Assessment Title
Practical Demonstration (AWS)

CCC603| Assessment-5

The Student Name | Geni Bahar
Student ID: 27085022

Tutor/Accessor | Ovesh Vohra

Programme
Diploma in Cloud Computing and Cyber Security (120 Credits)

Course
CCC603: Cyber Security in Cloud
(Level 6, 30 Credits)

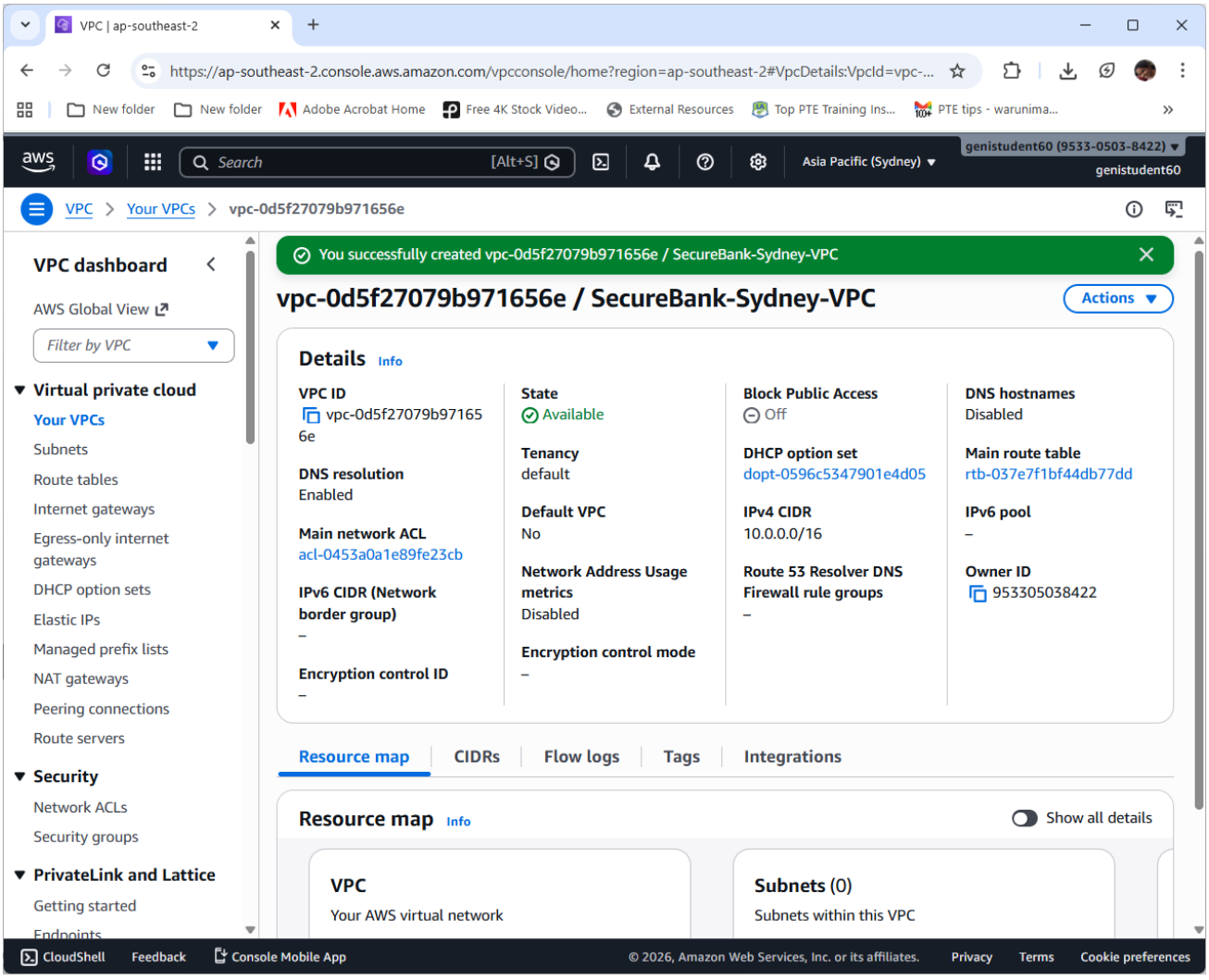
Date: 29th March 2026

Table of Contents

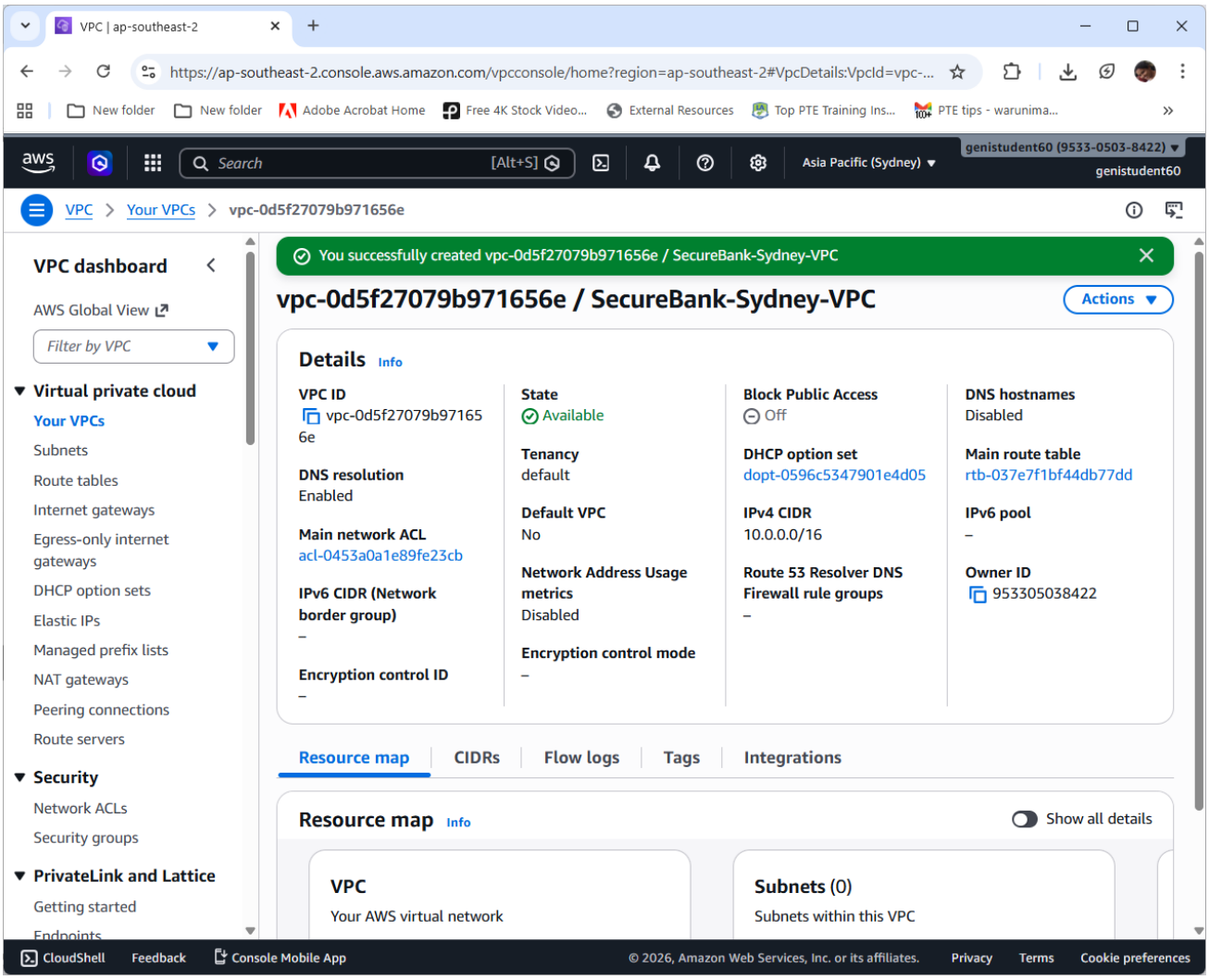
Task 1: Secure VPC Setup & Architectural Design

