Shajith Vignesh Ananthaneni
11691714

Final Exam Practical: Creating VPN/Tunneling Connection for Privacy & Security Protocol

Part 1: Preparing the environment

Understanding the concepts:

SSH (Secure Shell) is a protocol that provides a secure channel over an unsecured network by allowing users to remotely access and manage devices securely with encrypted communication (used for remote login and command execution)

VPN (Virtual Private Network) helps protect your privacy and secures data from hackers (especially on public Wi-Fi) by creating an encrypted tunnel between your device and a VPN server that masks your IP address and encrypts all transmitted data

Port Forwarding maps an external port on a network to an internal IP and port that allows external devices to connect to a specific device or service inside a private network (useful in hosting servers and remote access)

VPN Protocols Difference & Application:

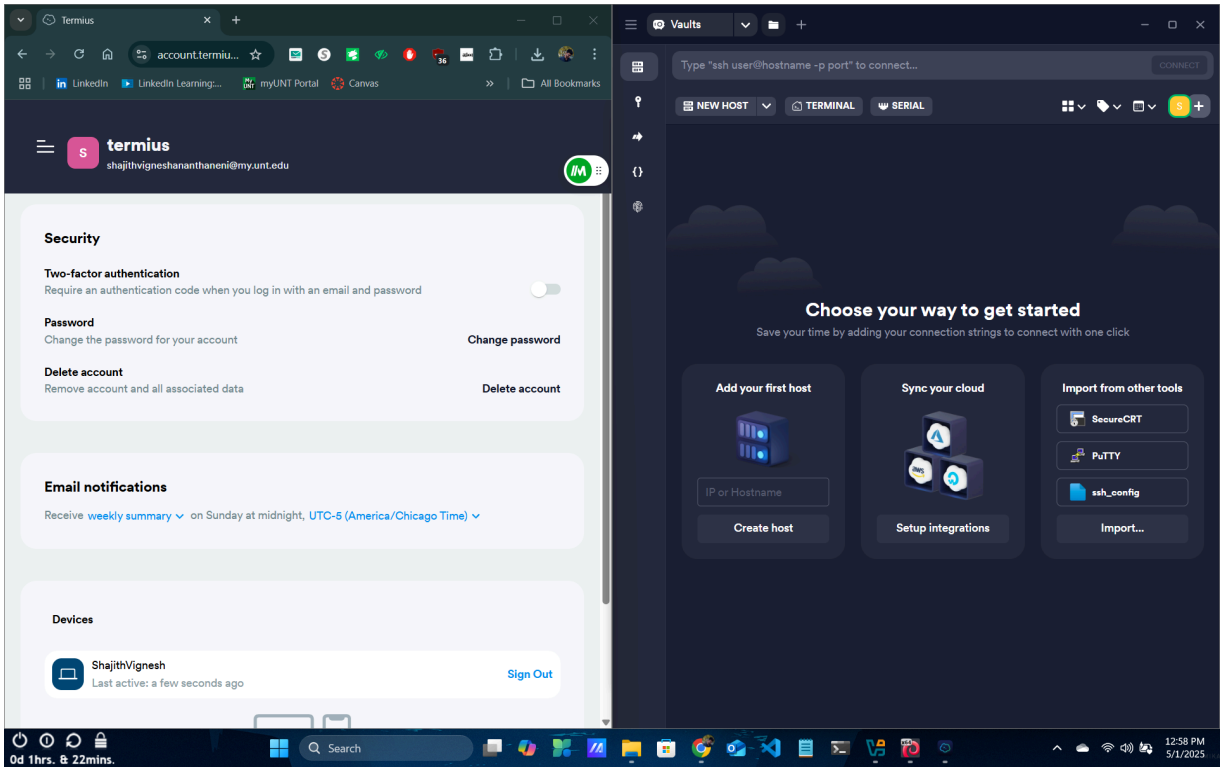
VPN Protocol	Security	Speed	Use Case	Notes
OpenVPN	Strong(SSL/TLS based encryption)	Moderate	General purpose	Open source and highly configurable
IKEv2/IPSec	Strong	Fast	Mobile devices (auto-reconnect)	Good for mobile due to stability
WireGuard	Very strong	Very Fast	Modern VPN solutions	Lightweight, newer protocol
L2TP/IPSec	Moderate	Slower than OpenVPN	Legacy systems	Easy to block with firewalls
PPTP	Weak (outdated)	Fast	Legacy use only	Insecure and deprecated

Applications of VPN and Port Forwarding:

VPN: Ensures secure browsing, bypasses geo-restrictions, and secures data on public networks

