Sri Lanka Institute of Information Technology



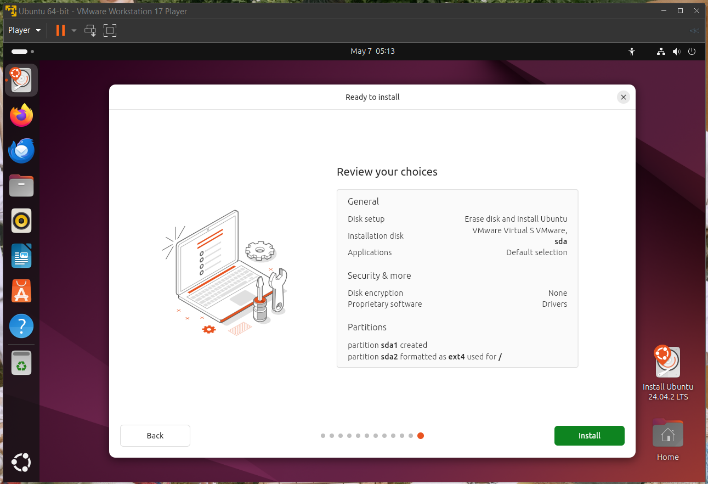
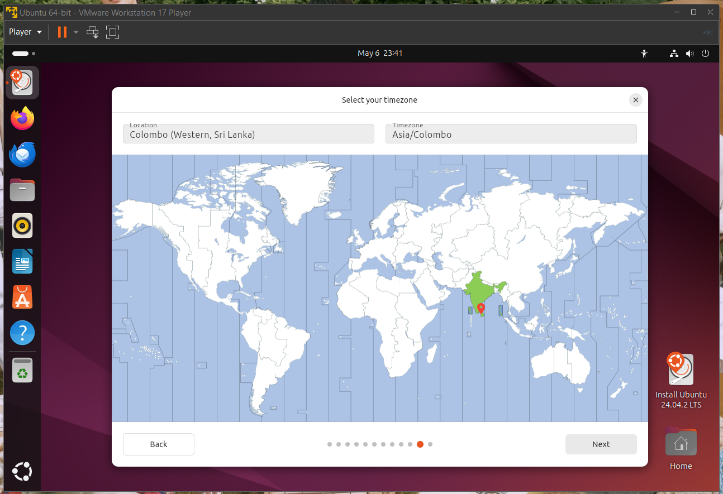
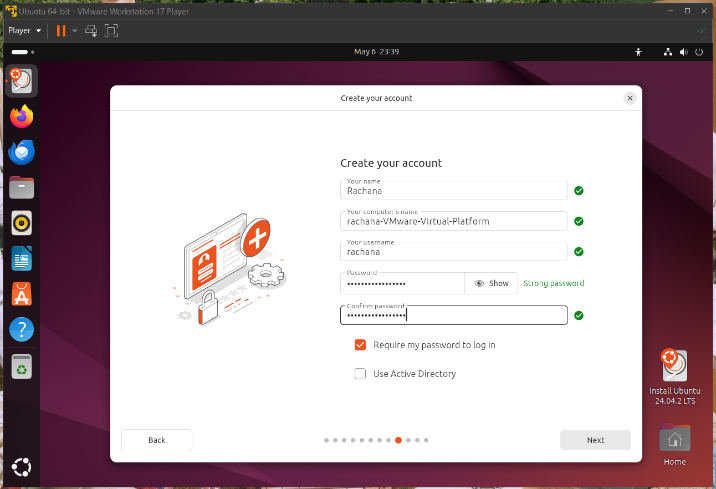
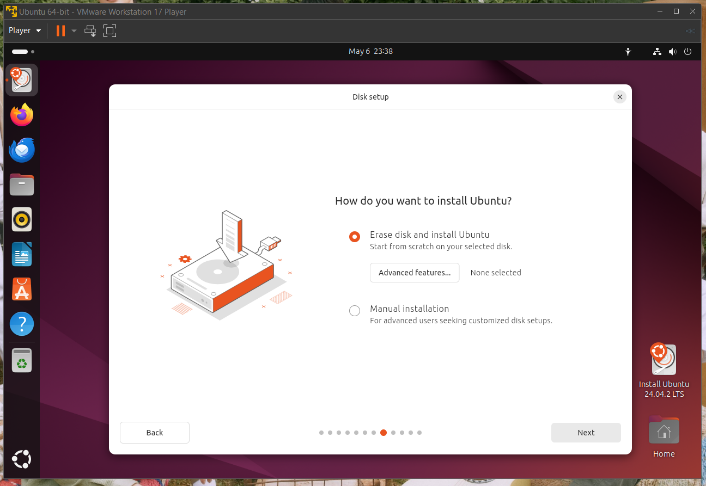
