CCT College Dublin

Assessment Cover Page

Module Title:	Networking & Virtualisation
Assessment Title:	Proof of Concept Linux Virtual Network Project
Lecturer Name:	Michael Weiss
Student Full Name:	Publio Lima de Mello Filho
Student Number:	2024028
Assessment Due Date:	29/09/2024
Date of Submission:	29/09/2024

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Summary

The project aims to establish a virtualized network infrastructure composed by two virtual machines: a Domain Controller Server and a Web server. Through a series of systematic steps, the network infrastructure is set up to facilitate efficient communication, centralized management, and reliable service hosting.

Important Actions:

Installation of Virtual Machine: Two virtual machines are constructed, each operating a Linux Server (ubuntuserver) and a Linux Client (ubuntuclient) VM. The ubuntuserver hosts a simple DCC web page, while the ubuntuclient tests connectivity and access to the web server. Additionally, remote connectivity via SSH and basic security configurations are implemented within the Linux environment.

Server Renaming: To provide consistent identification across the network, the UbuntuServer and UbuntuClient are renamed, respectively.

Static IP Address Assignment: In order to guarantee reliable network connectivity, static IP addresses are assigned to each server. IP addresses, default gateways, and DNS server addresses must all be specified in accordance with the guidelines provided during configuration.

Testing for Connectivity: The PING command is used to confirm that there is network connectivity between the two servers. This test verifies correct configuration and makes sure packets can move over the network.

Installation of Virtual Machine

Using virtualization software, two virtual machines have been created to establish a robust network infrastructure. The first virtual machine, designated as ubuntuserver, functions as the Domain Controller Server. It is responsible for managing user accounts, security policies, and network resources. This server plays a crucial role in centralizing authentication and authorization services, facilitating streamlined network management. The second virtual machine, called ubuntclient, is configured to operate as the web server, tasked with hosting and serving web applications and content to clients accessing the network. Both virtual machines are running Ubuntu, ensuring compatibility and interoperability within the network environment. Together, they are pivotal components of the infrastructure, working synergistically to deliver reliable services and support seamless communication across the network.

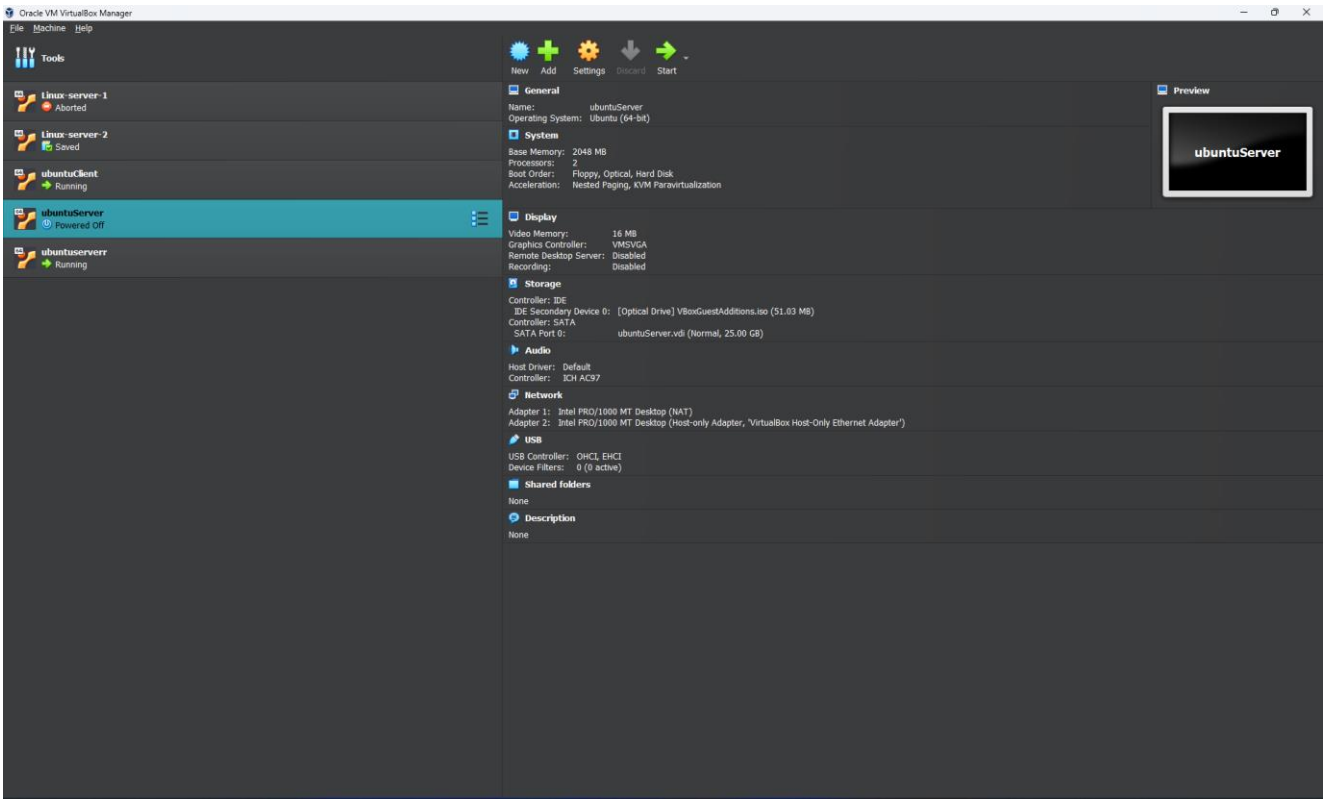

Image 1: First Virtual machine name ubuntuServer

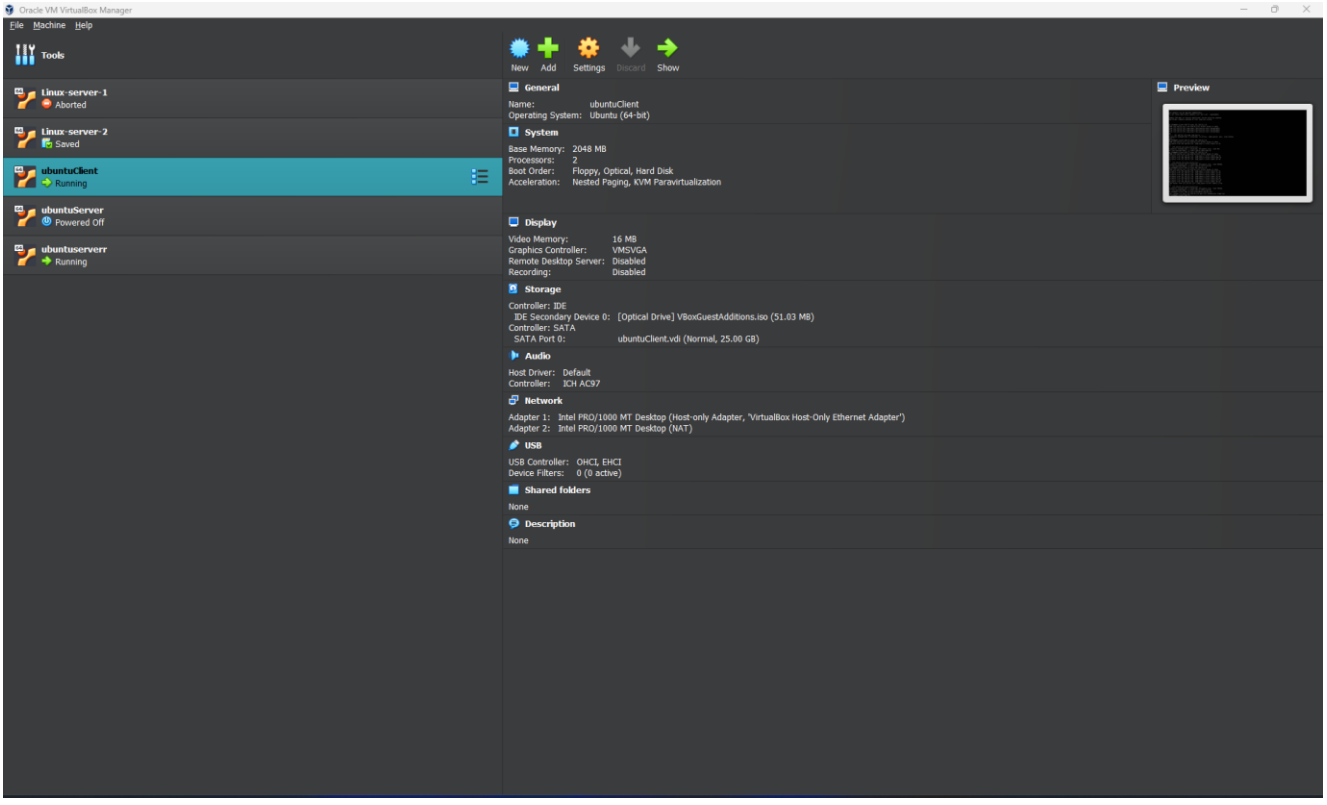

Image 2: Second Virtual machine named ubuntuClient

Part 1A: Virtual Linux Network and Web server

1 – Linux updates:

After using the command sudo apt update we have all the commands update and also tests our connectivity with the internet.

2 – Test Connectivity between both systems:

We used the internal IP addresses of the two virtual machines, ubuntuserver and ubuntclient, 192.168.56.108 and 192.168.56.109 respectively, to do a series of ping tests to check connectivity between them. Both virtual machines and the host computer can communicate with each other when the Host-Only network setup is used. The two virtual machines (VMs) successfully responded to ping requests, indicating that their internal network connection is intact. Additionally, we confirmed that there was no disruption in communication between the two Linux virtual machines (VMs) and the host computer by using ping. Furthermore, we verified that the virtual machines could access the internet without any problems by pinging popular websites like Google to test external connectivity. All of these tests show that the network configuration is stable and enables dependable internal and external connectivity.