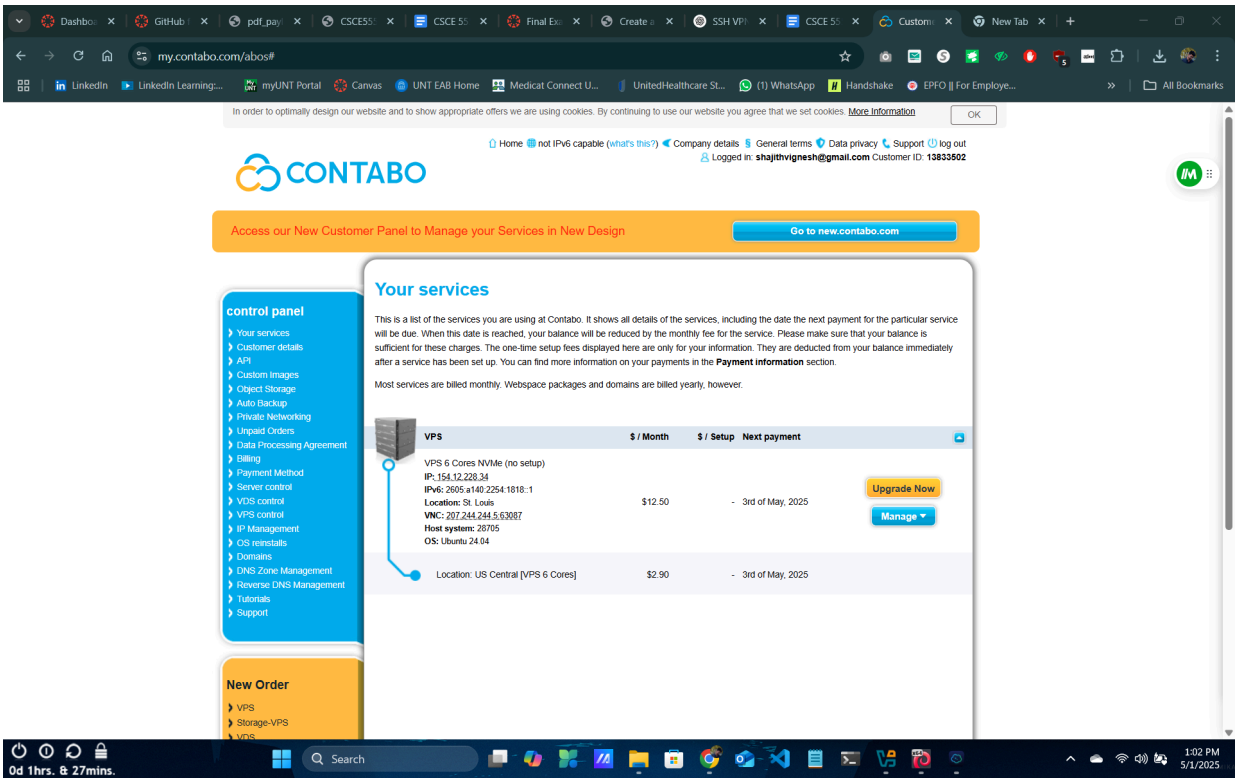
Port Forwarding: Allows access to internal network services (e.g., SSH, RDP, gaming servers) from outside a NAT/firewall

