CASE STUDY

Implementing a Robust VPN Solution for Secure Remote Access to Cloud Resources

Presented By.

Name	Registration No.
1. Ankit Kumar Panda	2241016052
2. Piyush Kumar Sahoo	2241019096
3. Barun Kumar Rout	2241018145

TABLE OF CONTENTS

INTRODUCTION

- Organizations increasingly depend on cloud resources (AWS EC2, RDS, etc.)
- Remote work culture requires employees to access cloud services from anywhere.
- Directly exposing resources to the internet leads to security threats.
- A VPN solution allows secure and private connectivity to AWS VPC resources.
- This project focuses on AWS Client VPN with IAM authentication to provide safe and scalable access.

MOTIVATION

- Protect sensitive business data from cyberattacks and unauthorized access.
- Provide remote teams with seamless and secure connectivity.
- Eliminate risks of public-facing SSH/RDP connections.
- Reduce operational complexity with IAM-based authentication.
- Ensure the solution is robust, scalable, and easy to manage for future needs.

WITHOUT VPN VS WITH VPN

Without VPN:

- Data travels in plaintext over the internet.
- Easy for attackers to sniff/modify traffic.
- No identity verification of users.

With VPN:

- Data is encrypted using protocols (AES, IPsec).
- Private tunnel → only authenticated users connect.
- Cloud resources hidden from direct internet exposure.
- Strong authentication: Certificates, IAM, MFA.

SOLUTION DEVELOPMENT

Our proposed VPN solution integrates modern security protocols and architectures:

- Site-to-Site VPN for organization-to-cloud secure link. (IKEv2 + IPsec)
- Point-to-Site VPN for individual remote user access.
- Robust Authentication & Key Exchange (OpenVPN, Certificates, MFA).

SOLUTION DEVELOPMENT

Site-to-Site VPN Architecture:

Use Case:

Connects an organization's on-premises data center securely with cloud VPC.

How it works:

- A VPN gateway (router/firewall) at each site.
- Tunnels established via Ipsec+IKEv2.
- Encrypted communication between data center and cloud resources.

Advantages:

- Permanent connection.
- Best for multiple users in branch offices.

SOLUTION DEVELOPMENT

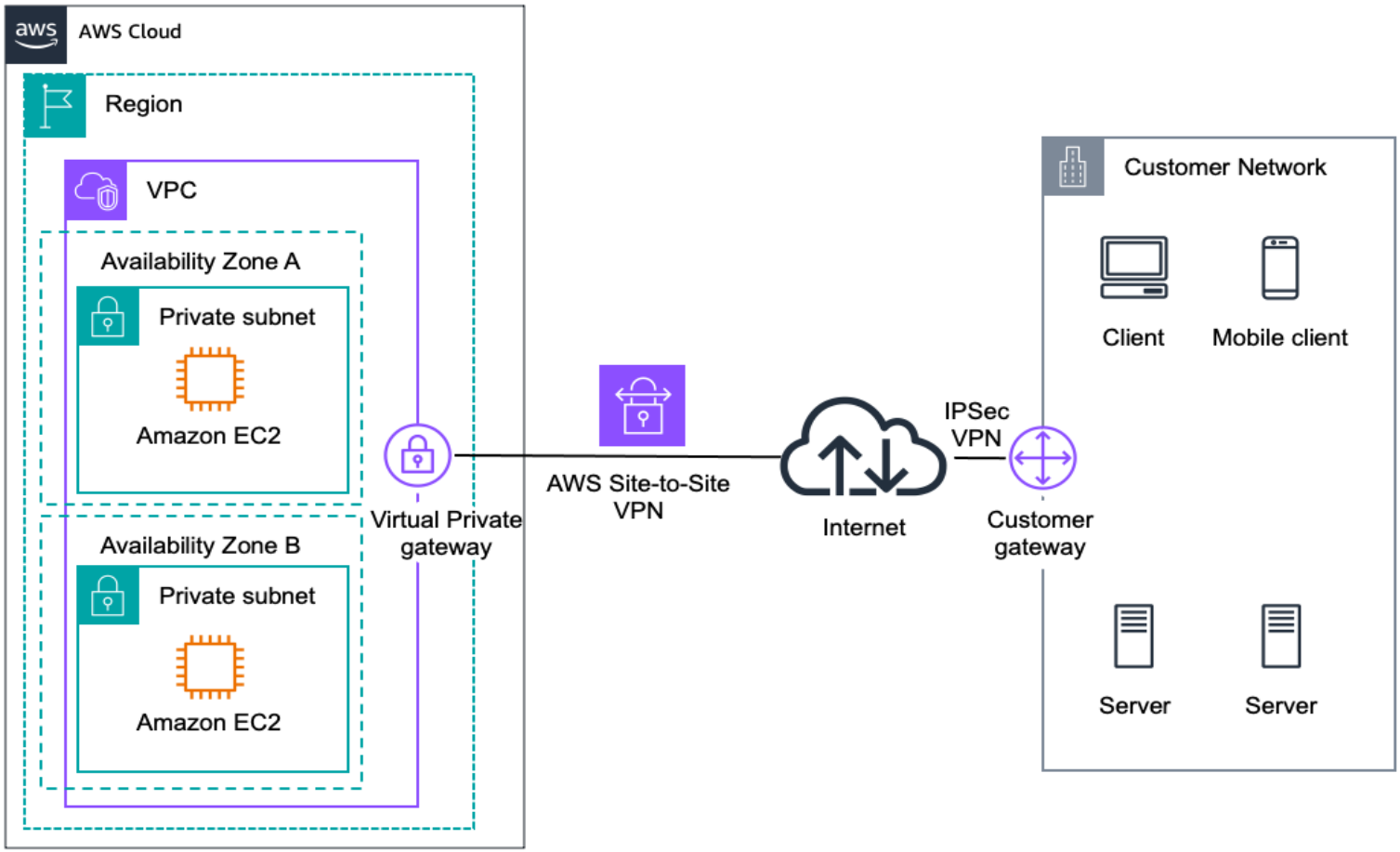


Fig : Site-To-Site VPN Architecture

SOLUTION DEVELOPMENT

Point-to-Site VPN Architecture:

Use Case:

Individual employees working remotely.

