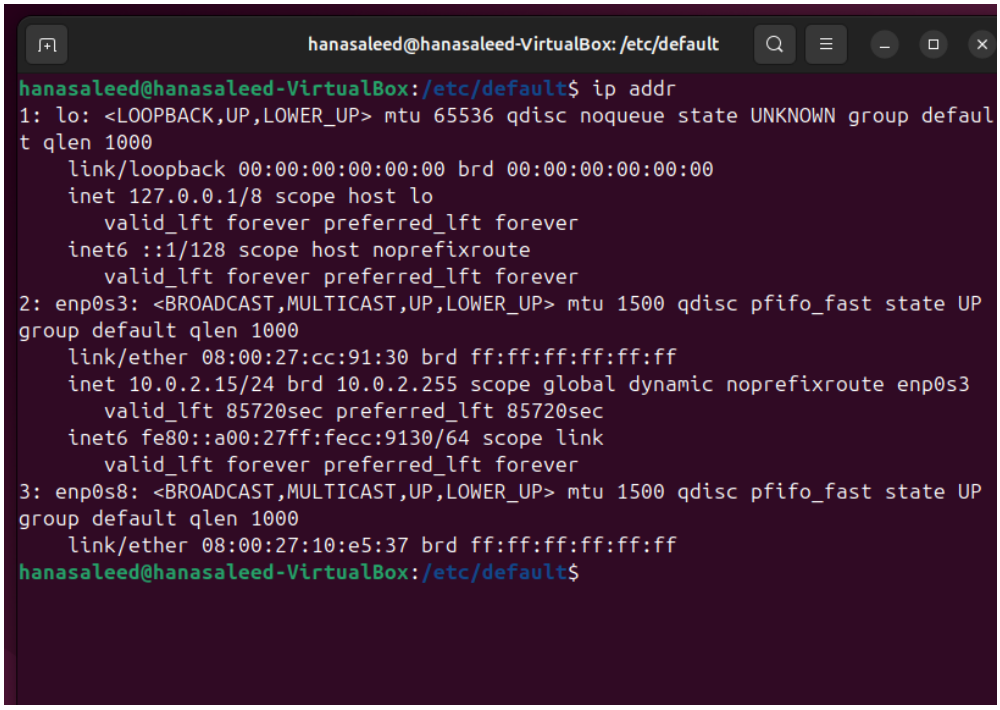


Then, one must navigate to the directory where the configuration files are located by **cd /etc/dhcp/** and backup the default file in case of any problems by typing the command **sudo cp dhcpd.conf dhcpd.conf.backup.** Next, dhcp file should be created using **sudo touch dhcpd.conf**



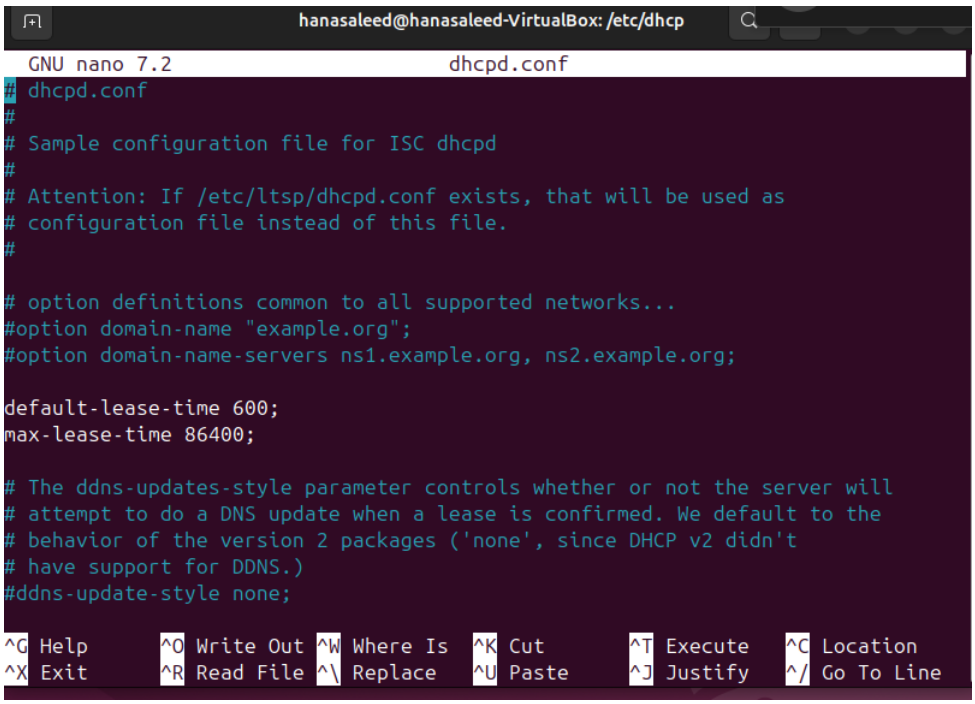


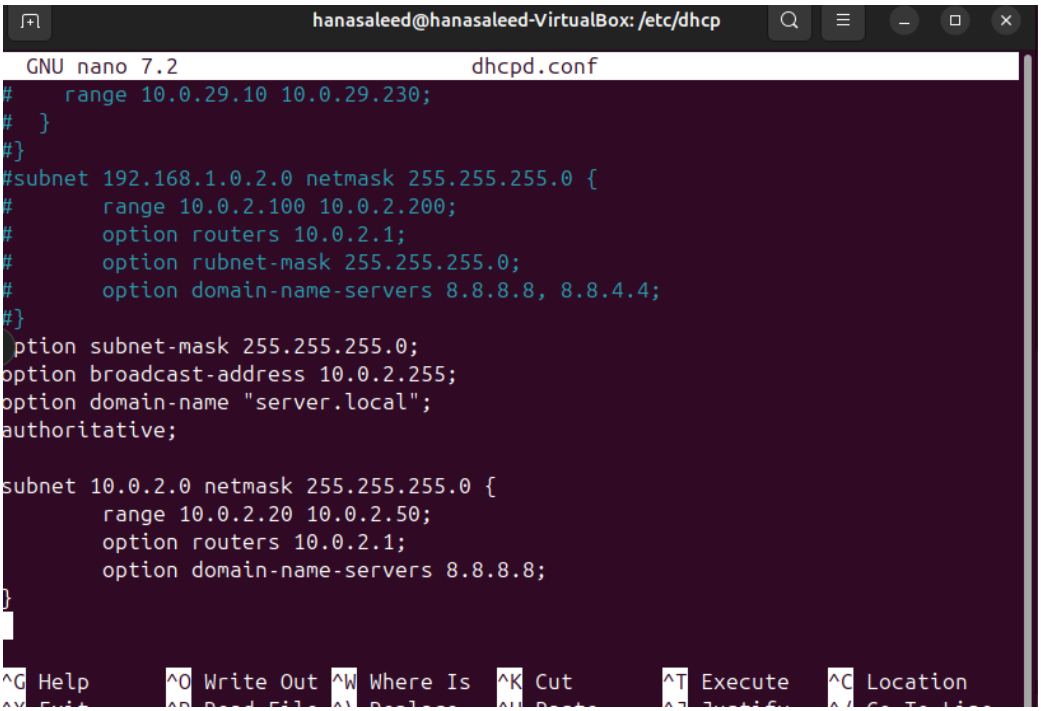
The ip address of the system has to be noted down. In order for that command **ip addr** has to be entered



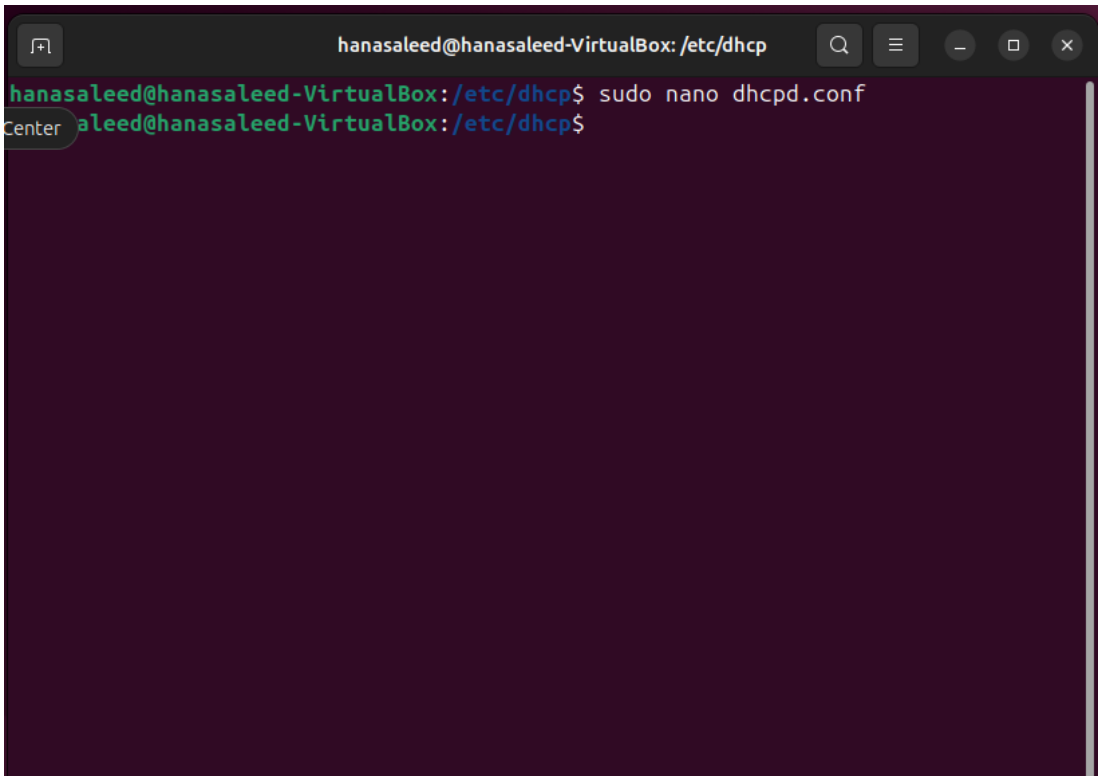
The Ip address here is **10.0.2.15/24** and **enp0s3**

The file has to be opened by the command **sudo nano dhcp.conf**. This will open the file **dhcpd.conf**. The file has to be updated as follows

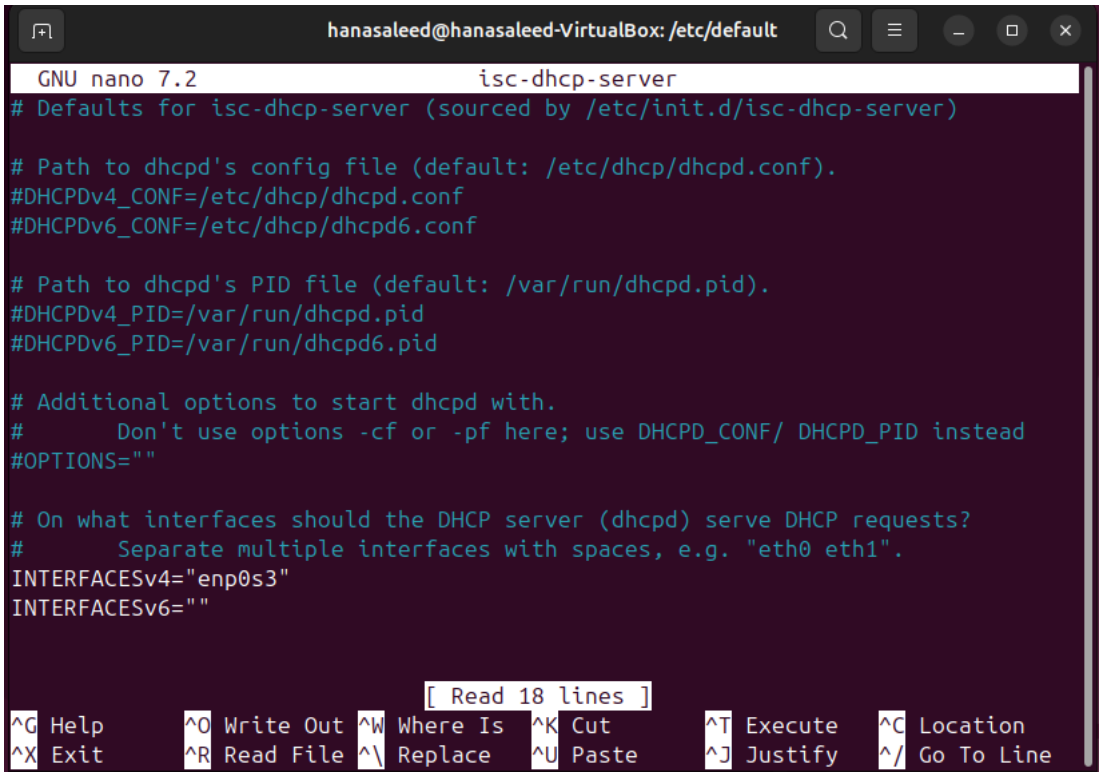




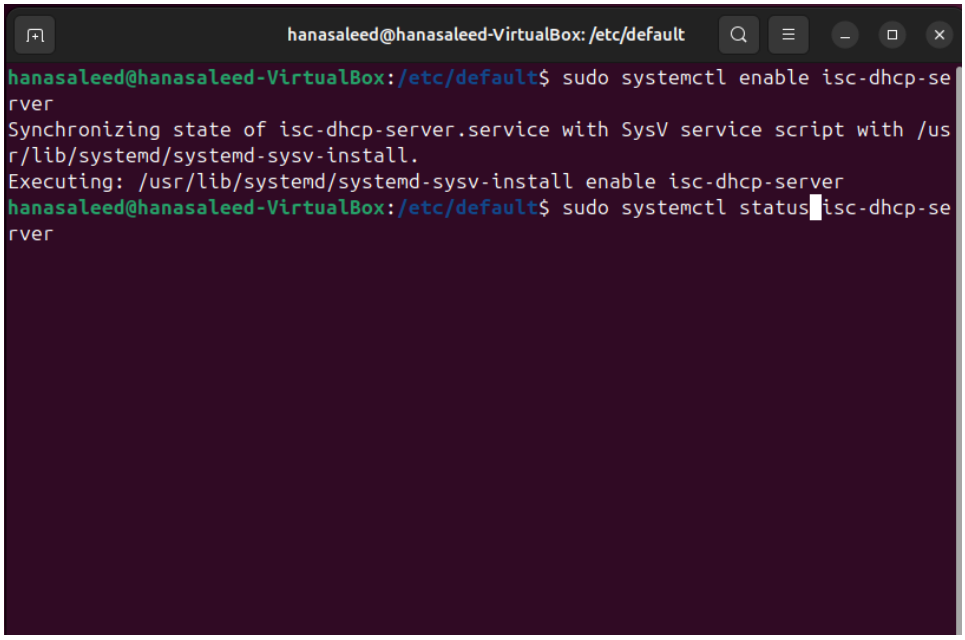
To assign the DHCP server on a network interface that it will be running on the DHCP file has to be opened by navigating **cd /etc/default/** and **sudo nano isc-dhcp-server**