Installing Terminus

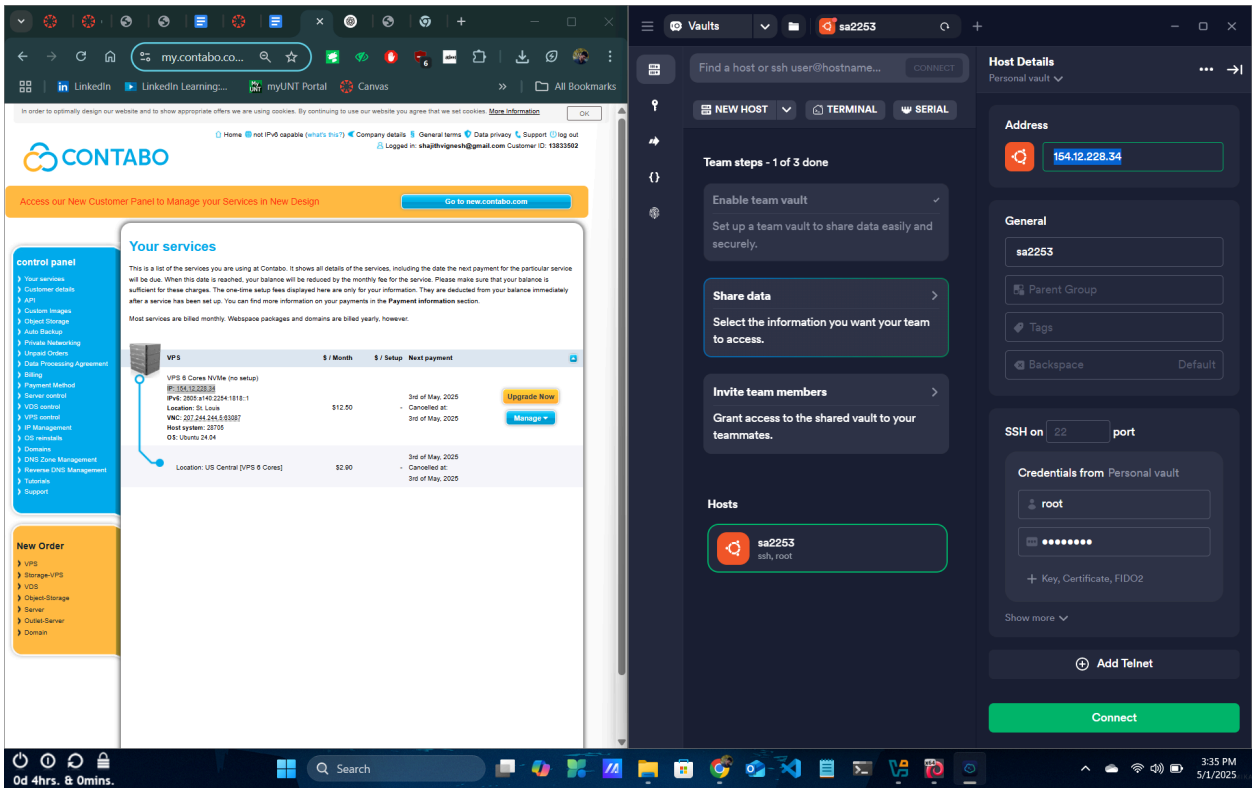


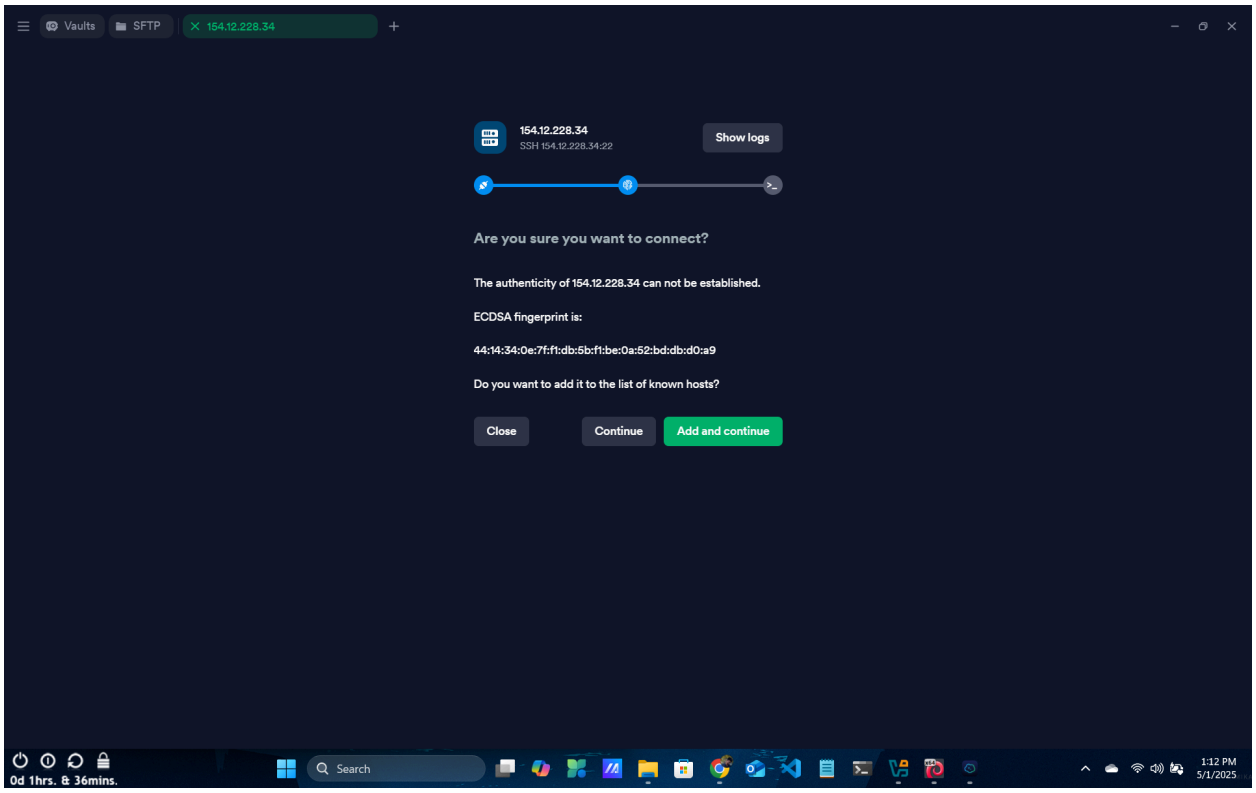
Part 2: Setting up the VPN/Tunnel

I'm using the same VPS server (IP: 154.12.228.34) used for Lab 2 (Mail Server) from Contabo



Added the server to the Termius application under the "Hosts" section and configured the necessary authentication details, including user ID and password for SSH access





Vaults SFTP 154.12.228.34

154.12.228.34
SSH 154.12.228.34:22

Hide logs

Starting a new connection to: "154.12.228.34" port "22"

Starting address resolution of "154.12.228.34"

Address resolution finished

Connecting to "154.12.228.34" port "22"

Connection to "154.12.228.34" established

Starting SSH session

Remote server: SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.9