How it works:

- User installs VPN client on laptop/phone.
- Client authenticates via certificate / username-password + MFA.
- Secure tunnel established directly to Cloud VPN gateway.

Advantages:

- Flexible – works from anywhere.
- Best for small teams or remote users.

SOLUTION DEVELOPMENT

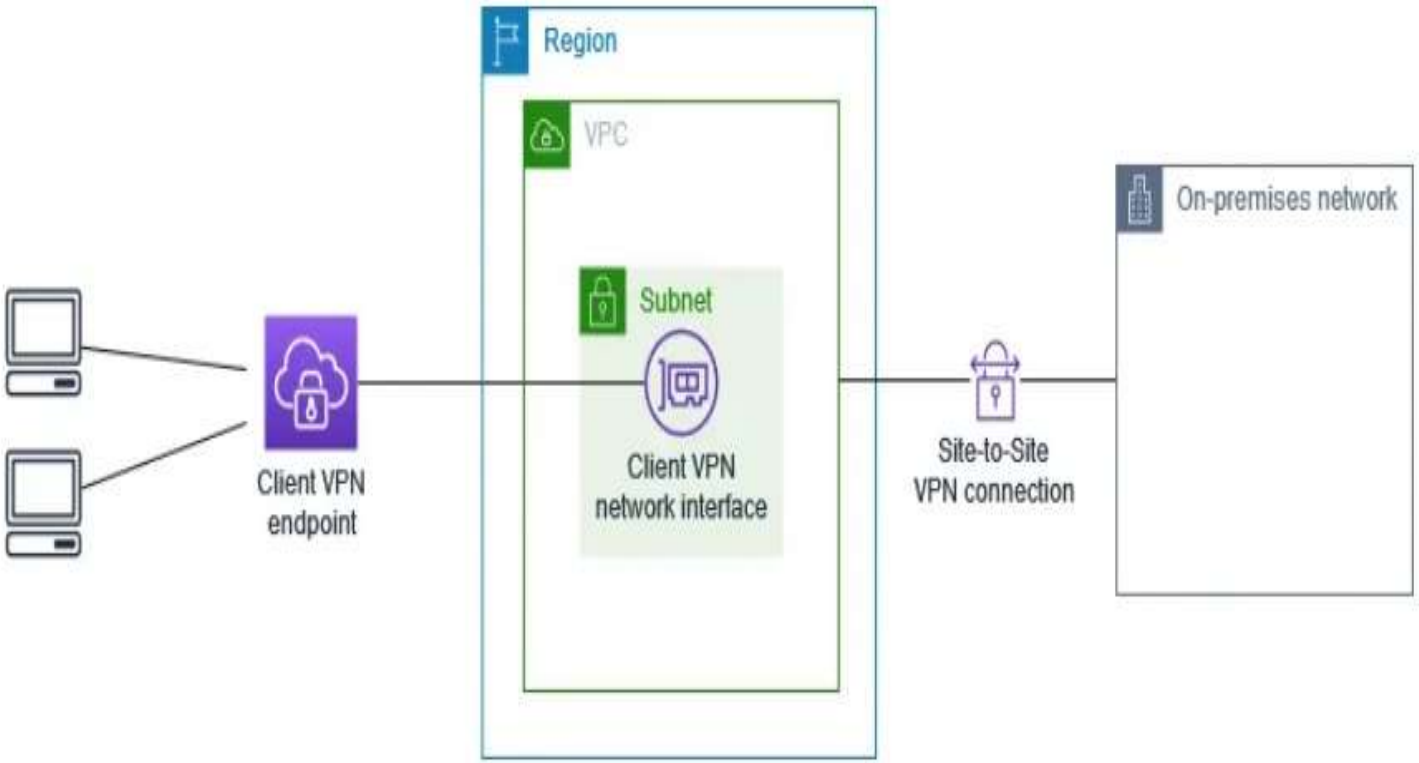


Fig : Point-To-Site VPN Architecture

SOLUTION DEVELOPMENT

Authentication & Key Management :

Key Exchange (TLS Handshake):

- Uses OpenVPN (TLS/SSL) for secure tunneling.
- Diffie–Hellman ensures unique keys per session.

Authentication Mechanisms:

- Using Digital Certificates: Each client and server has its own certificate issued by a trusted CA.
- Certificates validate identities.

Multi-Factor Authentication Integration Session Security:

- Perfect Forward Secrecy (PFS) ensures new keys per session(Uses Diffie–Hellman).
- Even if one session key is compromised, past/future sessions remain safe.

IMPLEMENTATION

Step 1: Set Up VPC, Subnets, and Route Tables.

VPC (VPN): (10.0.0.0/16)

Creating Subnets:-

Public Subnet (PUB): 10.0.1.0/24

Private Subnet (PVT): 10.0.2.0/24

Creating Internet Gateway:-

Internet Gateway (IGW): VPN-IGW

Creating Route Table:-

Public Route Table (PUB-RT)

Private Route Table (PVT-RT)

