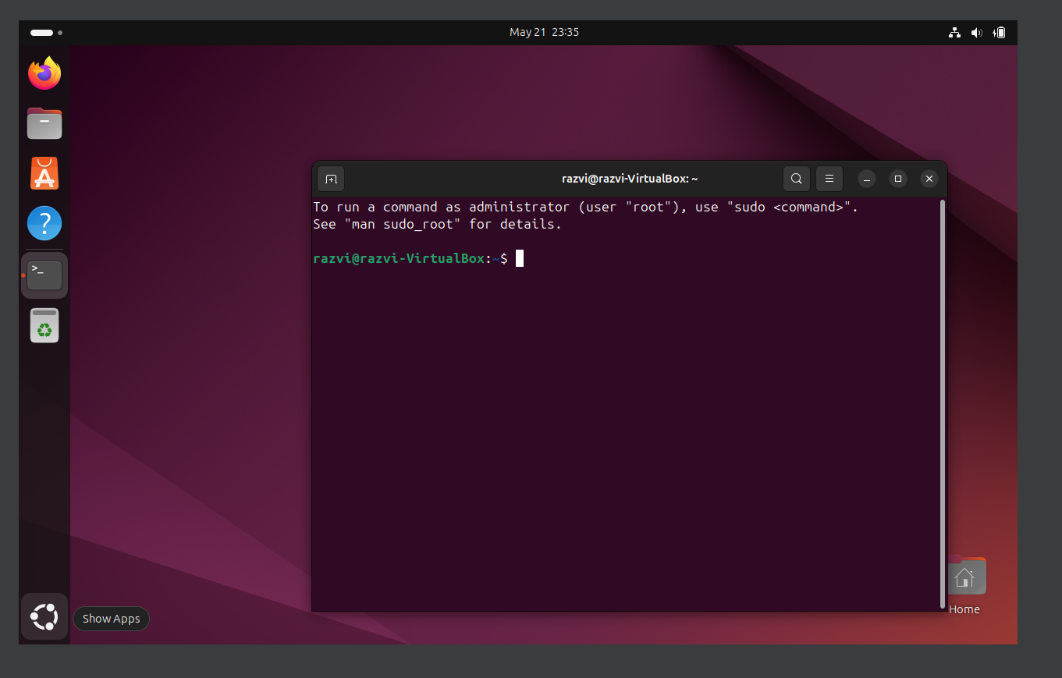
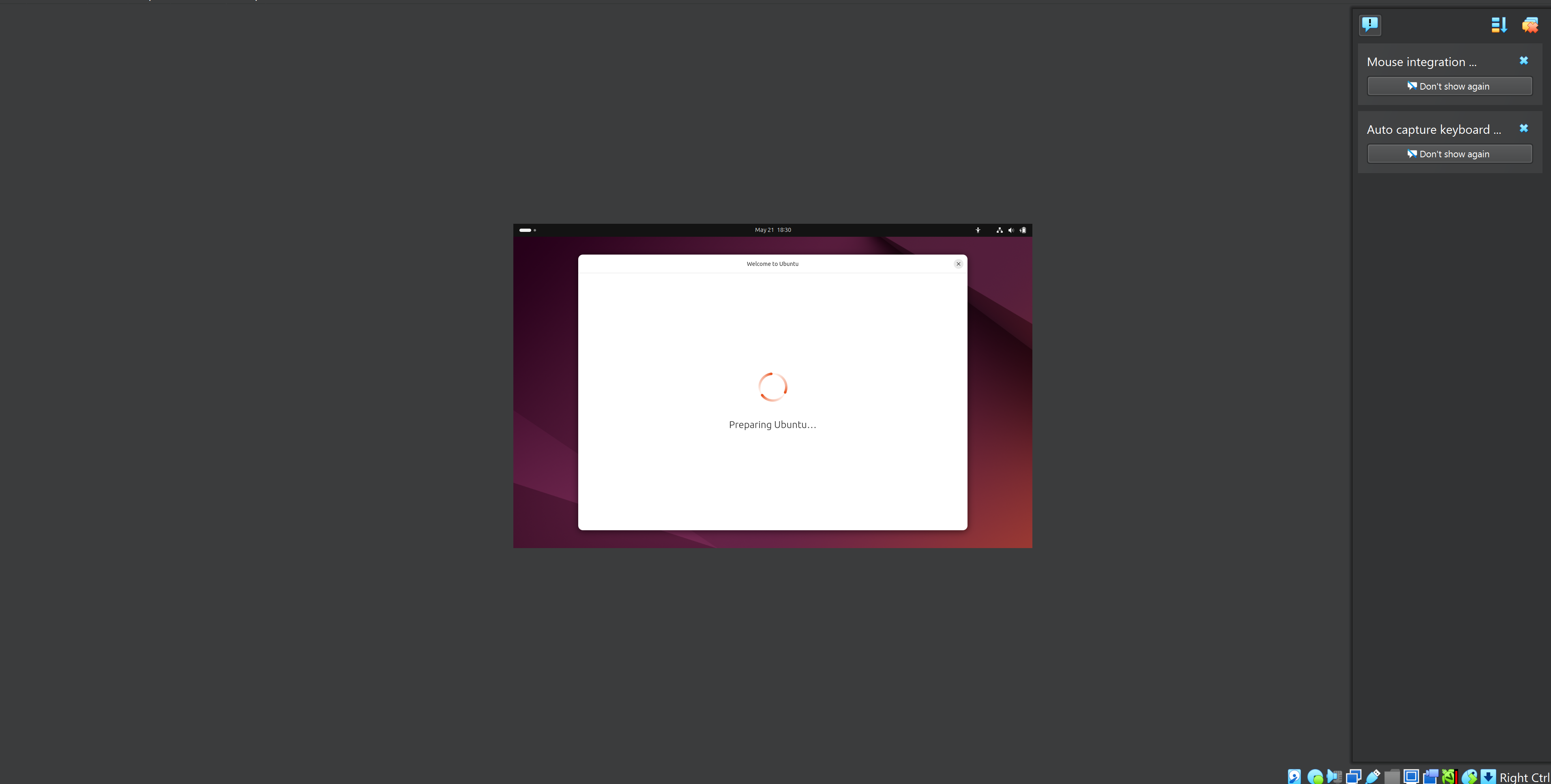
**1. Install Linux (Ubuntu)**

**Description:**  
Ubuntu is a free and open-source Linux distribution commonly used for development, servers, and learning system administration.