Agreed KEX algorithm: curve25519-sha256

Agreed Host Key algorithm: ecdsa-sha2-nistp256

Agreed server-to-client cipher: aes256-gcm@openssh.com MAC: INTEGRATED-AES-GCM

Close Start over

0d 1hrs. & 37mins. Search 1:12 PM 5/1/2025

Vaults SFTP 154.12.228.34

154.12.228.34
SSH 154.12.228.34:22

Hide logs

sha256

Agreed Host Key algorithm: ecdsa-sha2-nistp256

Agreed server-to-client cipher: aes256-gcm@openssh.com MAC: INTEGRATED-AES-GCM

Agreed client-to-server cipher: aes256-gcm@openssh.com MAC: INTEGRATED-AES-GCM

Agreed client-to-server compression: none

Agreed server-to-client compression: none

Handshake finished

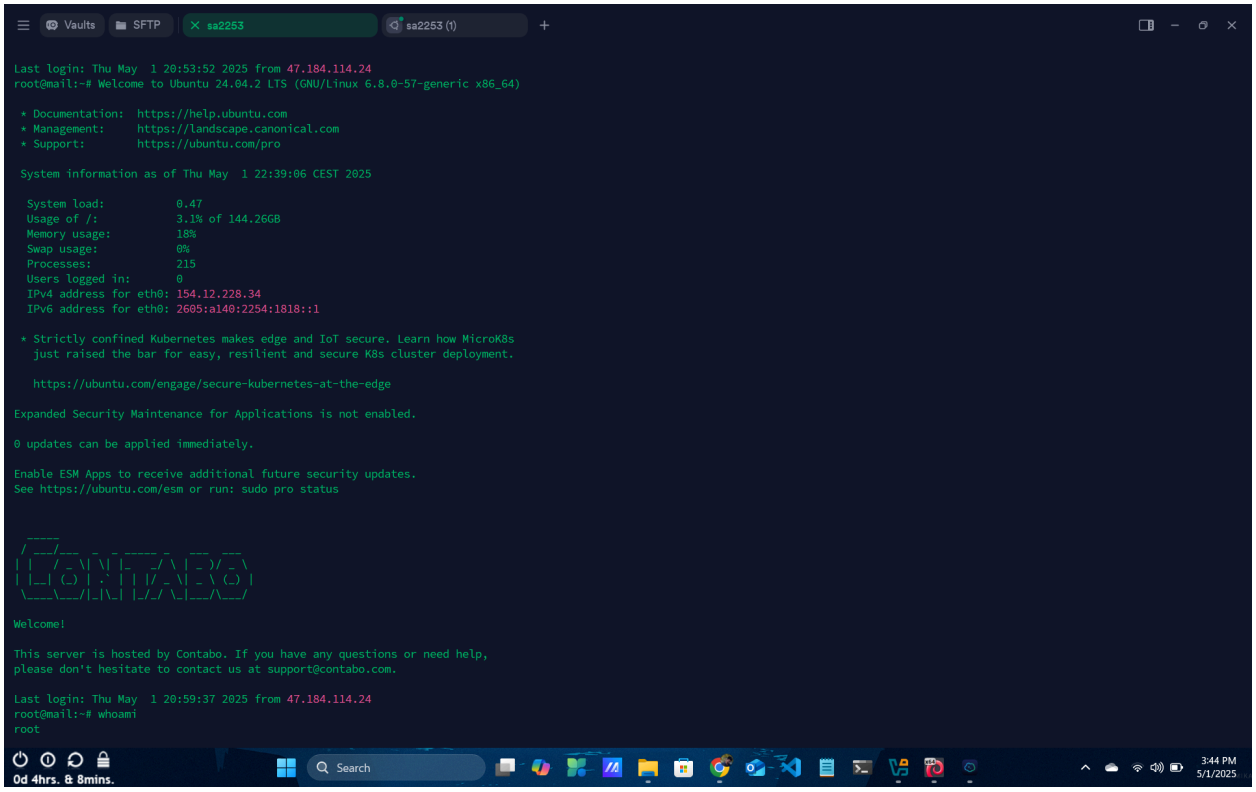
Checking host key: 44:14:34:0e:7f:f1:db:5b:f1:be:0a:52:bd:db:d0:a9