Creating Associations:-

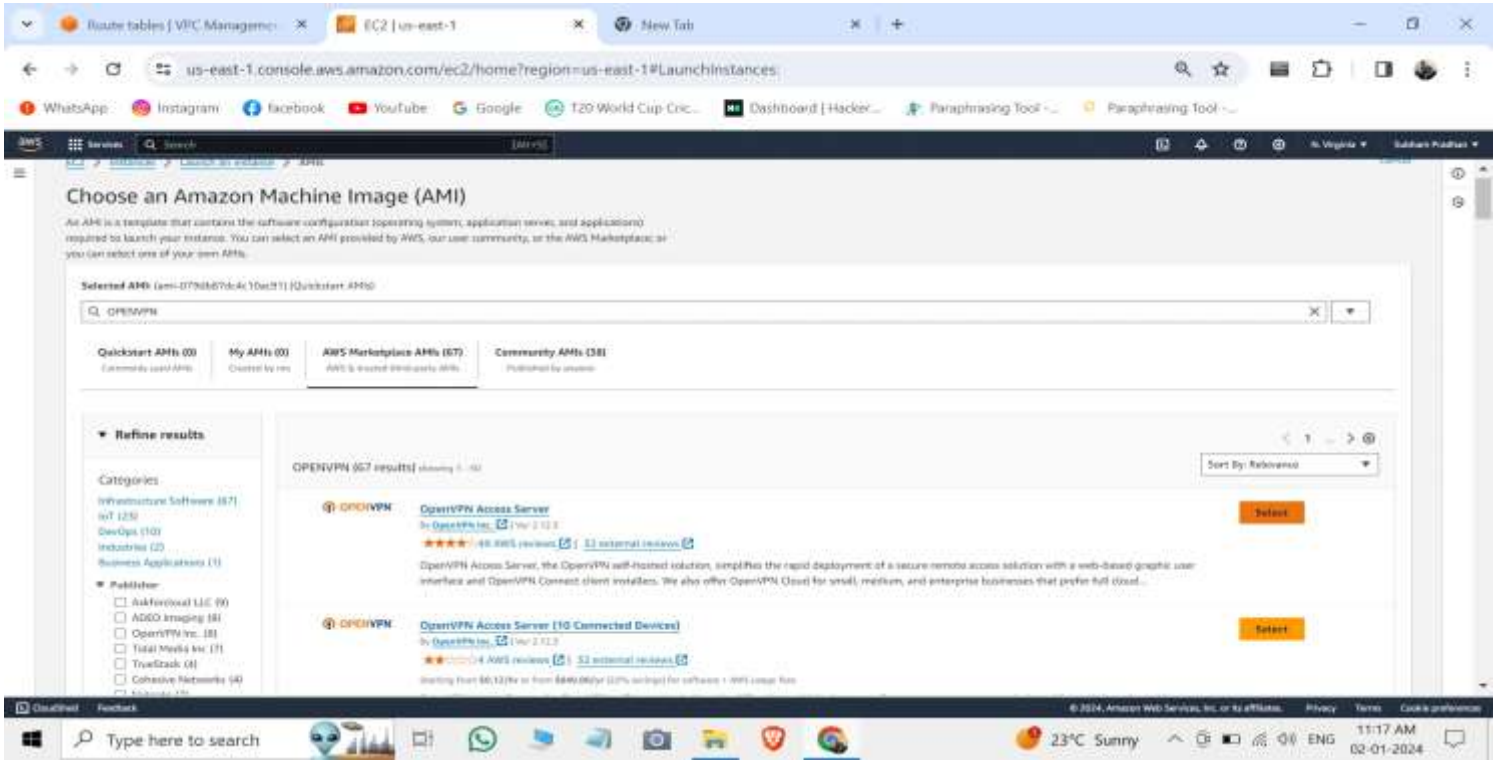
Associate IGW and PUB-RT with the PUB subnet

Associate PVT-RT with the PVT subnet

IMPLEMENTATION

Step 2: Launch OpenVPN Server

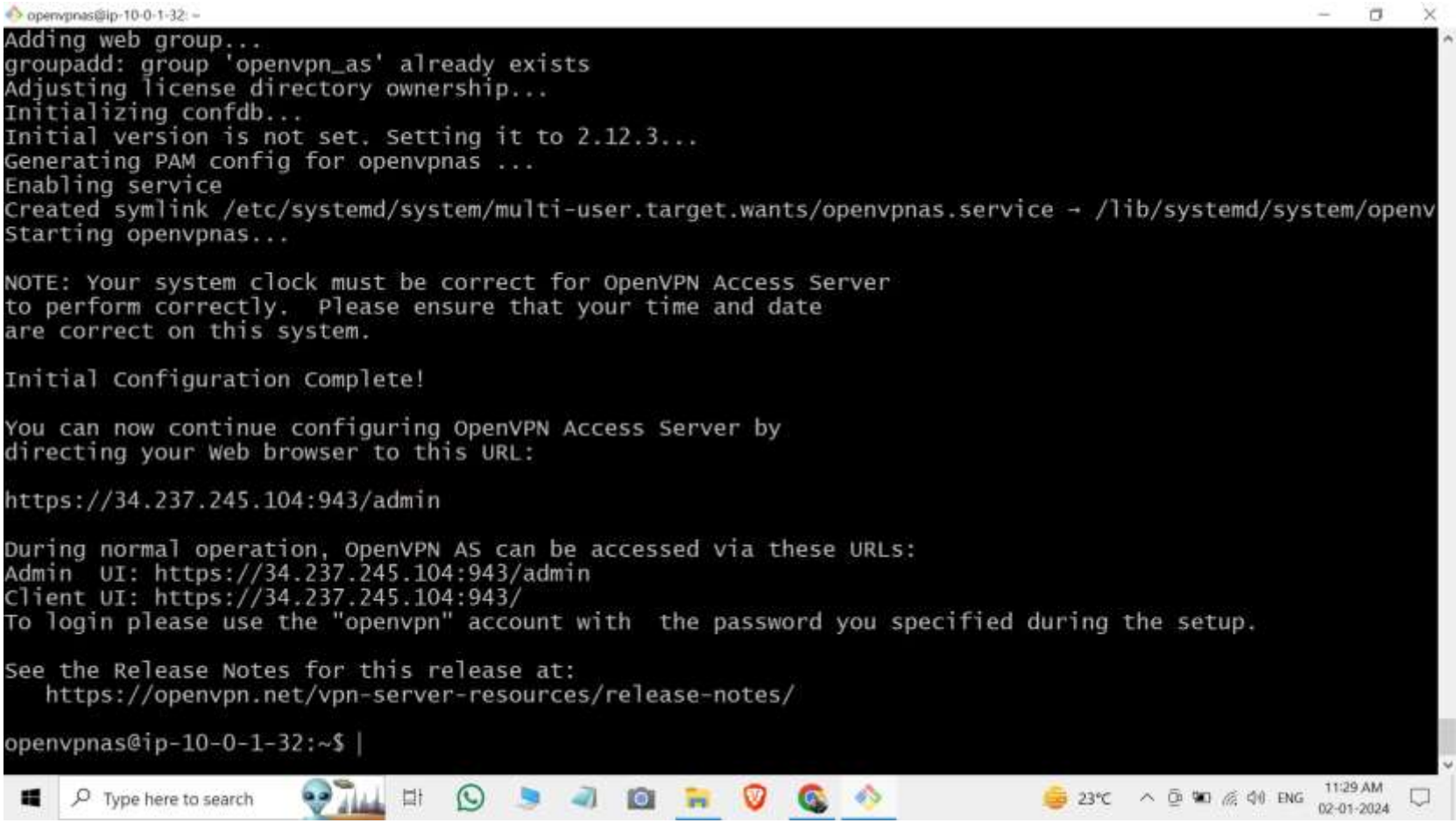
Launch an EC2 instance named “OPENVPN” using the OpenVPN AMI.