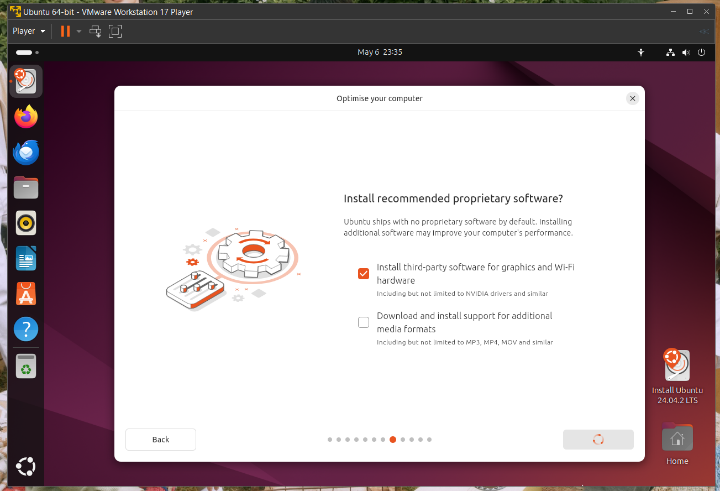
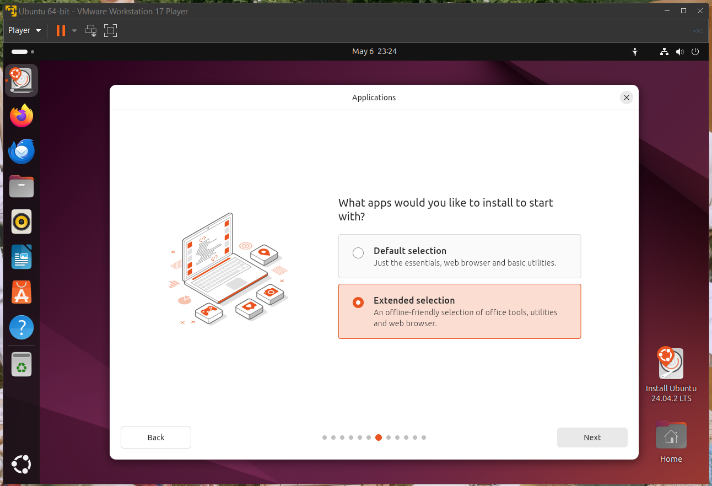
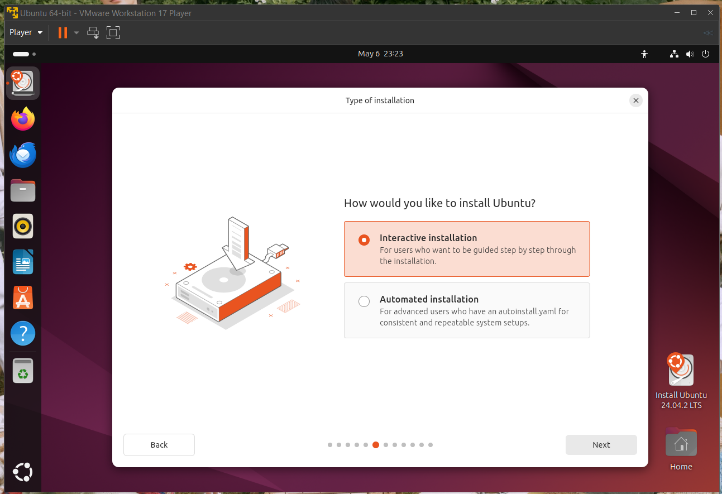
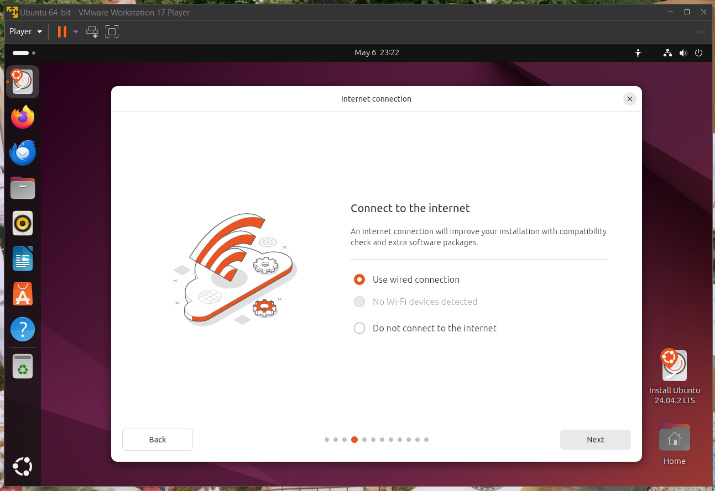
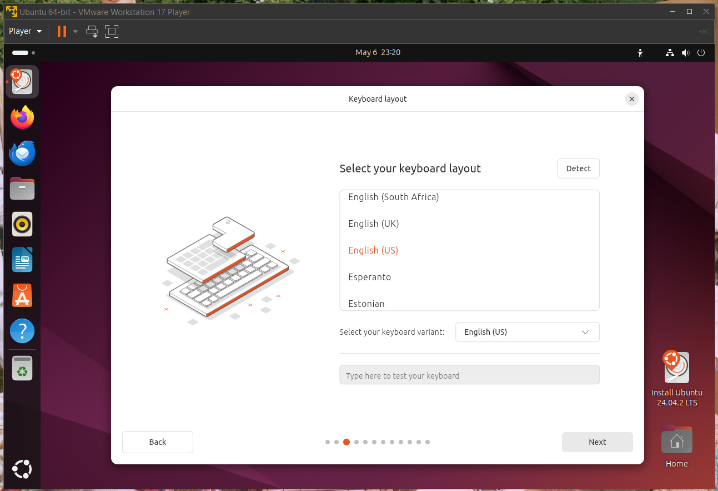
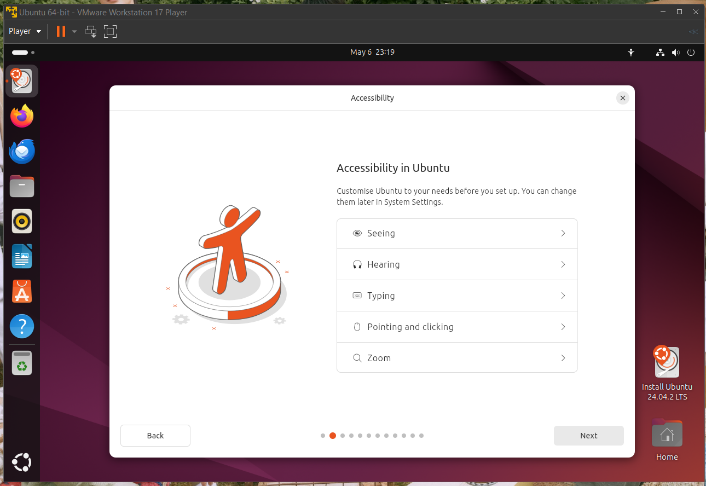