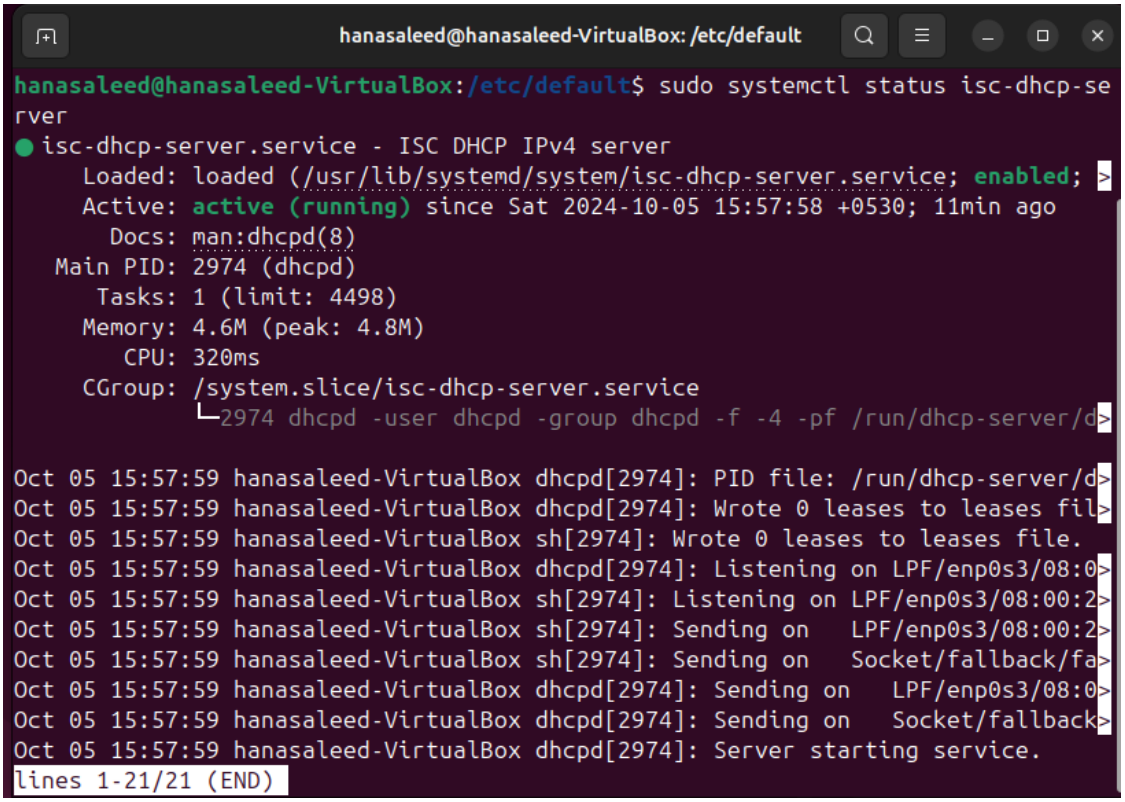


The empty ipv4 interface field should be filled with respect to the network interface available. Thereby, the network interface can be obtained by command **ip addr** and added it to the file. Here **enp0s3** has been chosen



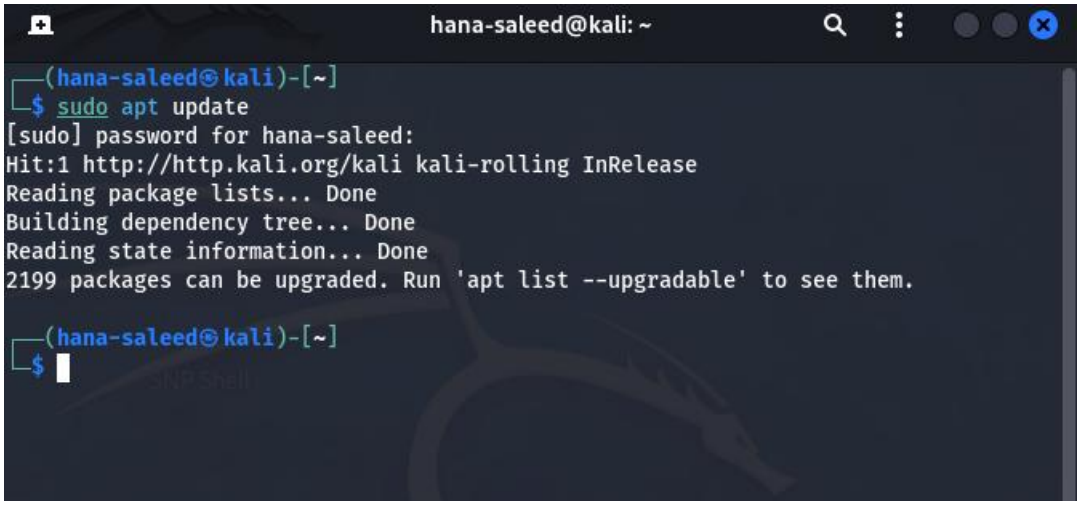
Then DHCP server should be enabled by **sudo systemctl enable isc-dhcp-server** and then restart by **sudo systemctl restart isc-dhcp-server.** Finally, the activation of the DHCP command be verified by **sudo systemctl status isc-dhcp-server.**

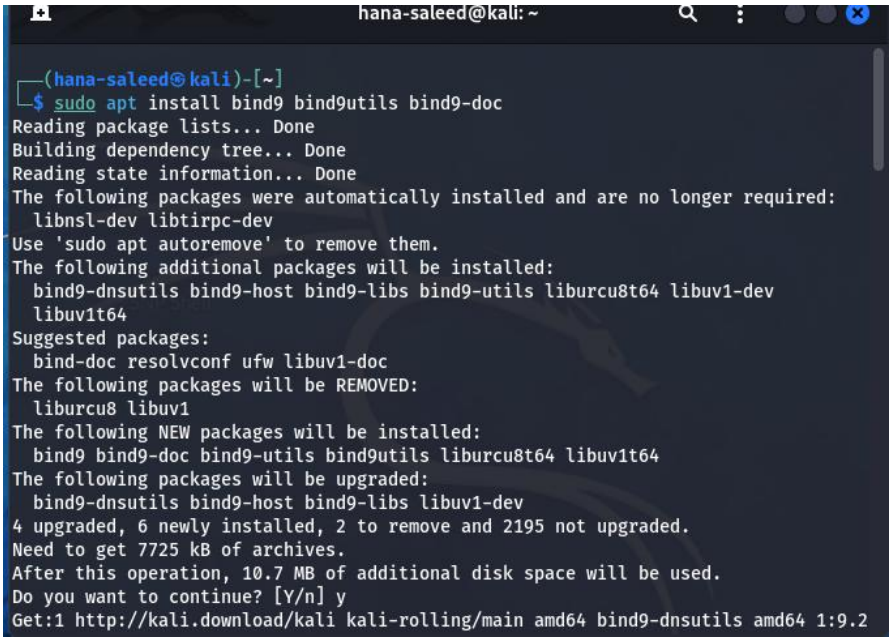



hanasaleed@hanasaleed-VirtualBox: /etc/default
hanasaleed@hanasaleed-VirtualBox:/etc/default$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled;
Active: active (running) since Sat 2024-10-05 15:57:58 +0530; 11min ago
Docs: man:dhcpd(8)
Main PID: 2974 (dhcpd)
Tasks: 1 (limit: 4498)
Memory: 4.6M (peak: 4.8M)
CPU: 320ms
CGroup: /system.slice/isc-dhcp-server.service
└2974 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/d
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: PID file: /run/dhcp-server/d
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Wrote 0 leases to leases fil
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Wrote 0 leases to leases file.
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Listening on LPF/enp0s3/08:0
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Listening on LPF/enp0s3/08:00:2
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Sending on LPF/enp0s3/08:00:2
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Sending on Socket/fallback/fa
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Sending on LPF/enp0s3/08:0
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Sending on Socket/fallback
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Server starting service.
lines 1-21/21 (END)

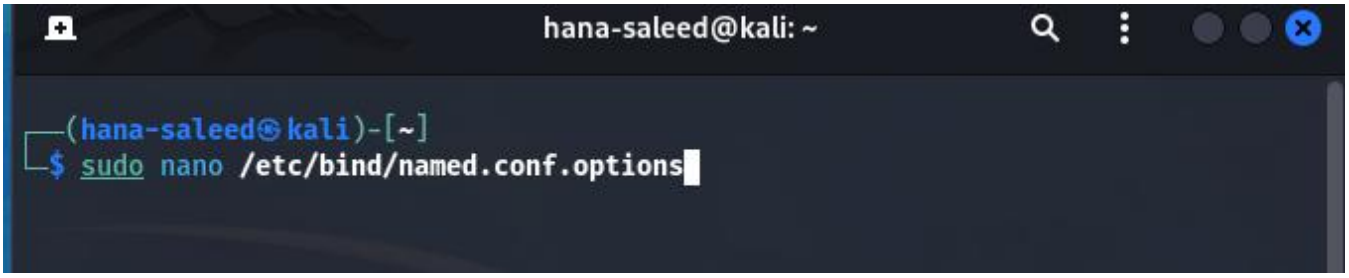
02.2 DNS (Domain Name System)

DNS is a naming system that translates human-readable domain names into IP addresses that computer identifies in the network. First, to install DNS server BIND (Berkeley Internet Name Domain) system has to be updated and BIND has to be installed.

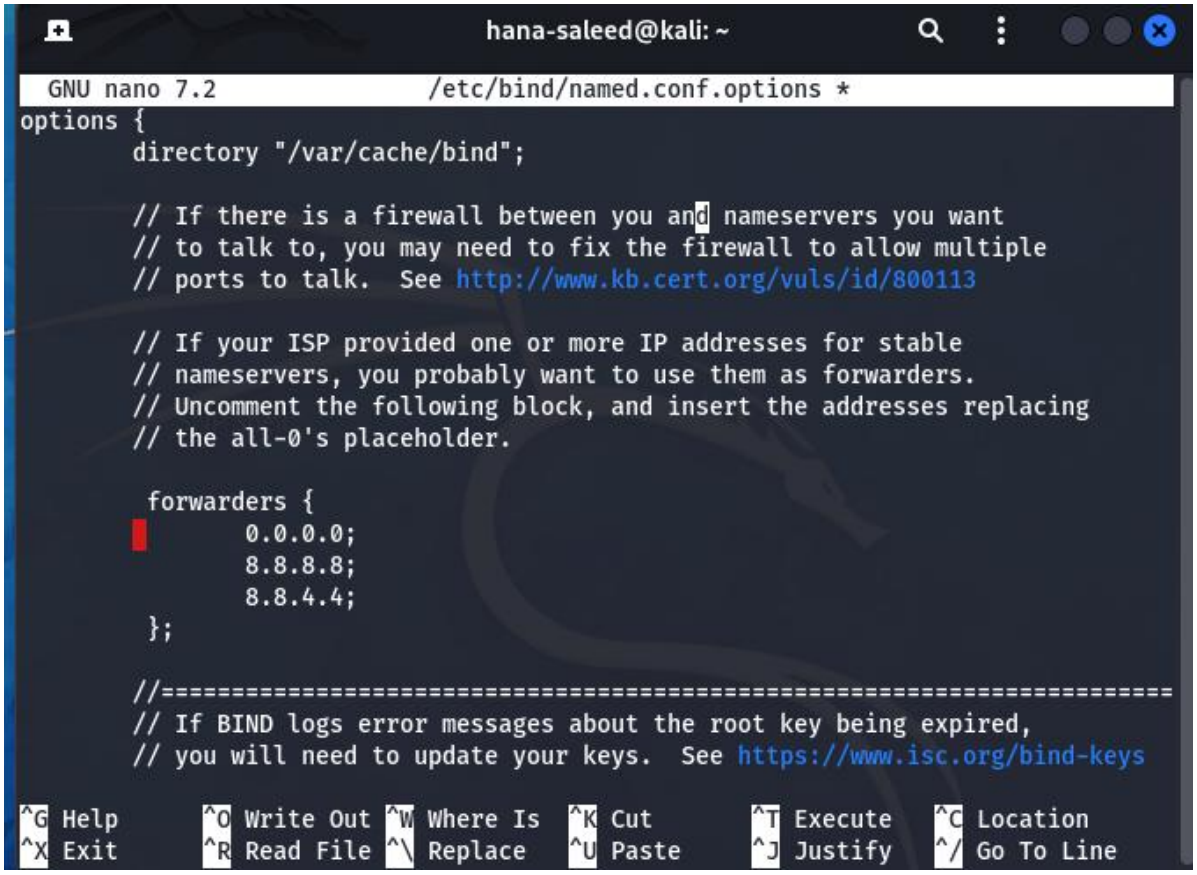




The main configuration file has to be edited by **sudo nano /etc/bind/named.conf.options**

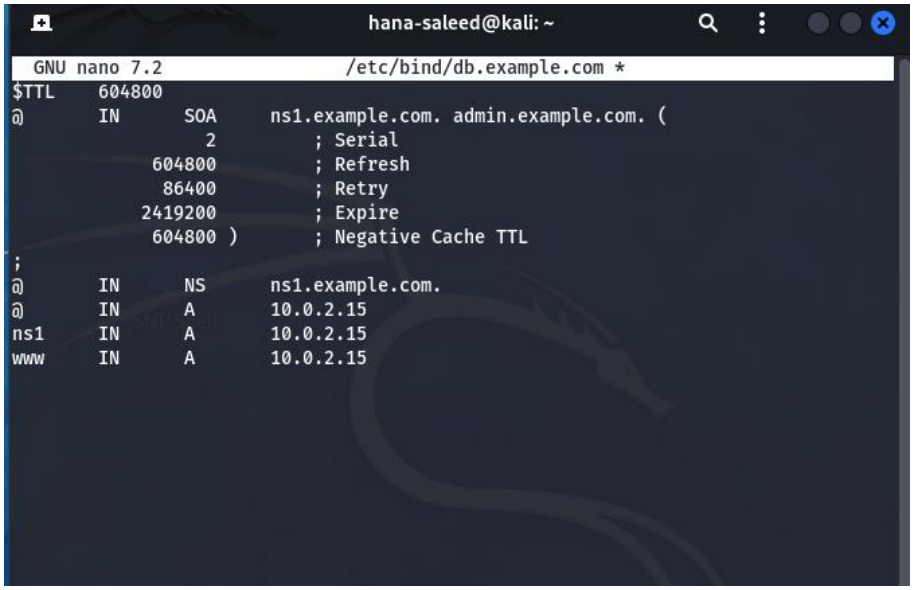


Thereafter, forwarders should be uncommented and set up a public server like google which has the DNS 8.8.8.8 and 8.8.4.4



The file should be saved and DNS zones should be edited in the file **named.conf** by the command **sudo nano /etc/bind/named.conf.local**