- Set a password for the OpenVPN Admin UI and Client UI.
- Copy the Admin UI and Client UI.

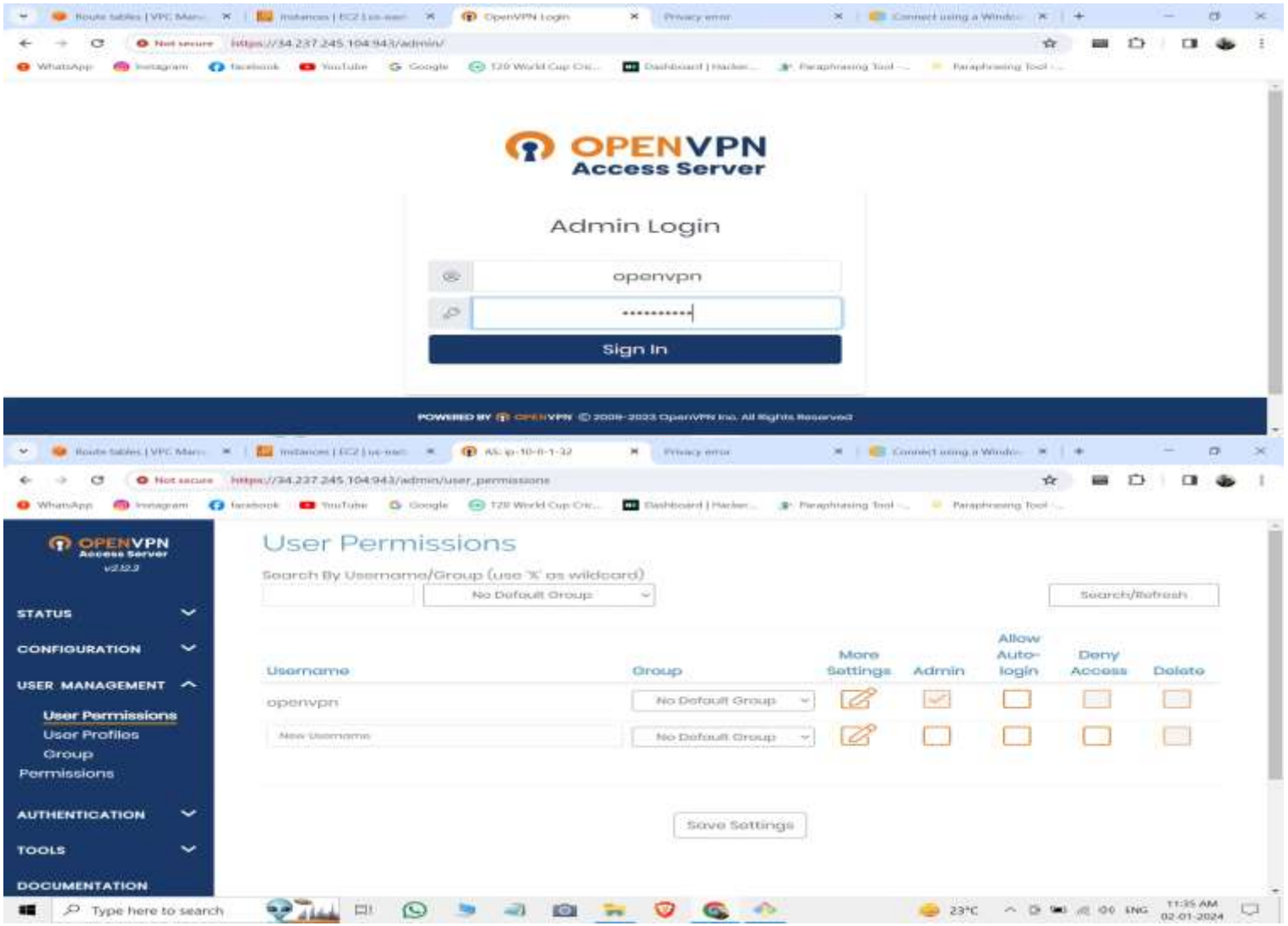
IMPLEMENTATION

Step 2: Launch OpenVPN Server



IMPLEMENTATION

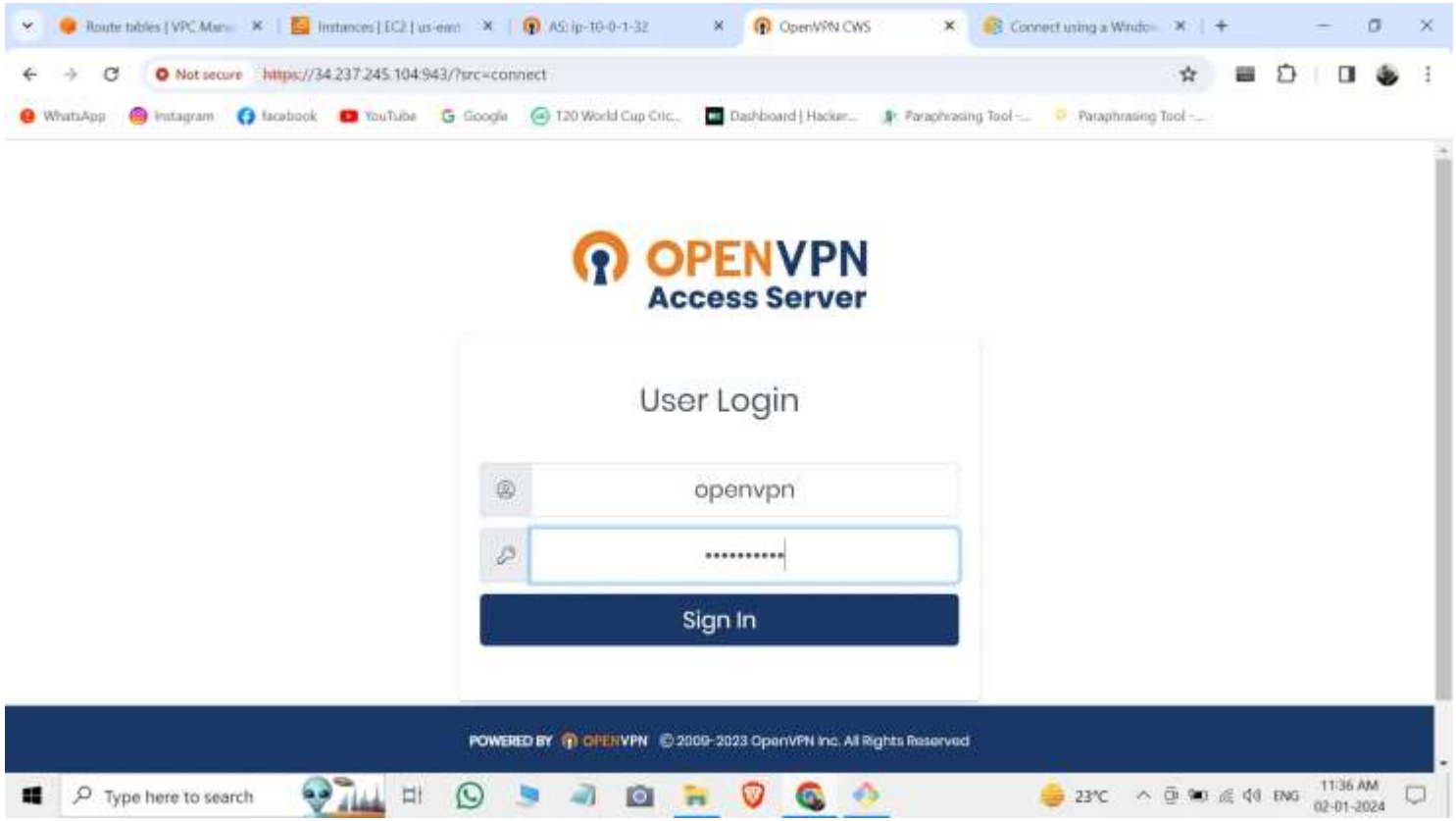
Step 3: Connect to Admin UI



IMPLEMENTATION

Step 4: Connect to Client UI & Download Connection Profile

- Connect to the Client UI.
- Install the OpenVPN client app on your device.
- Download the connection profile from the Client UI.



IMPLEMENTATION

Step 4: Connect to Client UI & Download Connection Profile

IMPLEMENTATION

Step 5: Test VPN Connection

- Ping the Private IP of the OpenVPN server; it will not work.

- Log in using the user ID and password in the OpenVPN client.
- Upload the downloaded profile file then connect.