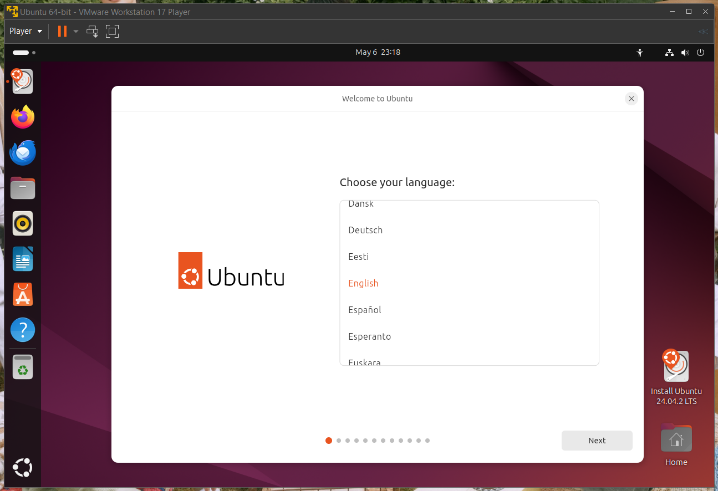
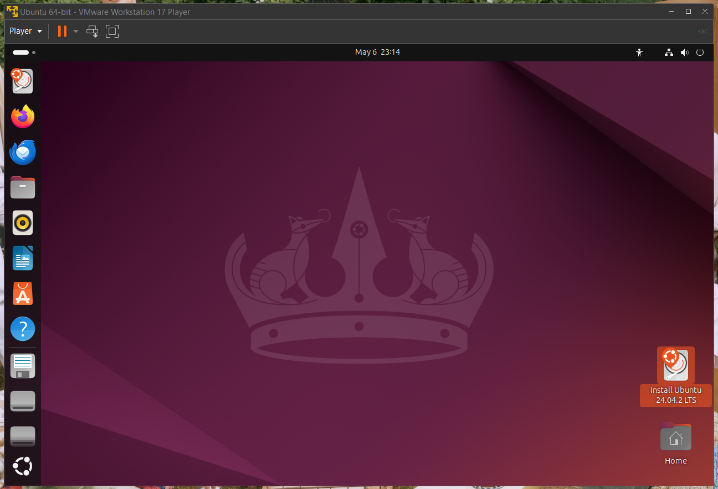
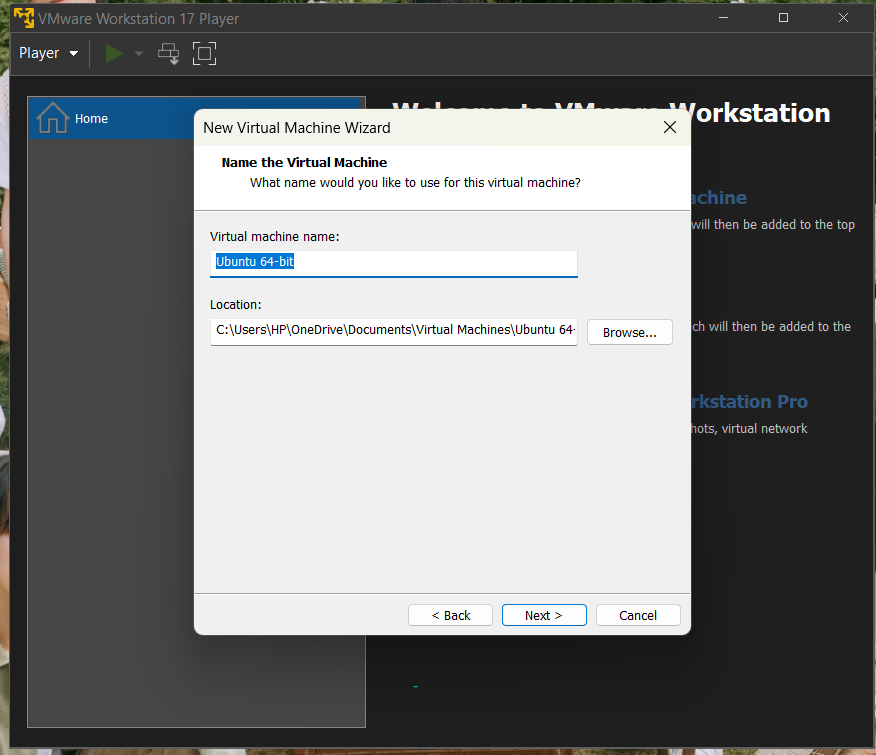
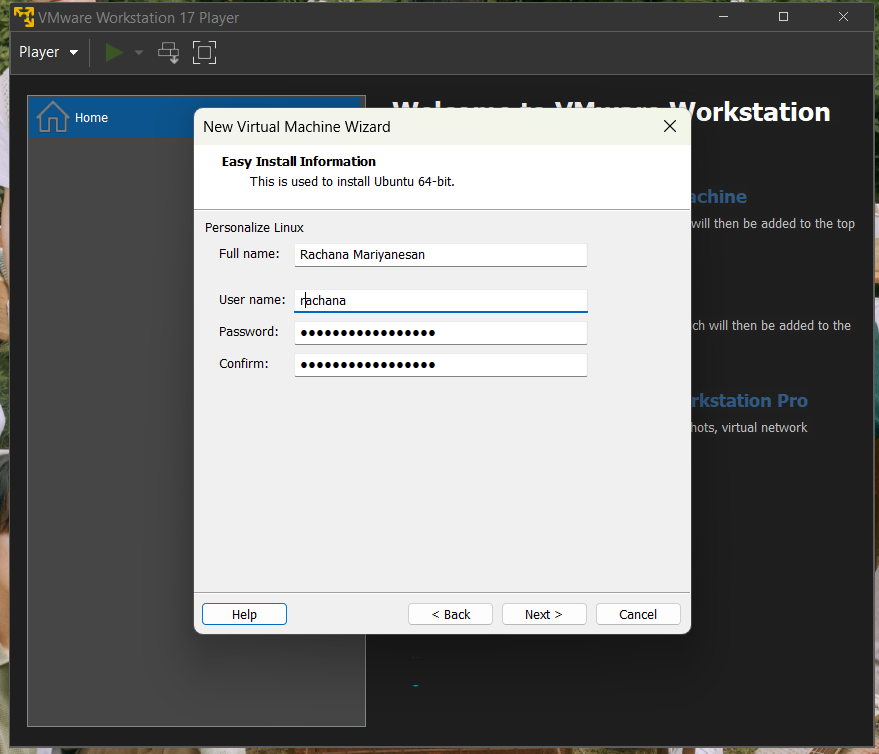
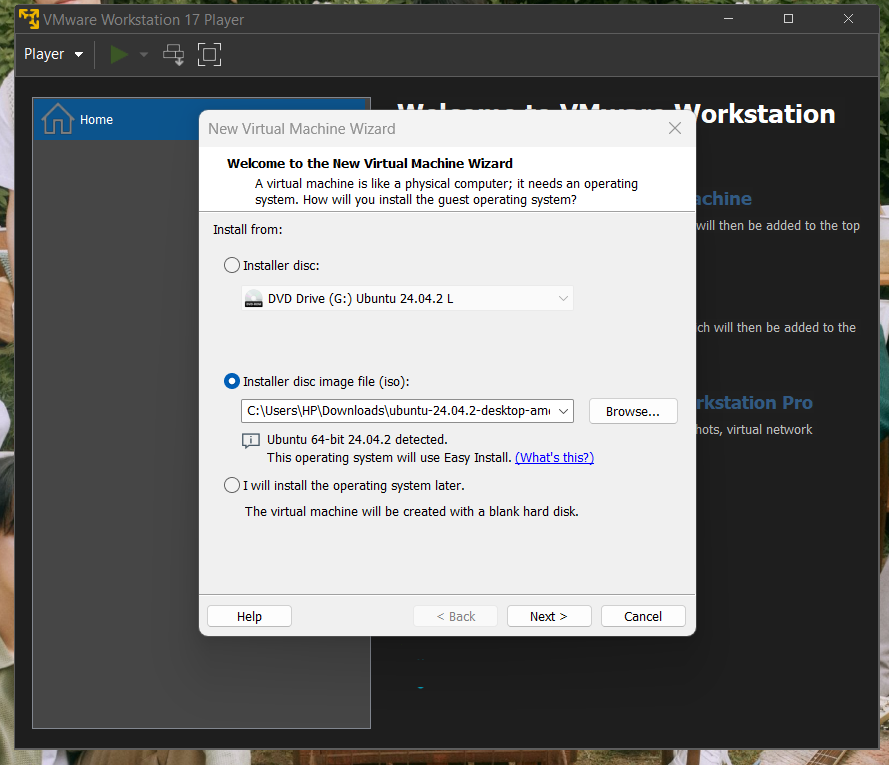
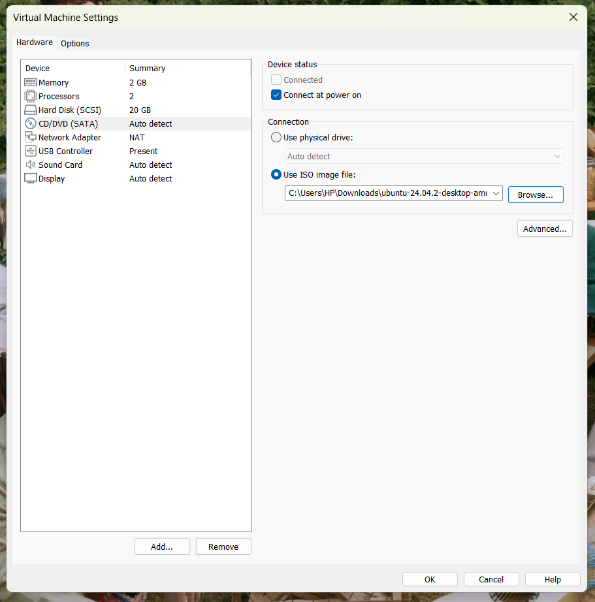