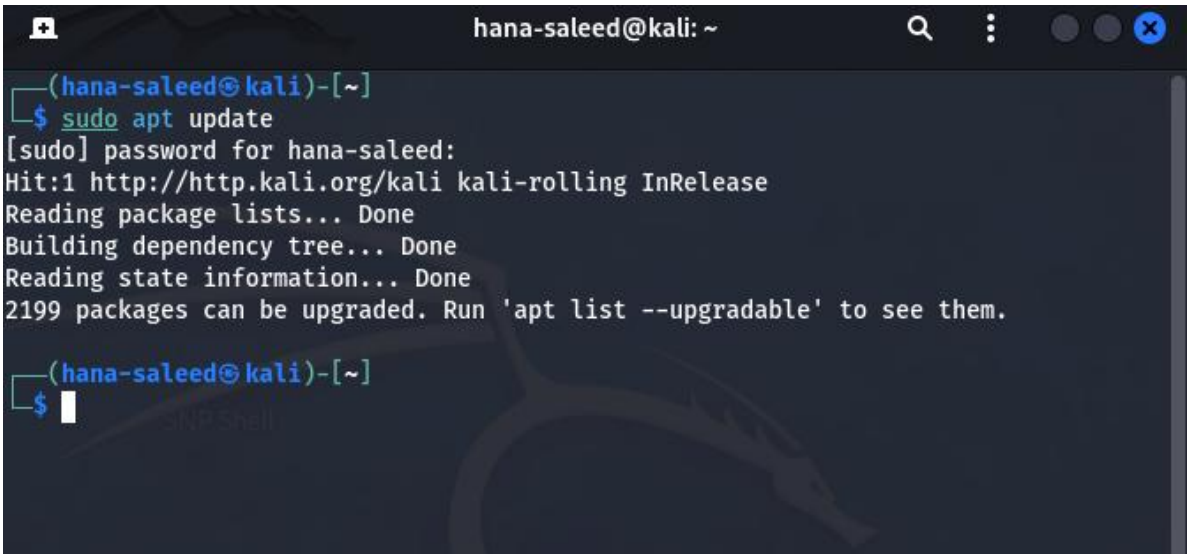
We should navigate to /etc/bind/db.example.com , edit the file as follow and check for syntax error

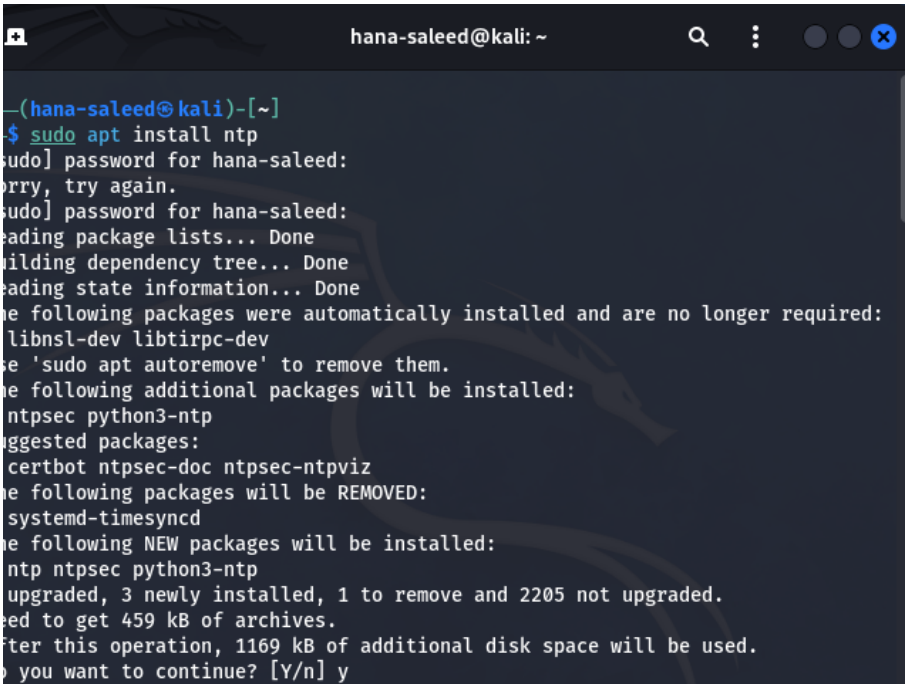


02.3 NTP (Network Time Protocol)

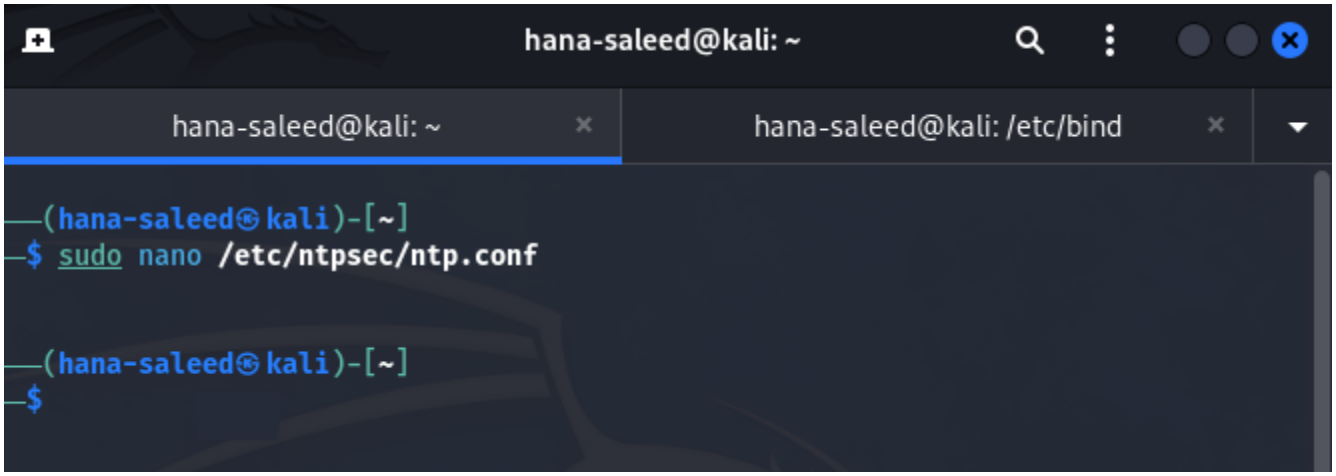
Network Time Protocol (NTP) is an internet protocol used to synchronize with computer clock time sources in a network. NTP helps ensure that all devices on a network maintain accurate and consistent time, which is essential for logging events, scheduling tasks, and coordinating activities across distributed systems.

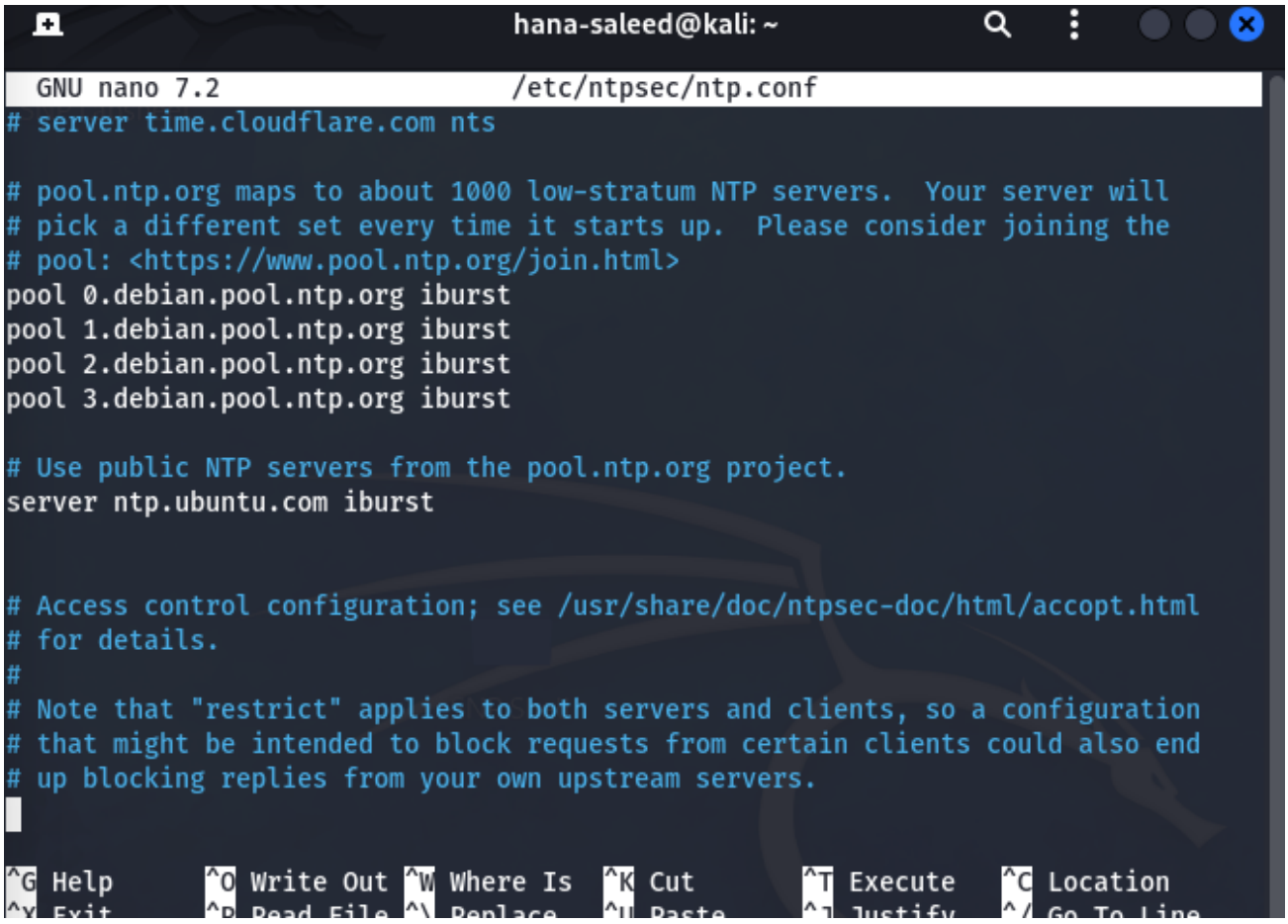
First update the package and install ntp





The ntp file has to be opened using the nano command it has to be updated as follows





Then the service should be started and the enabled and then we can check the status

```
└─$ sudo systemctl status ntp

● ntpsec.service - Network Time Service
     Loaded: loaded (/usr/lib/systemd/system/ntpsec.service; enabled; preset: dis>
     Active: active (running) since Sun 2024-10-06 18:32:32 +0530; 2h 43min ago
 Invocation: b6d3d4dca0214d4fbb6fbabbedb5dbe8
       Docs: man:ntpd(8)
   Main PID: 4371 (ntpd)
      Tasks: 1 (limit: 3529)
     Memory: 12M (peak: 12.7M)
        CPU: 3.450s
     CGroup: /system.slice/ntpsec.service
             └─4371 /usr/sbin/ntpd -p /run/ntpd.pid -c /etc/ntpsec/ntp.conf -g -N>

Oct 06 20:15:22 kali ntpd[4371]: DNS: dns_probe: 3.pool.ntp.org, cast_flags:1, fl>
Oct 06 20:15:22 kali ntpd[4371]: DNS: dns_check: processing 3.pool.ntp.org, 1, 20>
Oct 06 20:15:22 kali ntpd[4371]: DNS: Server skipping: 162.159.200.1
Oct 06 20:15:22 kali ntpd[4371]: DNS: Server skipping: 162.159.200.123
Oct 06 20:15:22 kali ntpd[4371]: DNS: dns_take_status: 3.pool.ntp.org=>good, 11
Oct 06 20:49:31 kali ntpd[4371]: DNS: dns_probe: 3.pool.ntp.org, cast_flags:1, fl>
Oct 06 20:49:31 kali ntpd[4371]: DNS: dns_check: processing 3.pool.ntp.org, 1, 20>
Oct 06 20:49:31 kali ntpd[4371]: DNS: Server skipping: 162.159.200.123
Oct 06 20:49:31 kali ntpd[4371]: DNS: Server skipping: 162.159.200.1
Oct 06 20:49:31 kali ntpd[4371]: DNS: dns_take_status: 3.pool.ntp.org=>good, 11
lines 1-22/22 (END)
```

To check the status and the servers we have synchronized with we can use the command **ntpq -p**

```
hana-saleed@kali: ~

┌──(hana-saleed㉿kali)-[~]
└─$ ntpq -p
     remote           refid      st t when poll reach   delay   offset   jitter
==============================================================================
 162.159.200.123  10.111.8.4       3 u 1367  512    0    0.0000   0.0000   0.0000
 162.159.200.1    10.111.8.4       3 u 1427  512    0    0.0000   0.0000   0.0000
 2606:4700:f1::123 .DNS6.          16 u    -  1024    0    0.0000   0.0000   0.0001
 3.pool.ntp.org   .DNS.            16 u    -  2048    0    0.0000   0.0000   0.0001
+LOCAL(0)         .LOCL.            5 l   14   64  377    0.0000   0.0000   0.0000

┌──(hana-saleed㉿kali)-[~]
└─$
```

03. Shell Scripting and Security

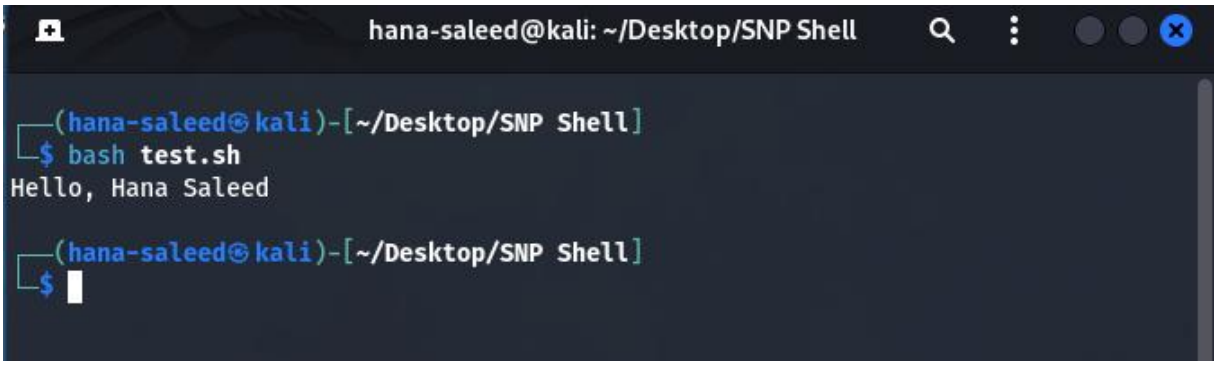
03.1 Shell Scripting

i. Basic Shell scripting syntax

Every shell script begins with a special character sequence **#!** (**shebang).** This tells the system which interpreter should be used to sun the script. That follows a specific path to the Bash Interpreter **/bin/bash.** Without shebang, the script will run using the default shell of the user executing it which may lead to unexpected behavior.