Activity 1.1: VPC Setup and Configuration

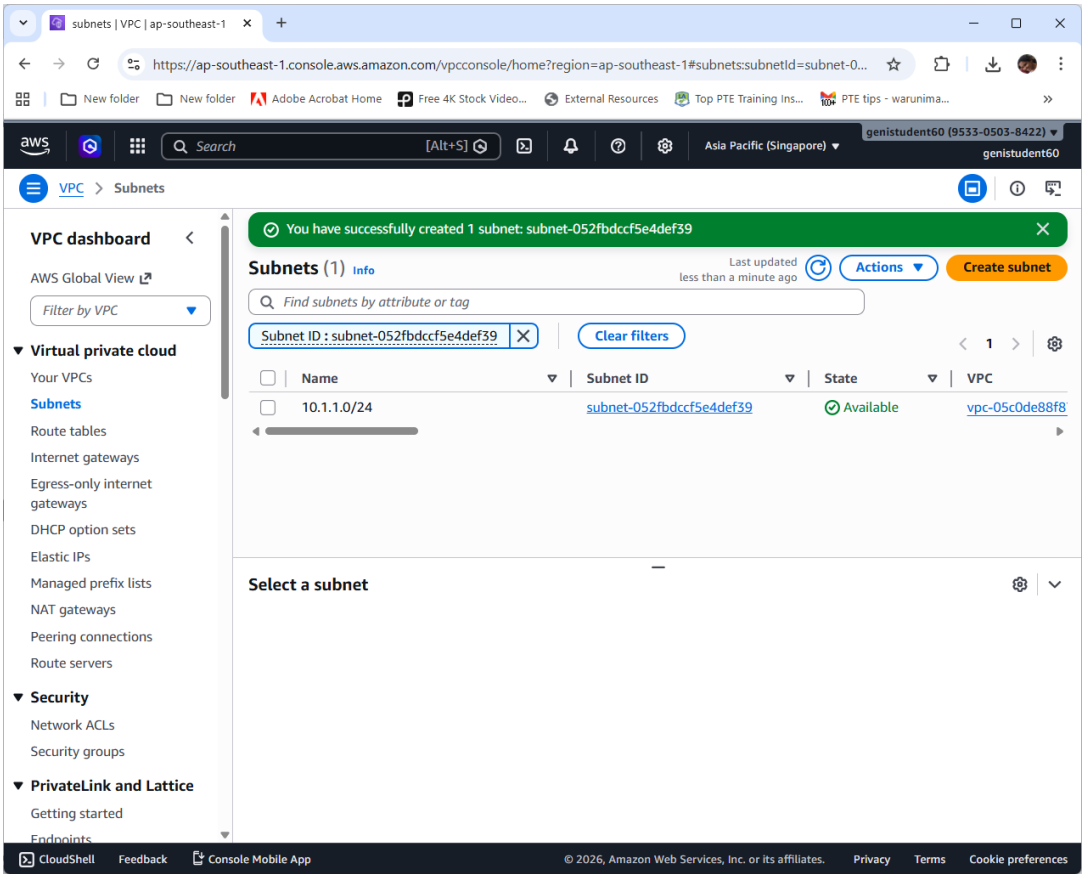
Created first VPC named SecureBank-Sydney-VPC