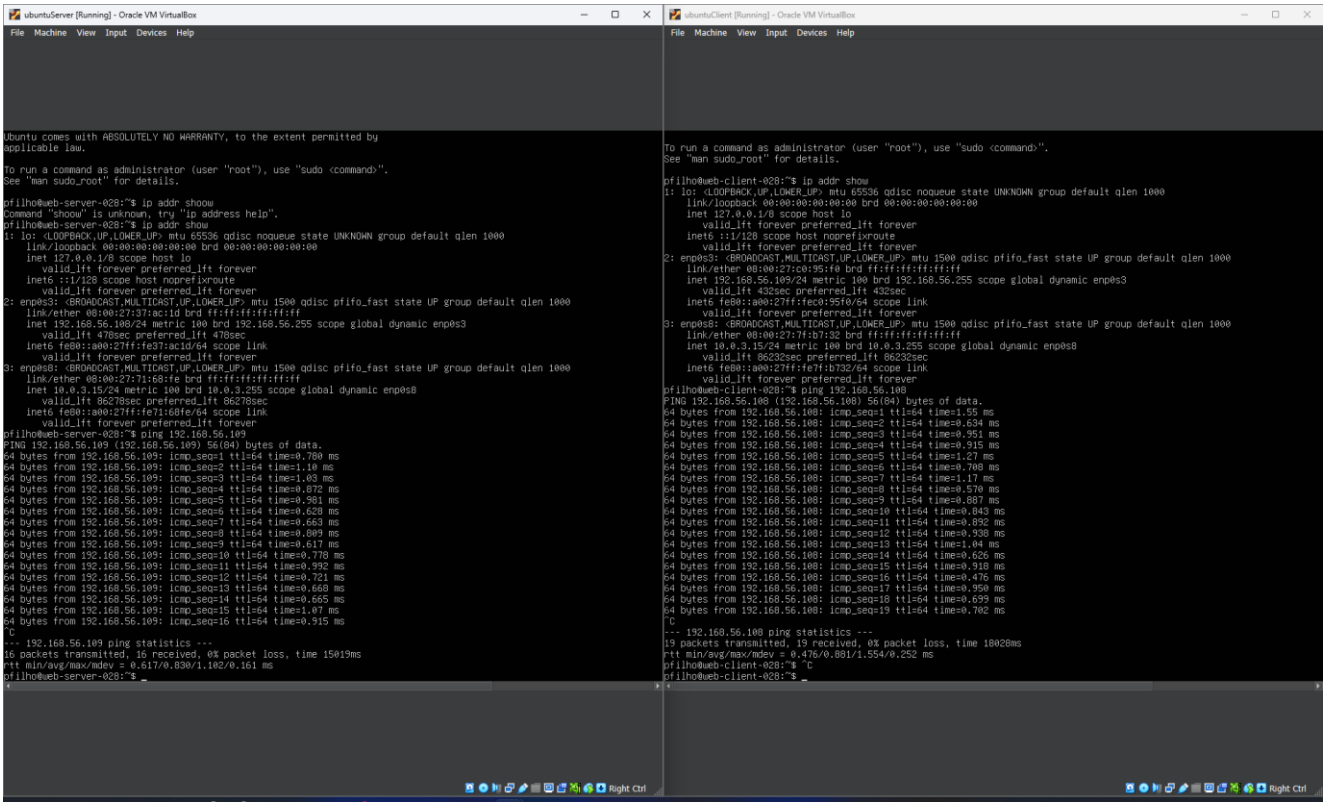

Image 3: Machines pinging to each other

3 – Testing Access to the DCC Website:

We tested access to the hosted web page labeled "DCC Website Under Construction" after successfully installing Apache on the Ubuntu server. To achieve this, open the Lynx browser on the Ubuntu client virtual machine and go to the Ubuntu server's website. The web page appeared in the Lynx browser when the URL was entered, indicating that the Apache web server was set up correctly and was serving content. This test made sure that clients could successfully access the hosted website and confirmed the web server's operation within the network infrastructure.

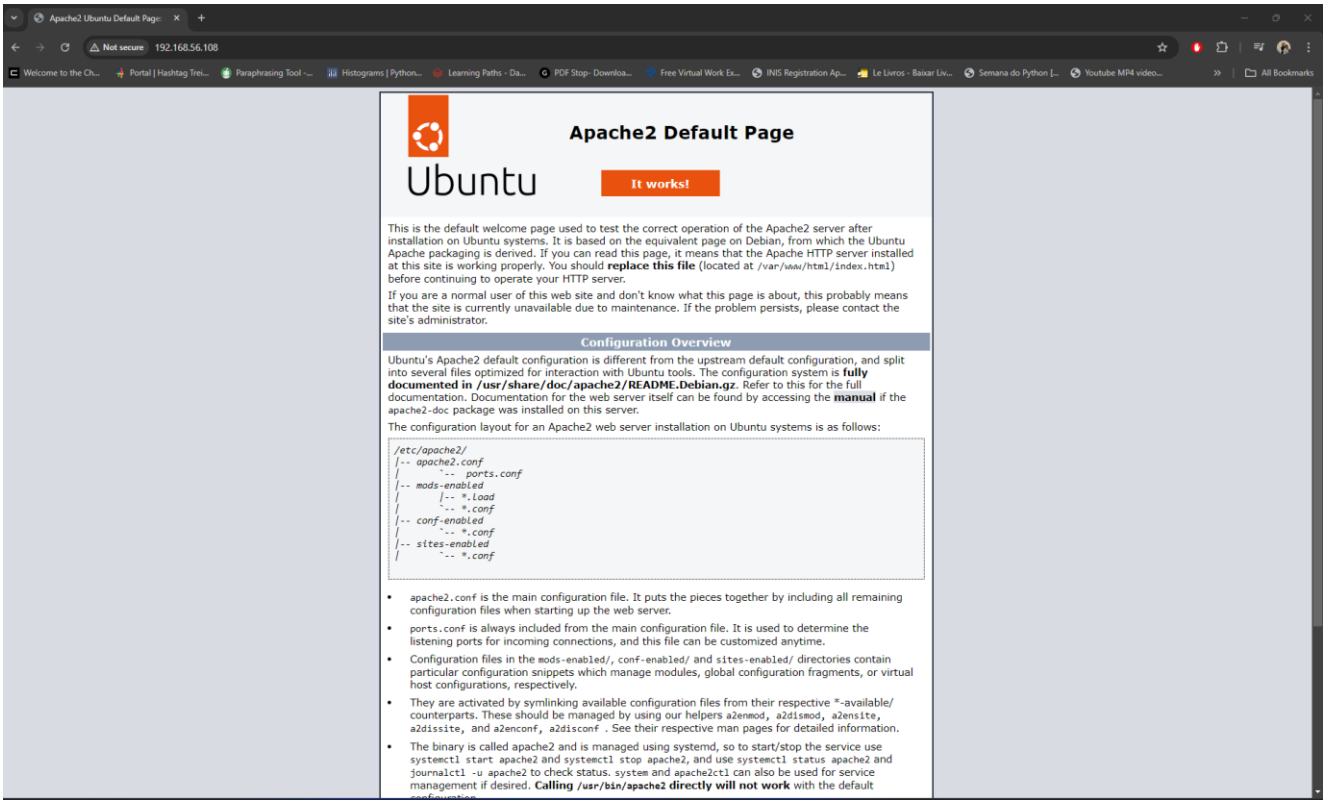

Image 4: Webserver working.

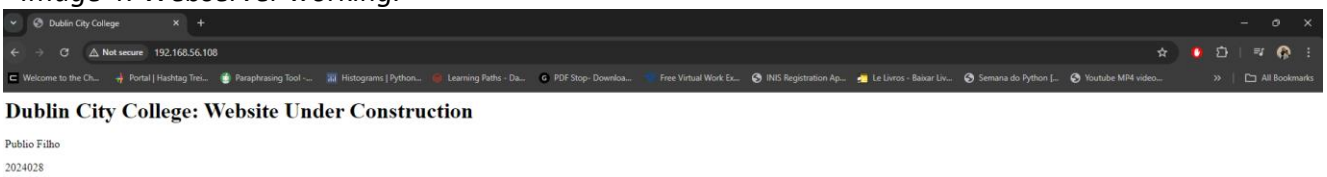

Image 5: Web page edited to show Under construction.

4 – Testing Access to the DCC Website from Host Operating System:

We tested the "DCC Website Under Construction" from the host operating system (Windows or Mac) in order to confirm its accessibility even further. We were able to visit the hosted web page by starting a web browser on the host computer and typing in the URL for the Ubuntuserver.

Through this test, it was verified that the Ubuntu server's Apache web server is operational and reachable from other networked devices. The fact that the host environment can access the "DCC Under Construction" page demonstrates how well the web server is configured and supports the reliability of the network infrastructure.

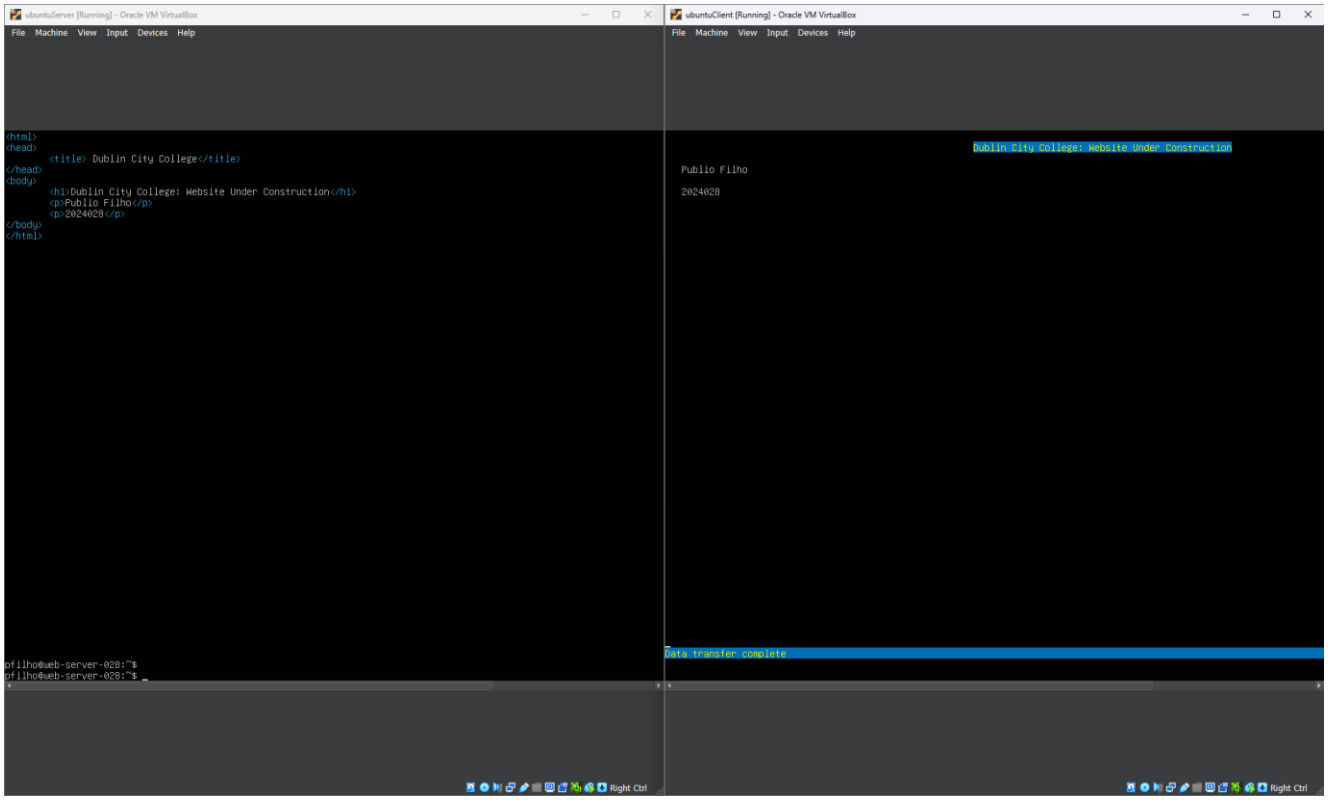

Image 6: Webpage accessed by the unbuntuclient.