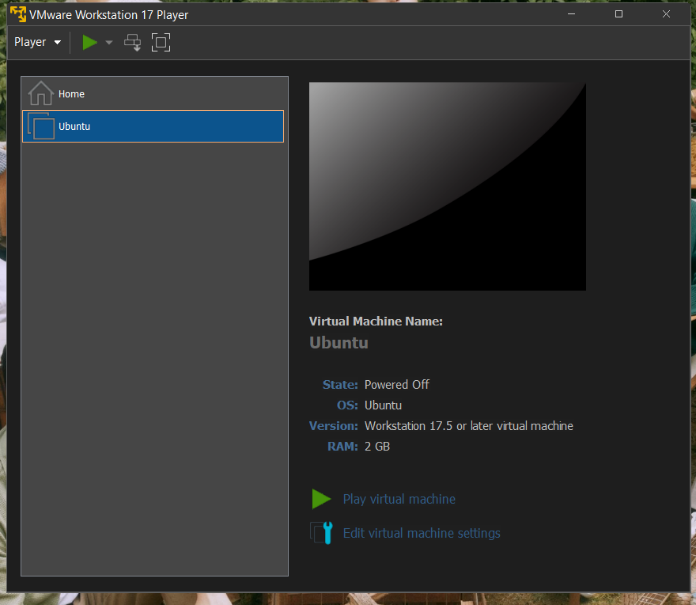
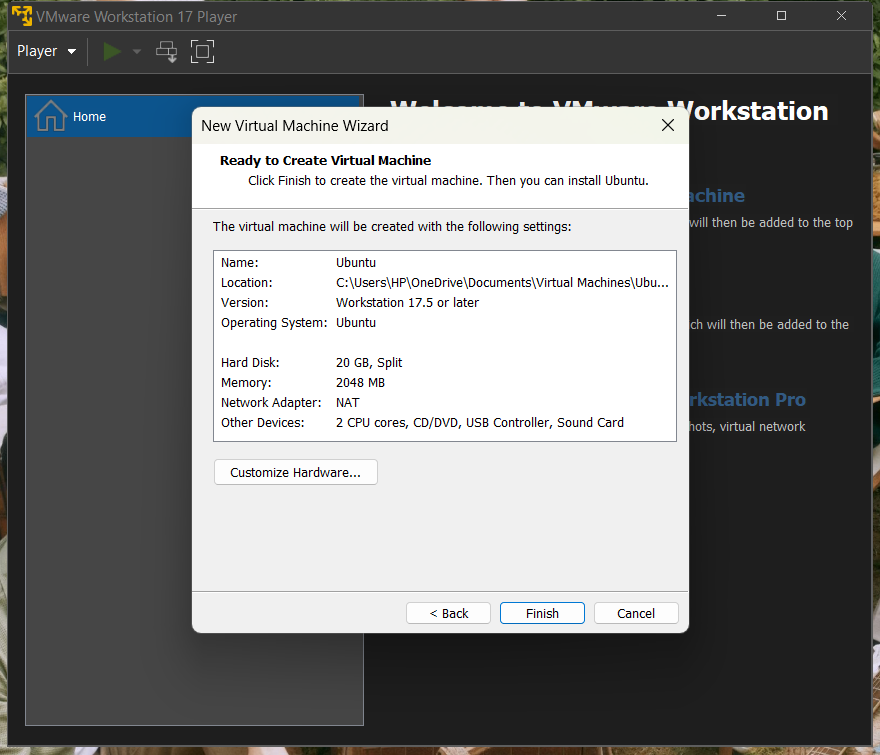
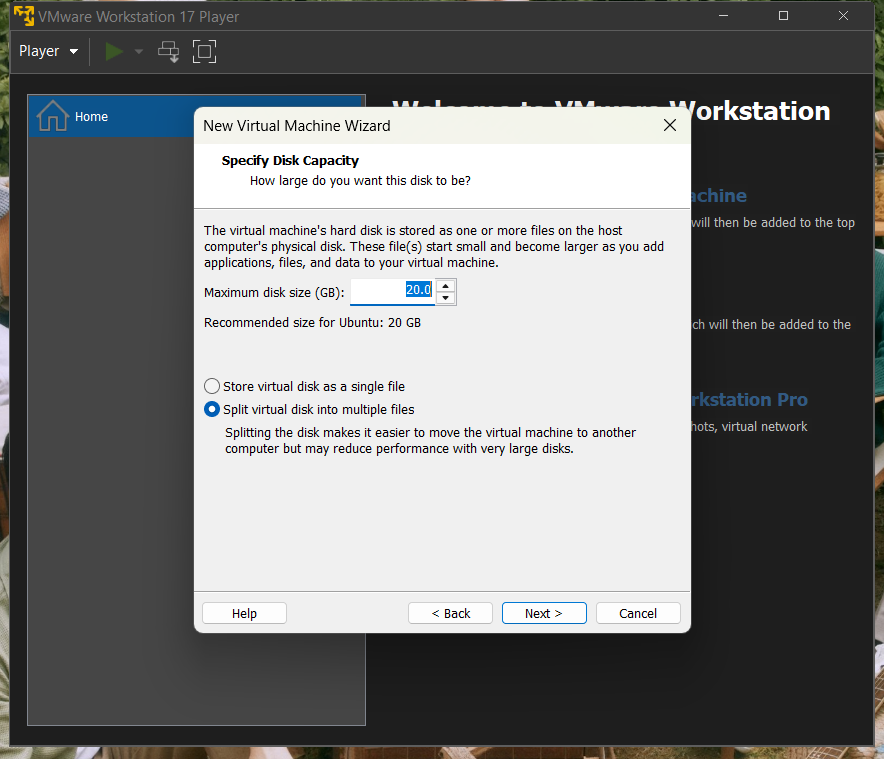
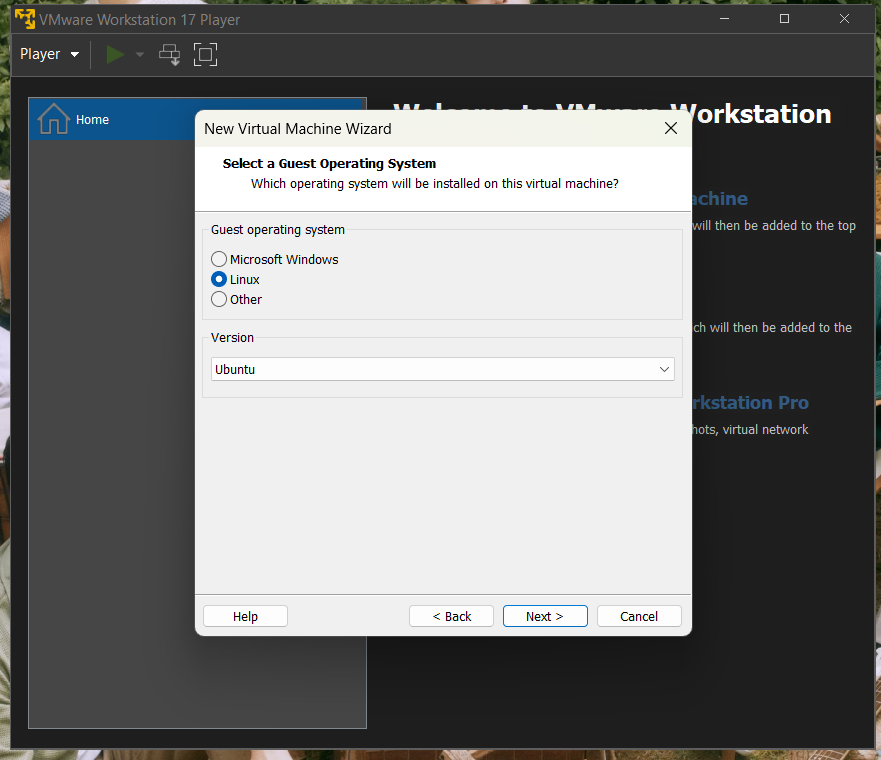
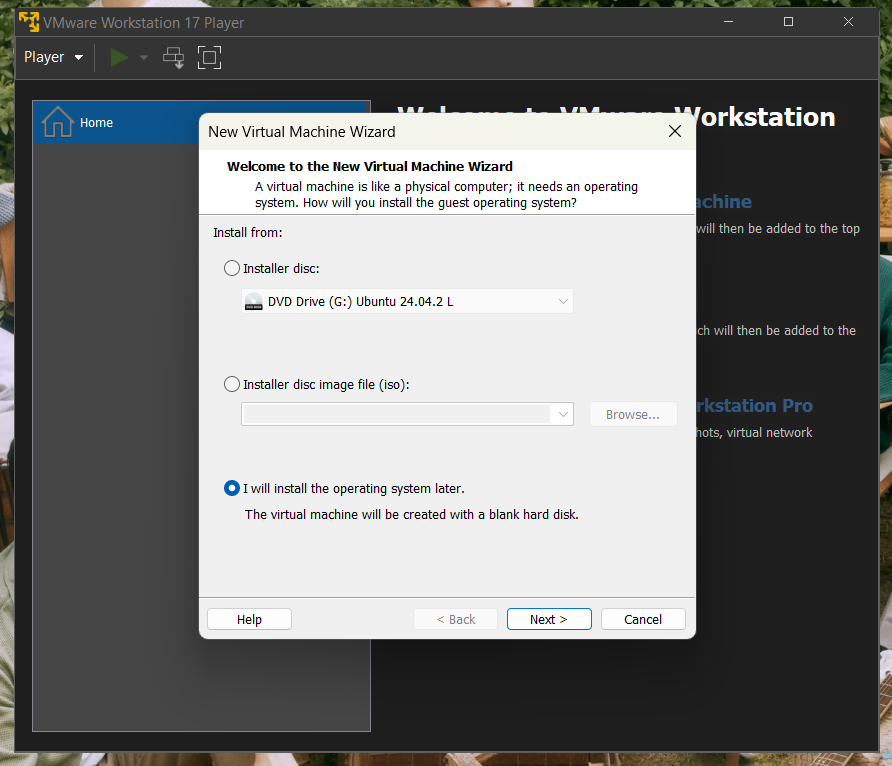