Host "154.12.228.34":"22" is unknown

Close Start over

0d 1hrs. & 37mins. Search 1:12 PM 5/1/2025

I was able to connect to my server through Termius successfully

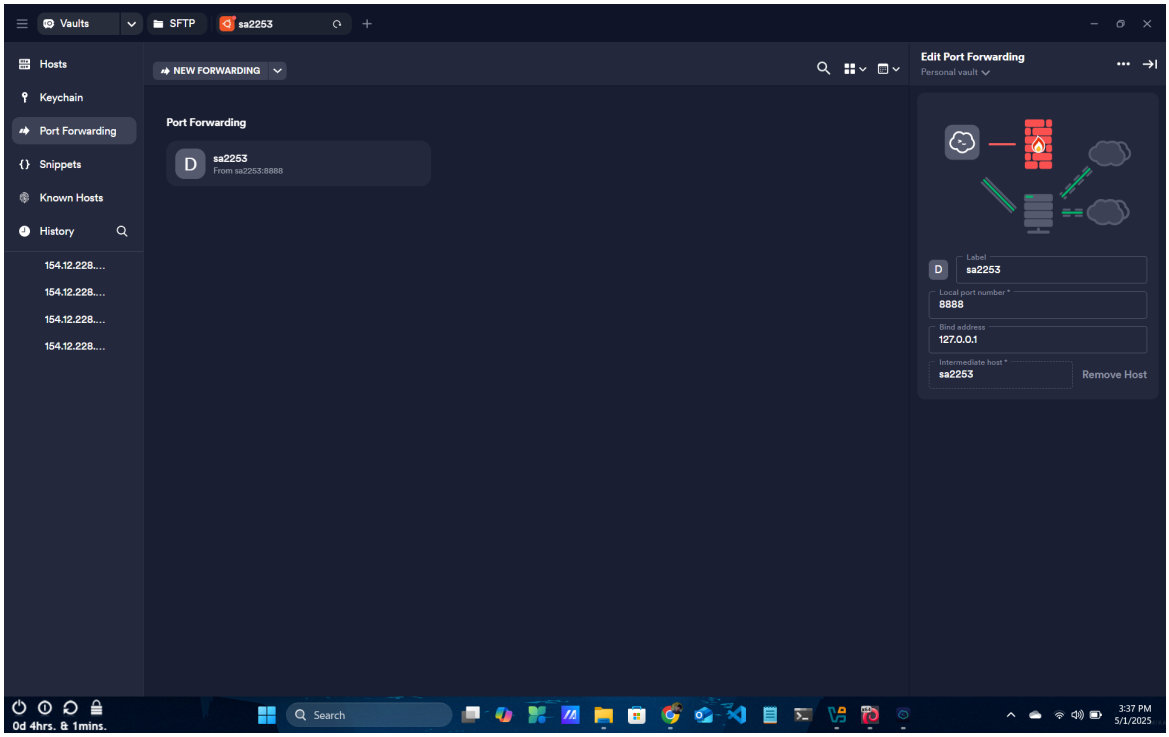


Understanding Port Forwarding Techniques in SSH:

Type	Description	Use Case
Local Forwarding	Forward a port from your **local machine** to a destination via the remote SSH server	Access an internal webpage or database behind a firewall
Remote Forwarding	Forwards a port from the **remote SSH server** back to your **local machine**	Share a local development site with someone remotely
Dynamic Forwarding	Creates a **SOCKS proxy** through the SSH tunnel, letting you route traffic from your browser/apps	Acts like a lightweight VPN for securing browsing, bypassing firewalls/geo-blocks