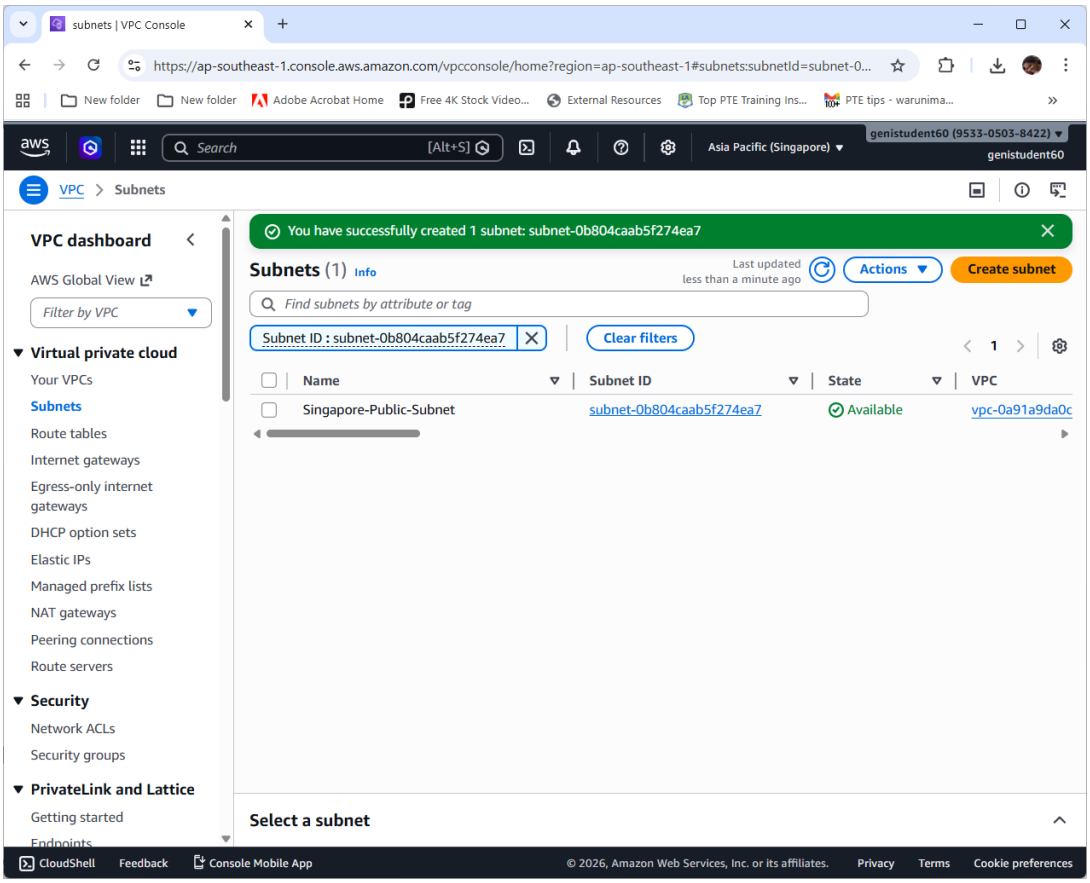


Created second VPC named SecureBank-Singapore-VPC