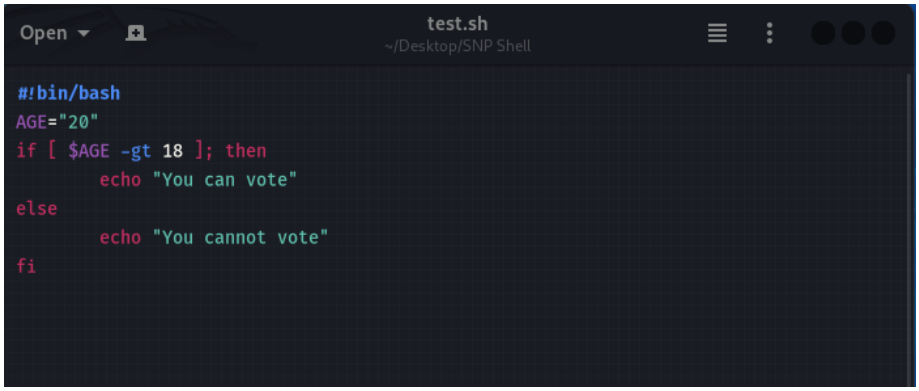
Comments in shell scripts start with the **#** symbol. Anything after the # in the line is ignored by the shell interpreter. Multi-line comments are created by putting **#** at the start of each line

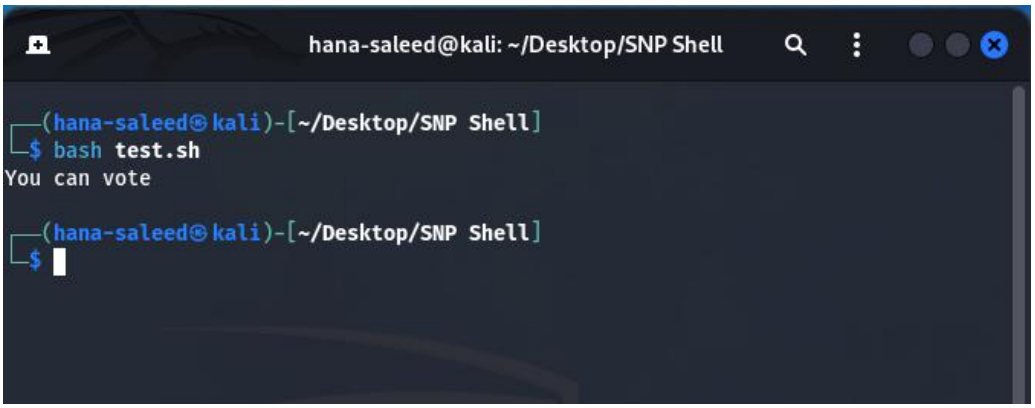
Variables in shell scripts are declared by assigning a value to a name without spaces around the equal sign. They are also case sensitive and every variable is treated as a string. In order to reference the variable a prefix **$** sign is used for the variable name. Without the $ sign the variable name is treated as a literal text. Moreover, **echo** can be used to print output to the console.



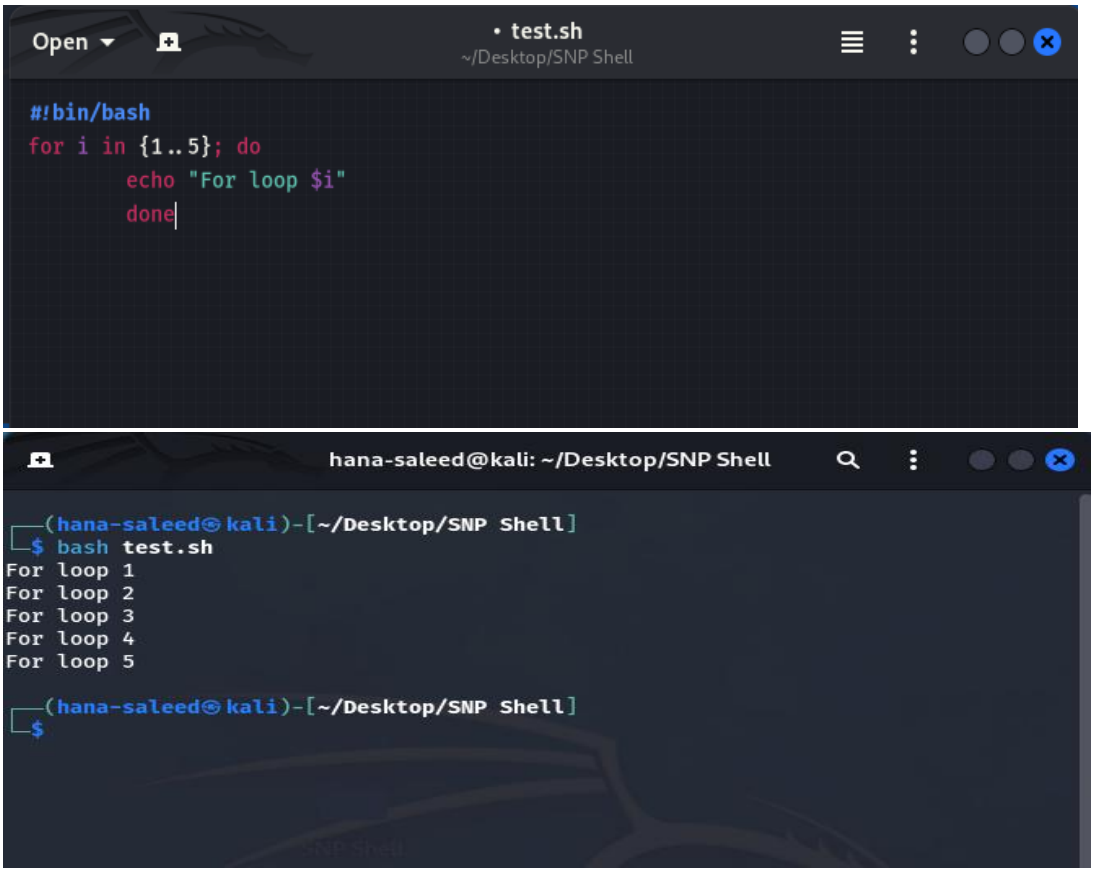
ii. Control flow statements

a. if command





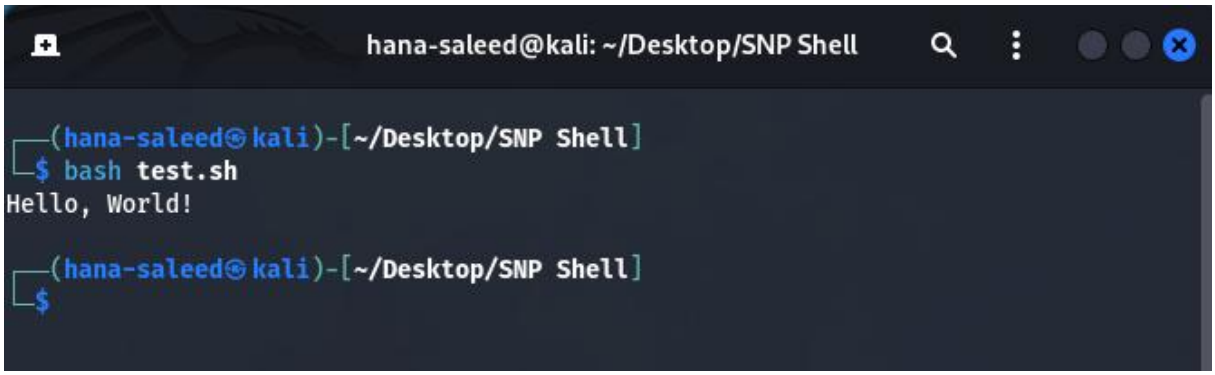
b. For command



C. While command

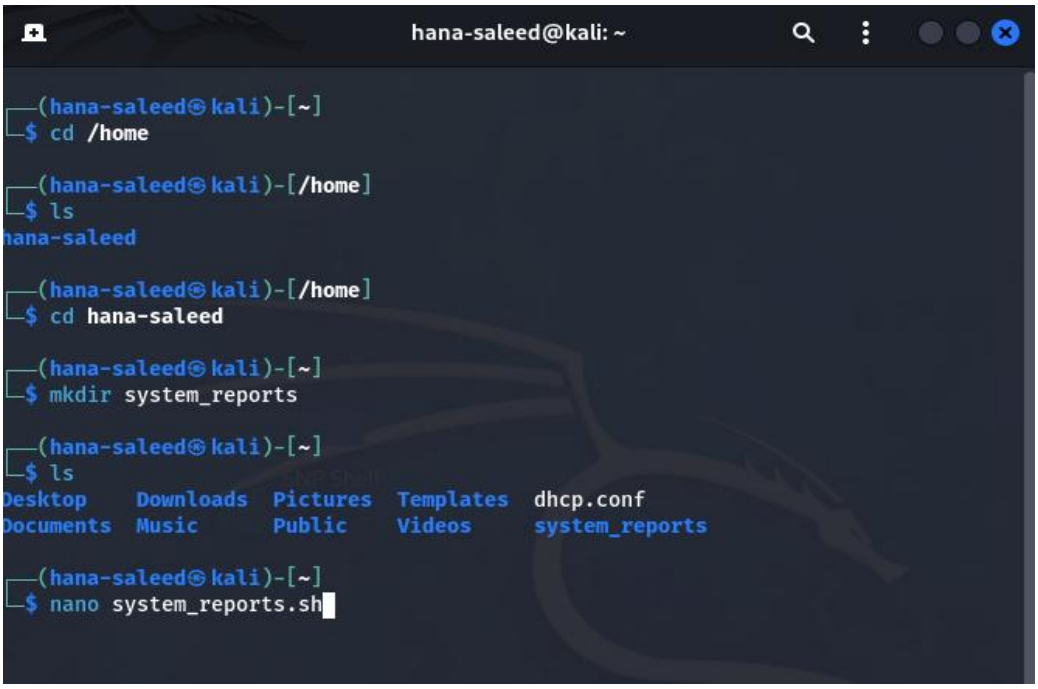
```
┌──(hana-saleed㉿kali)-[~/Desktop/SNP Shell]
└─$ bash test.sh
While loop: 1
While loop: 2
While loop: 3
While loop: 4
While loop: 5

┌──(hana-saleed㉿kali)-[~/Desktop/SNP Shell]
└─$
```

d.Functions



iii. Automating Daily System Reports

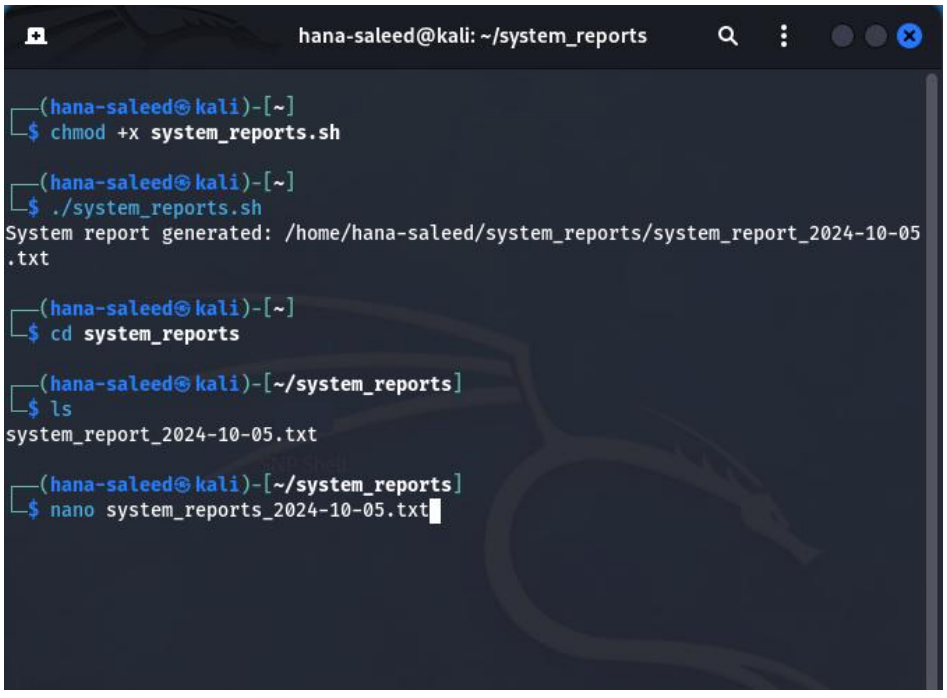
Automating daily system reports involves creating a script which collects key system information such as date, uptime, memory usage and disk storage. The script automatically runs on scheduled at specific intervals using cron jobs. First, we have to navigate to the destination directory given **/home/user/system_reports** using the command **cd**. A directory has to be creaded named **system_reports** by the command **mkdir.** Thereafter, a shell script has to be created as **system_reports.sh** by using the command **nano** text editor.



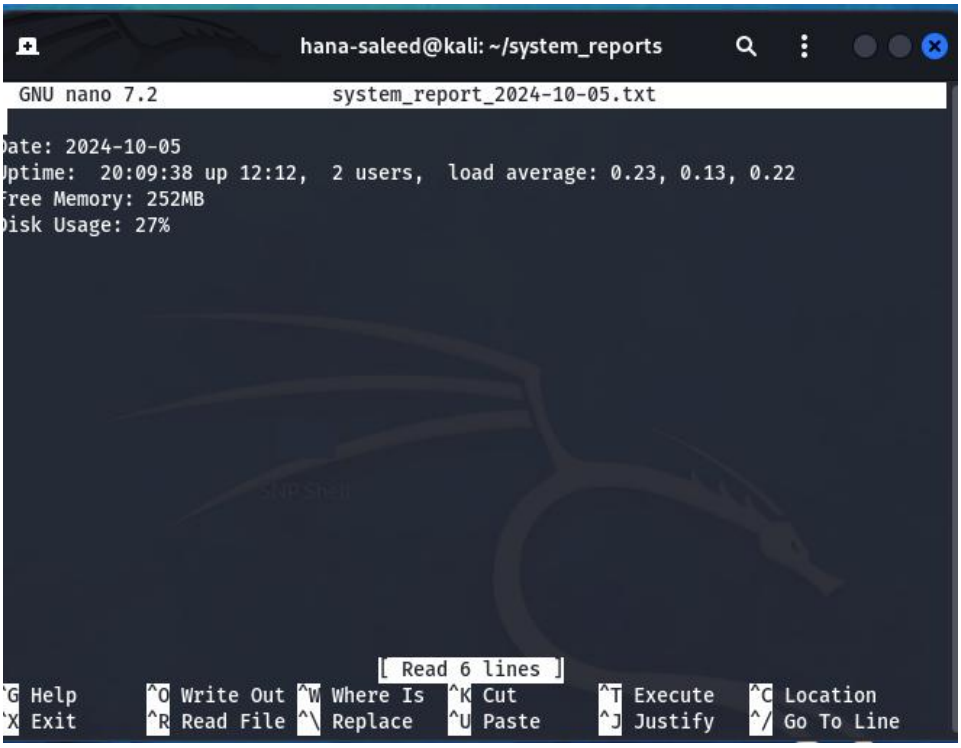
The command will create open a file and then the file should be edited as the following script