IMPLEMENTATION

Step 5: Test VPN Connection

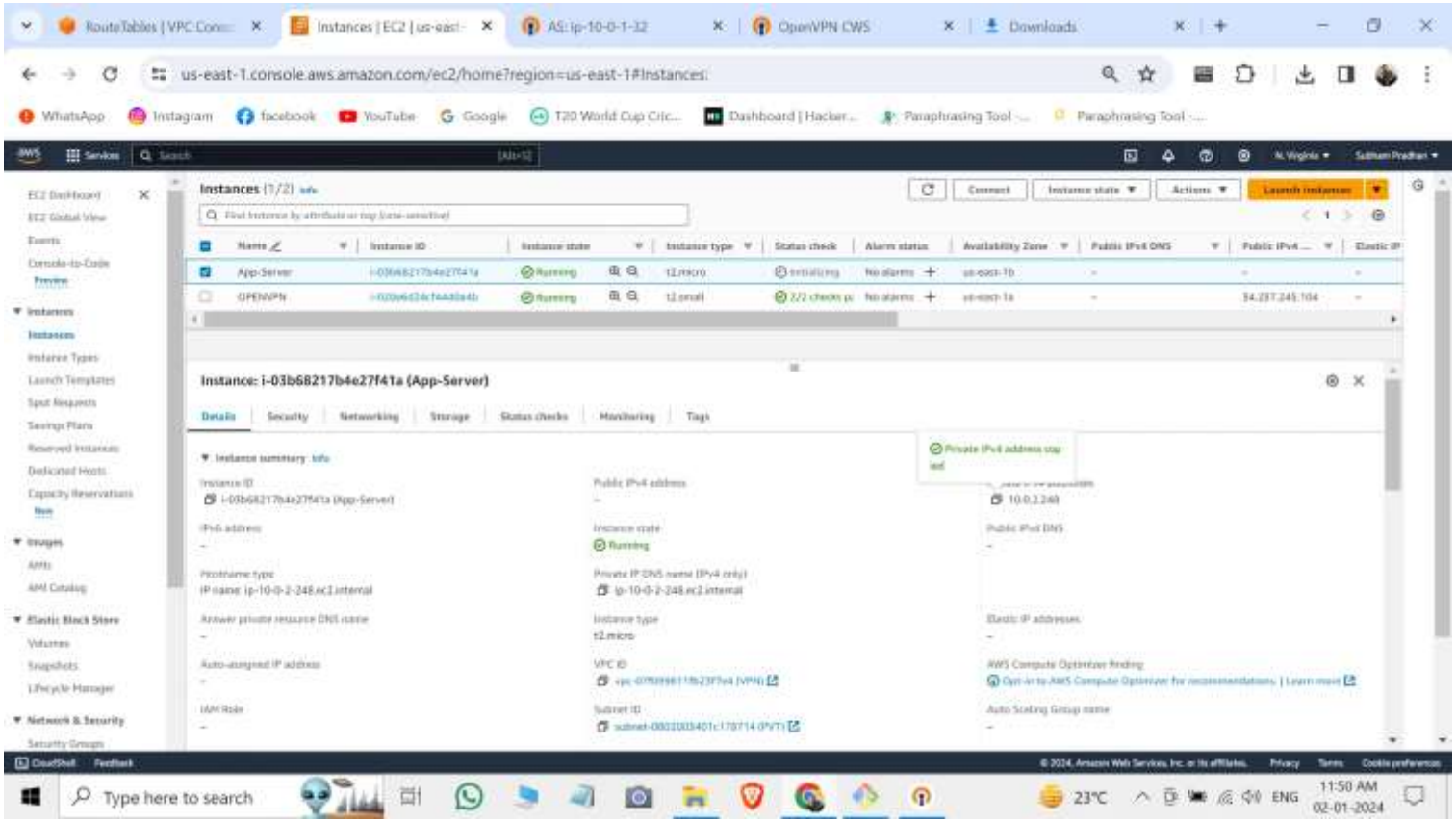
- Ping again

```
ASUS@LAPTOP-Q9QDRNKO MINGW64 ~/Downloads
$ ping 10.0.1.32

Pinging 10.0.1.32 with 32 bytes of data:
Reply from 10.0.1.32: bytes=32 time=272ms TTL=64
Reply from 10.0.1.32: bytes=32 time=322ms TTL=64
Reply from 10.0.1.32: bytes=32 time=394ms TTL=64
Reply from 10.0.1.32: bytes=32 time=424ms TTL=64

Ping statistics for 10.0.1.32:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 272ms, Maximum = 424ms, Average = 353ms
```

IMPLEMENTATION

Step 6: Create Additional Resources

- Create another EC2 instance (APP-SERVER) in the PVT subnet.
- Set up a NAT Gateway in the PUB subnet.
- Route the NAT Gateway in the PVT subnet.