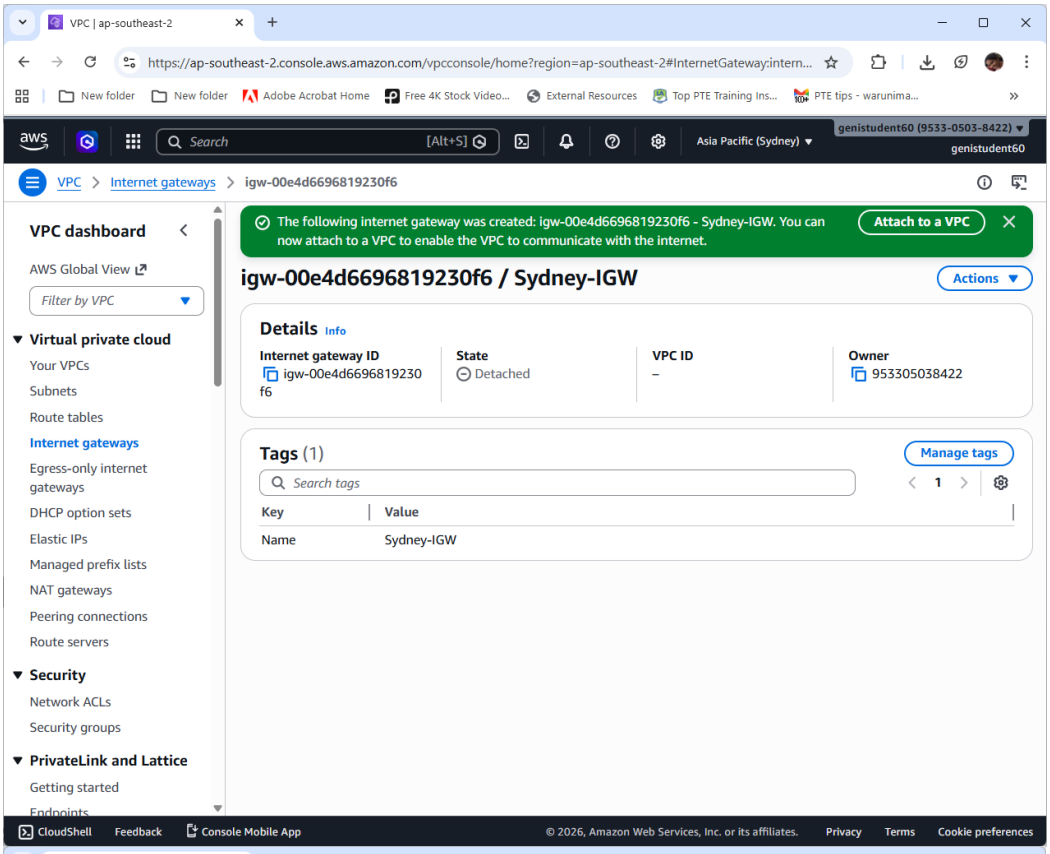


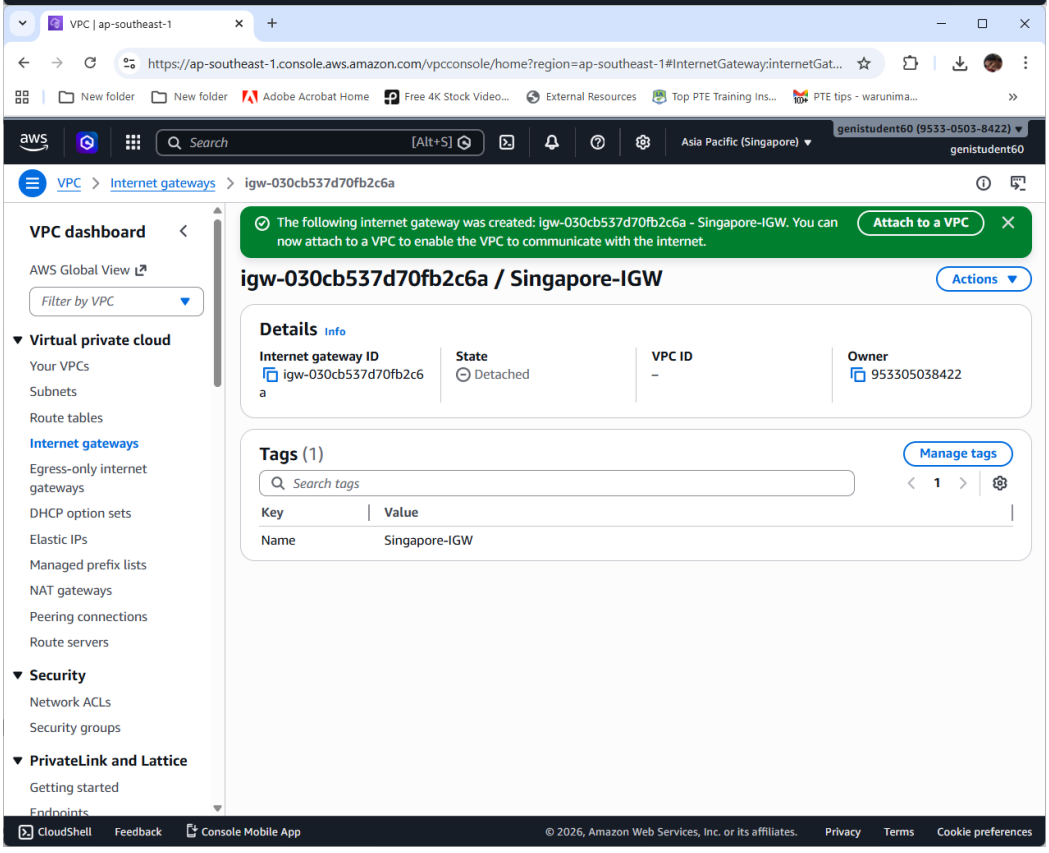
Created subnets in both VPCs