5 – Monitoring ICMP Traffic Using Wireshark:

We used Wireshark to record network traffic and ping the Ubuntuserver IP address from the host PC (Windows or Mac) in order to further examine network connectivity. We saw the ICMP traffic between the host operating system and the Ubuntu web server as the ping function ran. This data showed that the two devices were successfully communicating and responding to each other. As proof of the network communication, we took a screenshot of the Wireshark interface showing the ICMP packets for documentation reasons. This study validates that the host can easily communicate with the virtual machine and shows how successful the network arrangement is.

Image 7: Ping in my cmd.

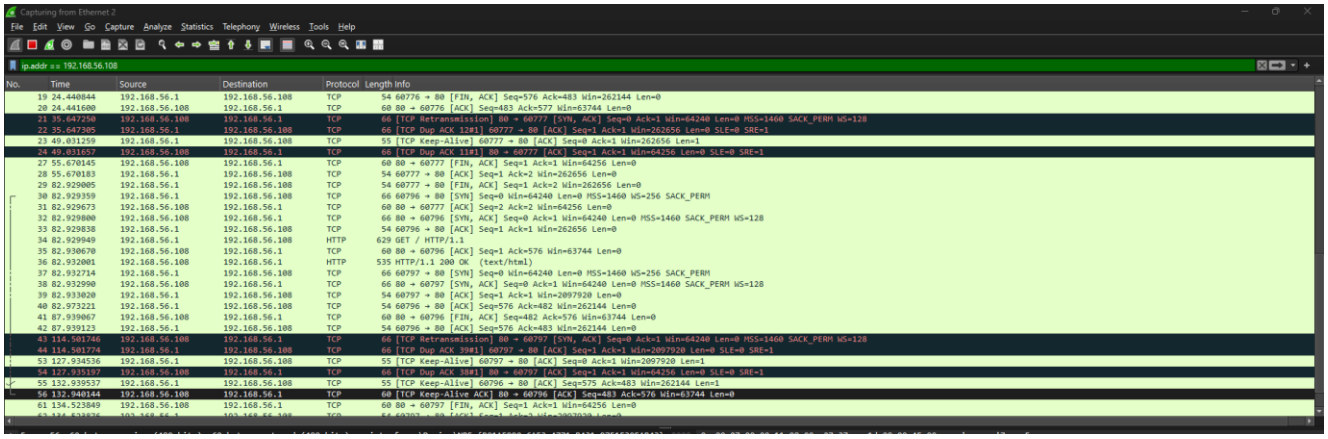

Image 8: Pinging my ubuntuServer.

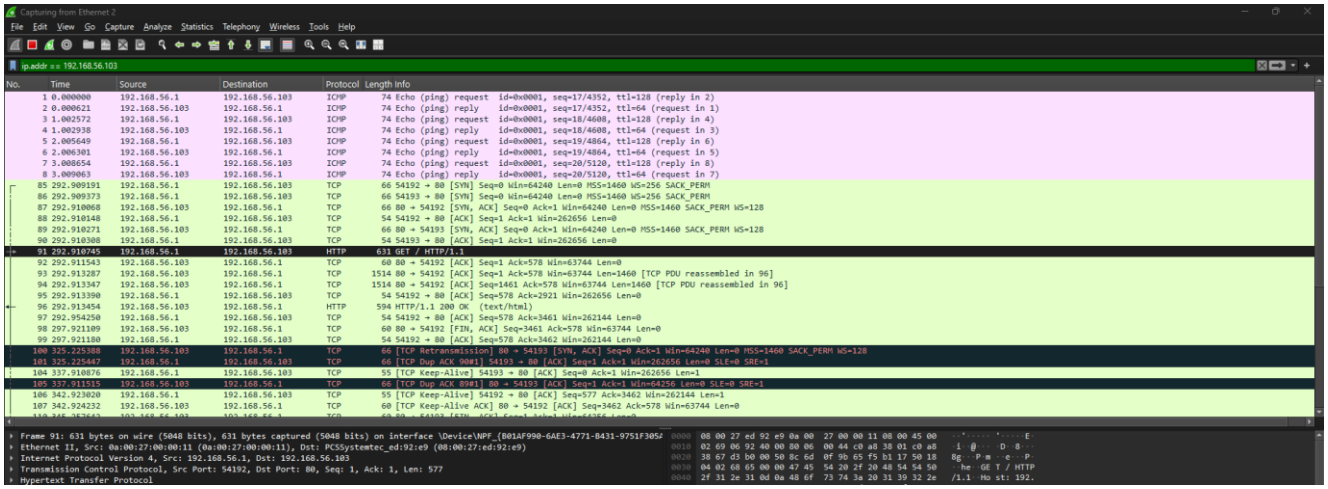

Image 9: Pinging my unbuntuClient.

Part 1B: Hostname Management: Renaming Linux Servers

We were able to successfully rename the Linux servers inside the network infrastructure in order to comply with DCC's requirements. Ubuntuserver and Ubuntuclient were renamed to web-server-028 and web-client-028, respectively, with "028" standing for my student number's final three digits. In order to guarantee that these modifications are irreversible, the relevant configuration files were updated, enabling the servers to keep their new hostnames even after rebooting. We took screenshots of the modified hostnames on both Ubuntu servers to record this procedure and ensure that the renaming was carried out properly. This step is vital for maintaining organized network management and guaranteeing clarity in server identification.

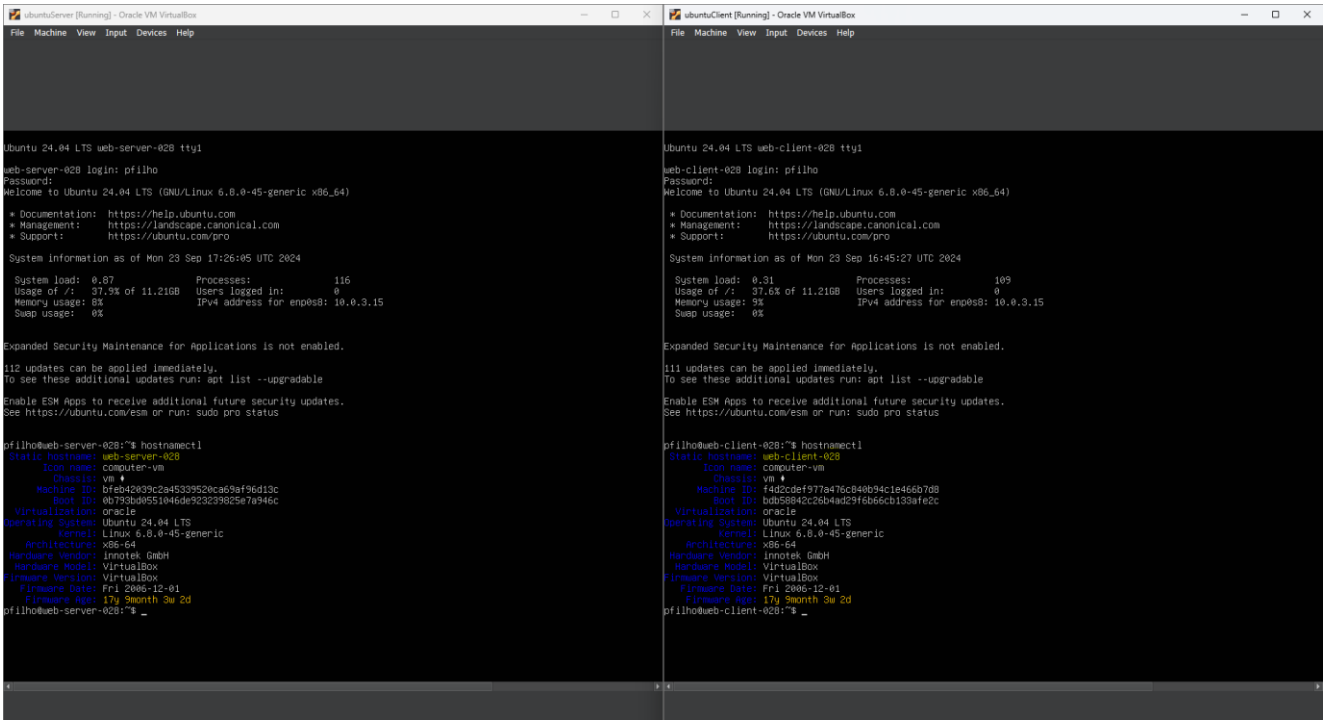

Image 10: Hostnames changed permanently.

Part 1C: IP Address Management: Configuring Permanent IP Addresses

We used the nano text editor to alter the Netplan configuration file in order to create a reliable network setup for the Linux servers. We gave the Ubuntu server a permanent IP address of 192.168.56.100/24. We implemented the new settings to make sure they take effect after making the required changes to the configuration files to make this change permanent. Next, we used the ping command to check connectivity between the two Linux virtual computers. The fact that both VMs could successfully communicate and that the IP address setting was accurate was validated by the successful ping responses. Ensuring that the servers are reachable within the network and preserving dependable network interactions depend on this phase.