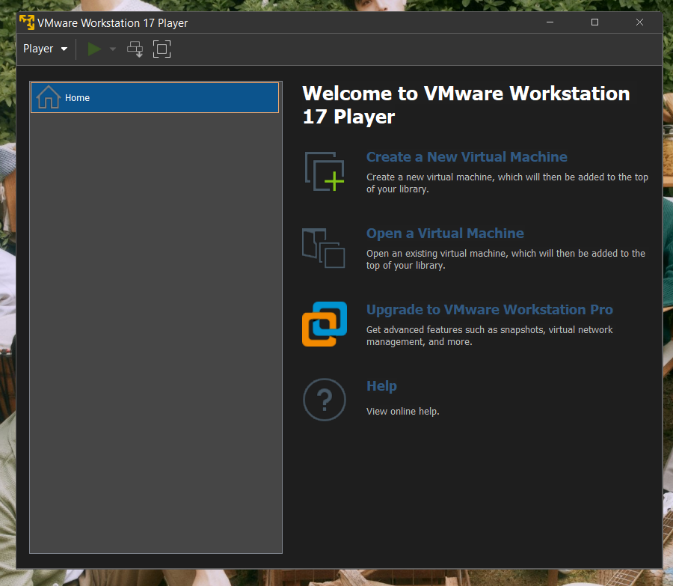
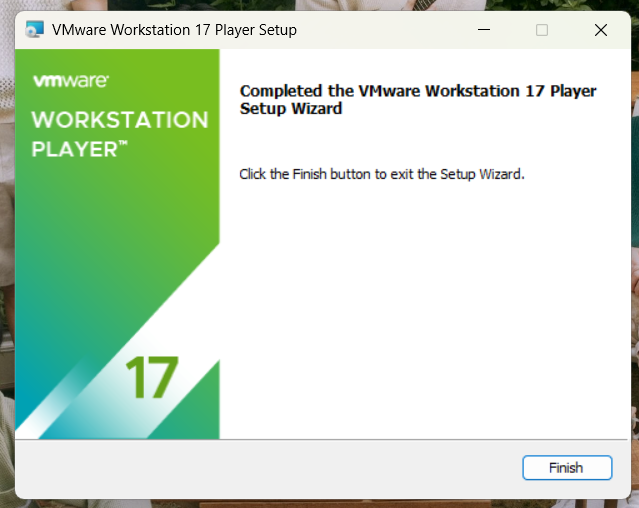
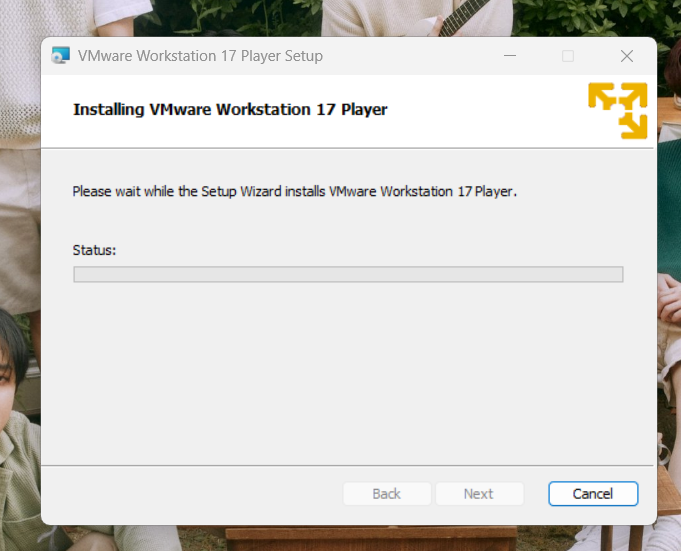
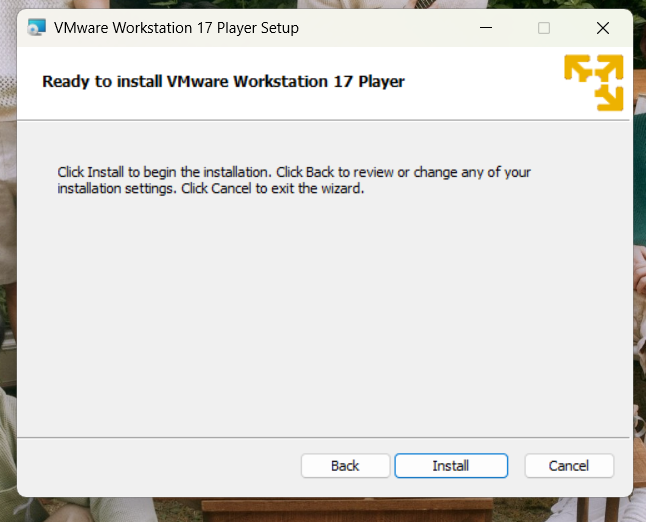
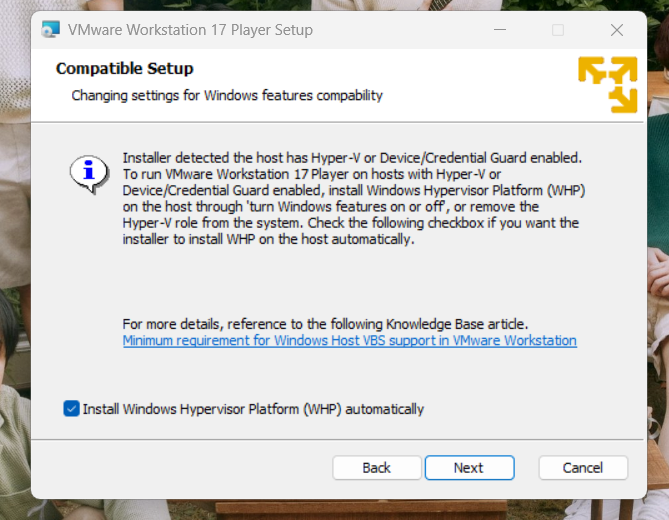
**Rachana Mariyanesan (IT23833920)**