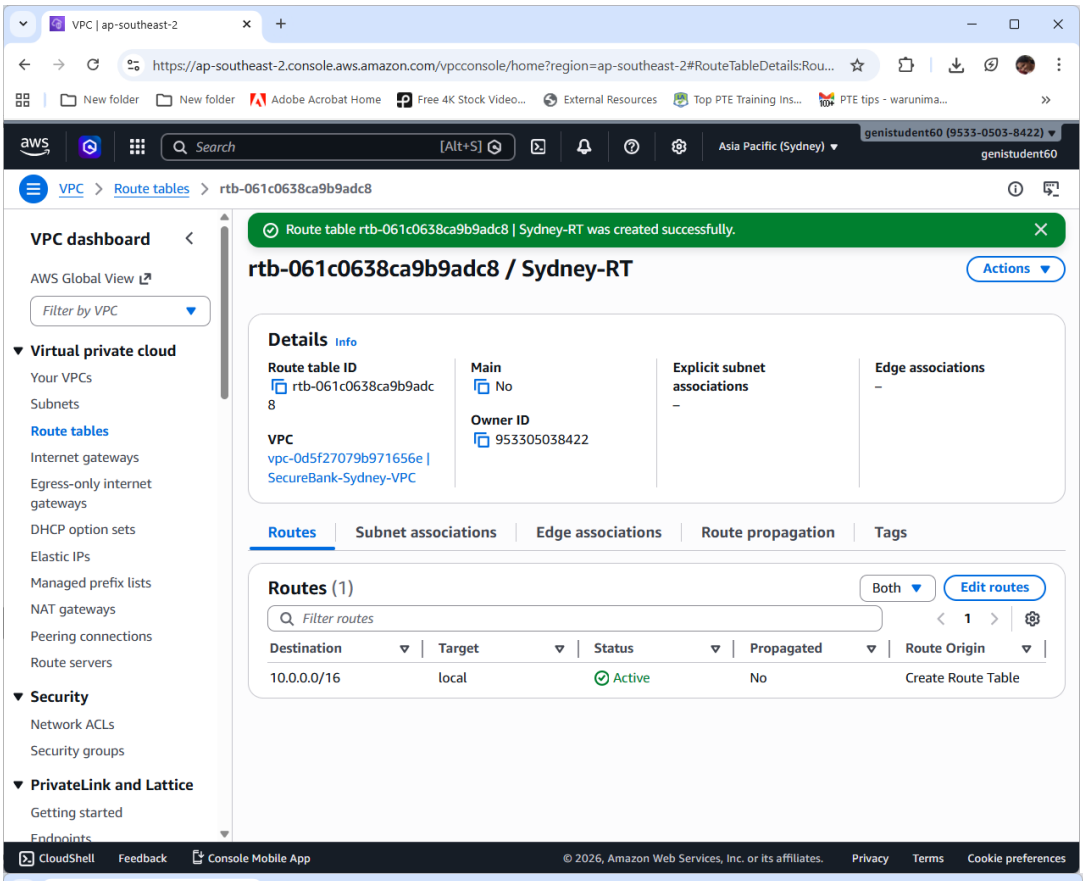


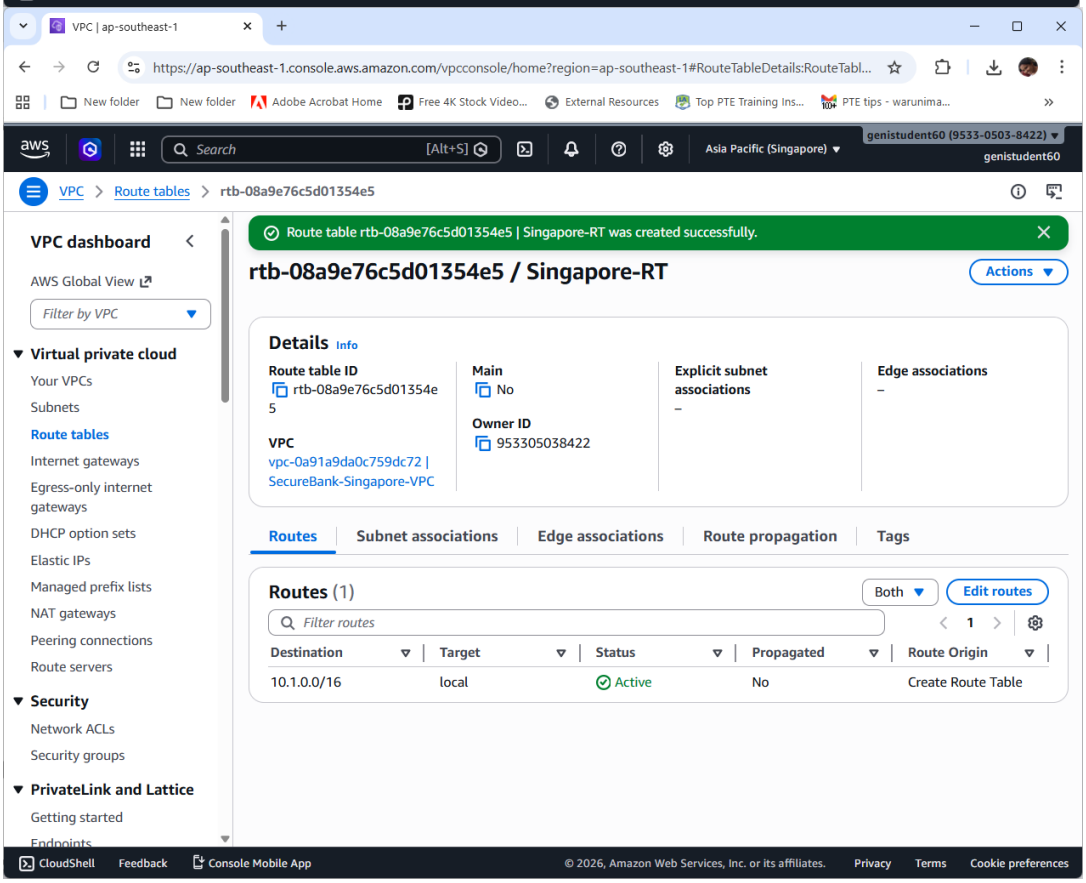