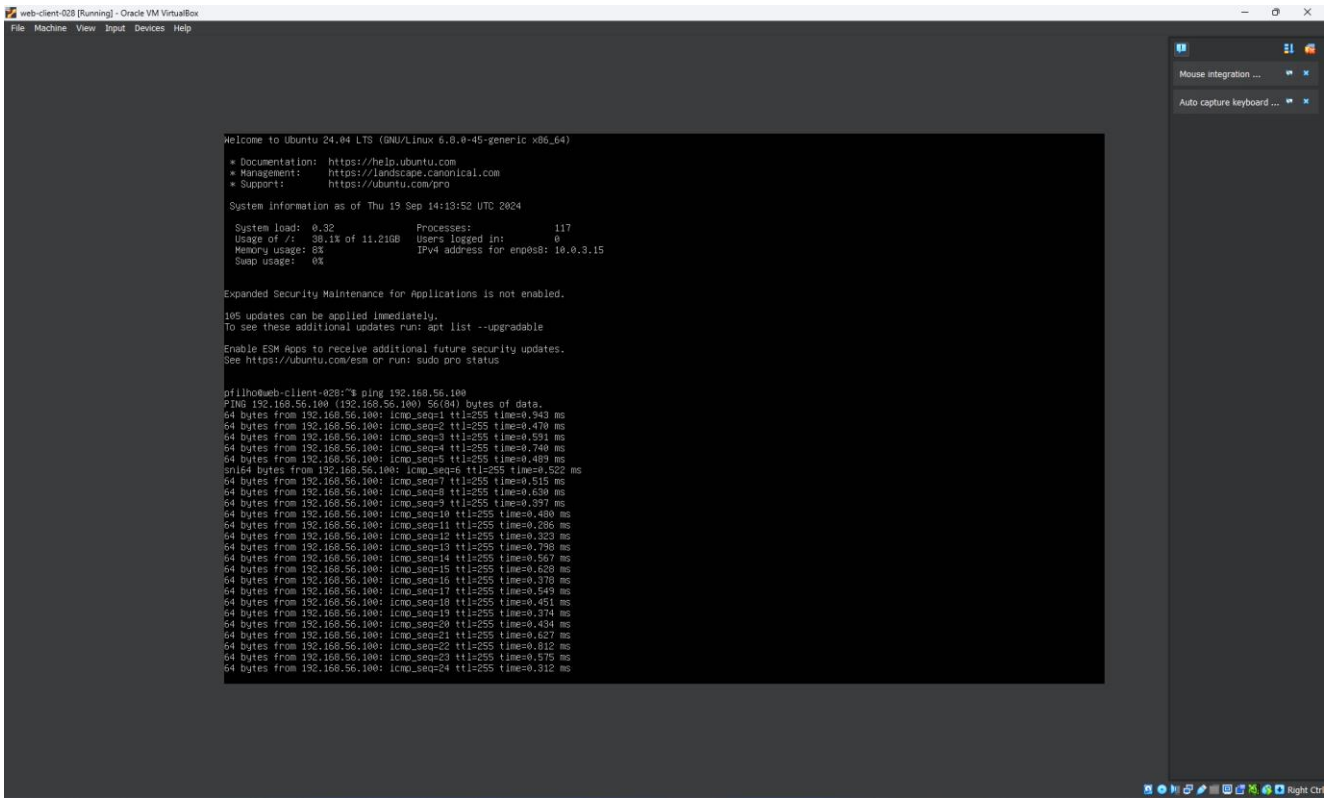

Image 11: Ip address changed and pinging with the UbuntuClient.

Part 1D: IP Address Management with DHCP Configuration

We installed and set up a DHCP server on web-server-028 to enable dynamic IP address allocation for the web-client-123 virtual machine. With a DHCP range of 192.168.56.150 to 192.168.56.200, the DHCP server was configured to administer the subnet 192.168.56.0/24. By doing this, the web-client-028 computer is guaranteed to get 192.168.56.150, which is the first IP address in the pool that becomes accessible. Furthermore, we set up 192.168.56.1 as the default gateway and DNS address to facilitate easy network routing and name resolution. Following configuration completion, the client was able to dynamically obtain the IP address from the DHCP server, validating that the network and DHCP functions were configured correctly.

Part 2: Using SSH for Secure Data Transfer Between Servers

We utilized SSH to create a remote connection between the two virtual machines in order to facilitate safe data transfer. We set up PuTTY (for Windows) on the host machine in order to establish a remote SSH connection to web-server-028. After entering the IP address of the server into PuTTY, we were able to log into web-server-028 remotely from the host computer. This proved that SSH was set up correctly and was operating as planned, enabling safe connection between the host and Ubuntu server. The successful remote connection is verified by a screenshot taken from PuTTY of the SSH session.

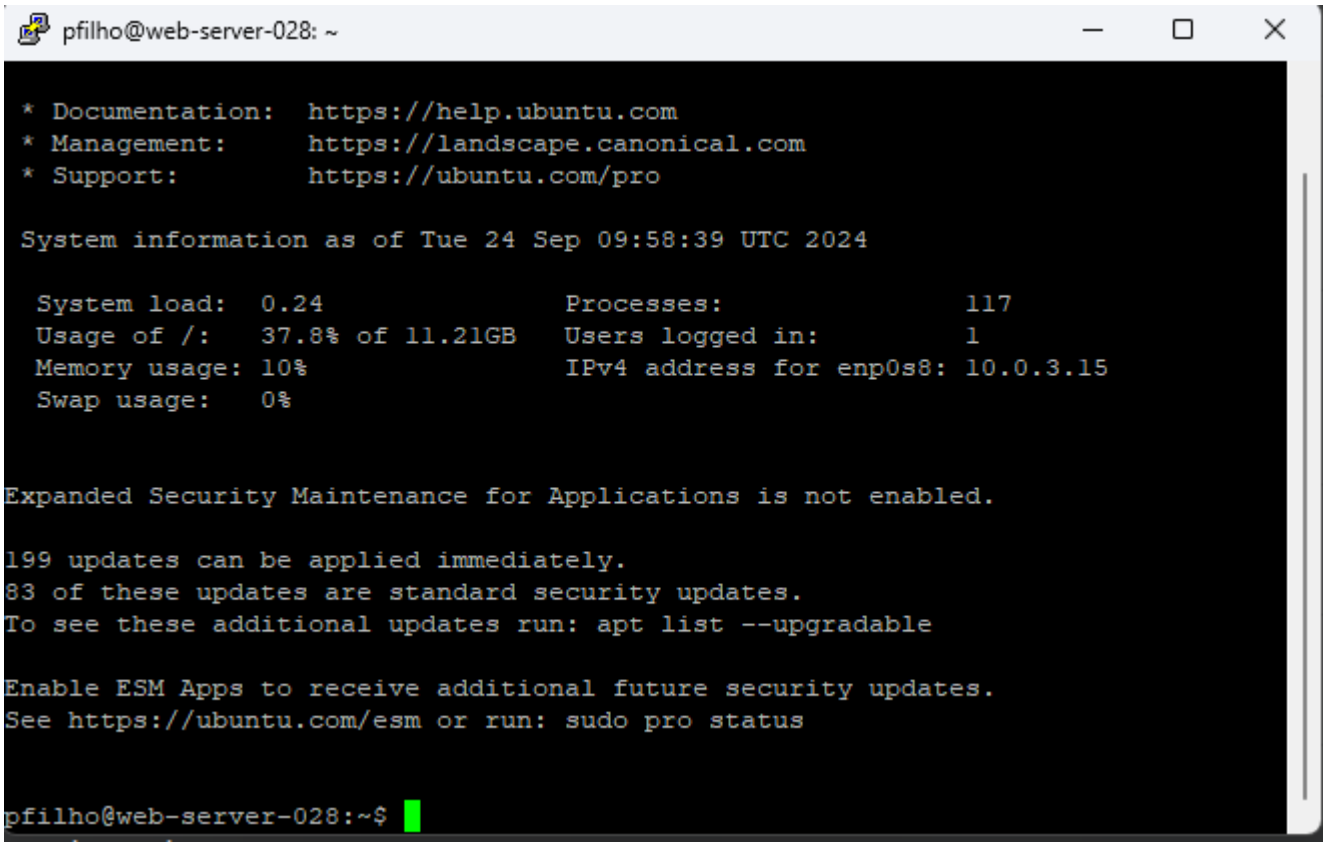

Image 12: UbuntuServer connected to Putty.

We utilized Wireshark on the host operating system to confirm the encryption of communication between the web-client-028 virtual machine and web-server-028. Wireshark was configured to record network activity when the client was connected to web-server-028 via SSH. We were able to ensure that all data exchanged over the connection was securely encrypted during the session by observing encrypted SSH traffic between the client and the server. To identify the precise packets displaying the SSH encrypted traffic, a screenshot was obtained. The correct security of data exchanges between the client and server is confirmed by this examination using SSH.