**IE2012 – Systems and Network Programming**

**Year 2 Semester 1 – 2025**

**Virtual Machine Installation Steps**

* Download and install a virtualization tool **VMware Workstation Player.**
* Download a Linux distribution **Ubuntu**
* Create a new virtual machine, allocate CPU, RAM, and disk space.
* Mount the ISO and follow the installation prompts to install Ubuntu in VMware Workstation Player.
* Set up a user account and password during installation.



**Basic Navigation Commands**

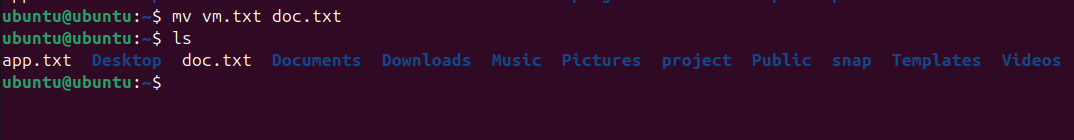
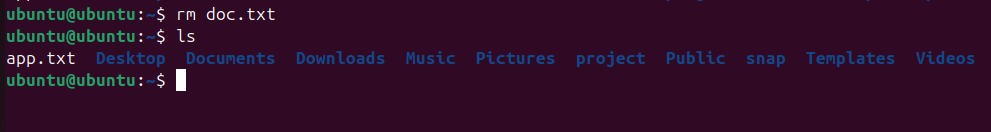
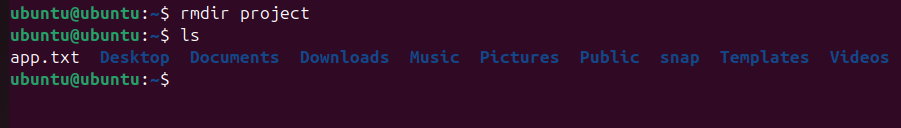
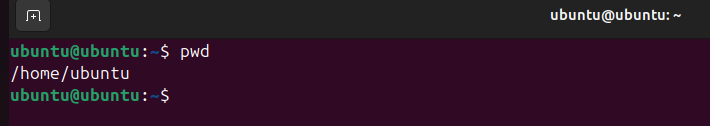
* cd: Change directory.
* ls: List directory contents.
* pwd: Print working directory.
* mkdir: Make a new directory.
* rmdir: Remove an empty directory.

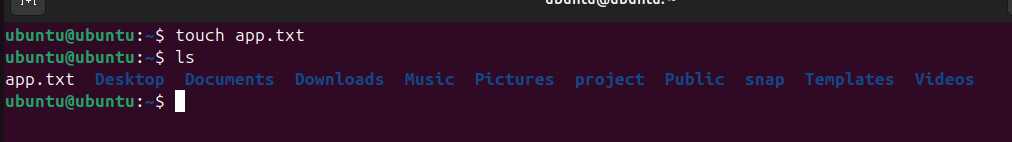
**File Manipulation Commands**

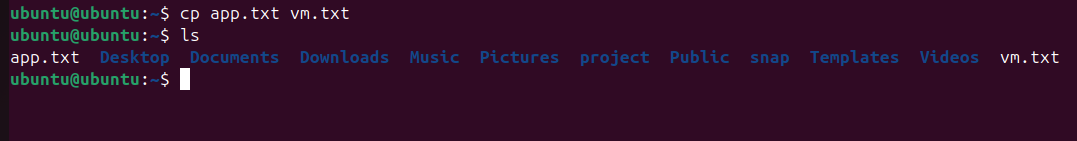
* cp: Copy files or directories.
* mv: Move or rename files or directories.
* cat: Display file contents
* rm: Remove files or directories.
* touch: Create an empty file.

**15 Basic Linux Commands with Descriptions**

|  |  |
| --- | --- |
| Command | Description |
| Pwd | Print current working directory |
| ls | List files and directories |
| cd | Change Directories |
| mkdir | Create a new directory |
| rmdir | Remove an empty directory |
| touch | Create an empty file |
| cat | Display file contents |
| cp | Copy files or directories |
| mv | Move or rename files/directories |
| rm | Remove files or directories |
| echo | Display a line of text |
| nano | Simple text editor |
| vi | Advanced text editor |
| chmod | Change file/directory permissions |
| chown | Change file/directory owner and group |



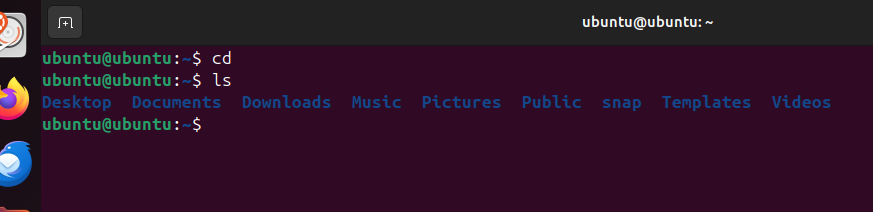












**DHCP (Dynamic Host Configuration Protocol)**

* **Role in Network Configuration:** Automatically assigns IP addresses to devices in a network, reducing manual configurations.
* **Installation & Configuration:**
  + **Install DHCP server:** 
    - sudo apt update
    - sudo apt install isc-dhcp-server -y
  + **Edite the DHCP configuration file:**
    - sudo nano /etc/dhcp/dhcpd.conf
  + **Define a subnet configuration:** 
    - subnet 192.168.1.0 netmask 255.255.255.0 { range 192.168.1.100 192.168.1.200; option domain-name-servers 8.8.8.8; option routers 192.168.1.1; option broadcast-address 192.168.1.255;default-lease-time 600; max-lease-time 7200;}
  + **Restart the DHCP server:** 
    - sudo systemctl restart isc-dhcp-server
  + **Configure the client virtual machines to use DHCP:** 
    - sudo nano /etc/netplan/00-installer-config.yaml
    - network:

ethernets:

ens33:

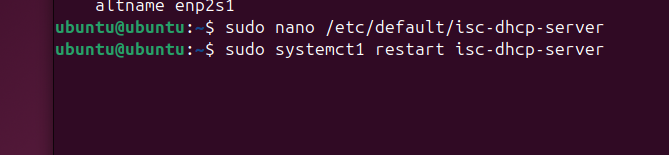
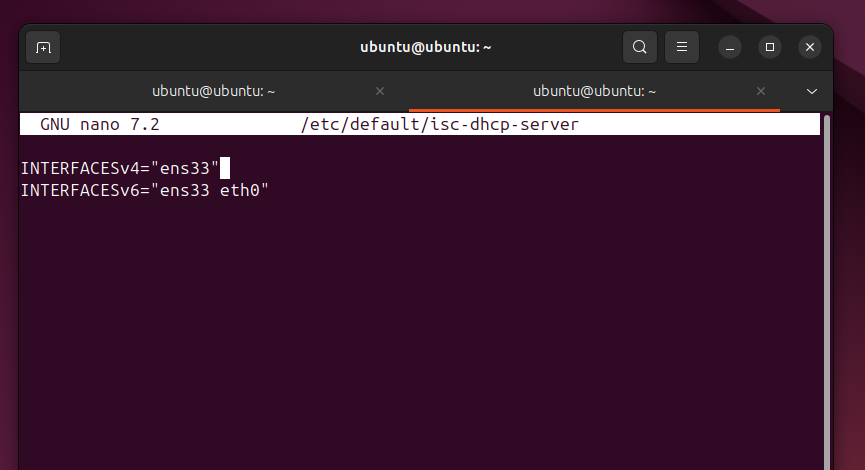
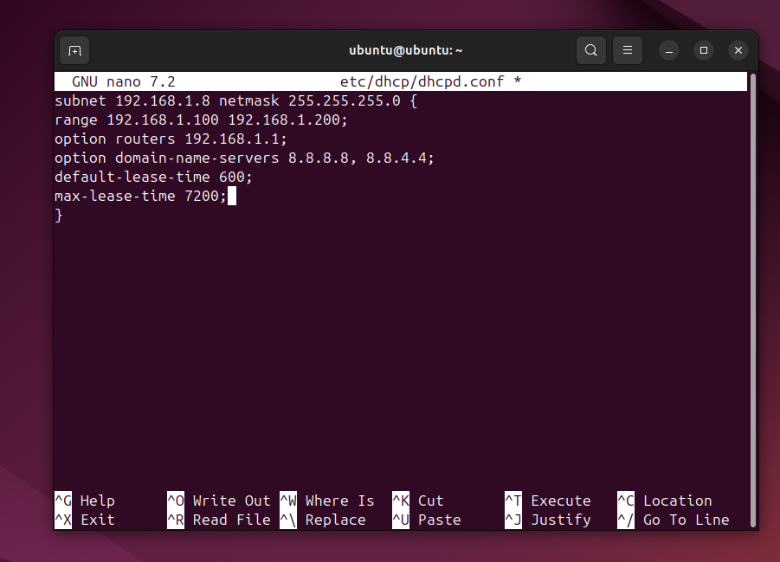
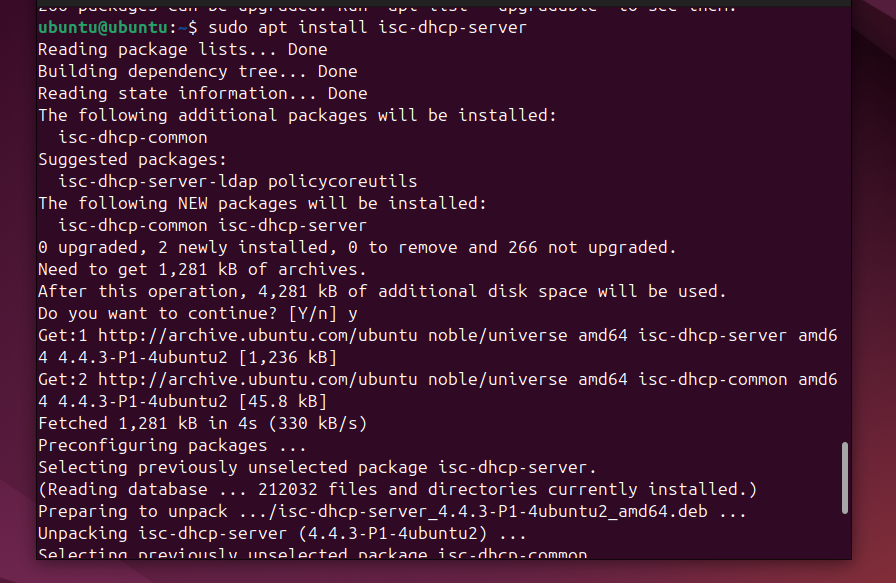
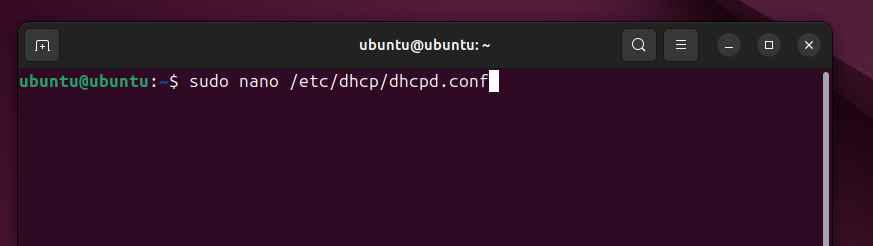
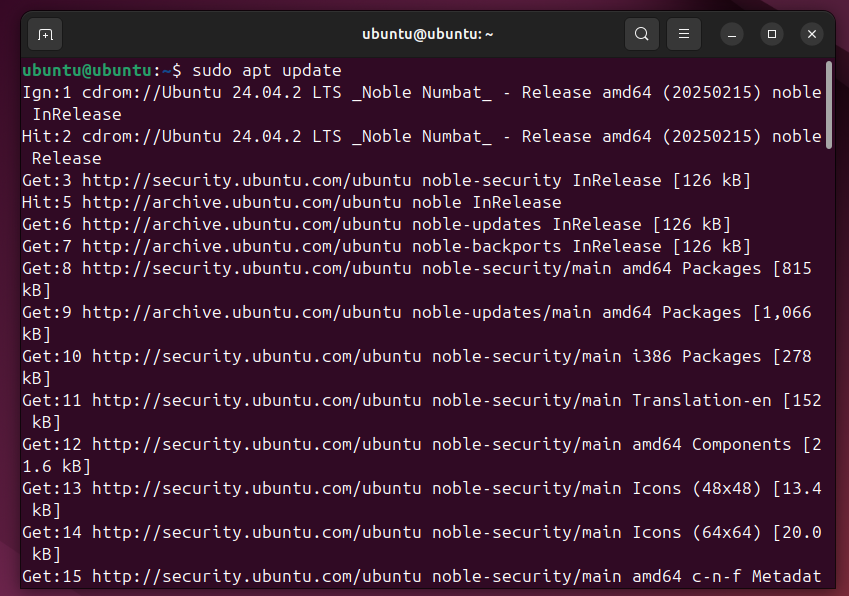
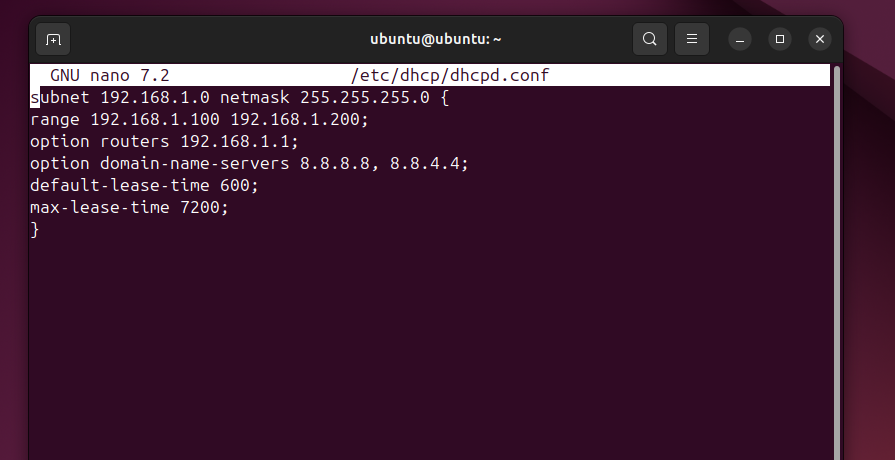
dhcp4: true

version: 2

* + - sudo netplan apply
  + **Set DHCP:** 
    - auto eth0

iface eth0 inet dhcp

* + **Restart the network service:**
    - sudo systemctl restart networking
* **Verify IP allocation:**
  + **Dynamically:** ip a | grep inet
* **Testing Connectivity:**
  + ping -c 5 192.168.1.1