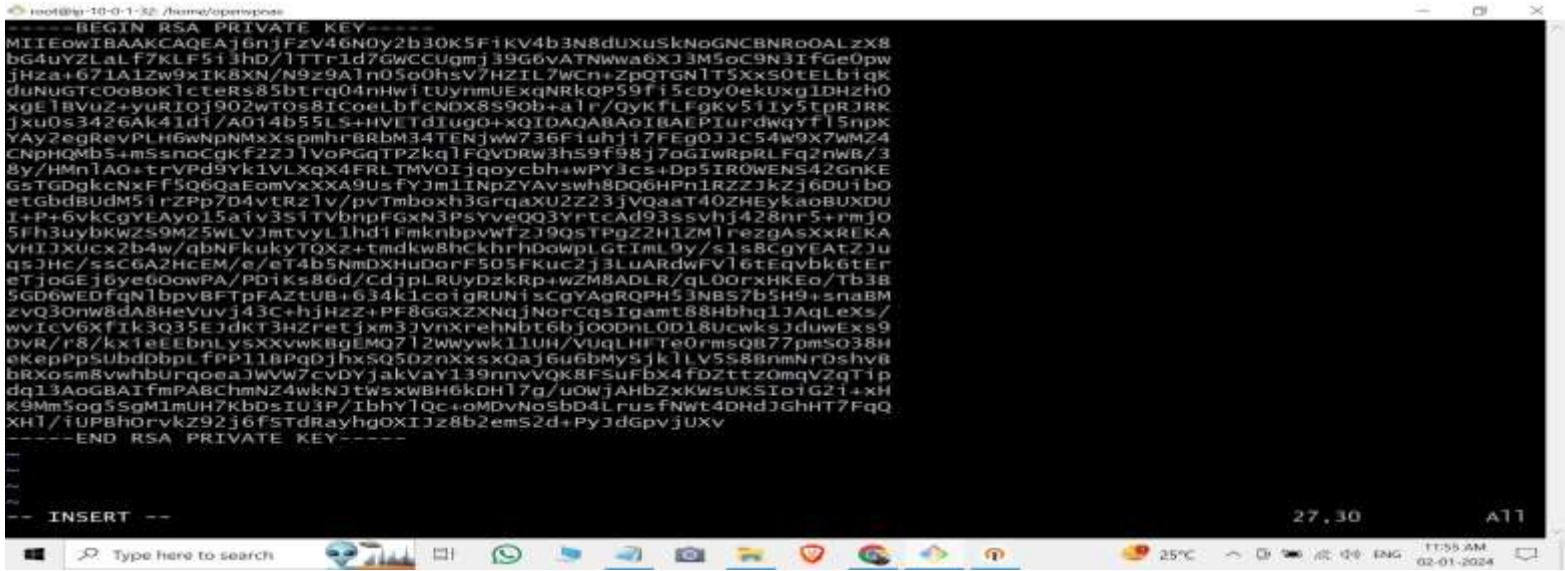
IMPLEMENTATION

Step 7: Test Communication with NAT Gateway

- Reconnect to the OpenVPN server.
- Ping the private IP of the APP-SERVER; it should ping due to the NAT Gateway.

```
openvpnas@ip-10-0-1-32:~$ sudo su
root@ip-10-0-1-32:/home/openvpnas# ping 10.0.1.32
PING 10.0.1.32 (10.0.1.32) 56(84) bytes of data.
64 bytes from 10.0.1.32: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 10.0.1.32: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.0.1.32: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 10.0.1.32: icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from 10.0.1.32: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 10.0.1.32: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 10.0.1.32: icmp_seq=7 ttl=64 time=0.044 ms
```

- Create a test-key on APP-SERVER, change permissions, and perform an SSH connection.

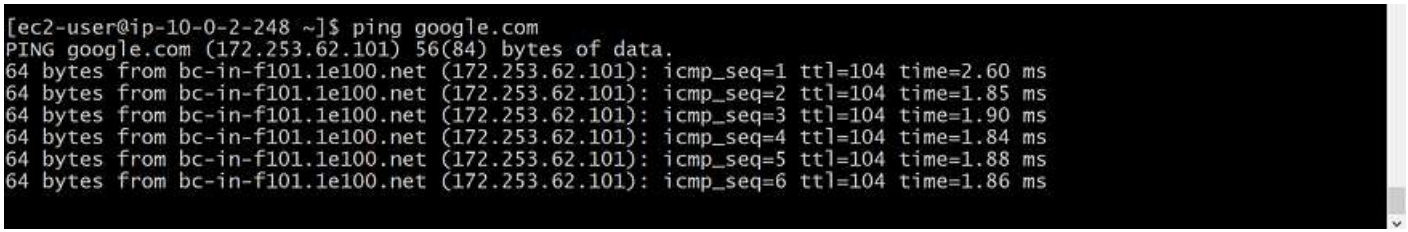


IMPLEMENTATION

Step 8: Install Apache Server on APP-SERVER

- connect to the APP-SERVER.