Attached Internet Gateways to both VPCs



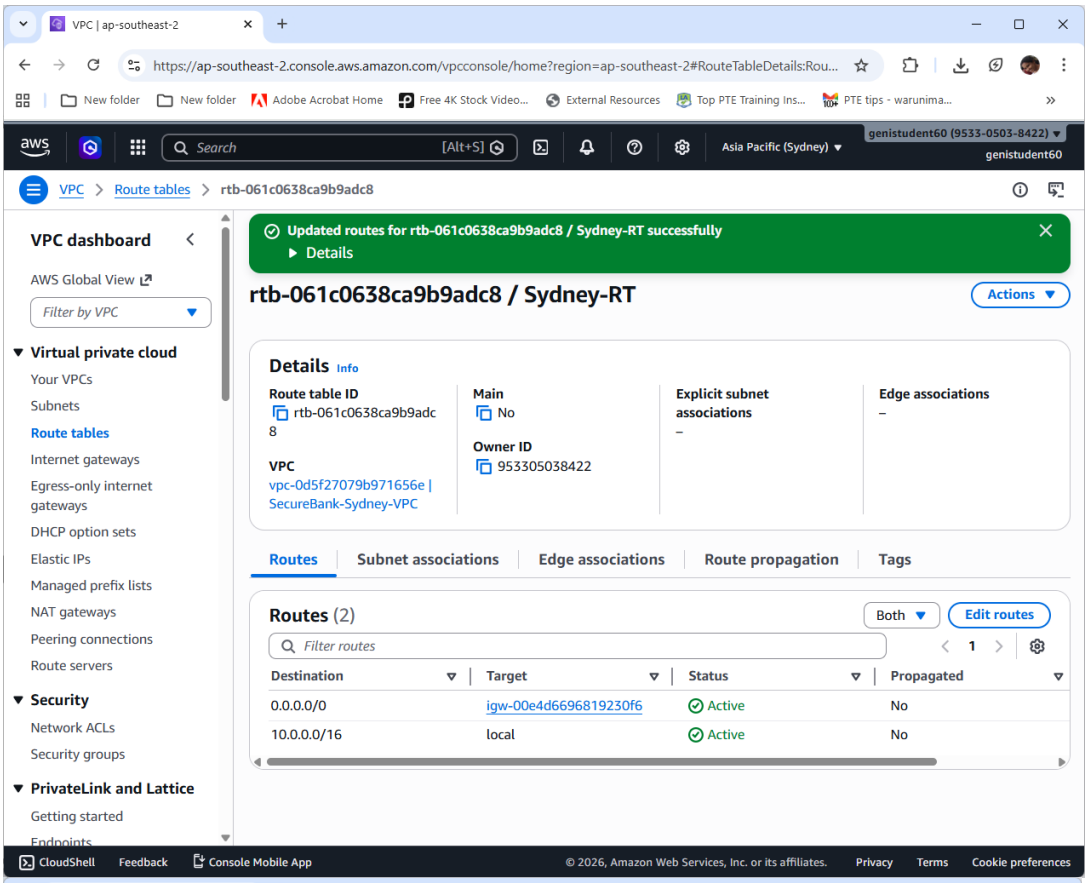