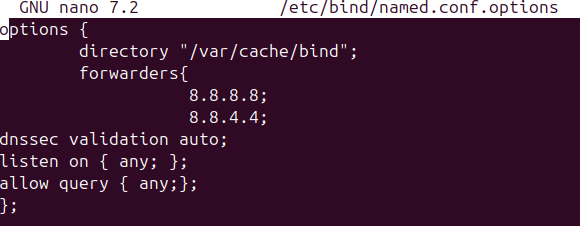
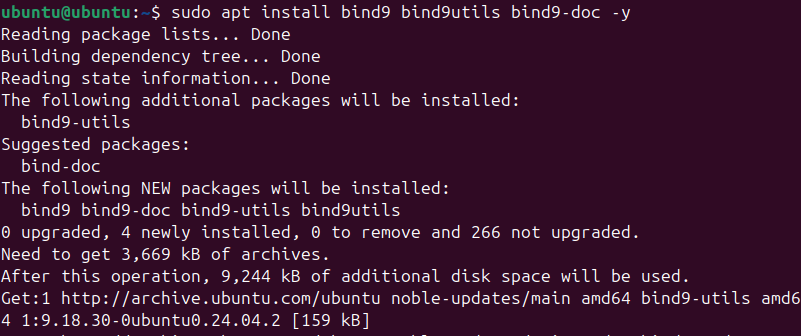
**DNS (Domain Name System)**

* **Purpose:** Resolves human-readable domain names into IP addresses**.**
* **Installation:**
* **Install BIND DNS server:** sudo apt install bind9 -y
* **Backup default config file:** sudo cp /etc/bind/named.conf.options /etc/bind/named.conf.options.backup
* **Configure the DNS server:** cat <<EOT | sudo tee /etc/bind/named.conf.options

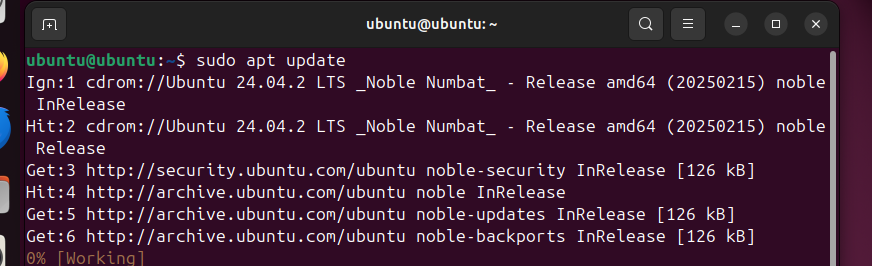
options {directory "/var/cache/bind"; forwarders { 8.8.8.8; # Google DNS 8.8.4.4; }; dnssec-validation auto; listen-on { any; };allow-query { any; };};EOT

* **Restart the service**: sudo systemctl restart bind9
* **Enable BIND to start on boot:** 
  + sudo systemctl enable bind9
  + echo "BIND DNS Server is configured and running."
* **Run the script:** chmod +x setup\_dns\_server.sh

./setup\_dns\_server.sh



**NTP (Network Time Protocol)**

* **Importance: Ensures accurate time synchronization across systems.**
* **Installation & Configuration:**
  + **Install NTP client:** sudo apt install ntp
  + **Configure** /etc/ntp.conf **to sync with an internet NTP server.**
  + **Start service**: sudo systemctl restart ntp
* **Validation:** Check time sync using ntpq -p.

****

**Need & Use of These Services**

* **DHCP**: Automates IP assignments, reducing misconfigurations**.**
* **DNS**: Essential for hostname-to-IP resolution in networks.
* **NTP:** Prevents clock drift, ensuring time-sensitive operations function correctly.

**System Details Script & Cron Job**

**Sample Script:**

#!/bin/bash

DATE=$(date)

UPTIME=$(uptime -p)

MEMORY=$(free -h | grep Mem | awk '{print $3 "/" $2}')

DISK=$(df -h / | tail -1 | awk '{print $3 "/" $2}')

echo "Date: $DATE"

echo "Uptime: $UPTIME"

echo "Free Memory: $MEMORY"

echo "Disk Usage: $DISK"

**Cron Job:**

* + **Edit crontab:** crontab -e
  + **Add:** 0 0 \* \* 0 /path/to/script.sh >> /var/log/sysdetails.log (runs every Sunday at midnight)

**Cron Job**

* **Use** crontab -e **to edit cron jobs.**
* **Syntax:** \* \* \* \* \* command
* **Example:** 0 12 \* \* 1 /home/user/backup.sh **(runs every Monday at noon).**
* **Save and exit to activate the job.**
* **Check jobs:** crontab -l

**Connecting to VM Remotely Using SSH**

* **Ensure SSH server is installed:** sudo apt install openssh-server
* **Find VM IP :** ip a
* **From another computer:** ssh username@VM-IP
* **Accept the fingerprint and enter your password to connect**

**iptables Rule Description with Command**

|  |  |
| --- | --- |
| Rule Description | Command Example |
| Allow SSH | sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT |
| Allow HTTP | sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT |
| Allow HTTPS | sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT |
| Block all incoming traffic by default | sudo iptables -P INPUT DROP |
| Allow established connections | sudo iptables -A INPUT -m state –state ESTABLISHED,RELATED -j ACCE PT |

**Web Server**

**Apache:**

* **Install:** sudo apt install apache2
* **Start:** sudo systemctl start apache2 or sudo systemctl start httpd
* **Enable:** sudo systemctlenable apache2 or sudo systemctl enable httpd
* **Place web files in** /var/www/html/
* **Access via browser:** http://[VM-IP]/

**Email Server**

**Postfix Example Ubuntu**

* **Install:** sudo apt install postfix
* **Choose** "Internet Site" **during setup**
* **Set system mail name (FQDN)**
* **Configure** /etc/postfix/main.cf **as needed.**
* **Start and enable:** sudo systemctl start postfix , sudo systemctl enable postfix
* **Test with:** echo "Test mail" | mail -s "Test" user@example.com

**Linux GDB**

**Execution process**

* **Check System Architecture:** uname -m
* **Choose the executable:** x86\_64
* **Run the selected executable:** sudo ./x86\_64
* A new executable named IT23833920 got generated when entered IT number as an input.
* **Analyze the Generated Executable:**
  + **Run the new executable:** ./IT23833920
  + **Inspect the generated file:** cat data.txt

**Debugging process**

* **Start GDB**: gdb ./IT23833920
* **Track file modification:** Set breakpoints on file writing functions.
  + **Break when files are opened:** (gbd) break fopen
  + **Break when data is written:** (gbd) break fwrite

**File System Analysis**

* **Before execution:**
  + **List files:** ls -l
  + **Check metadata:** stat data.txt
* **During execution:**
  + **Monitor directory charges in real-time:** inotifywait -m .
* **After execution:**
  + **Recheck files and metadata:** ls -l
  + **Verify creation/ modification time:** stat data.txt
  + **Extract printable strings:** strings data.txt
  + **View hex and ASCII:** hexdumb -C data.txt

**Analysis of “data.txt” file**

* **Check for encoding/encryption:**
  + **Base 64:** base64 -d data.txt
  + **Hex:** xxd -r -p data.txt
* **Hidden data:**
  + **Use binwalk to detect embedded files:** binwalk data.txt