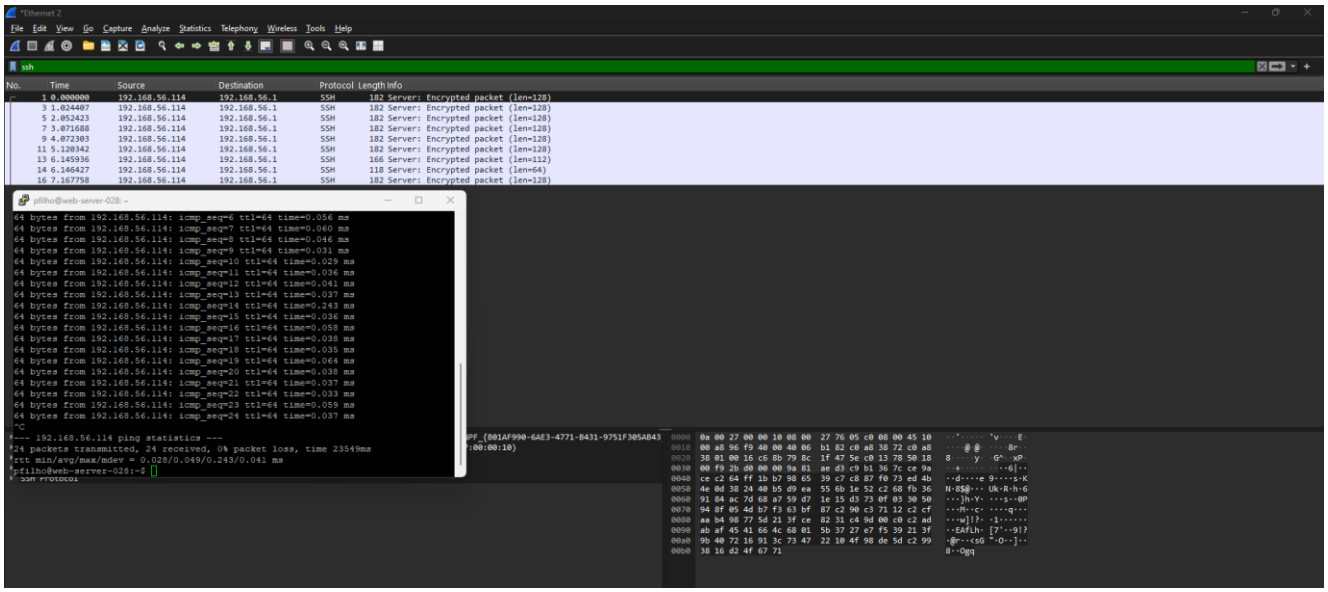

Image 13: UbuntuServer pinging in Putty confirmed in Whiteshark.

Part 3: Security Configuration

In this task, I configured the Uncomplicated Firewall (UFW) on Ubuntu to control both HTTP and SSH traffic. UFW is a user-friendly firewall utility that allows the creation of host-based firewall rules. By default, UFW is disabled, and it requires manual configuration to control network traffic. The following steps outline how I blocked and allowed both HTTP and SSH traffic, along with evidence in the form of screenshots to demonstrate the firewall's functionality.