**Steps:**

* Downloaded ubuntu-24.04.2-desktop-amd64.iso from the official Ubuntu website
* Opened VirtualBox and clicked New
* Created a VM named ubuntu and selected Ubuntu (64-bit)
* Set 4096 MB RAM and 25 GB virtual hard disk
* Attached the Ubuntu ISO under Storage > IDE Controller
* Clicked Start
* Followed installation steps:
  + Chose "Install Ubuntu"
  + Selected language and keyboard layout
  + Chose "Minimal installation"
  + Accepted disk formatting
  + Created a user account
  + Waited for installation to finish and restarted the VM
* After reboot, logged into Ubuntu and saw the welcome screen



**2. SSH Server (OpenSSH)**

**Description:**  
SSH (Secure Shell) allows remote command-line access to the Ubuntu system from another machine, using an encrypted connection. It is commonly used for secure administration and file transfers.

**Steps:**

sudo apt update

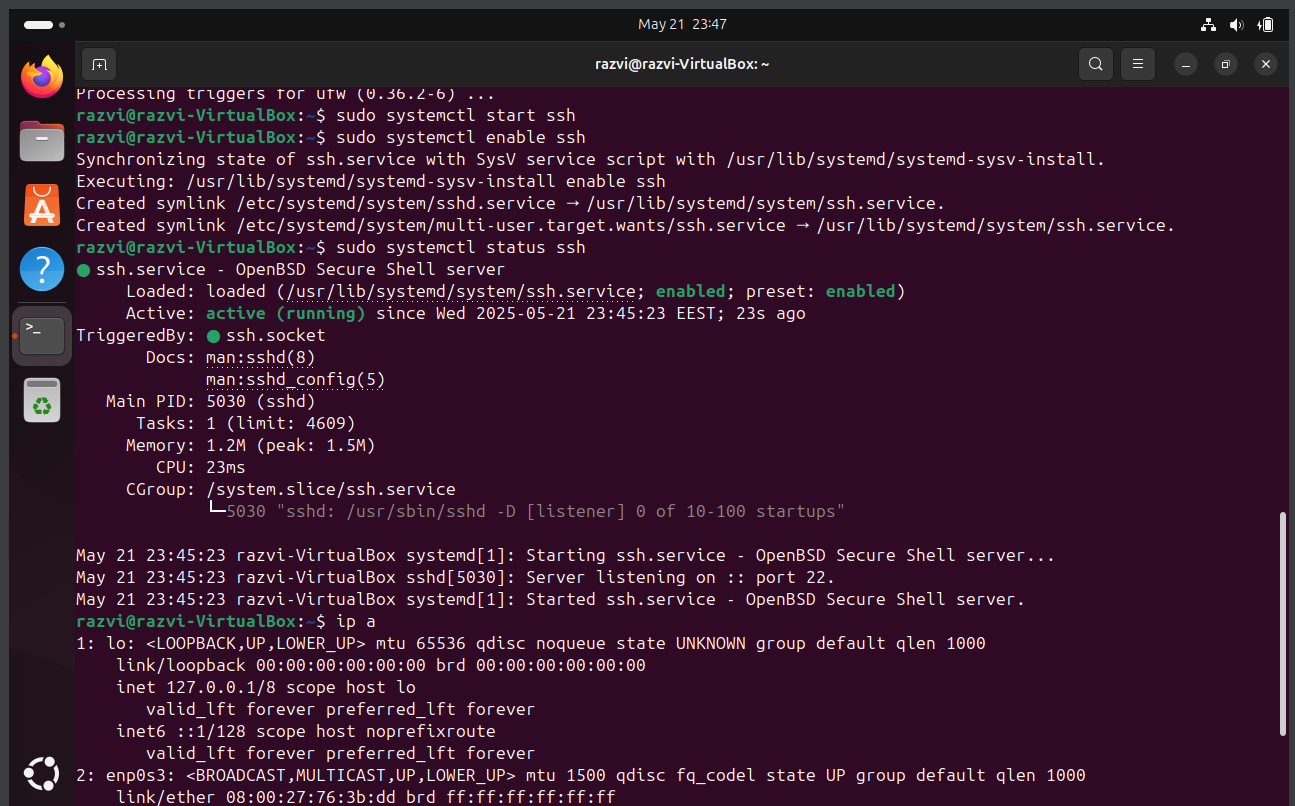
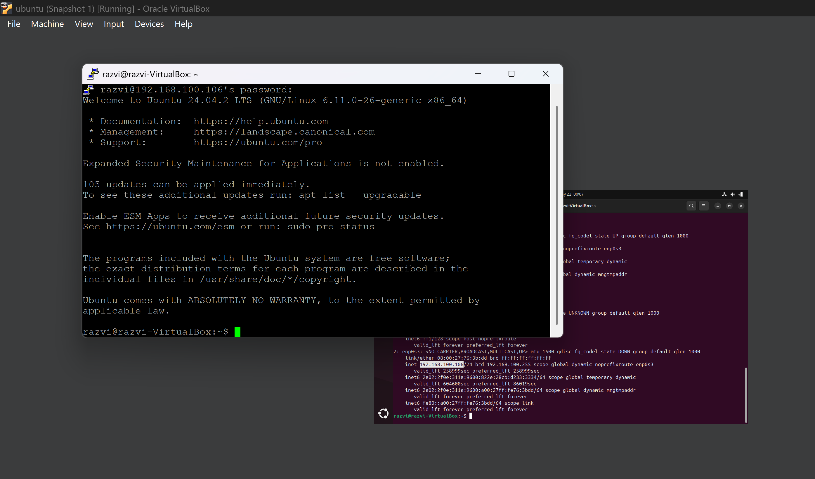
sudo apt install openssh-server

sudo systemctl start ssh

sudo systemctl enable ssh

sudo systemctl status ssh

* Verified that the SSH service was running with:  
  systemctl status ssh → showed active (running)
* Checked the machine’s IP address with:  
  ip a → found inet 192.168.100.106 as the active IP
* Switched network adapter in VirtualBox to **Bridged Adapter** to allow access from the host system
* Used **PuTTY** from Windows to connect to the VM via SSH:
  + Host: 192.168.100.106
  + Port: 22
  + Protocol: SSH
* Accepted the key fingerprint when prompted
* Logged in successfully to Ubuntu from Windows terminal via PuTTY



**3. SCP – Secure File Transfer**

**Description:**  
SCP (Secure Copy Protocol) is a secure way to transfer files between two systems over SSH. In this example, files were copied between the Windows host and the Ubuntu virtual machine.