- **#!/bin/bash :** This tells the system to use the Bash shell to interpret the script
- **destination="/home/hana-saleed/system_reports" :** This sets the path where the report will be saved
- **date=$(date +"Y-%m-%d") :** The **date** gets the current date in YYYY-MM-DD format. The $ is a syntax used to execute the command and it stores its output in the variable **date.**
- **uptime=$(uptime) :** This command captures how long the sytem has been running since the last boot
- **free_mem=$(free -m | awk 'NR==2{print $4}') :** The command **free -m** shows the memory usage in megabytes. **N==2** prints the fourth line column from the precessed output which shows the free memory available. This value will be stored in the variable **free_mem**
- **disk_usage=$(df -h / | awk 'NR==2{print $5}') : df -h** command outputs disk space usage in human readable form.
- **mkdir -p $destination :** Creates the directory if it does not already exist. The **-p** ensures no error is show if the directory exist
- **report_file="$destination/system_report_$date.txt" :** Creates a filename of the report with the date
- **echo "$report_content" > $report_file :** This will output the contents in the report to the file

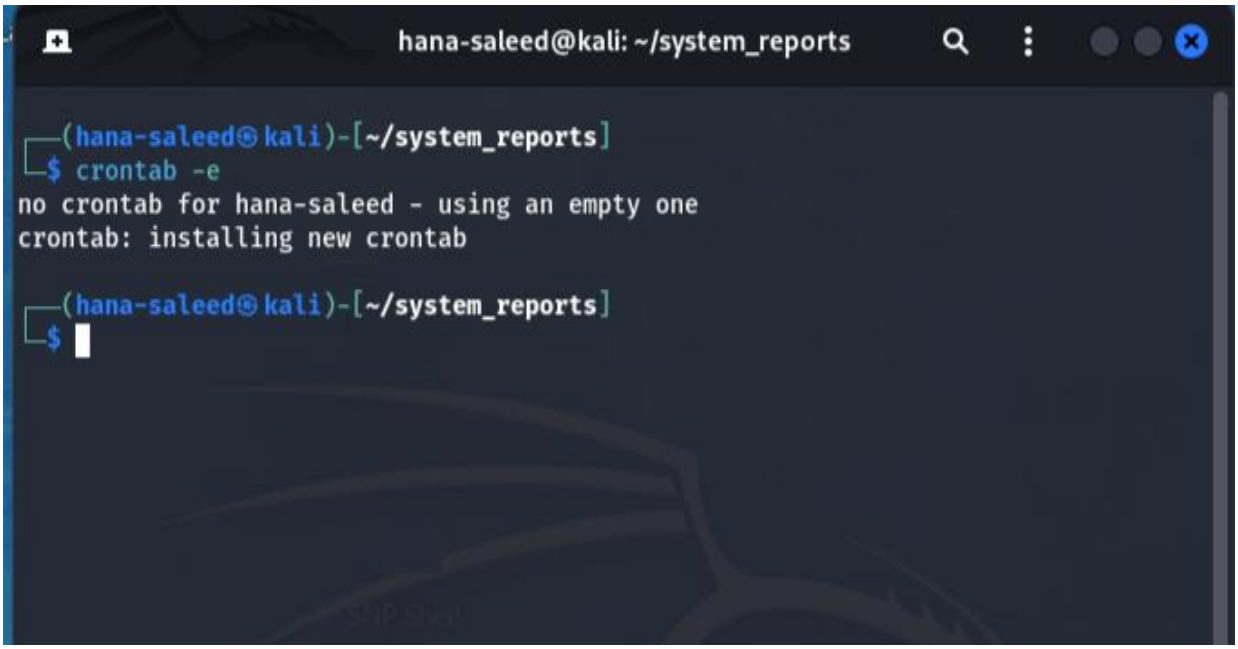
After the script is edited it has to be saved and exited. Then, permission has to be enabled to make it executable and the system has to run by the command **./system_reports.sh.** This will display the directory of report generated. The file with date has to be opened by the command nano.



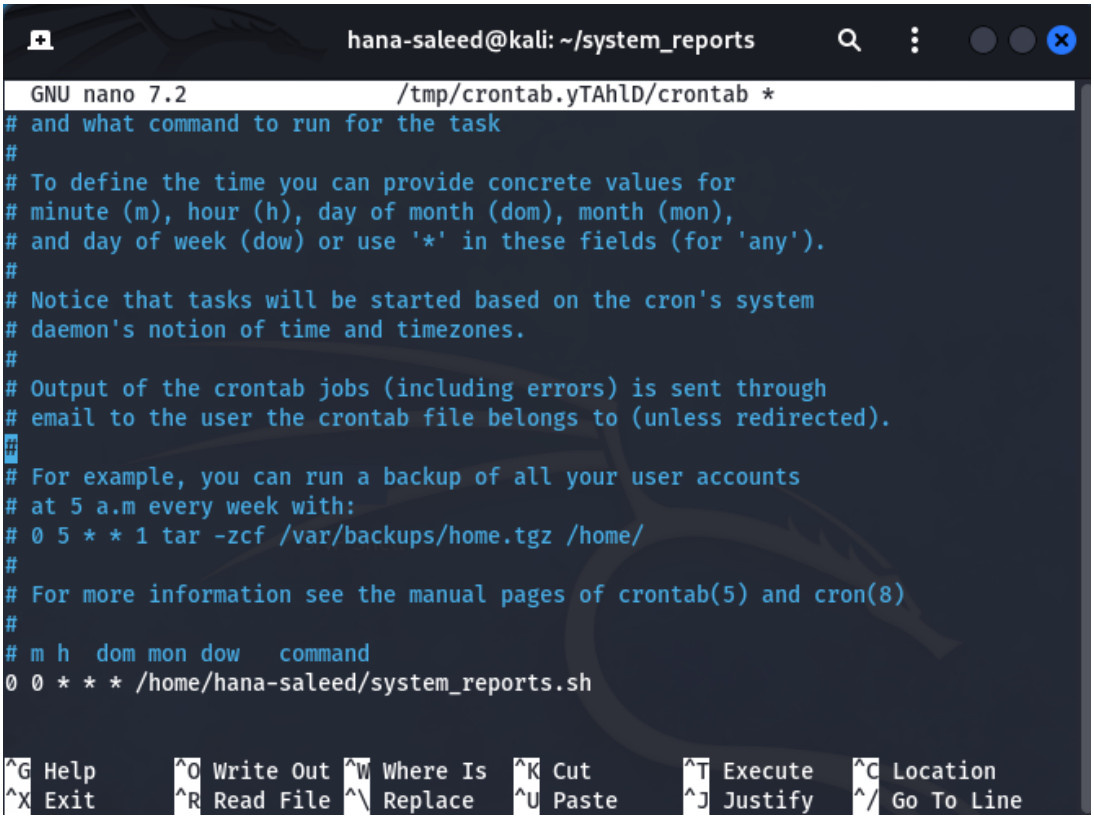
After the execution of the previous command, it will output the report as



To schedule it to run daily, cron can be used. It can be opened with the command



The above command will open a file and it has to be edited in a way that it execute the script every midnight, generating a report with system details in the specific directory.

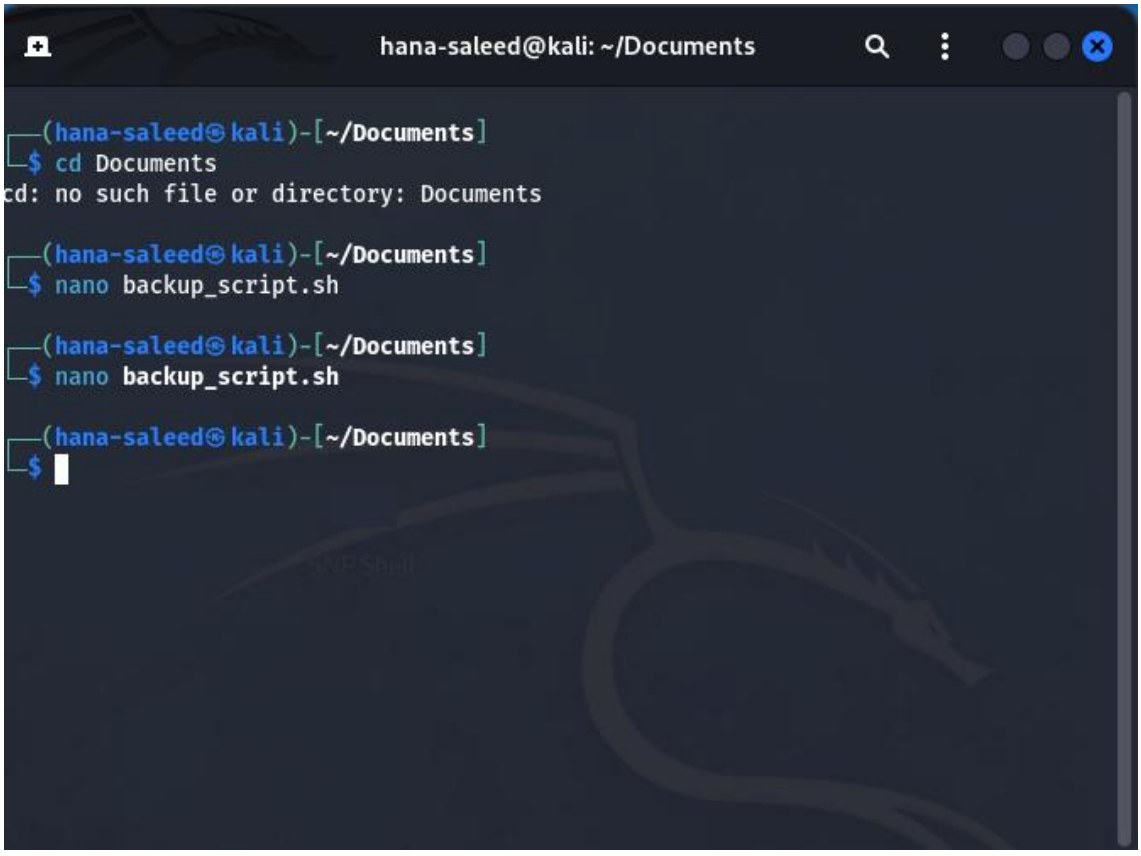


The line **0 0 * * * /homr/hana-saleed/system_reports.sh** has to be added to the file

iii. Automating Backup Of Critical Files

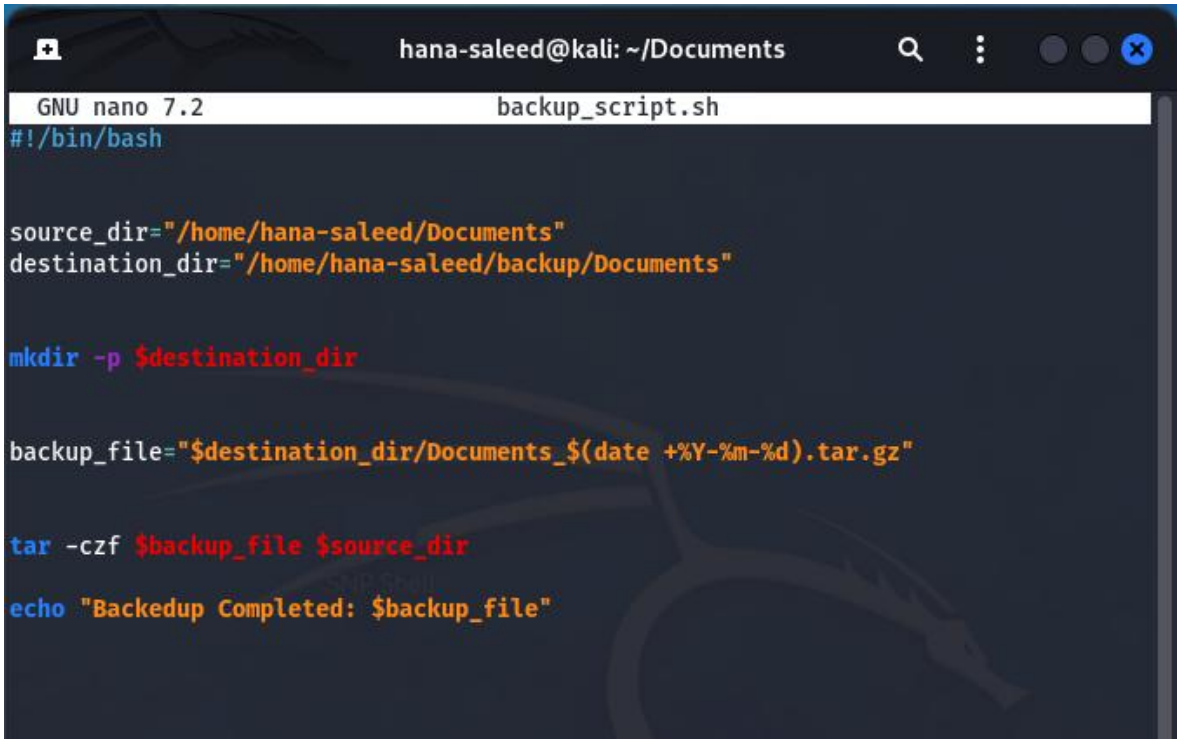
This shell scripting is to ensure that important files are regularly copied and stored safely. In this case it will be stored in the directory /**home/user/documents.** The script can be scheduled to run periodically by using cron jobs. This helps with data loss.