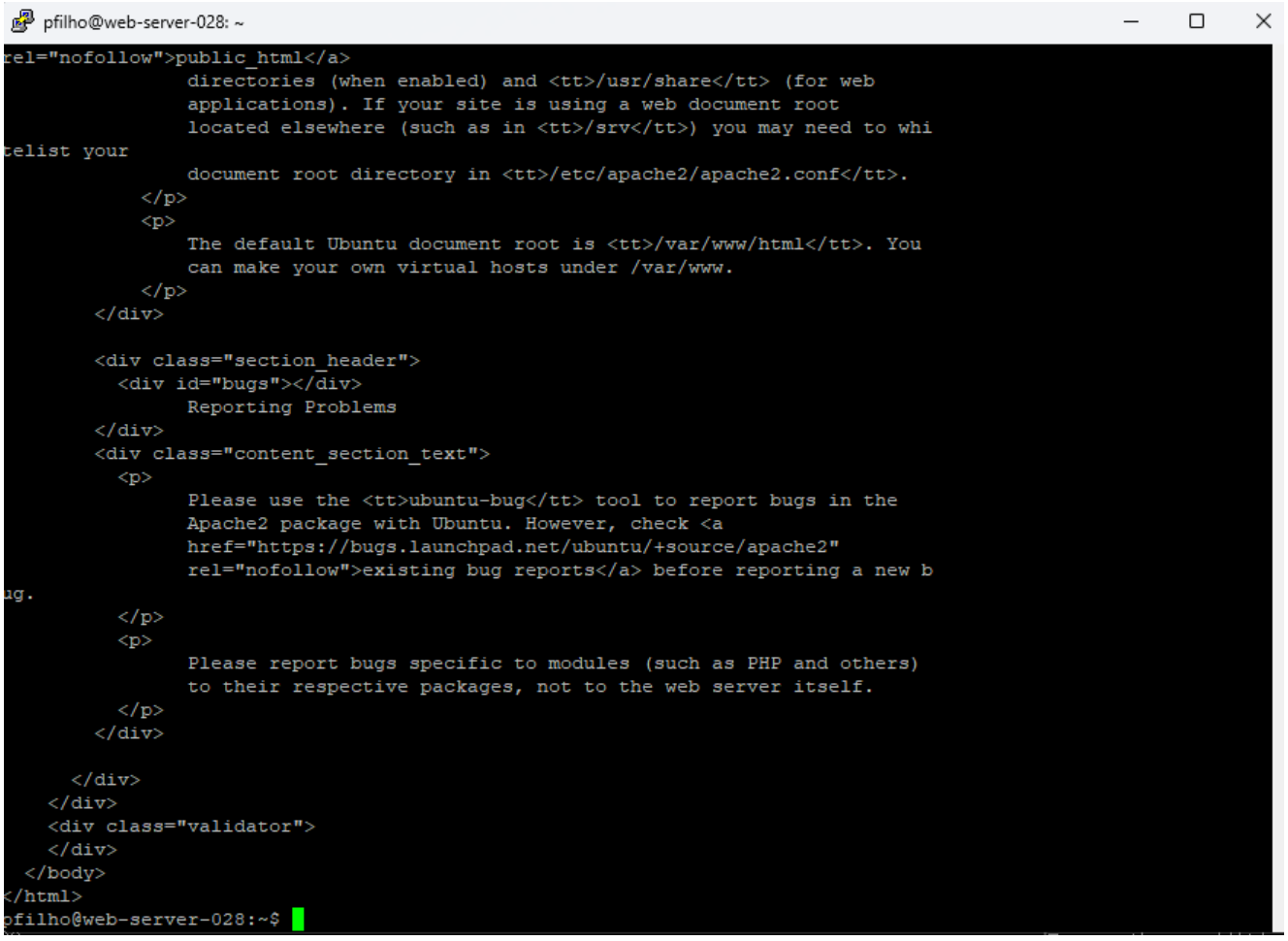

Image 14: Connection HTTP working

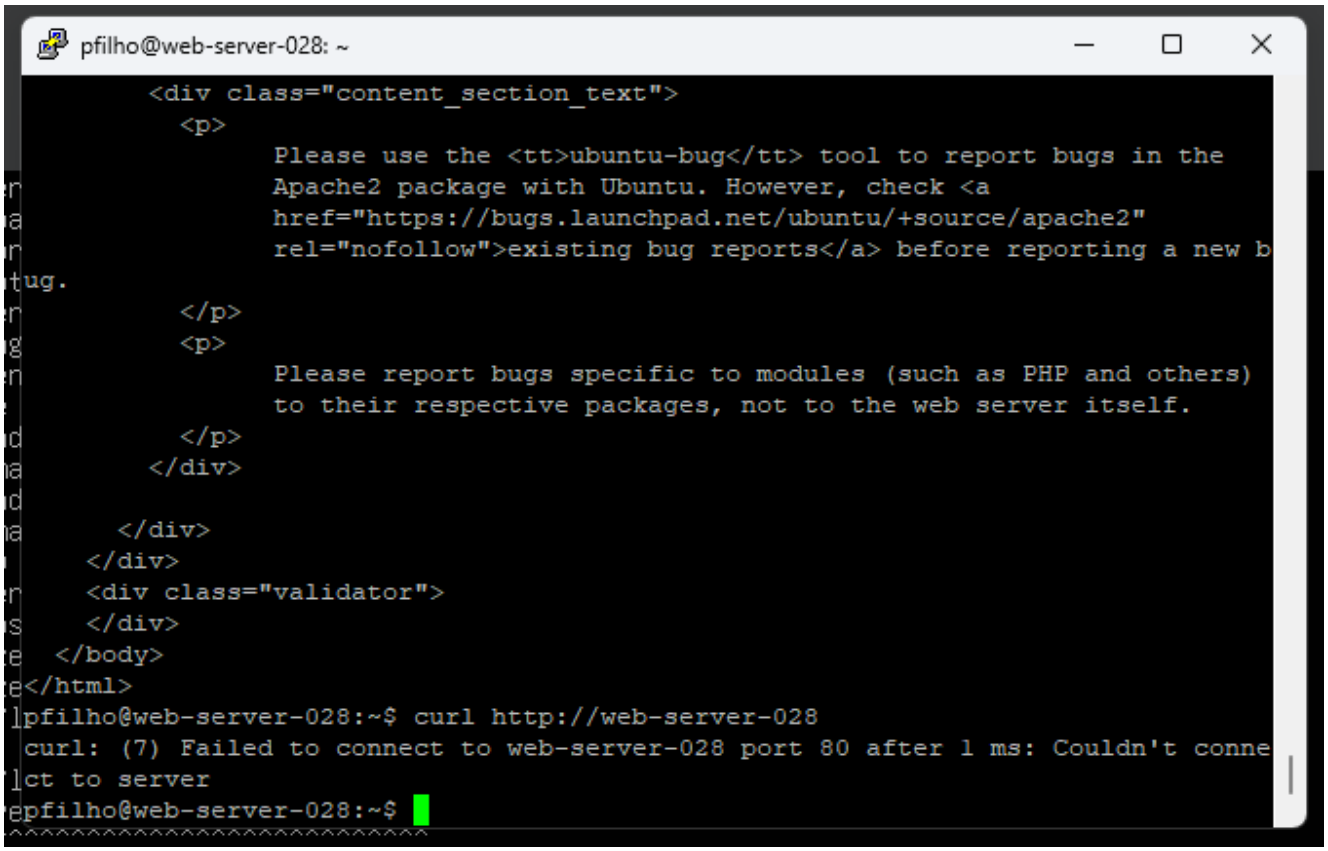

Image 15: Connection HTTP denied

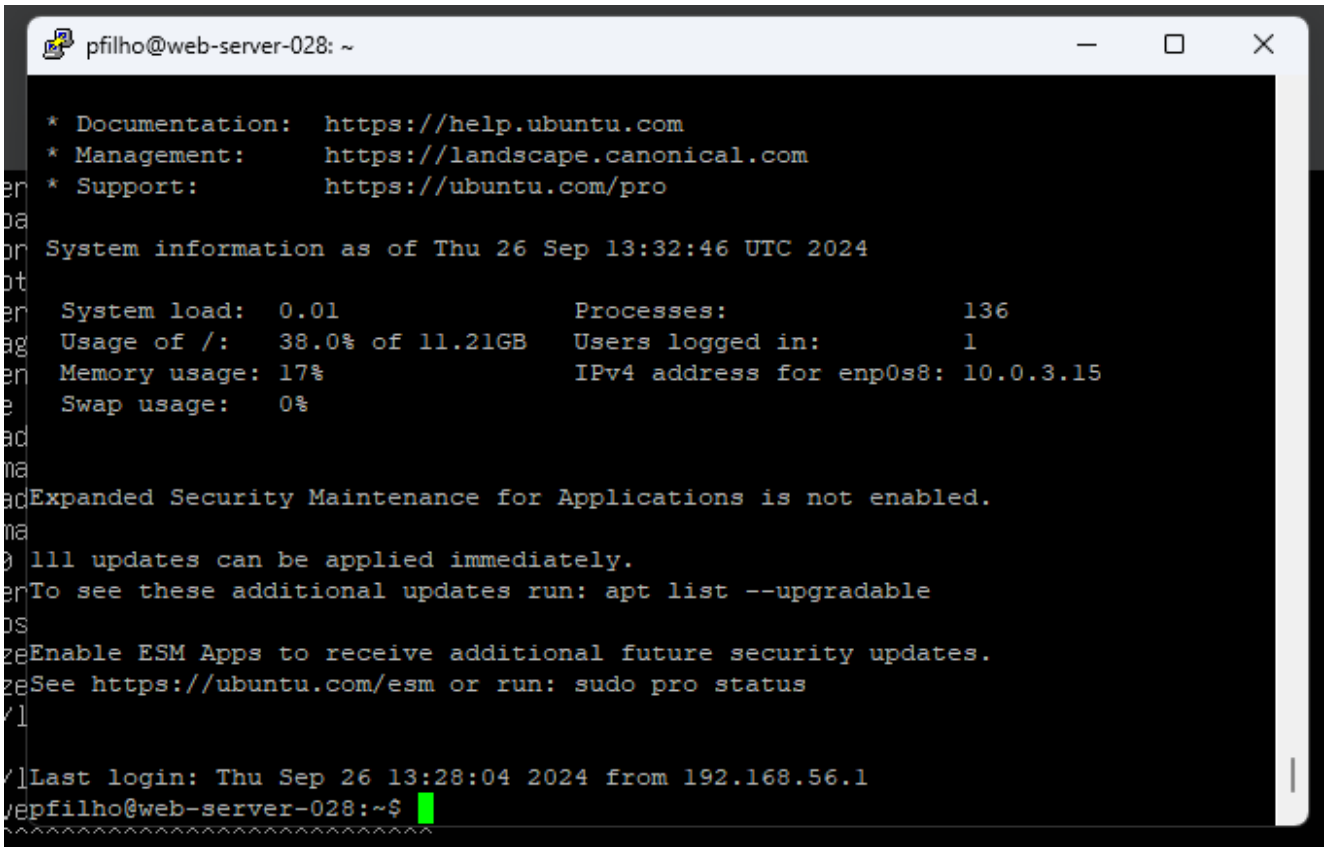

Image 16: Connection SSH working

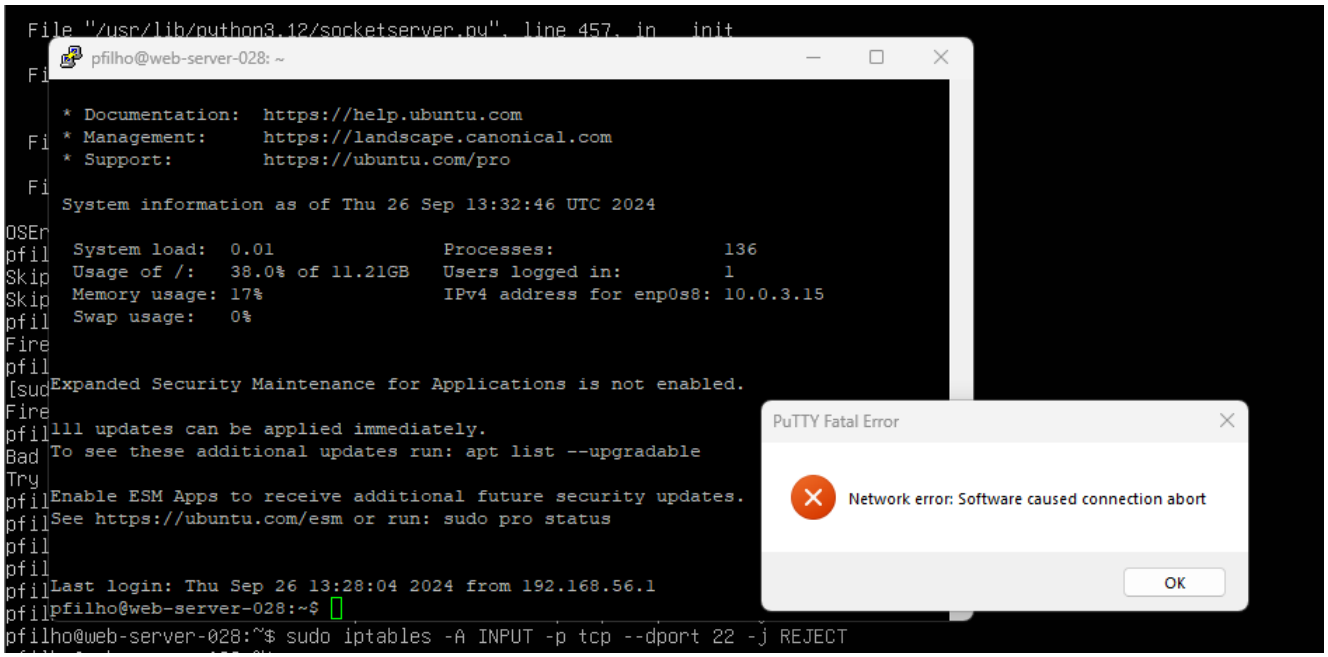

Image 17: Connection SSH denied

Part 4: The SAMBA file server

On web-server-123 (Ubuntu), we set up and installed Samba in order to facilitate file sharing via the SMB protocol between Ubuntu and Windows PCs. Nano was used to add a test file, my-samba-file.txt, to the shared folder samba-folder. To ensure secure access, permissions were set to need a password from users in order to access the folder. The host computer was able to access the Samba share with success, and we verified that we could add new files and modify existing ones, such my-samba-file.txt. Samba is an essential tool for mixed OS situations since it allows cross-platform file and printer sharing.

Image 18: Giving permission.

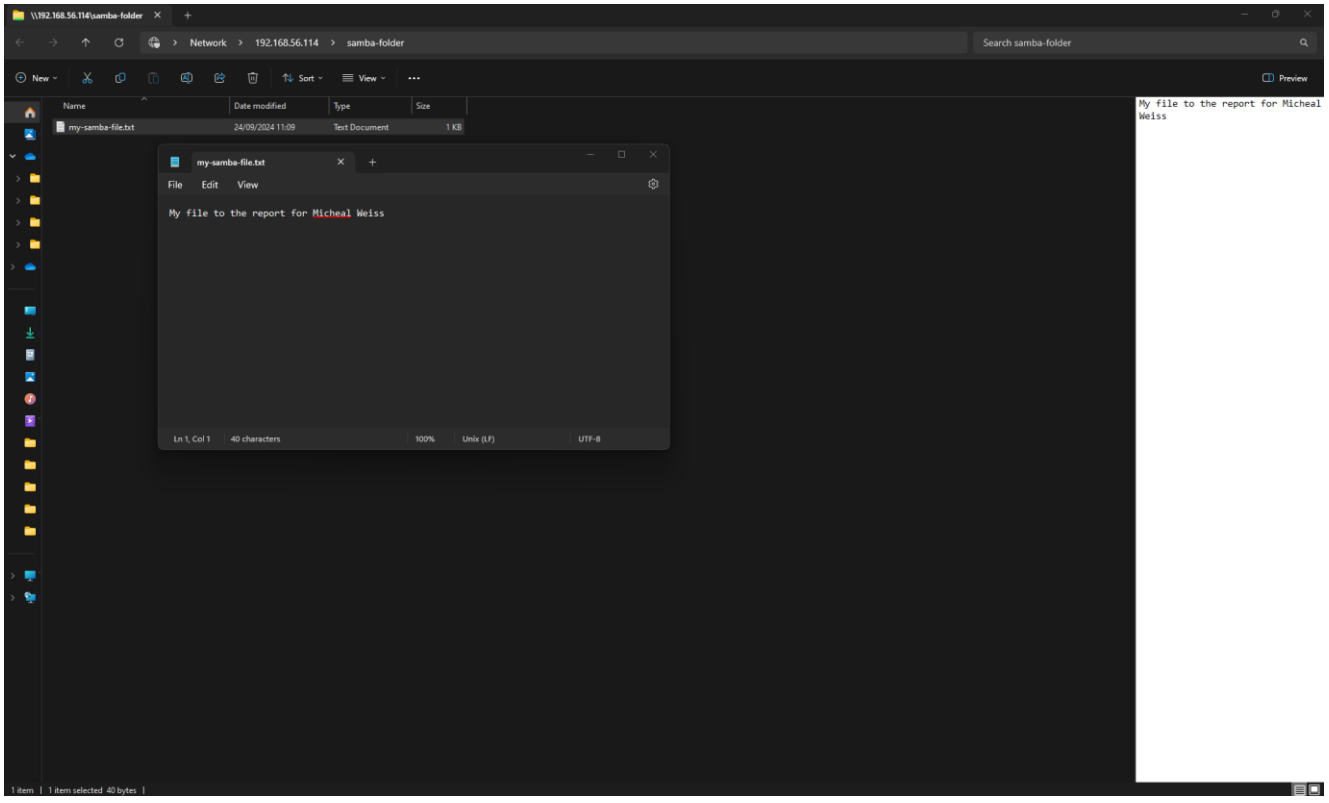

Image 19: Samba file opened in my computer.

Image 20: file edited in my computer showed in my ubuntuserver.

Part 5: Research and challenge activities

Part A: Security configuration #2

In this task, I configured iptables, a traditional firewall utility, to control the HTTP and SSH traffic entering and exiting the server. Before the introduction of Uncomplicated Firewall (UFW), Linux administrators commonly used iptables to manage firewall rules. At the request of my supervisor at 'Consult & Connect Ltd', I demonstrated the use of iptables to block and allow both HTTP and SSH traffic. Below are the steps I followed to configure iptables, along with evidence of their effectiveness.