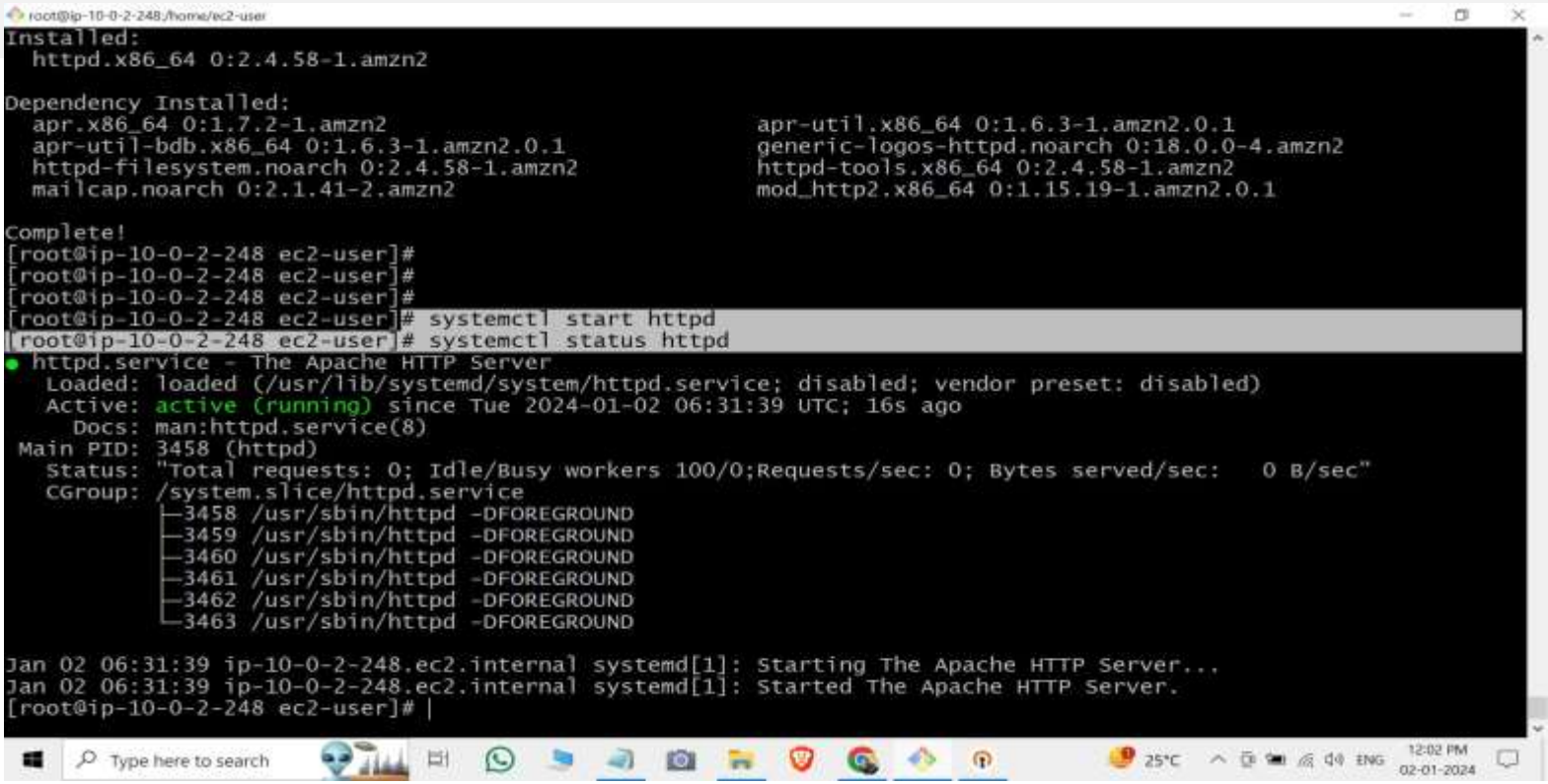
- ping google.com



IMPLEMENTATION

Step 8: Install Apache Server on APP-SERVER

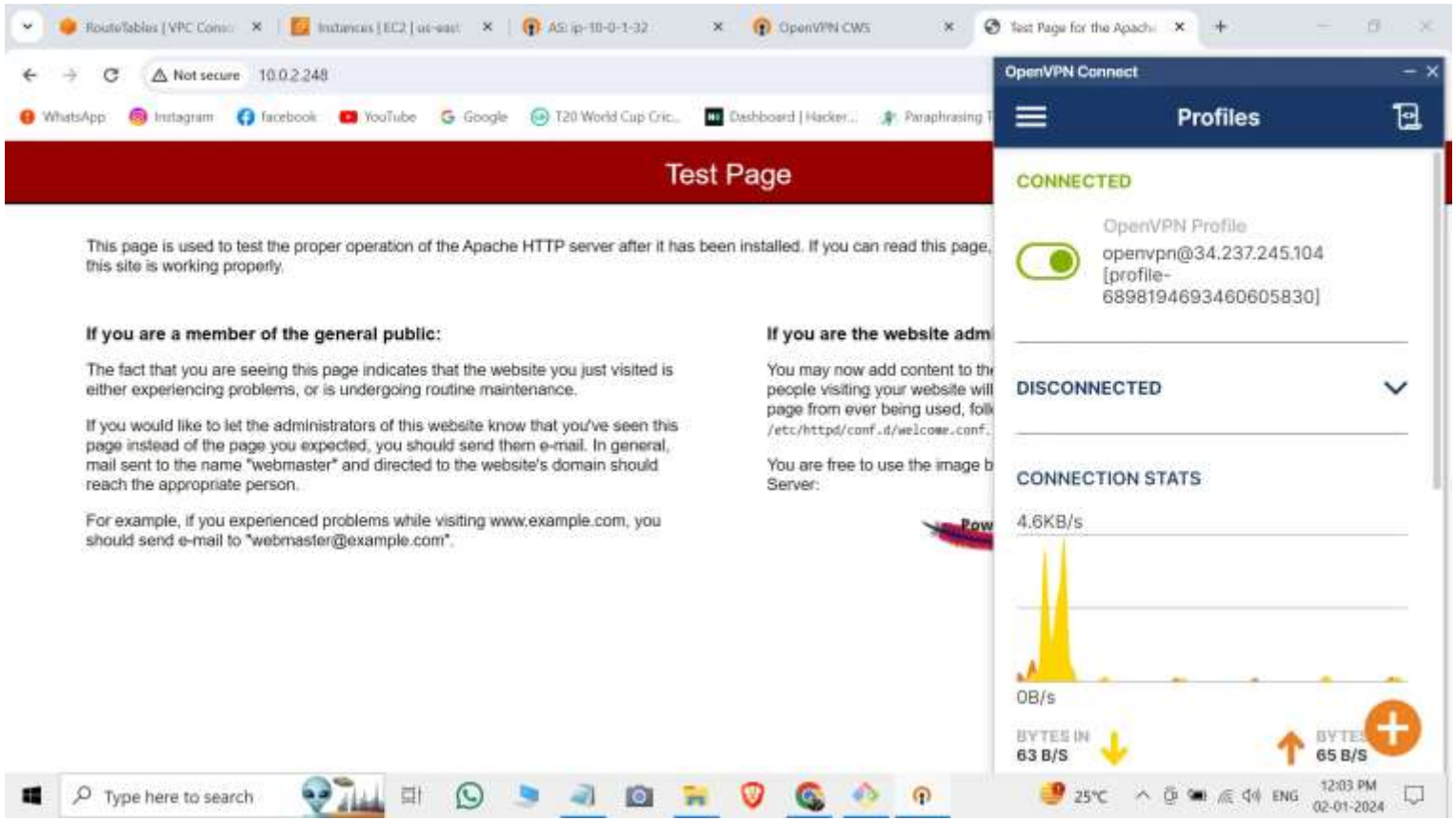
- Switch to sudo su on APP-SERVER.
- Install Apache: yum install httpd -y.
- Start Apache: systemctl start httpd.
- Check status: systemctl status httpd.



RESULT

Access Apache by pasting the private IP of APP-SERVER; it should display the Apache server.

*(**Note**: Ensure that the VPN connection is active during testing.)*



CONCLUSION

•Implemented a secure Point-to-Site VPN using AWS Client VPN service.

•Enabled remote user access to AWS VPC resources via an encrypted, OpenVPN-based tunnel.

•Ensured confidentiality, integrity, and controlled access with mutual TLS authentication and AWS IAM/ACM integration.

•Achieved scalability, flexibility (split-tunnel/full-tunnel), and observability.

•Final outcome: Remote users can securely connect and manage AWS cloud resources as if they were inside the VPC without exposing them directly to the internet.

REFERENCES

- **AWS Documentation – Client VPN**
  https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/what-is.html
- **Medium – Point to Site Connection**
  https://medium.com/@subhampradhan966/setting-up-point-to-site-vpn-connection-in-aws-a-step-by-step-guide-9edd65e6ca44
- **AWS Documentation – Tutorial: Create and Configure a Client VPN Endpoint**
  https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/cvpn-getting-started.html
- **Nord VPN - What is it, and how does it work?**
  https://nordvpn.com/blog/point-to-site-vpn/?srsltid=AfmBOoruoxZL6qczyCqvfo1h1imnnMaWL4FkDnzhw1AZ2yWUOlRPsyG5