The directory has to be changed using the **cd** command and open a new script under the name **backup_scrip.sh** by the command **nano**

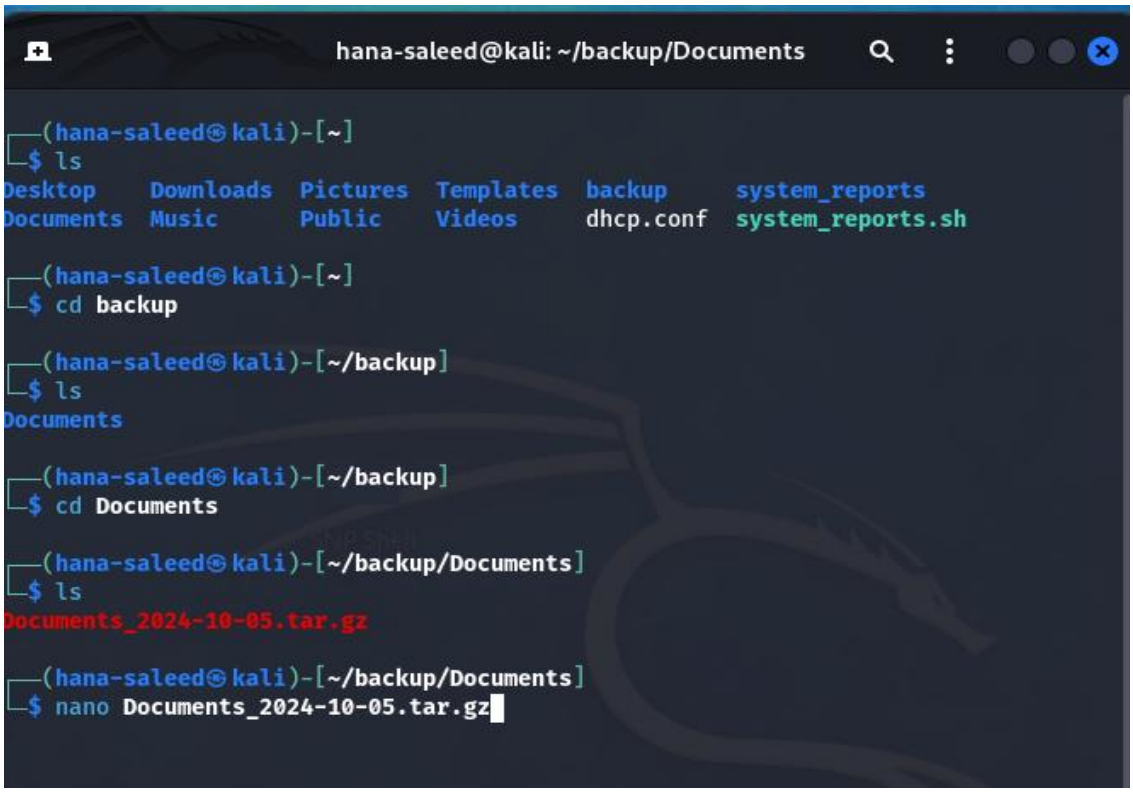


This will open the new script file and it has to be edited as such:

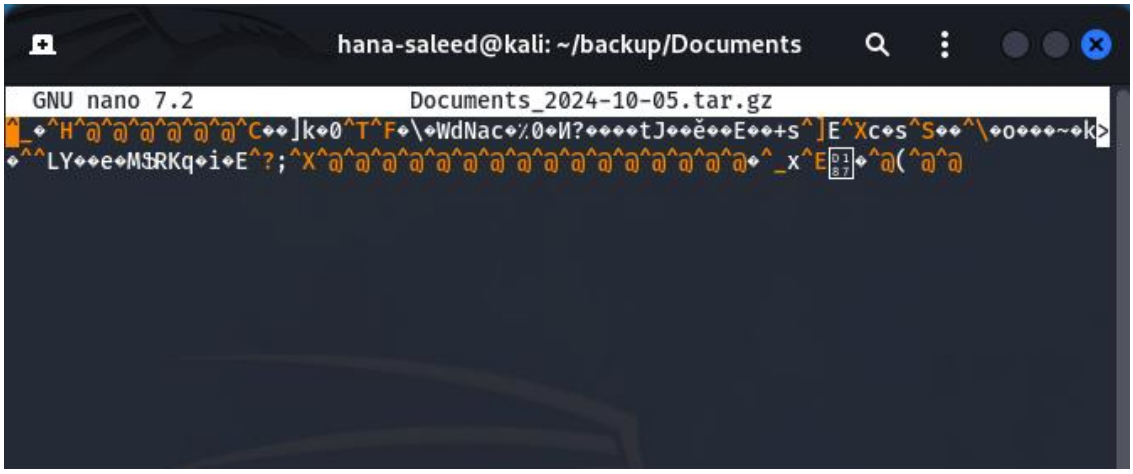
- **DATE=$(date '+%Y-%m-%d') :** This command will get the date in YYYY-MM-DD format
- **BACKUP_DIR="/home/user/documents":** Variable assigned to get the directory that needs to be backed up.
- **DEST_DIR="/home/user/backups":** This command is to assign the destination directory where the backp up will be save
- **BACKUP_FILE="backup_$DATE.tar.gz":** Creates the file name of the backup file with the current date
- **mkdir -p $DEST_DIR:** Creates the destination directory
- **tar -czvf $DEST_DIR/$BACKUP_FILE $BACKUP_DIR:** Command that creates the backup



The file has to be given permission to make it executable and run using **./backup_script.sh** command. Then the backup path should be followed

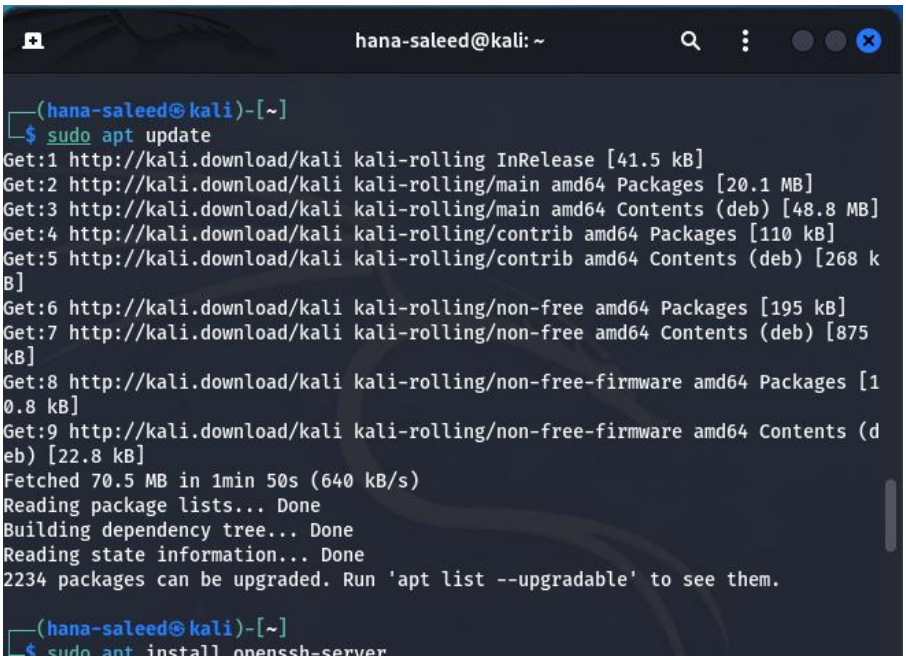


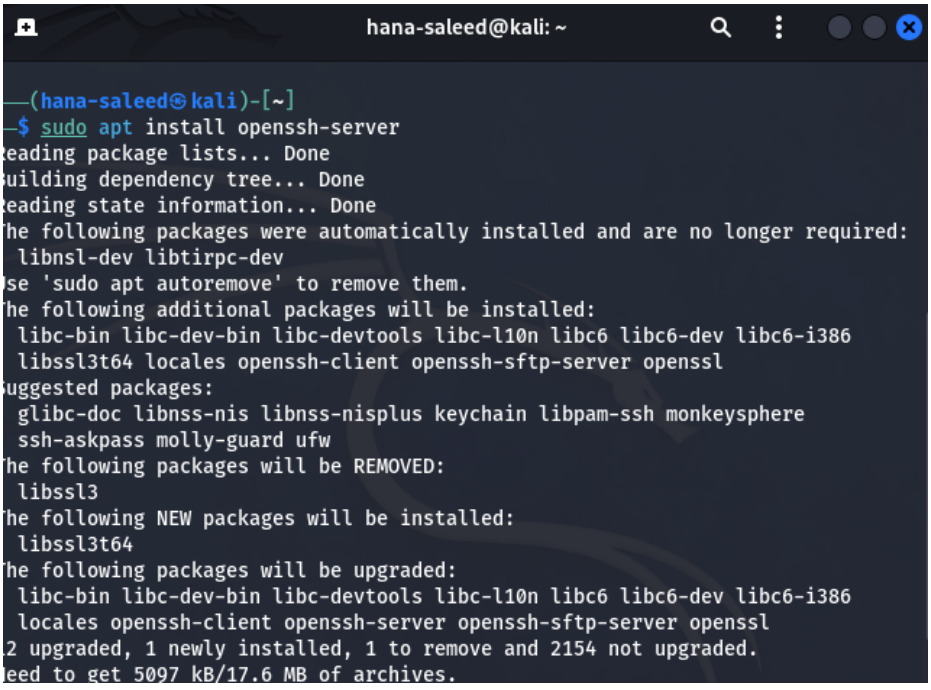
The file can be opened with the date can be opened with the command **nano** and the view of the backup file be as



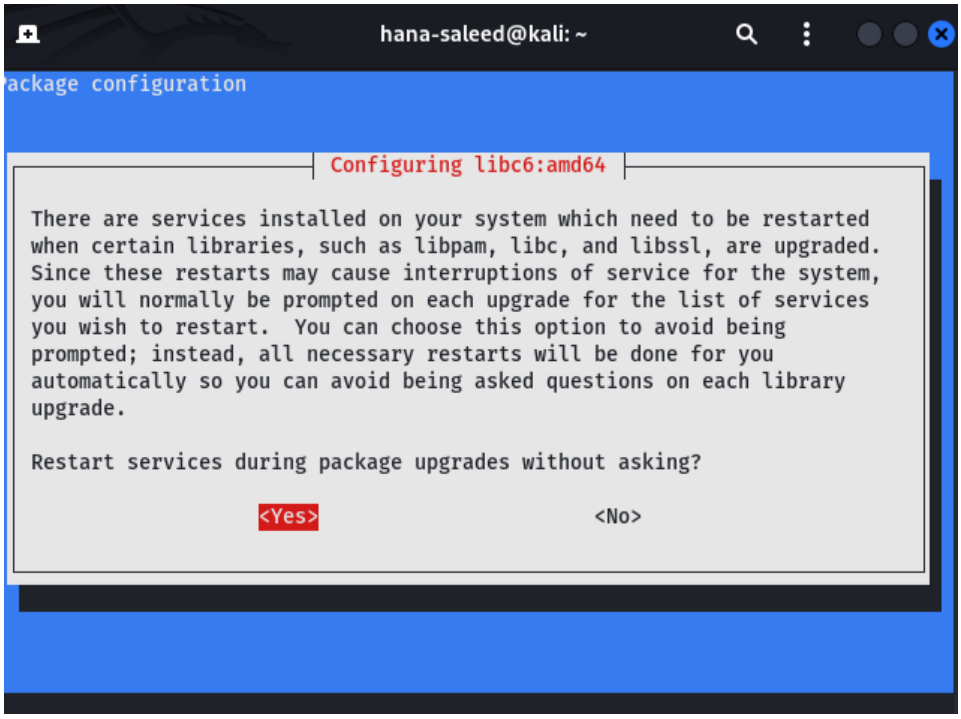
03.2 SSH (Secure Shell)

SSH is a cryptographic network protocol that allows you to securely communicate with a remote system. It provides a secure way to access and manage remote devices over an unsecured network. Initially, kali has to be updated by **sudo apt update** and OpenSSH server package has to be installed by **sudo apt install openssh-server**

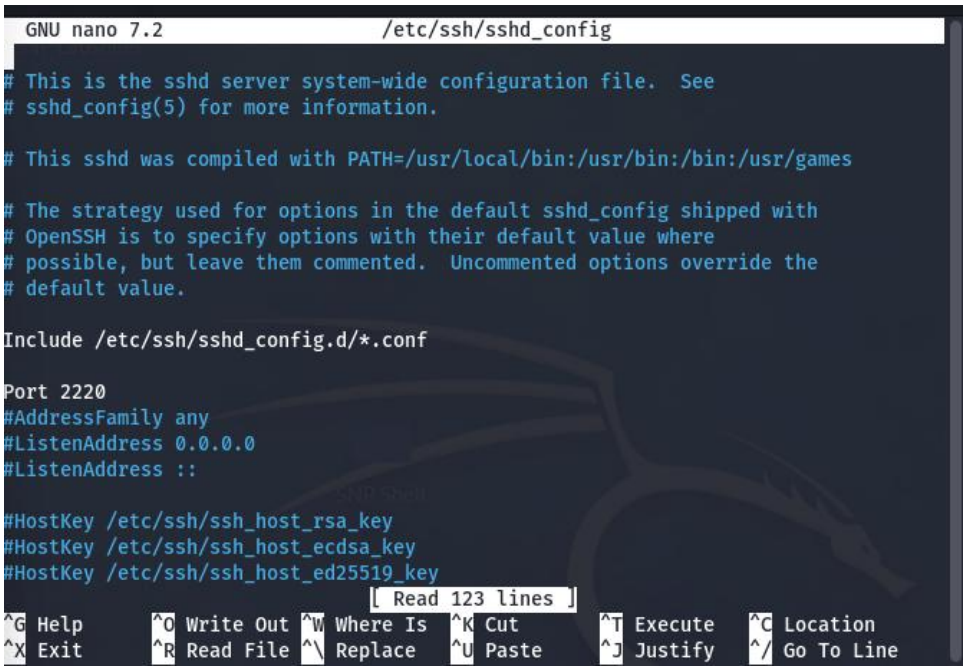




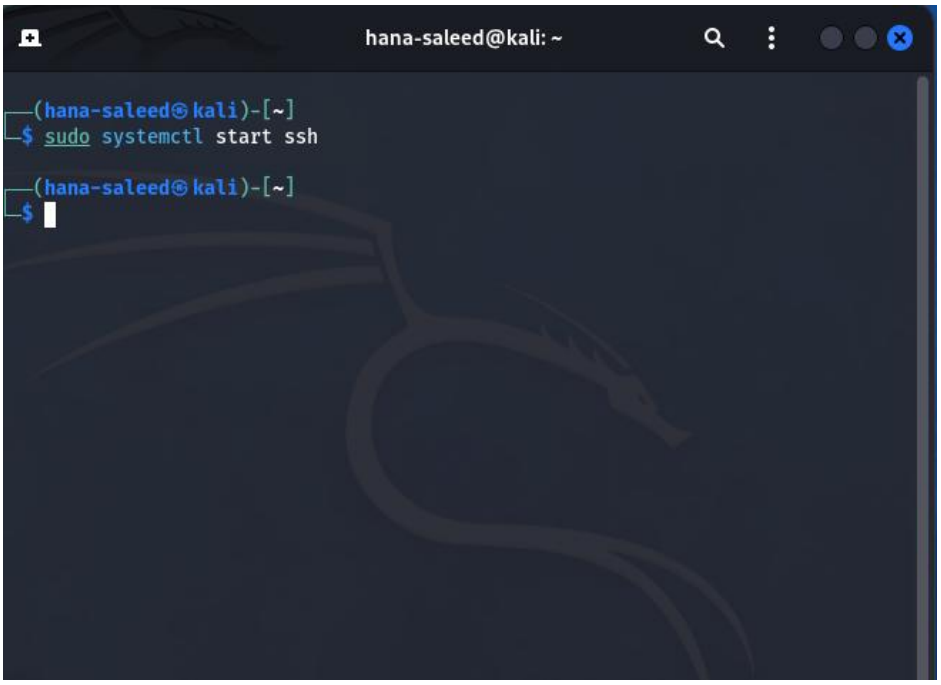
After the installation it requests to restart and confirm by giving yes.