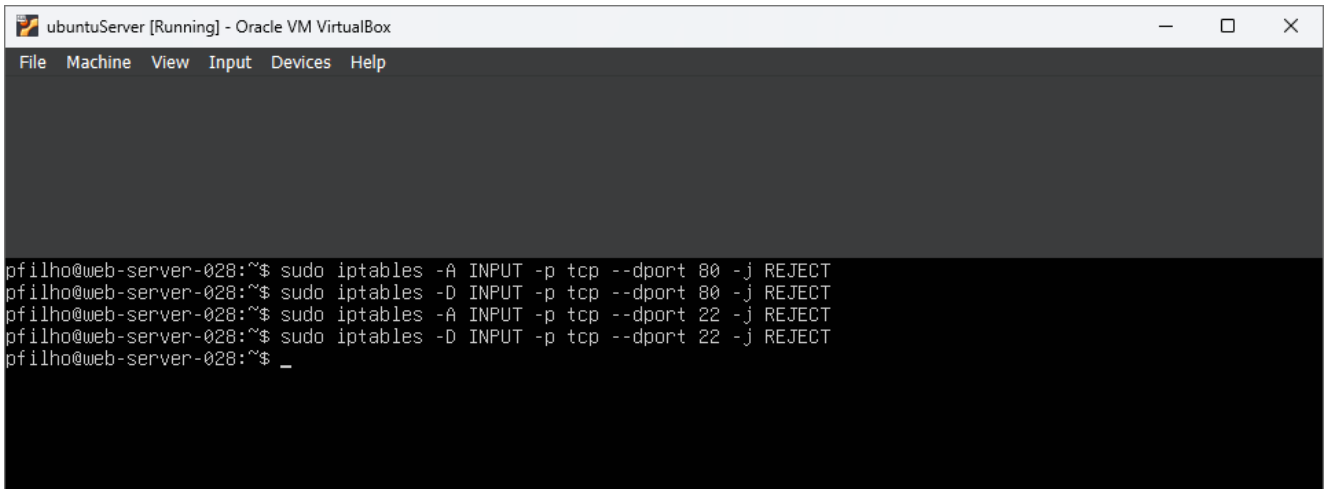

Image 21: codes used to block and allow SSH and HTTP.

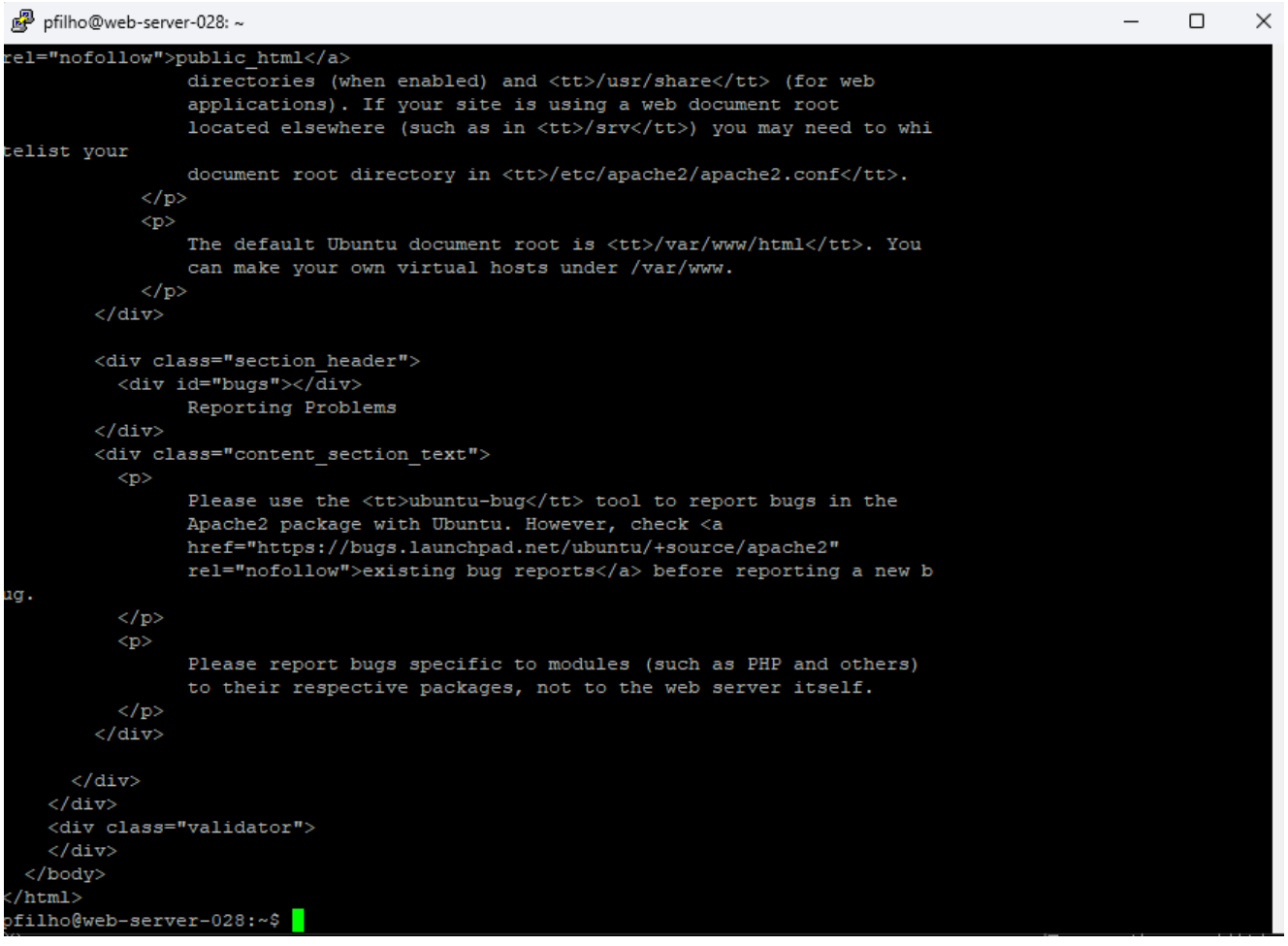

Image 22: Connection HTTP working.

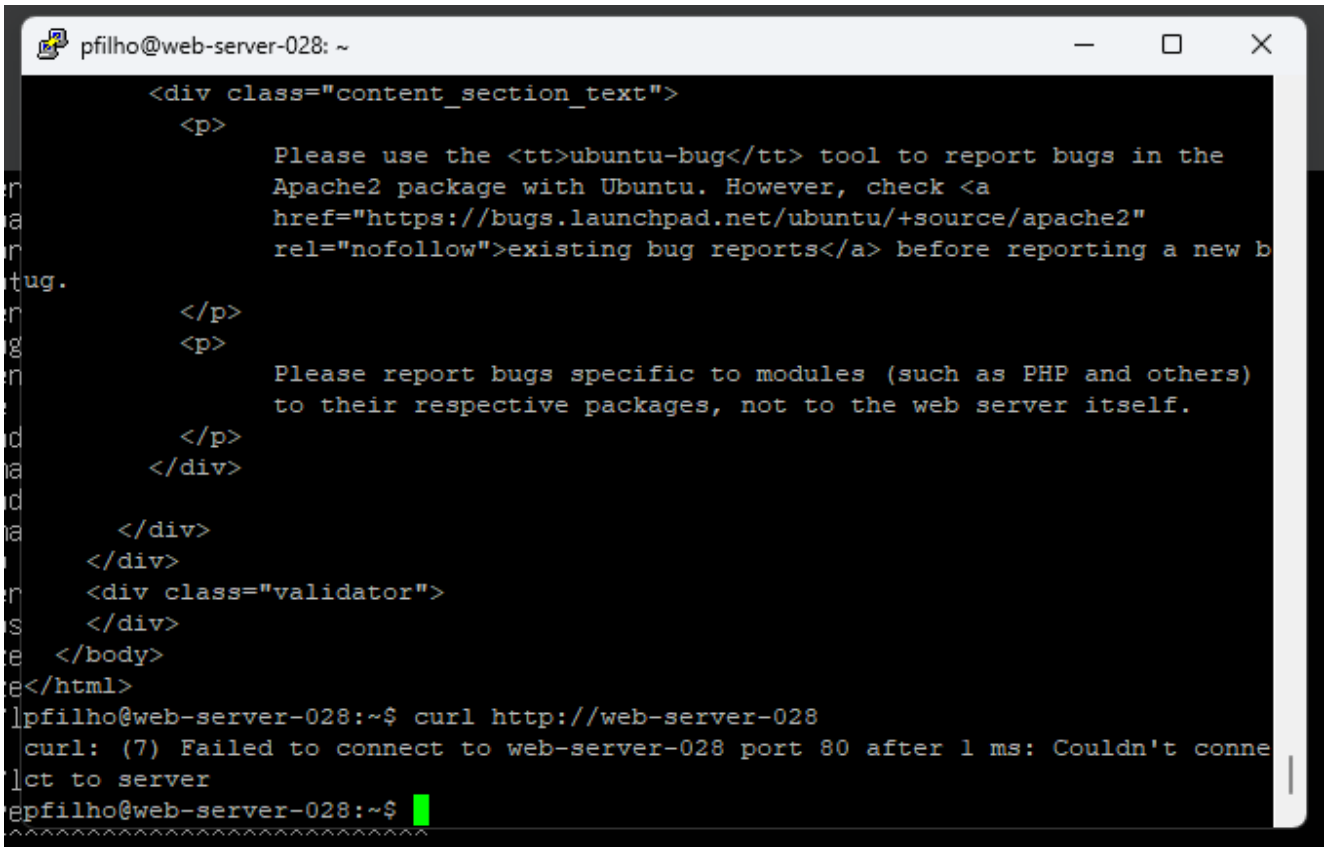

Image 23: Connection HTTP denied.