**Steps:**

1. Created a text file in Ubuntu:

echo "This is a test from ubuntu" > test.txt

1. Verified the file:

cat test.txt

1. Copied the file from Ubuntu VM to Windows host using CMD/PowerShell:

scp razvi@192.168.100.106:/home/razvi/test.txt C:\Users\rzvbo\Desktop\

1. Created a new file in Windows:

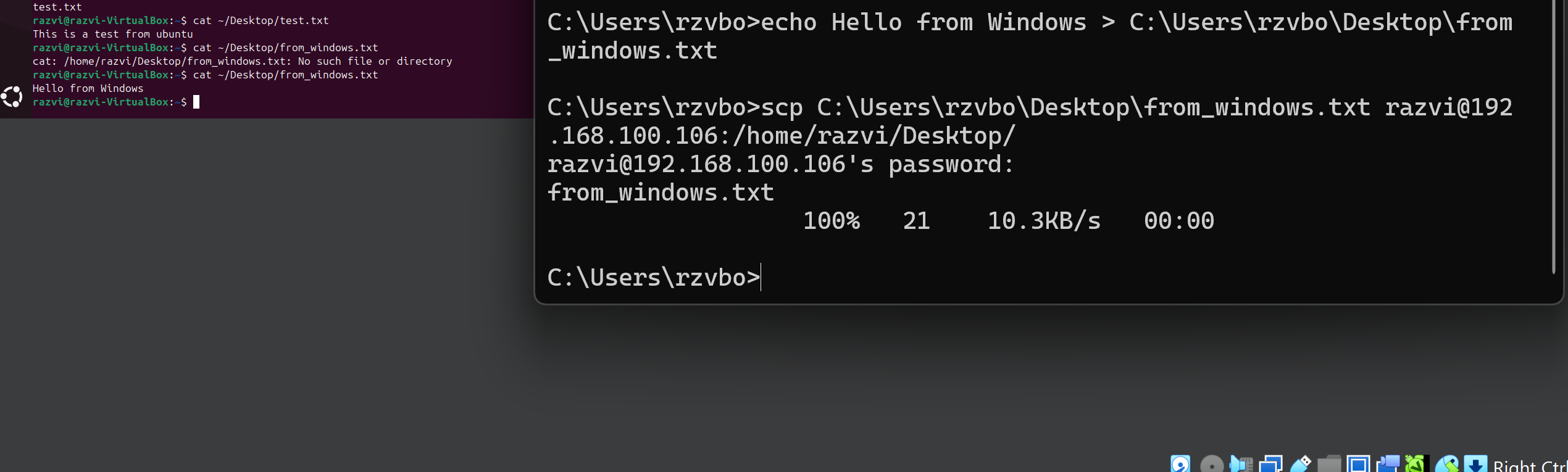
echo Hello from Windows > C:\Users\rzvbo\Desktop\from\_windows.txt

1. Copied the file from Windows to Ubuntu VM:

scp C:\Users\rzvbo\Desktop\from\_windows.txt razvi@192.168.100.106:/home/razvi/Desktop/

1. Verified the transferred file in Ubuntu:

cat ~/Desktop/from\_windows.txt



**4. Install Apache Web Server**

**Description:**  
Apache is one of the most widely used open-source HTTP web servers. It allows the Ubuntu VM to serve web pages that can be accessed from any browser on the local network.

**Steps:**

1. Updated package list and installed Apache:

sudo apt update

sudo apt install apache2

1. Checked that the service is running:

sudo systemctl status apache2

The status showed active (running)

1. Checked the VM's IP address:

ip a

1. Opened the browser on Windows and accessed:

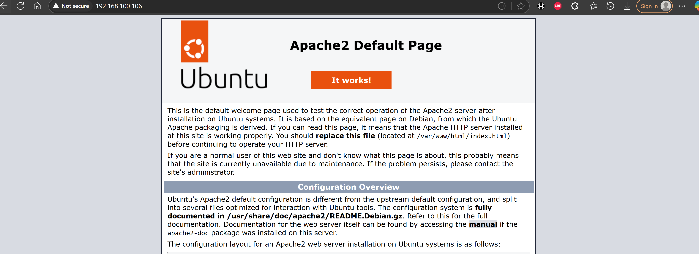
http://192.168.100.106

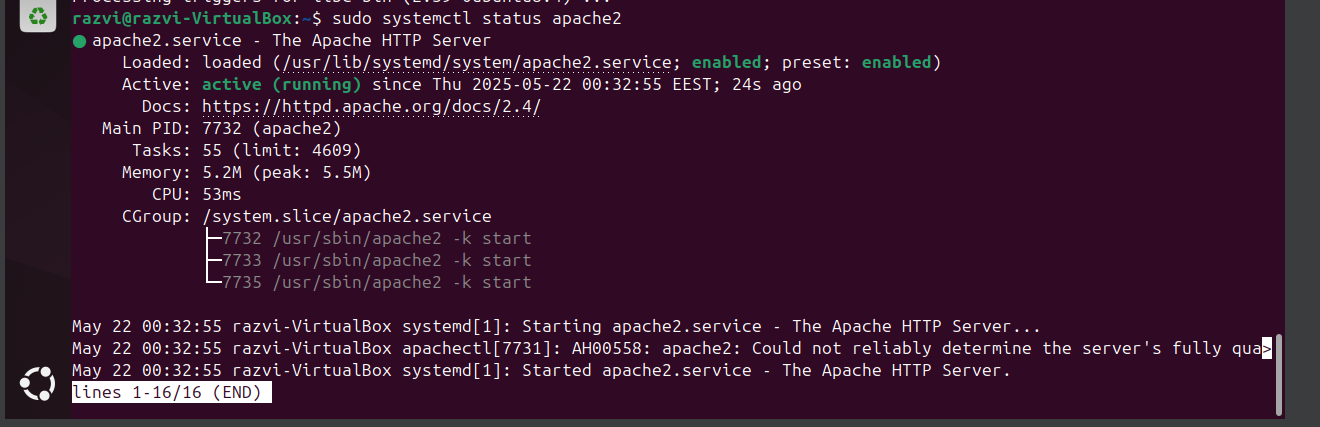
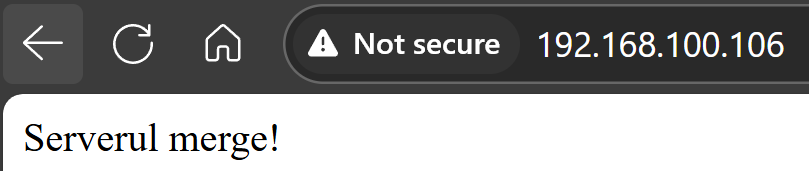
The default Apache welcome page appeared, confirming successful setup.

1. Replaced the homepage content with a custom message:

echo "Serverul merge!" | sudo tee /var/www/html/index.html

After refreshing the page, the custom message was displayed.