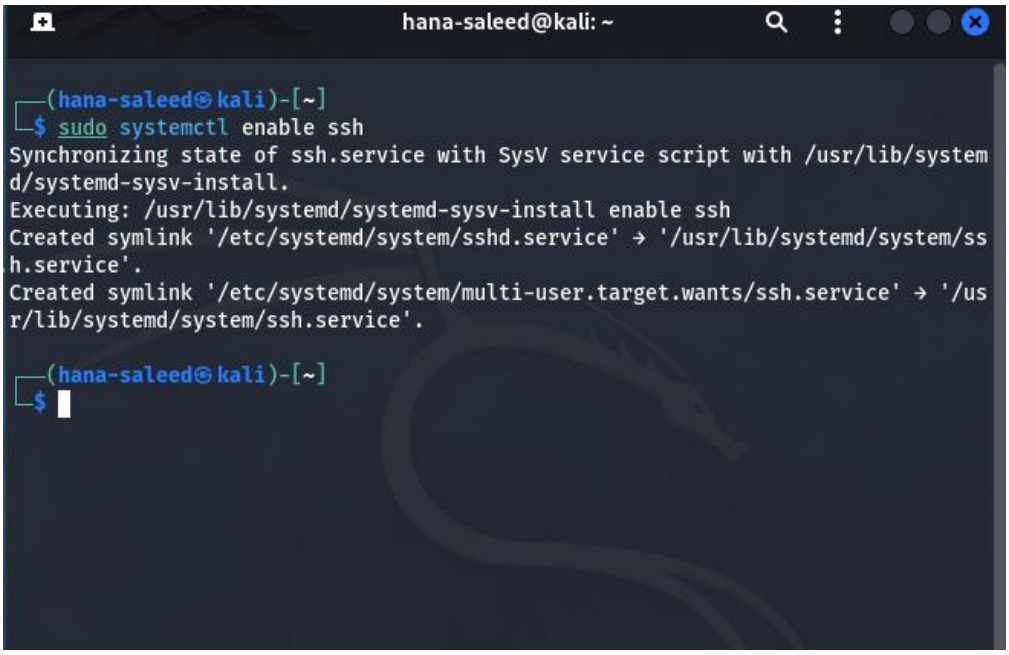
Navigate to **/etc/ssh/sshd_config** and add Port 2220



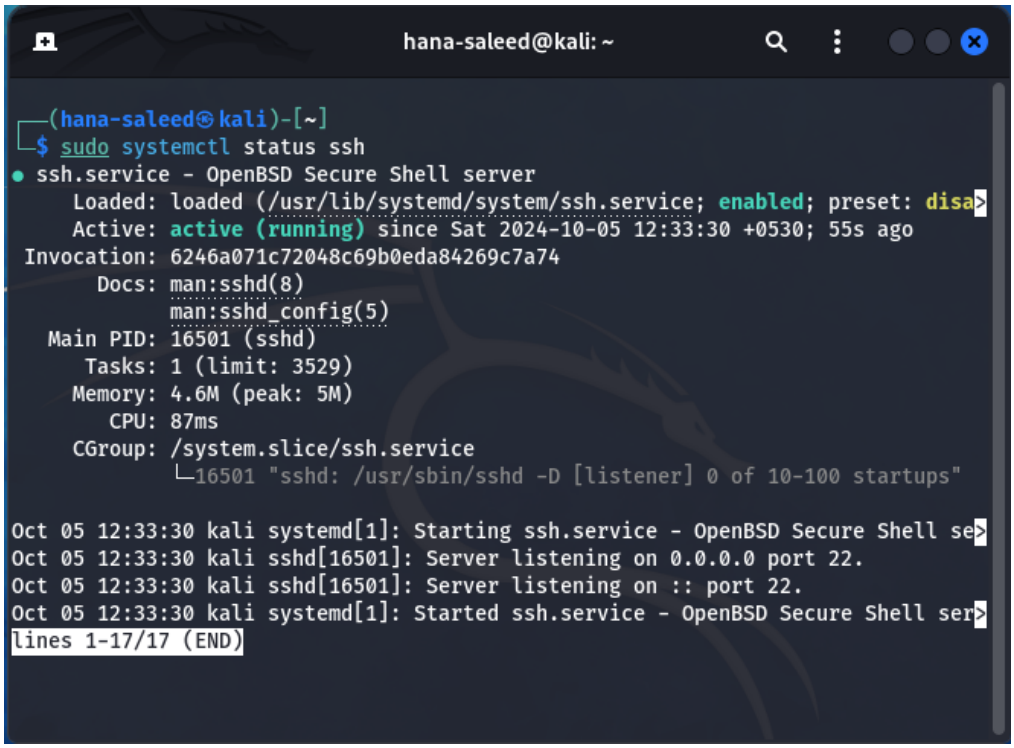
Starting SSH launches the service, allowing connections to the service. If SSH is not started, no remote user can access the machine securely.



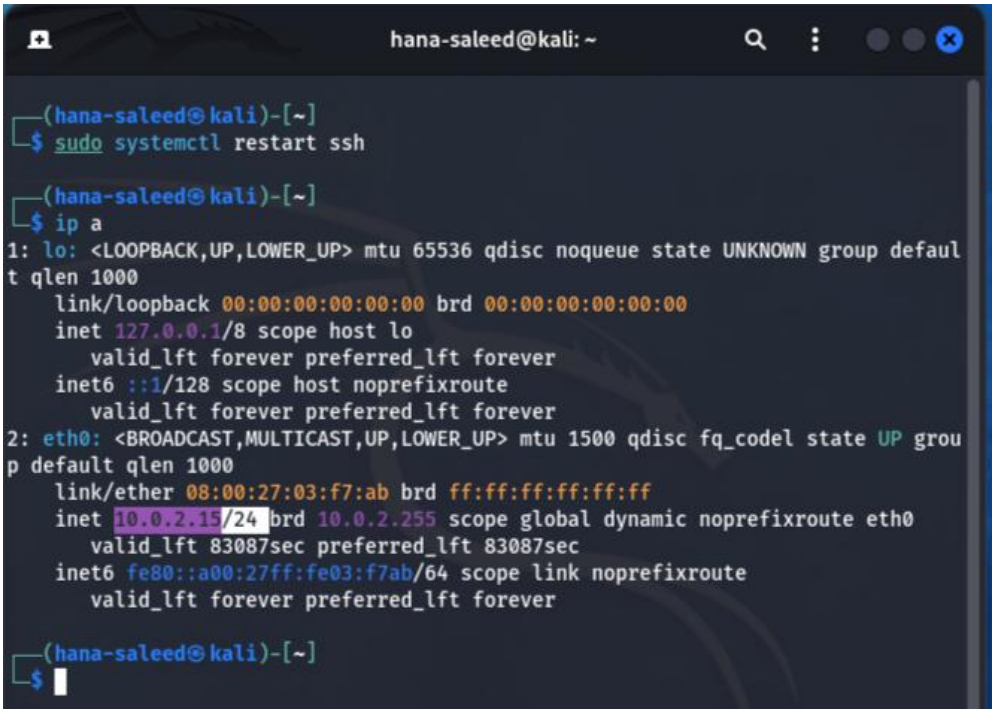
To ensure that the SSH service starts automatically every time the system boots, **sudo sysemctl start ssh** has to be run.



After the activation of the server the status of SSH has to be checked



The ip address of kali is needed. Therefore, to find this information the command **ip a** has to run



To connect from a client machine the command (windows)

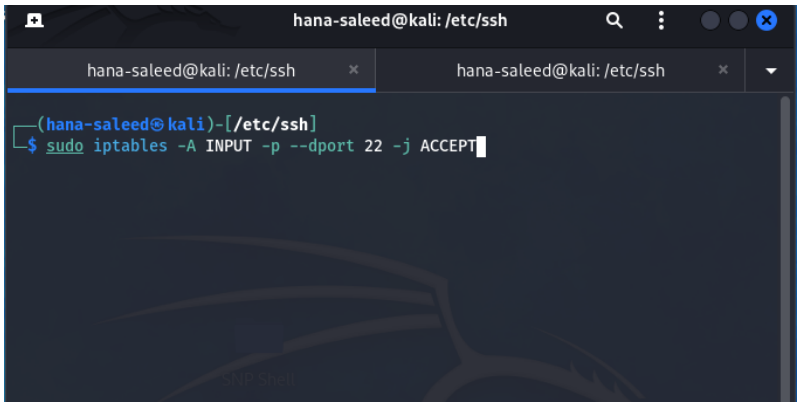
```
PS C:\Users\DELL> ssh hana-saleed@10.0.2.15 -p 2220
ssh: connect to host 10.0.2.15 port 2220: Connection timed out
```

03.3 iptables and ACLs

Iptables is a command-line firewall utility that uses policy chains to allow or block traffic that will be enforced by linux kernels netfilter framework. Iptables packet filtering mechanism is organized into three different kinds of structures:

- DefaultTables:
  a. Filter : The default table for most filtering operations
     - Input – This chain handles all packetes that are destined to te server and also controls the behaviour for incoming connections.
     - Forward – This chain is used for packets routed through the system.
     - Output – This chain contains rules for packet generated locally.
  b. NAT (network Address Translation)
     - Prerouting – This chain rule alters a packet as soon as it's received.
     - Postrouting – This chain rule alters a packet as it is about to go out
     - Output – This chain rule alerts locally generated traffic.
  c. Raw : This table is used to exempt packets from connection tracking
     - Output – To alter locally generated packets.
     - Prerouting – to alter incoming connections.
  d. Mangles : This table adjusts the IP header properties of packets
     - Input – for incoming packets
     - Output – To alter locally generated packets.
     - Forward =- for packets routed through the linux box
     - Prerouting – To alter incoming connections
     - Prostrouting – To alter outgoing connecteions.

- Basic iptables commands syntax:
  a. **sudo iptables -L :** List current rules**.**
  b. **sudo iptables -A <CHAIN> <CONDITIONS> -j <TARGET> :** Add a rule



c. **sudo iptables -D <CHAIN> <CONDITIONS> :** Delete a rule

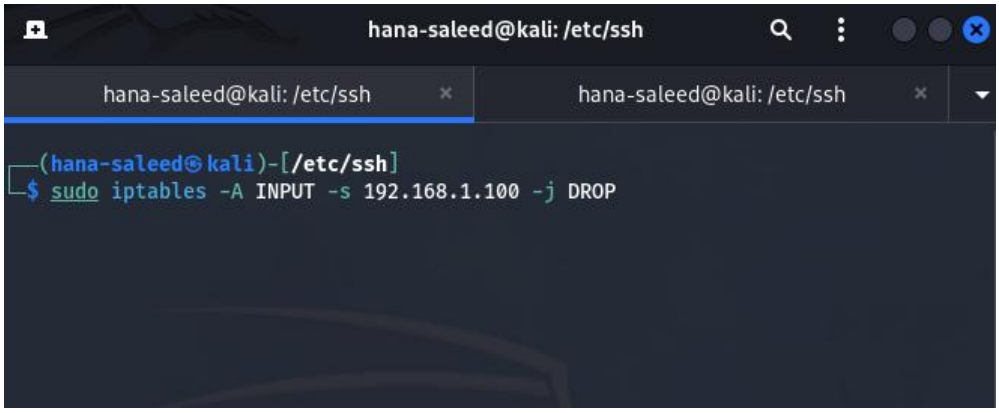
d. **sudo iptables-save > /etc/iptables/rules.v4 :** Save the rules

❖ Basic Firewall Rules

a. Allow specific traffic on a port

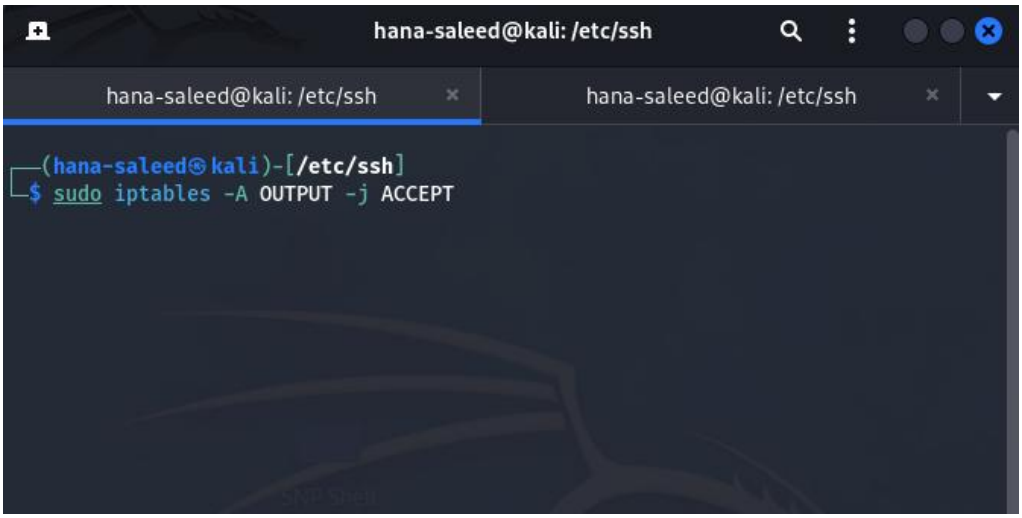
This command allows incoming TCP traffic on port 22 (SSH) through firewall.

b. Block traffic from a specific IP address.



This example will block the incoming IP address 192.168.1.100

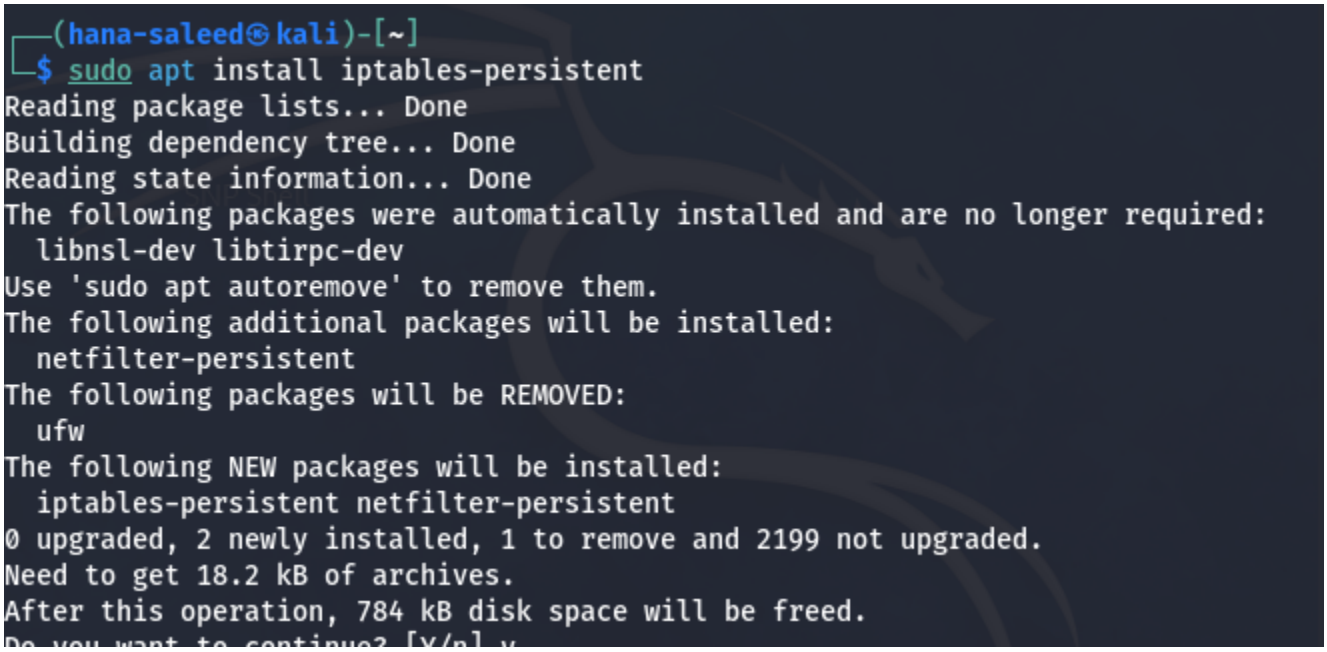
c. Allow all outgoing traffic.