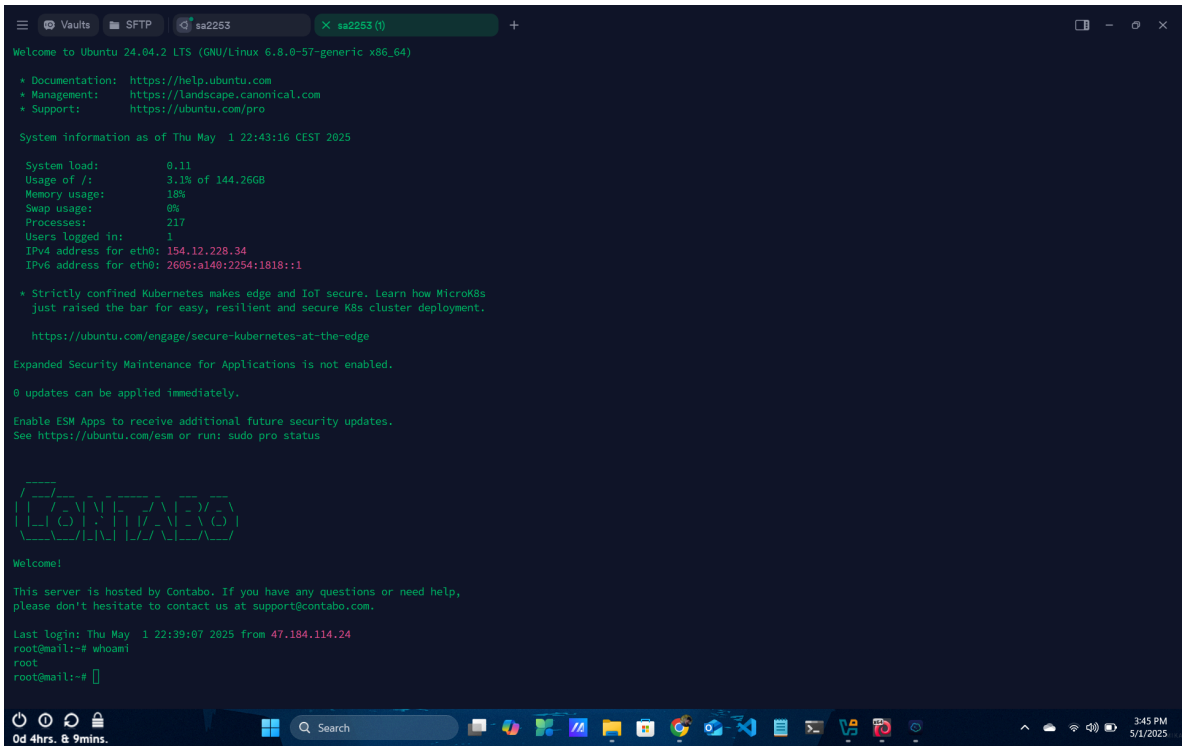
Setting up port forwarding in Termius

I'm using Dynamic Port Forwarding to set up my browser's proxy settings to use SOCKS5 on Host: 127.0.0.1 & Port: 8888 with my server as the Intermediate Host to simulate VPN