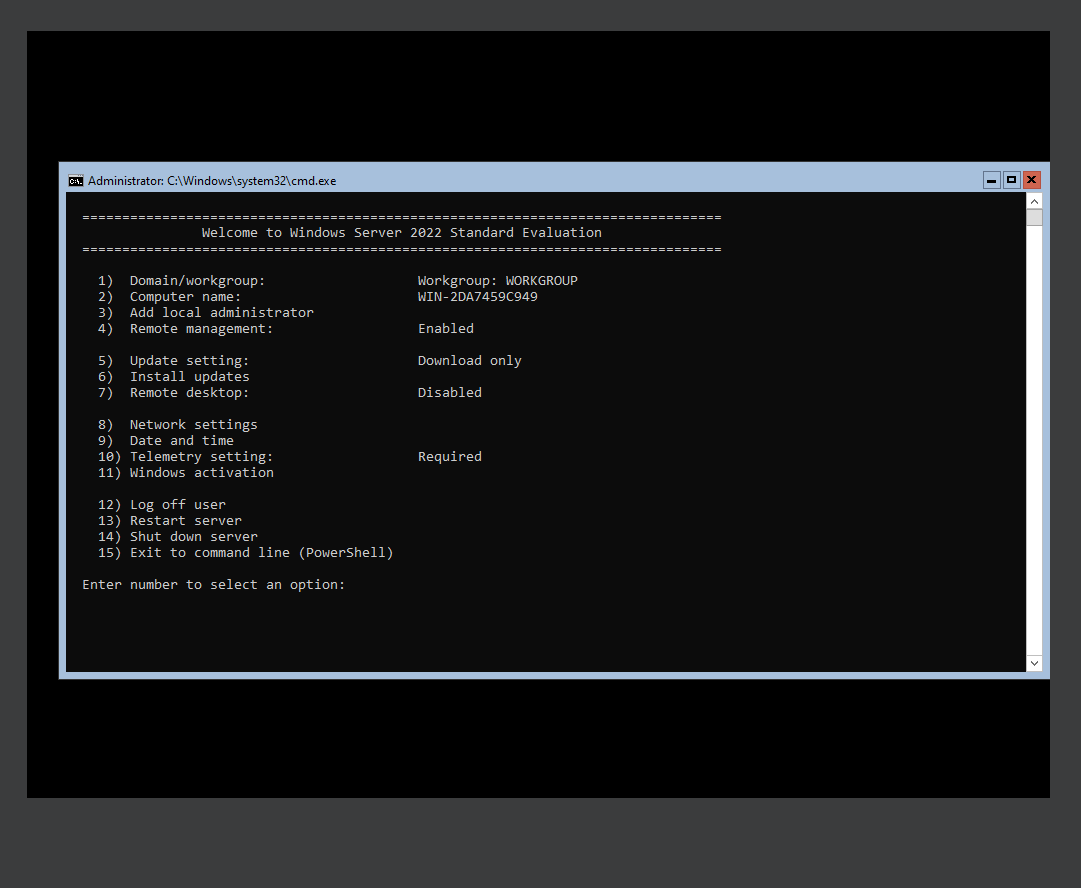


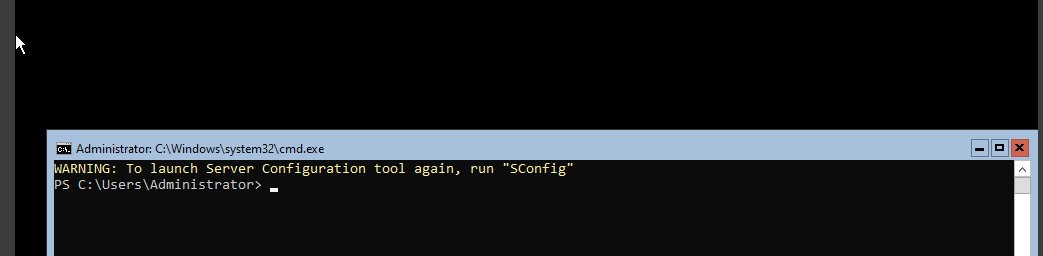


**5. Install Windows Server (2022)**

**Description:**  
Windows Server is a specialized operating system designed for servers, offering features like Active Directory, file sharing, DNS, and remote management. In this task, we install Windows Server 2022 on a virtual machine using VirtualBox.

**Steps:**

* Downloaded the ISO image of Windows Server 2022 from Microsoft.
* Created a new VirtualBox VM with the following settings:
  + OS type: Windows Server 2022 (64-bit)
  + RAM: 4096 MB
  + Virtual disk: 50 GB (VDI format)
* Mounted the Windows Server ISO under Storage > Controller: IDE > Optical Drive.
* Started the VM and booted from the ISO.
* Followed the installation wizard:
  + Selected language and keyboard layout
  + Chose Windows Server 2022 Standard Evaluation (Desktop Experience)
  + Selected “Custom: Install only Windows”
  + Formatted the disk and continued the installation
* After setup completed and the VM rebooted:
  + Set Administrator password
  + Logged in and saw the sconfig configuration menu
* Tested basic features such as:
  + Remote management = Enabled
  + PowerShell access confirmed



**6. Configure Samba File Sharing on Ubuntu**

**Description:**Samba is an open-source software that allows file and print sharing between Unix/Linux and Windows machines, using the SMB/CIFS protocol.

**Steps:**

* Installed Samba package:

sudo apt update

sudo apt install samba

* Created a shared directory:

mkdir ~/sambashare

* Added write permissions for demonstration:

sudo chmod 0777 ~/sambashare

* Edited the Samba configuration file:

sudo nano /etc/samba/smb.conf

At the bottom of the file, added:

[sambashare]

path = /home/razvi/sambashare

available = yes

valid users = razvi

read only = no

browsable = yes

public = no

writable = yes

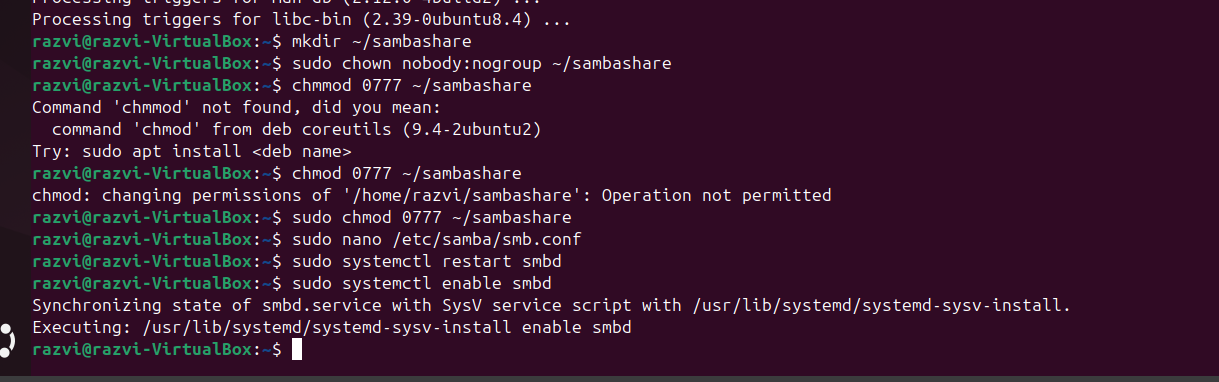
* Set a Samba password for the user:

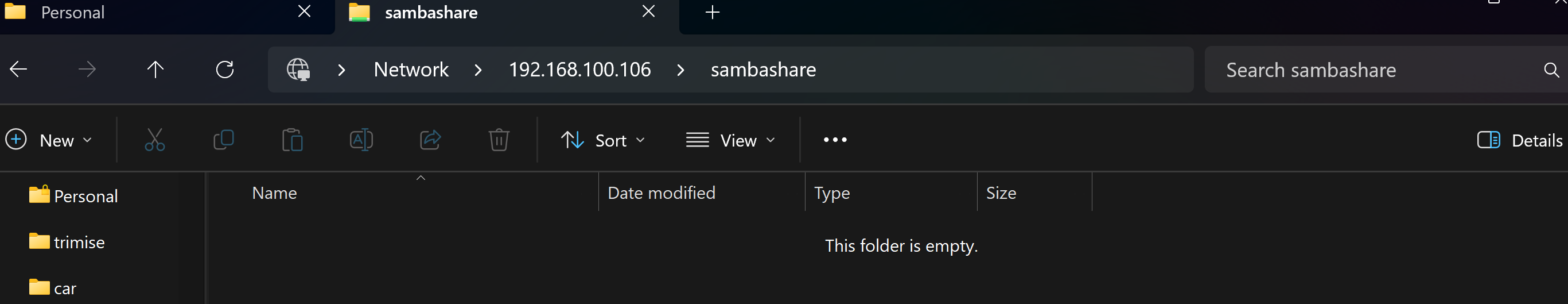
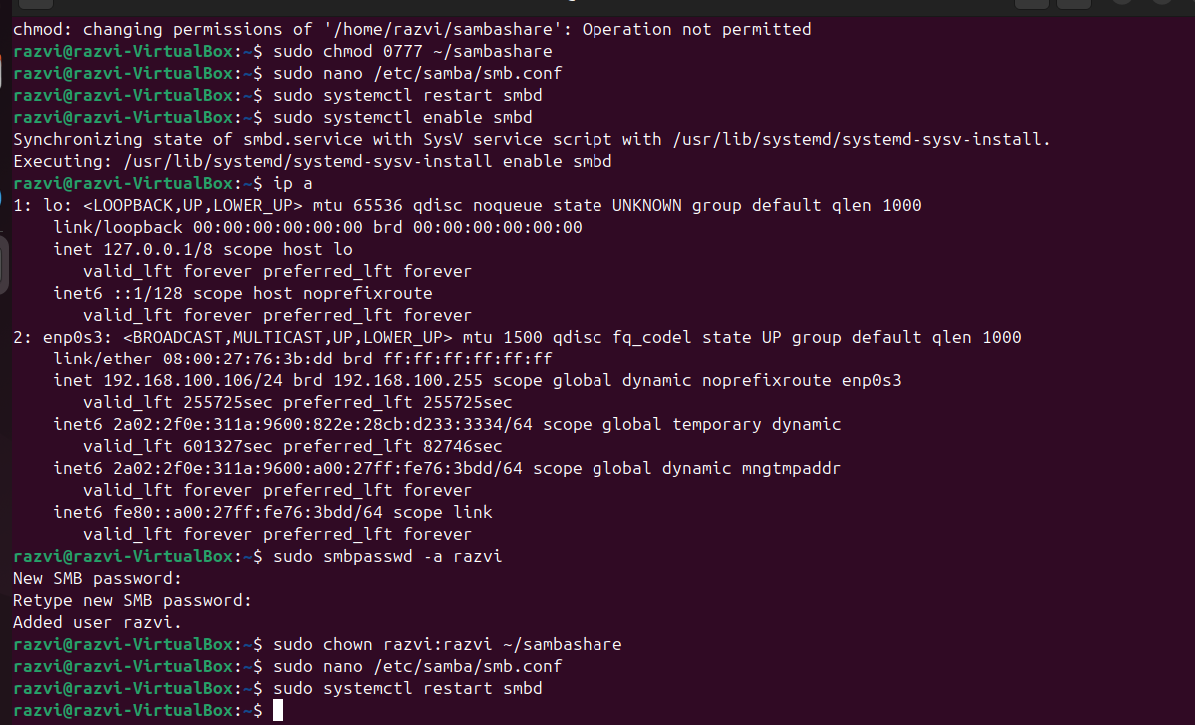
sudo smbpasswd -a razvi

* Restarted Samba service:

sudo systemctl restart smbd

* Verified the share from Windows:
  + Opened \\192.168.100.106\sambashare in File Explorer
  + Entered credentials for user razvi
  + Successfully accessed the shared folder

****

****

**7. FTP Server (vsftpd)**

**Description:**The File Transfer Protocol (FTP) allows file sharing between systems over a network. It is useful for transferring files between virtual machines and physical devices.

**Steps:**

* Installed the FTP server using:

sudo apt install vsftpd

* Verified that the vsftpd service is active and running:

sudo systemctl status vsftpd

* Created a directory to be shared via FTP:

mkdir ~/ftpfiles

* Created a test file:

echo "Hello FTP" > ~/ftpfiles/test.txt

* Set proper permissions:

chmod -R 755 ~/ftpfiles

Configuration:

* Edited the FTP config:

sudo nano /etc/vsftpd.conf

And ensured the following lines are set:

local\_enable=YES

write\_enable=YES

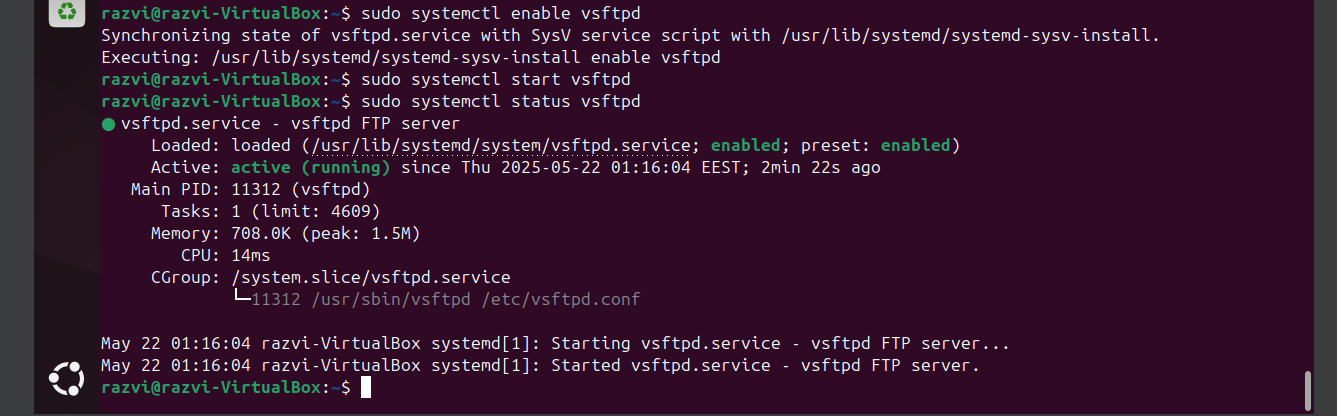
local\_umask=022

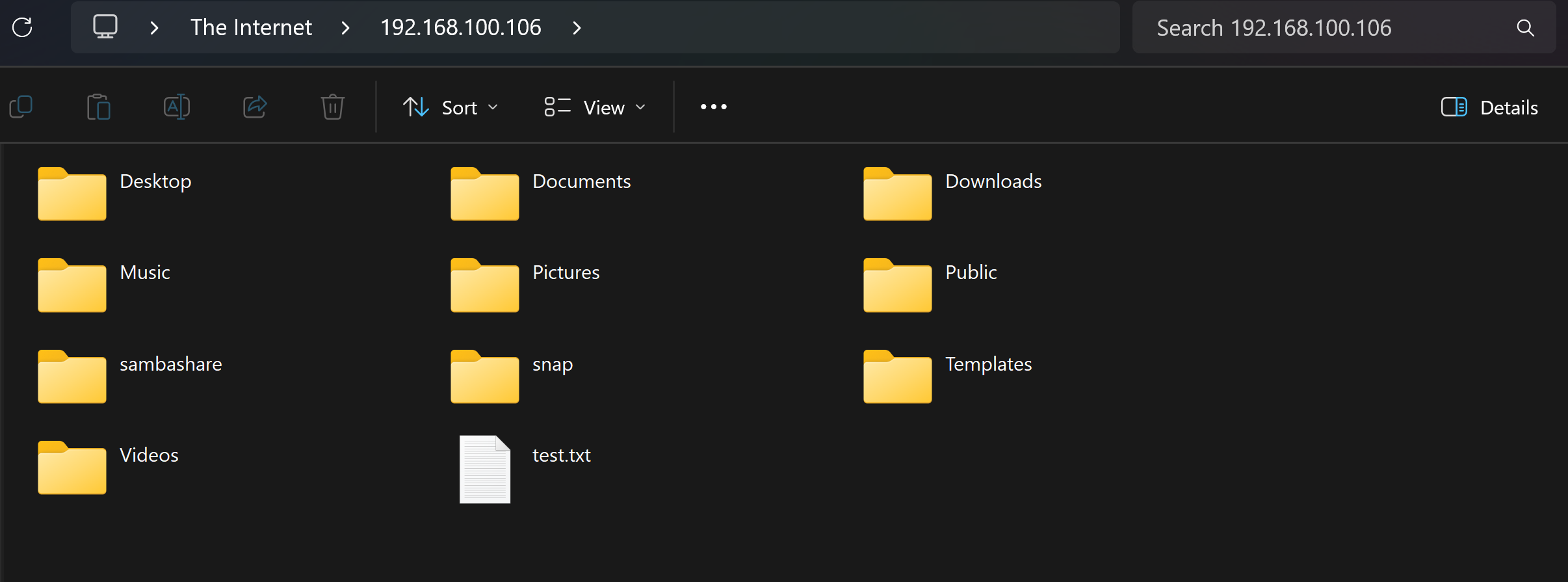
* Restarted the service:

sudo systemctl restart vsftpd

Test from Windows:

* Opened File Explorer and typed the server IP (e.g., ftp://192.168.100.106) in the address bar.
* Logged in using the Ubuntu username and password.
* The shared directory and file were visible and accessible.

****

****

**8. Firewall Configuration (UFW)**

**Description:**

The Uncomplicated Firewall (UFW) provides an easy way to manage firewall rules on Ubuntu systems. It’s especially useful for allowing only specific traffic like SSH, FTP, web, or Samba services.