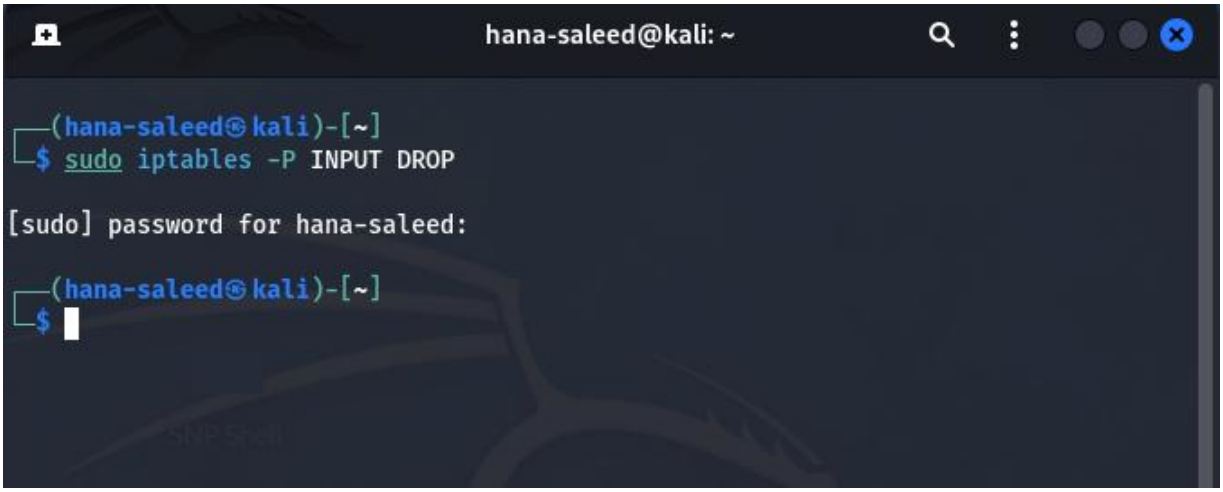
An access control list (ACL) contains rules that grant or deny access to certain digital environments. It is a list of permissions that dictate what a user has access to and what types of operations they are allowed to do with that access. An **Allow** or **Deny** rules either permit or deny access.

i. Web Server Security

Install the package using the command



To allow only incoming traffic port 80 (HTTP) and port 443(HTTPS), First we have to block all incoming traffic unless explicitly allowed

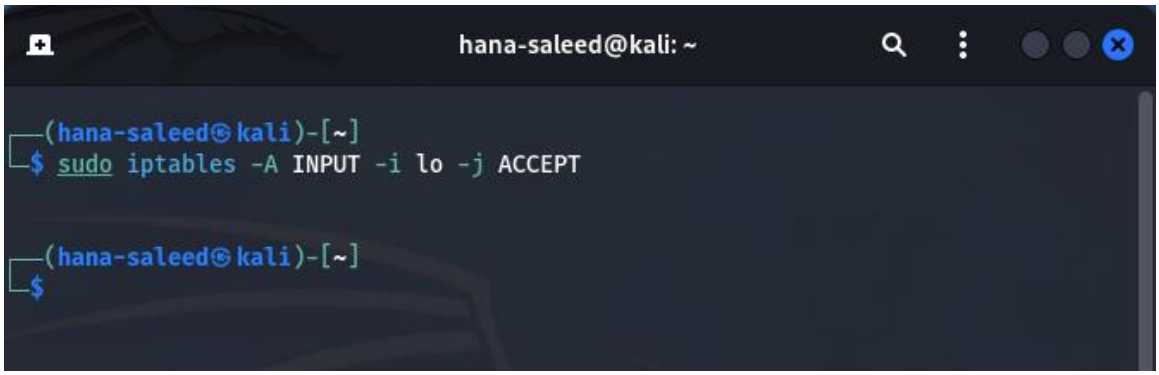


Then we to allow web traffic in port 80 using the following command

Then we to allow web traffic in port 443 using the following command

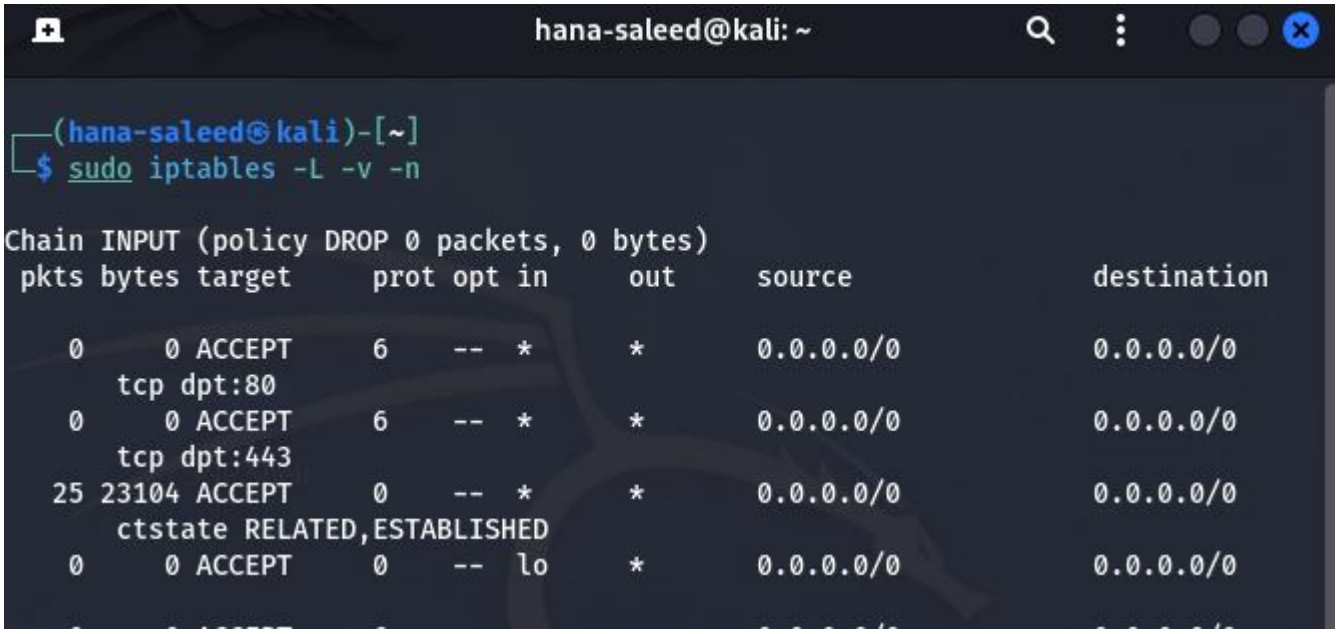
Established connections responses like HTTP/HTTPS are allowed back in with the command

Thereafter, we have to ensure the server can communicate with itself.



To make sure these rules persist after a reboot we have to save after editing respective permissions. **sudo chmod 644 /etc/iptables/rules.v4**

Finally, we can confirm that the web server security configuration is correctly allowing incoming traffic on ports 80 and 443 as **tcp dpt:80 tcp dpt :443**

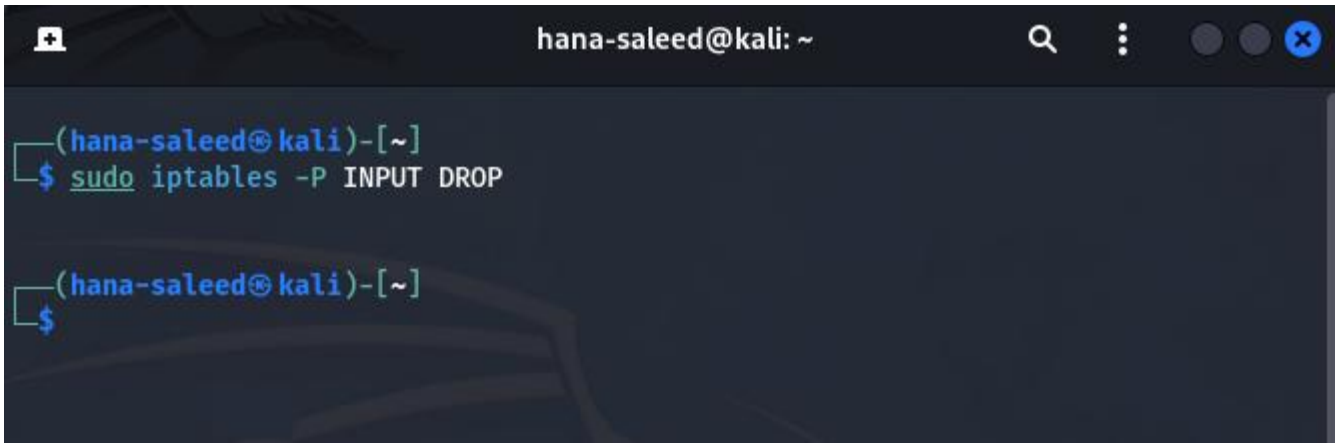


ii. Remote Administration Access

Allows only a specific ip address to SSH

iii. Allow specific applications

Allows specific applications, permitting traffic on their associated port numbers. First, we have to drop all incoming traffic



The port 443 should be allowed

Then we have to allow established connections

iv. Allow Pings (ICMP Echo Request)

To allow ping requests we can start off with allowing incoming ICMP traffic

Next we can allow related ICMP traffic and save the rule so that it persist after a reboot

v. Printer Server Access

To secure the printer server, port 9100 traffic only has to be allowed from our IP address. First and foremorst, we have to only allow traffic to the printer from our ip addresse range.

Then all other traffic to port 9100 has to be blocked and the ongoing establish connections has to be allowed

We can check if the rules have been applied by the command **sudo iptables -L -v -n**

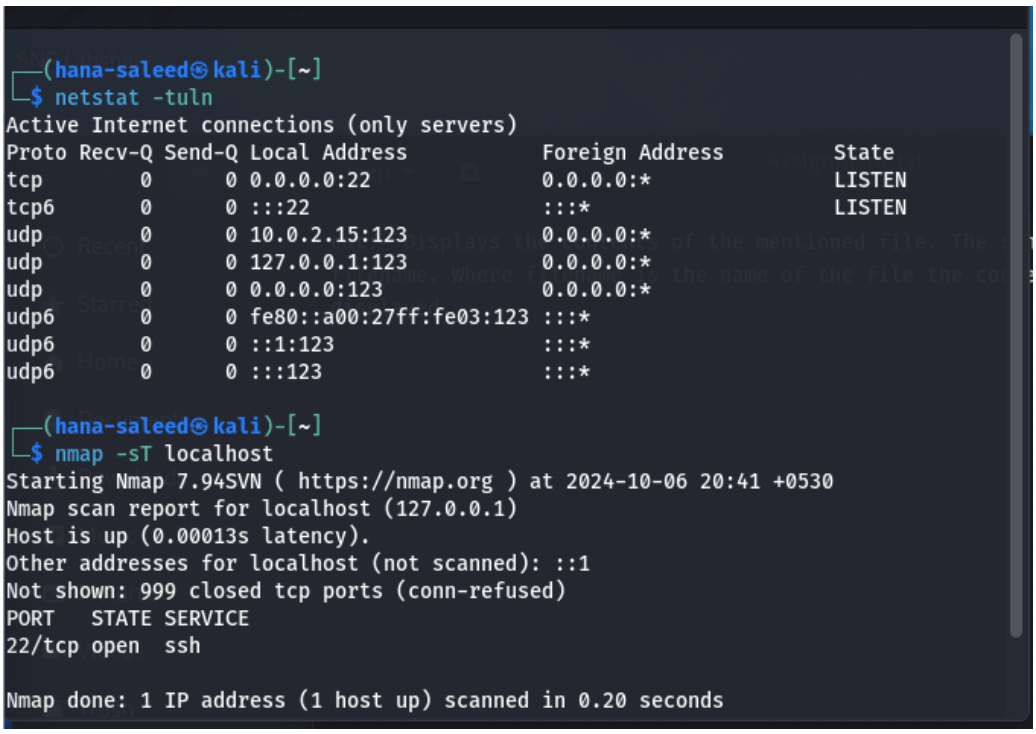
```
0     0 ACCEPT     6    --  *      *       10.0.2.0/24          0.0.0.0/0
   tcp dpt:9100
0     0 DROP       6    --  *      *       0.0.0.0/0            0.0.0.0/0
   tcp dpt:9100
0     0 DROP       6    --  *      *       0.0.0.0/0            0.0.0.0/0
   tcp dpt:9100
0     0 ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0
   ctstate RELATED,ESTABLISHED
```

04. Best Practices

Understanding the security aspects of network interface configuration is crucial for protecting your system from unauthorized access, data breaches, and various network attacks

04.1 Close unused ports

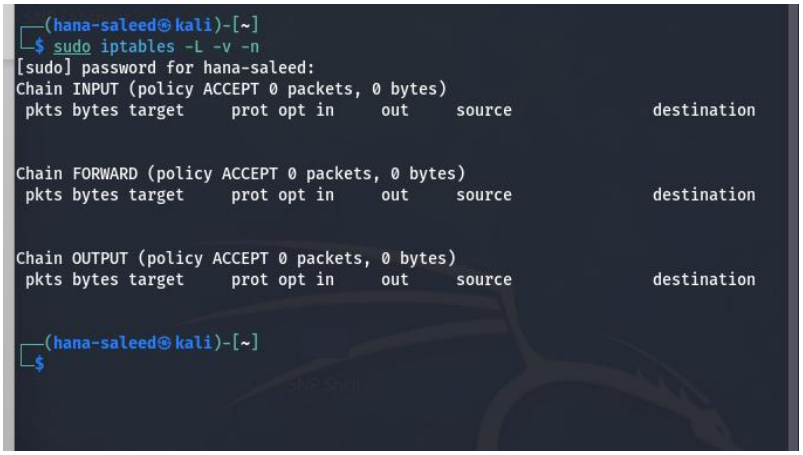
We can find unsused ports sing the commands like netstat,nmap or ss



We can close the port with the syntax **sudo iptables -A INPUT -p tcp --dport *port_number* -j DROP**

04.2 Configure Firewalls

We can find the current rules by command **sudo iptables -L -v -n**



We can set some default policies like dropping all incoming traffic

```
┌──(hana-saleed㉿kali)-[~]
└─$ sudo iptables -P INPUT DROP
```

04.3 Limit SSH Access

Editing the SSH configuration file to limit access by editing the file as

```
┌──(hana-saleed㉿kali)-[~]
└─$ sudo nano /etc/ssh/sshd_config


┌──(hana-saleed㉿kali)-[~]
└─$
```

PermitRootLogin no

AllowUsers *user1 user2*

04.4 Enable Automatic Security Updates

Ensure that your system automatically installs security updates.

```
┌──(hana-saleed㉿kali)-[~]
└─$ sudo apt install unattended-upgrades
```

04.5 Change File Permissions

Set appropriate file permission to protect sensitive files

```
┌──(hana-saleed㉿kali)-[~]
└─$ sudo chmod 600 /etc/secret.conf
```