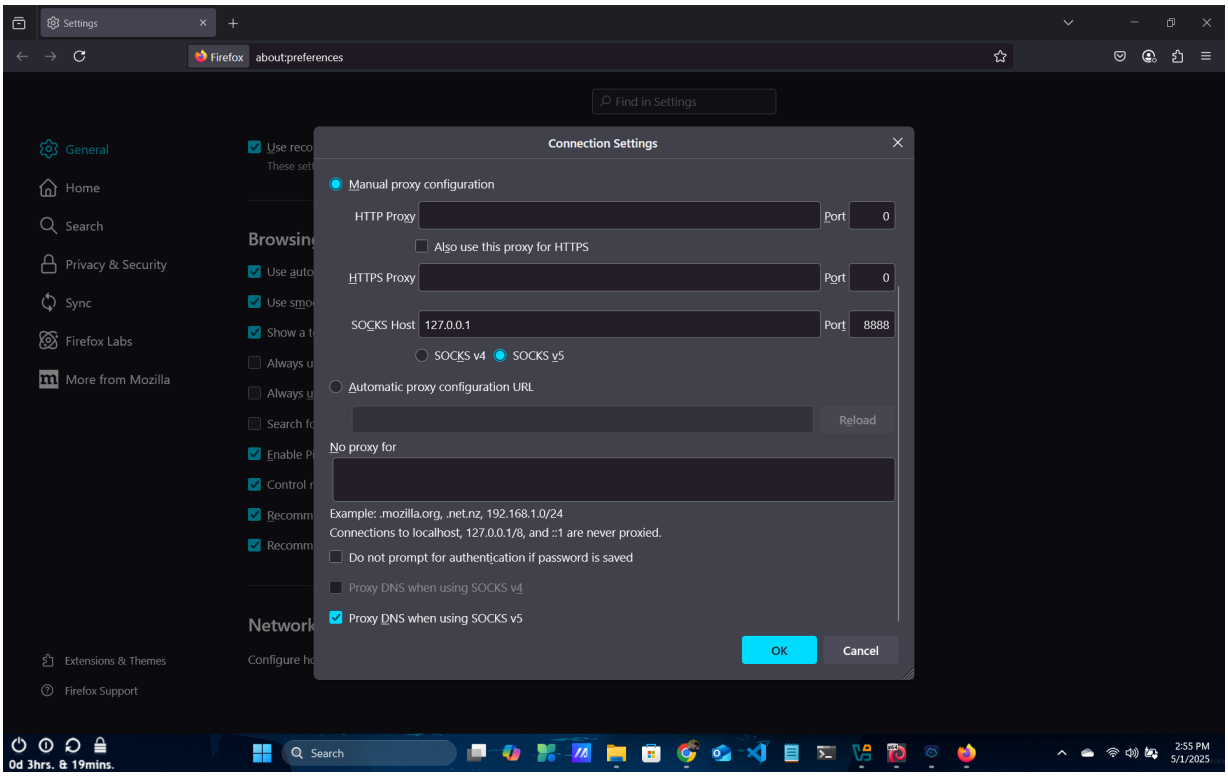
Part 3: Establishing the VPN/Tunnel Connection