**Steps performed:**

1. Enabled UFW:

sudo ufw enable

1. Allowed necessary services through the firewall:

sudo ufw allow ssh

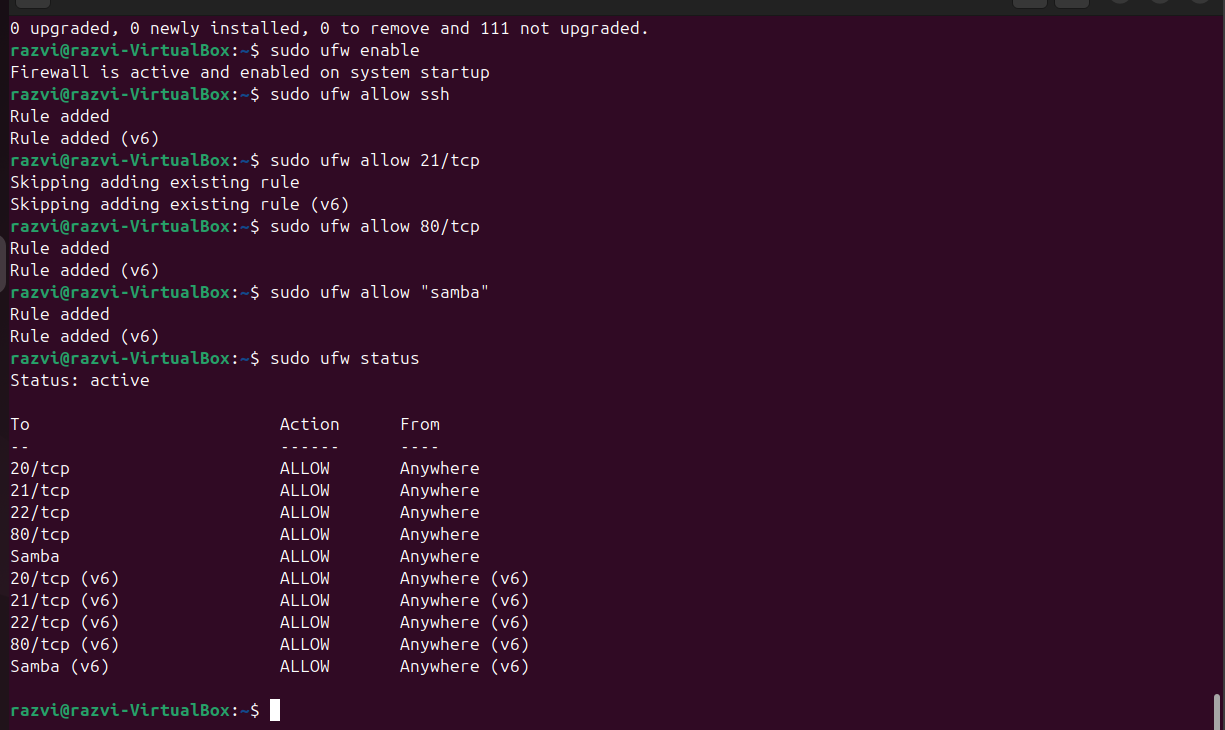
sudo ufw allow 21/tcp # FTP

sudo ufw allow 80/tcp # Apache (HTTP)

sudo ufw allow "samba" # File sharing

1. Checked firewall status to confirm all rules:

sudo ufw status

****

**9. DHCP Server Configuration (isc-dhcp-server)**

**Description:**DHCP (Dynamic Host Configuration Protocol) is a network management protocol used to dynamically assign IP addresses and other network configuration to devices.

**Steps:**

1. Install the DHCP server:

sudo apt install isc-dhcp-server

1. Configure the network interface to be used (enp0s3):

sudo nano /etc/default/isc-dhcp-server

Edit the following line:

INTERFACESv4="enp0s3"

1. Edit the DHCP configuration:

sudo nano /etc/dhcp/dhcpd.conf

And insert this configuration:

subnet 192.168.100.0 netmask 255.255.255.0 {

range 192.168.100.50 192.168.100.150;

option routers 192.168.100.1;

option domain-name-servers 8.8.8.8, 8.8.4.4;

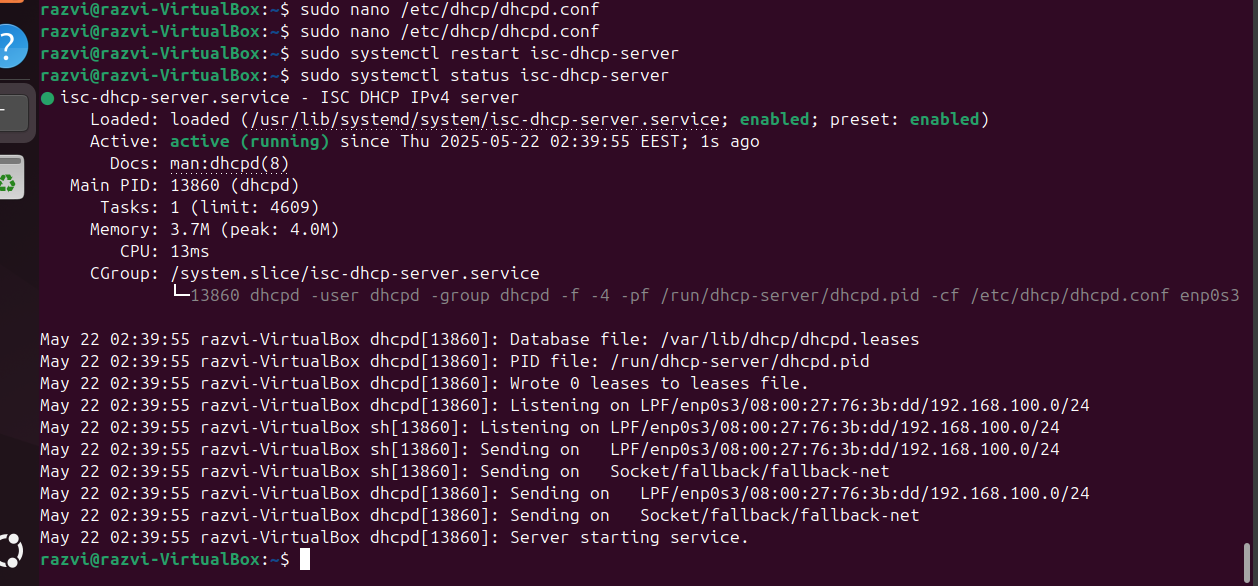
default-lease-time 600;

max-lease-time 7200;

}

1. Restart and enable the DHCP service:

sudo systemctl restart isc-dhcp-server

****sudo systemctl enable isc-dhcp-server

1. Verify the service is running:

sudo systemctl status isc-dhcp-server

**10. Proxy Server (Squid)**

**Description:**A proxy server routes and filters traffic between clients and external networks. It can be used for caching, access control, and logging network activity.

**Steps performed:**

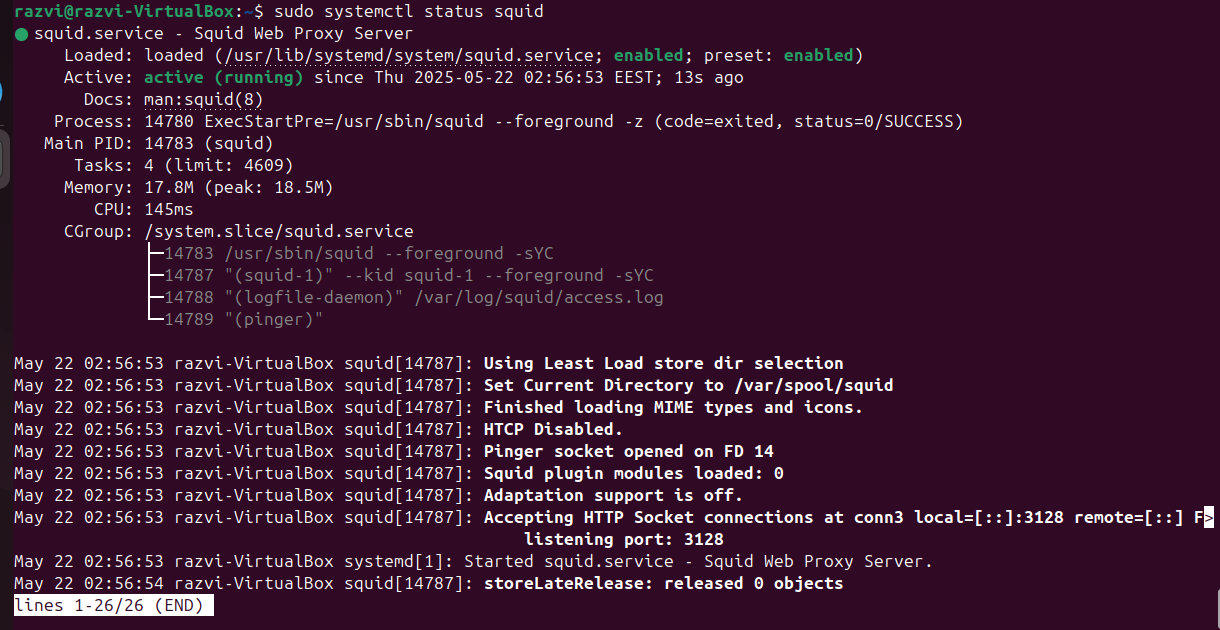
1. Installed the Squid proxy package:

sudo apt update

sudo apt install squid -y

1. Verified service status to confirm it’s running:

sudo systemctl status squid

****

**11. Network Traffic Capture (tcpdump)**

**Description:**

tcpdump is a command-line packet analyzer tool used for network diagnostics and traffic capture.