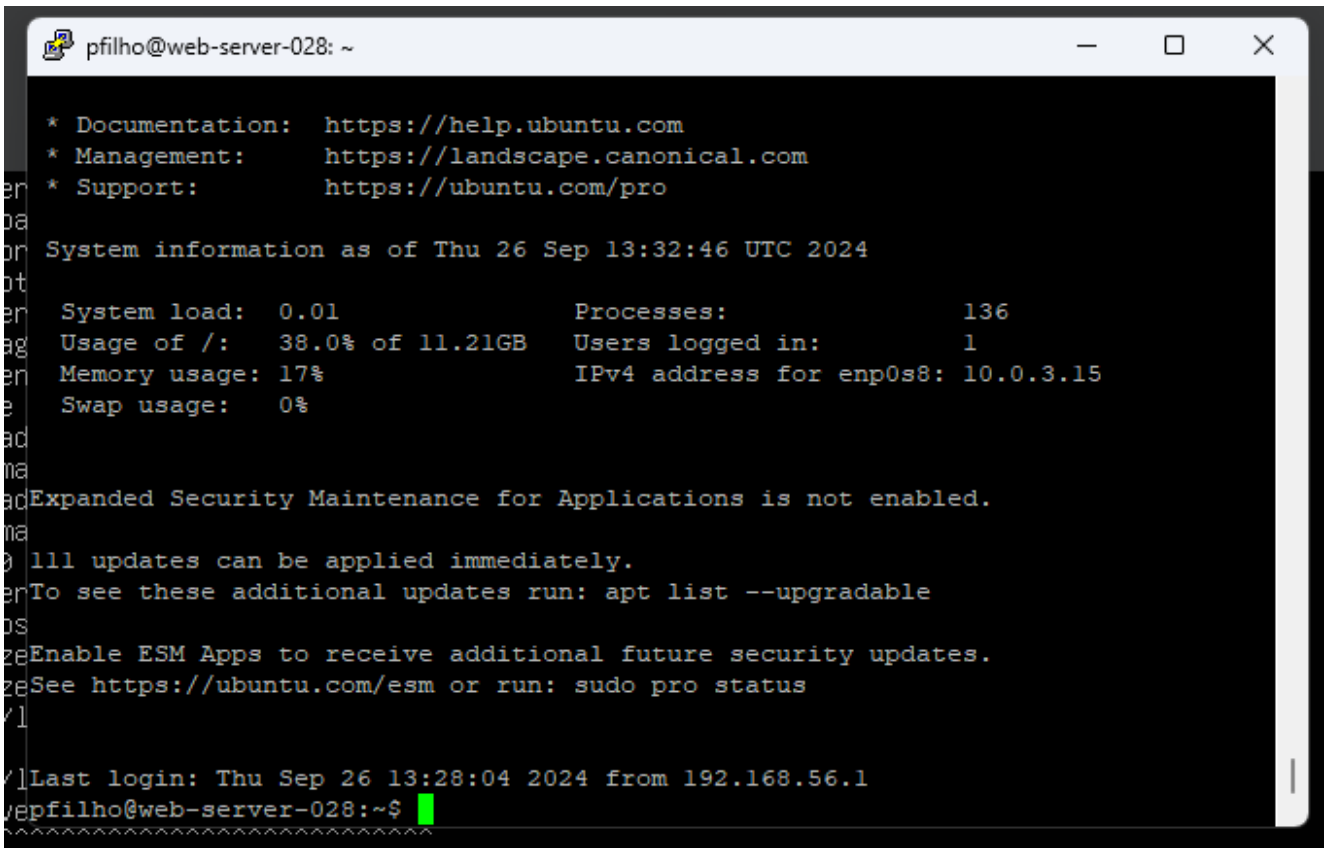

Image 24: Connection SSH working.

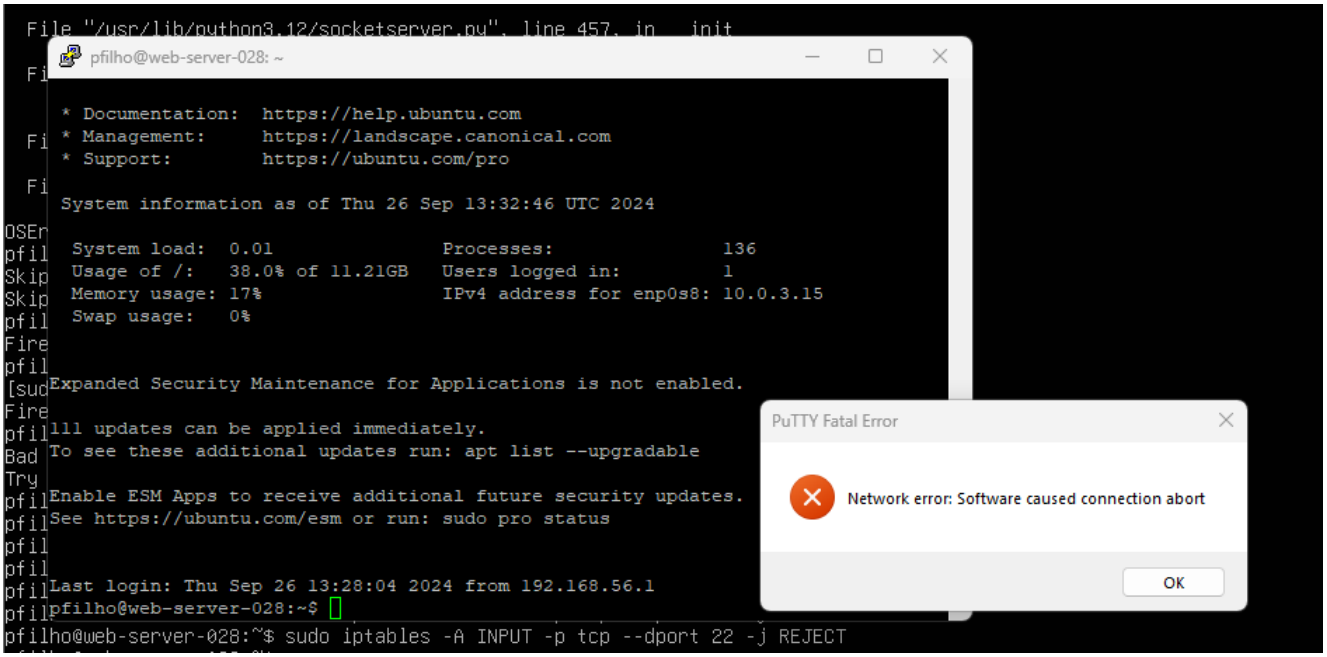

Image 25: Connection SSH denied.

Part D: Forgotten password

In this task, I assisted an associate who had forgotten the password to virtual machines running both Linux and Windows operating systems in Oracle VirtualBox. Rather than creating new virtual machines, I demonstrated how to regain access by resetting the passwords on both systems. For Linux, I used the GRUB boot loader to access recovery mode and reset the password through the root shell. For Windows, I used the installation disk to access the command prompt and reset the password via the Ease of Access tool. Below are the steps I followed for each system, along with evidence of their effectiveness.

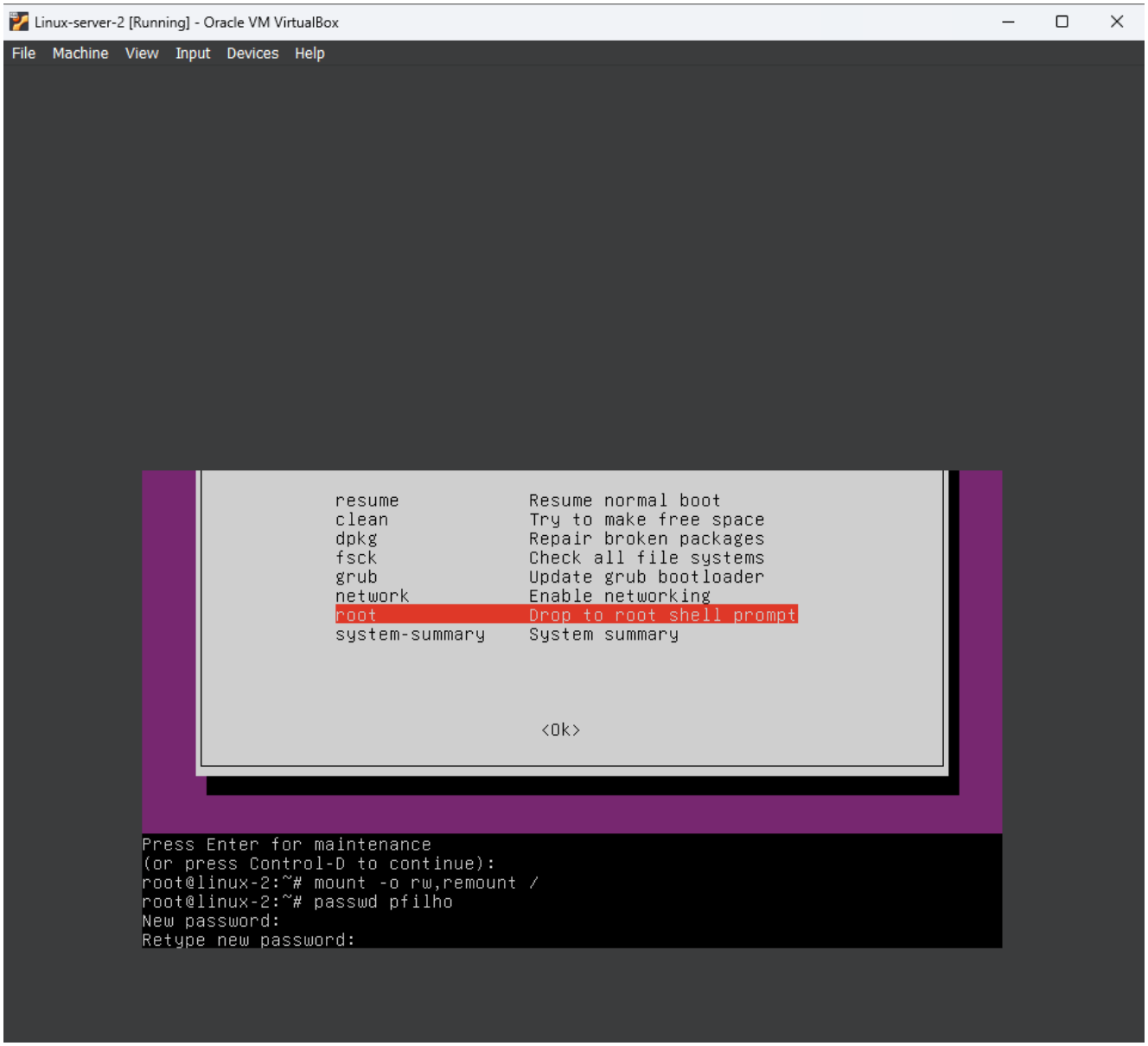

Image 26: Password reset.