Successfully connected to the VPN server by using Termius

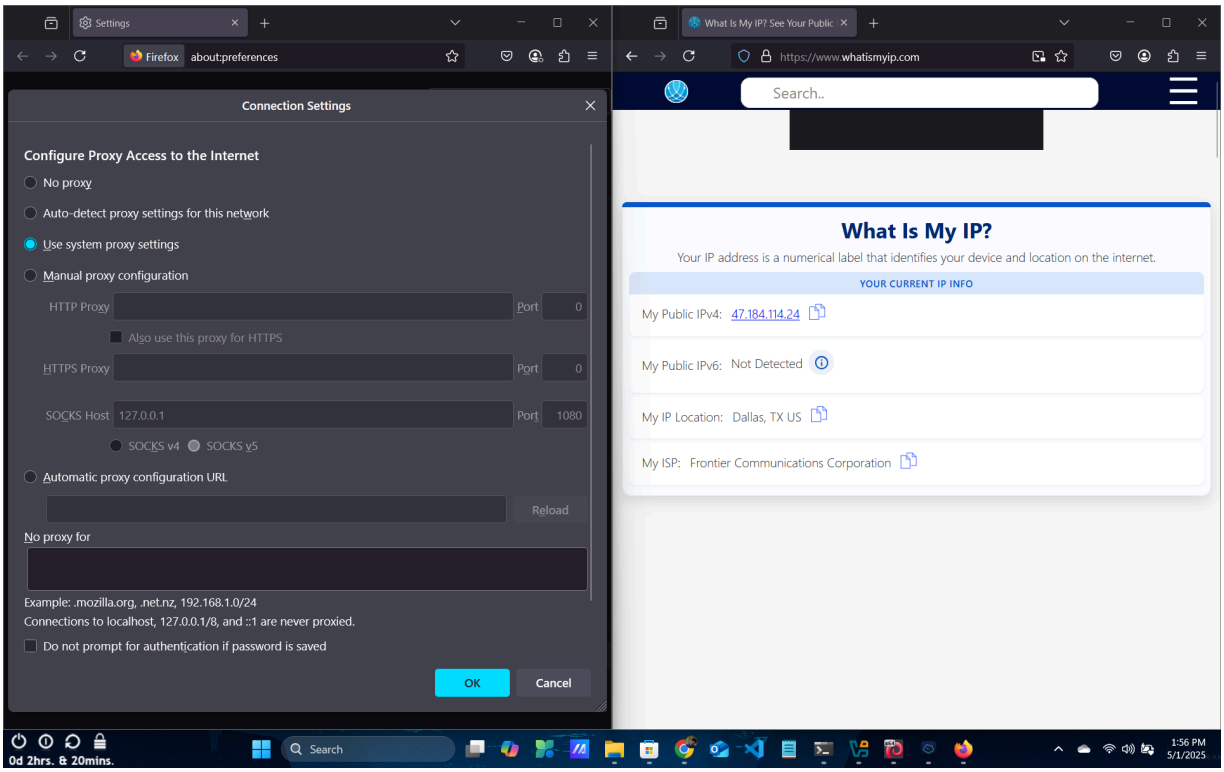


Part 4: Application Integration

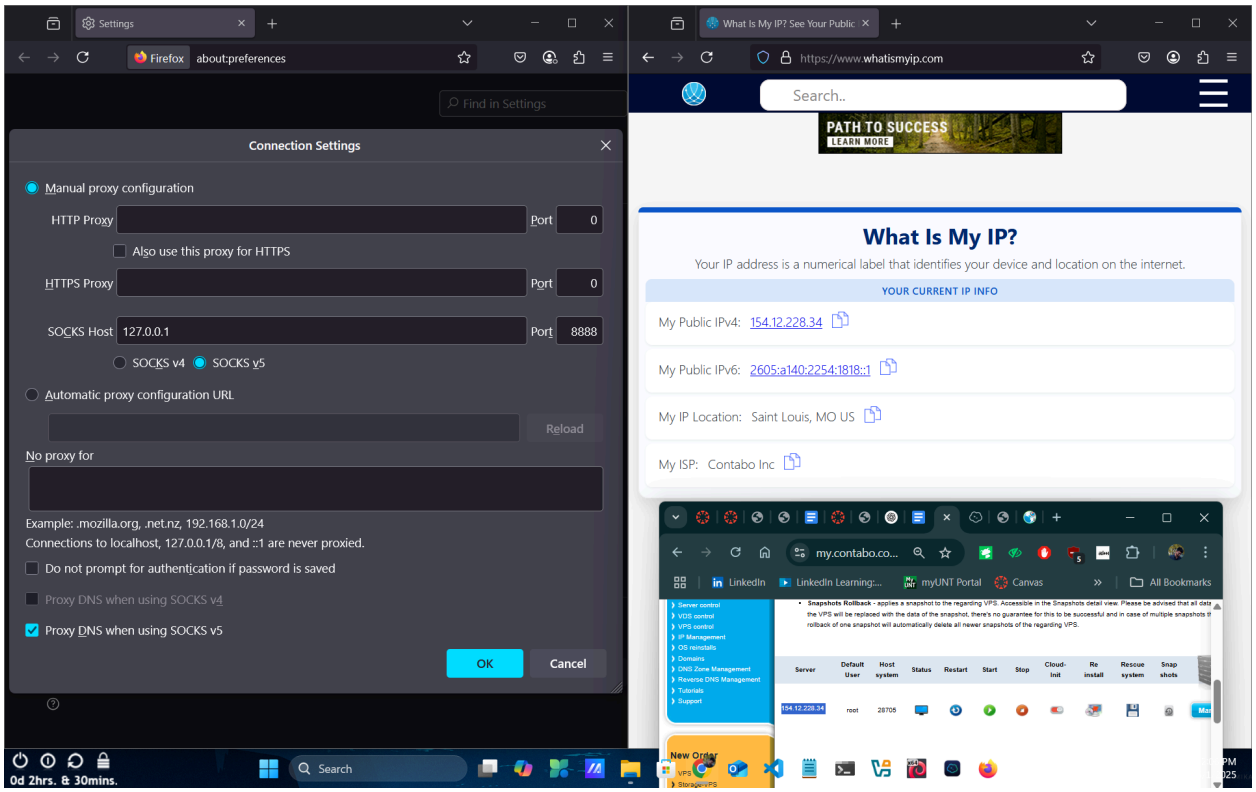
I configured my web browser (Mozilla Firefox) to use the VPN tunnel as a SOCKS proxy



Before the tunnel setup, my local ISP IP is shown



After the tunnel setup, the remote server's IP is shown & confirms all my browser traffic is being routed securely via SSH tunnel



Part 5: Reporting

Using Termius tools for VPN/Tunnel as a privacy and security protocol setup

Advantages:

- Easy GUI for managing SSH connections and tunnels
- Cross-platform availability (Windows, macOS, mobile)
- Student-friendly (free plan)
- Simplifies dynamic port forwarding

Limitations

- Not a true VPN (more like a proxy)
- No traffic encryption outside SOCKS-supported apps
- Manual setup for browser only, not system-wide
- Requires an external SSH server

Comparison with Alternatives

OpenVPN or **WireGuard**: True VPNs with system-wide traffic tunneling
PuTTY: Can also forward ports, but no modern UI
Tailscale/**ZeroTier**: Easy mesh VPNs with auto-configuration