**Steps performed:**

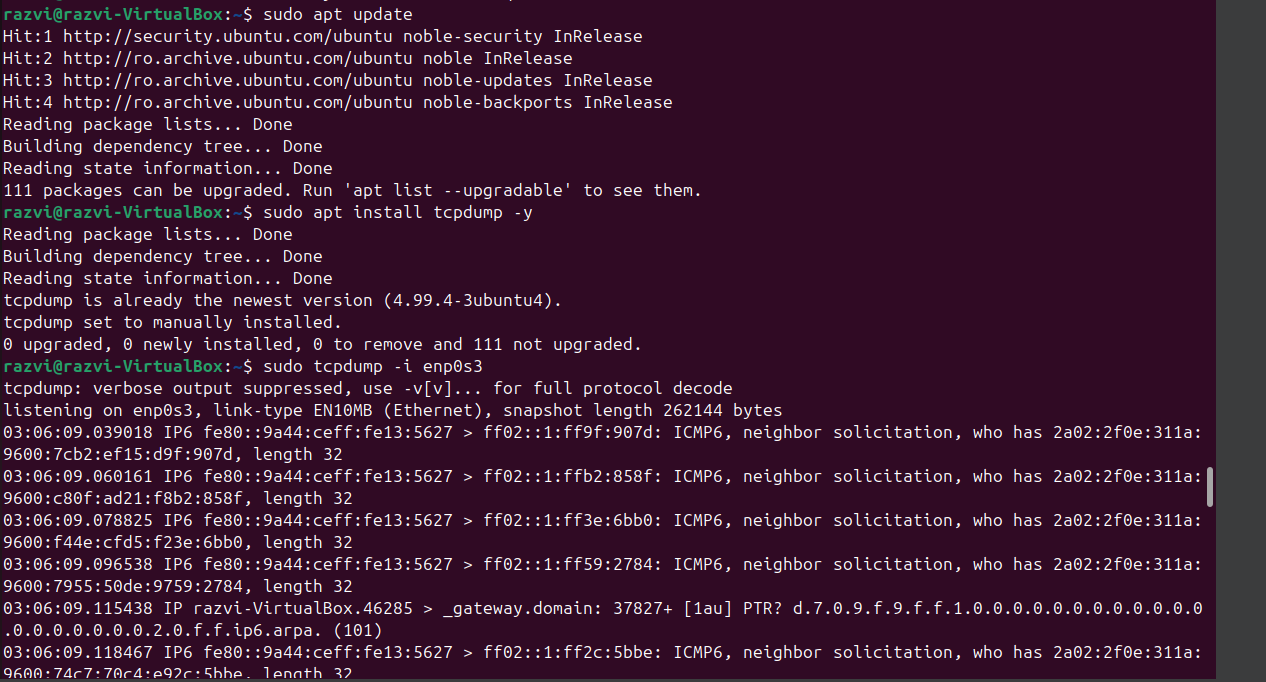
1. Installed tcpdump:

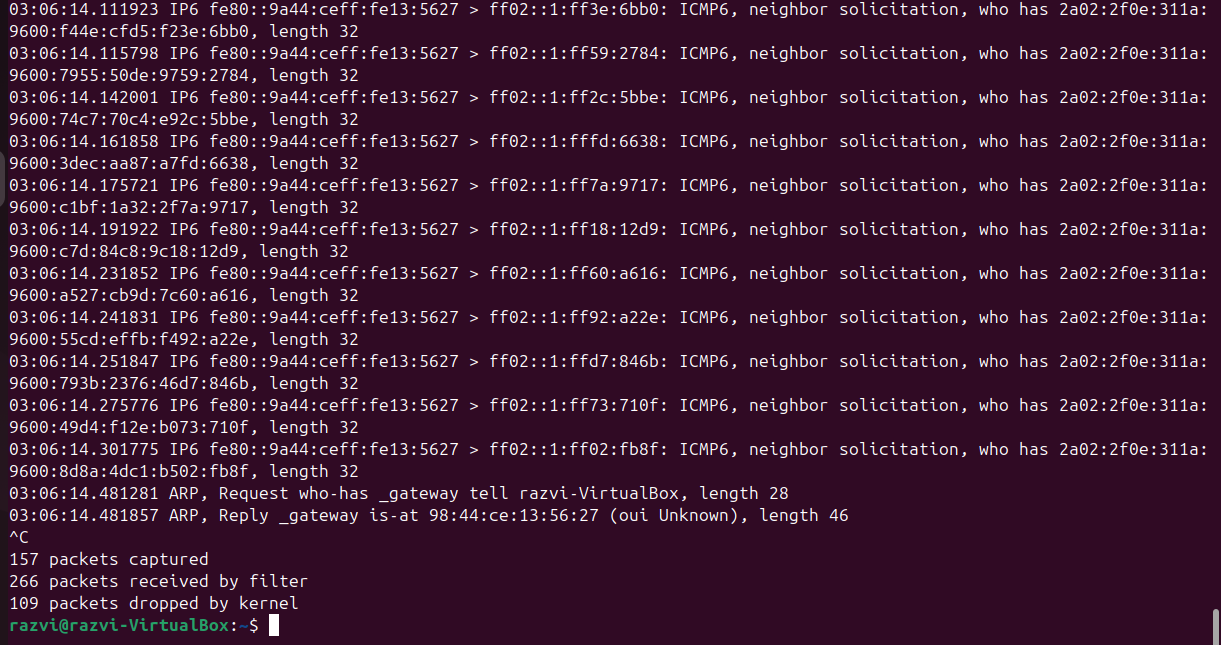
sudo apt install tcpdump -y

1. Captured packets on the primary network interface:

sudo tcpdump -i enp0s3

1. Accessed a website from the host system to generate traffic.

****

****