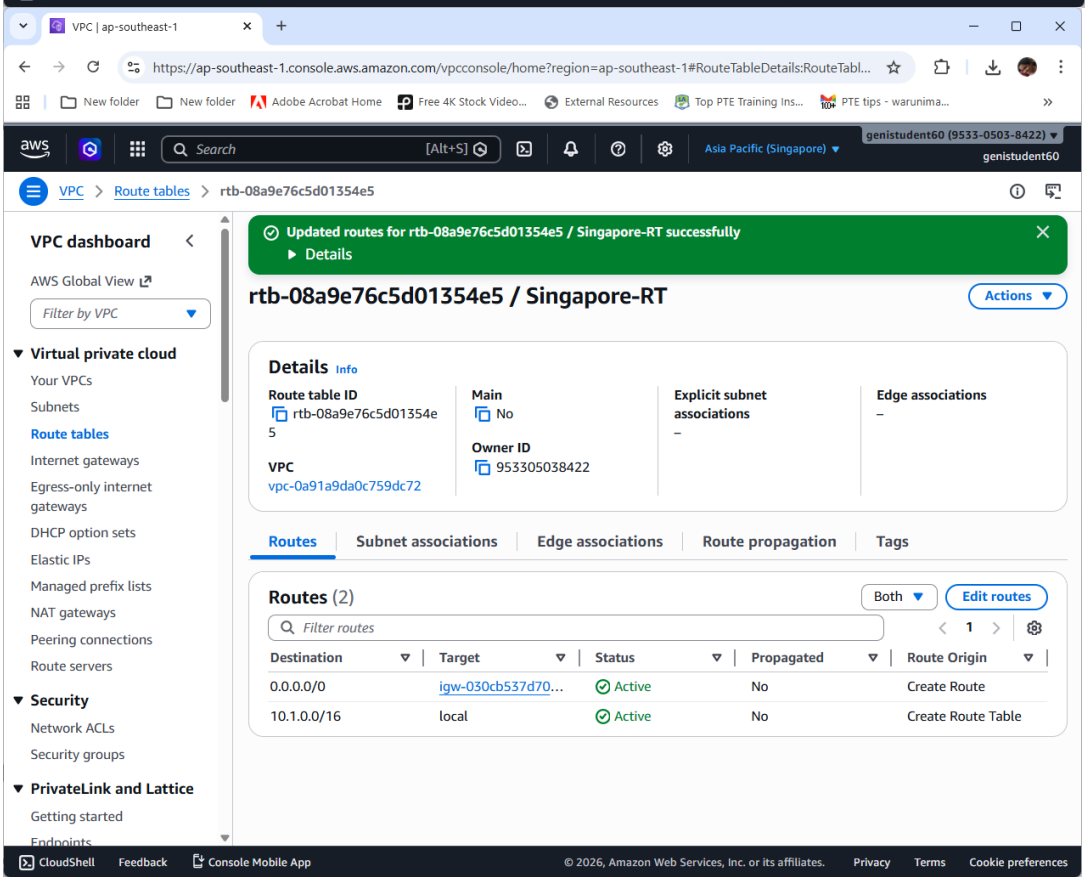
Configured Route Tables



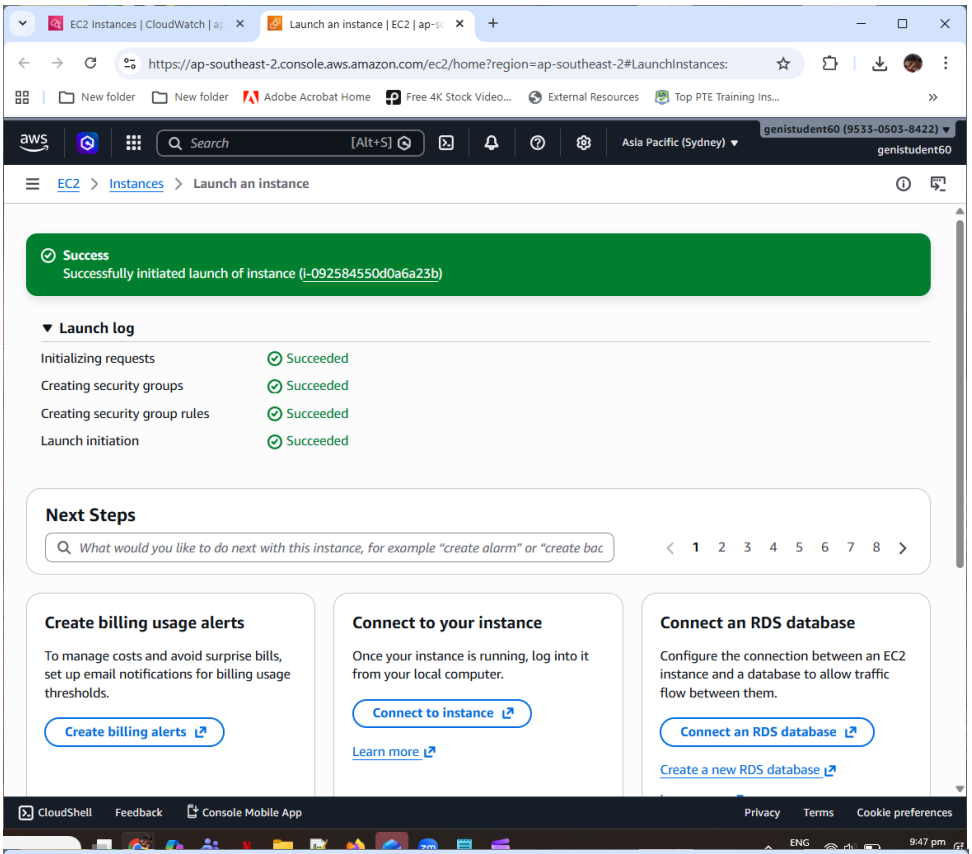


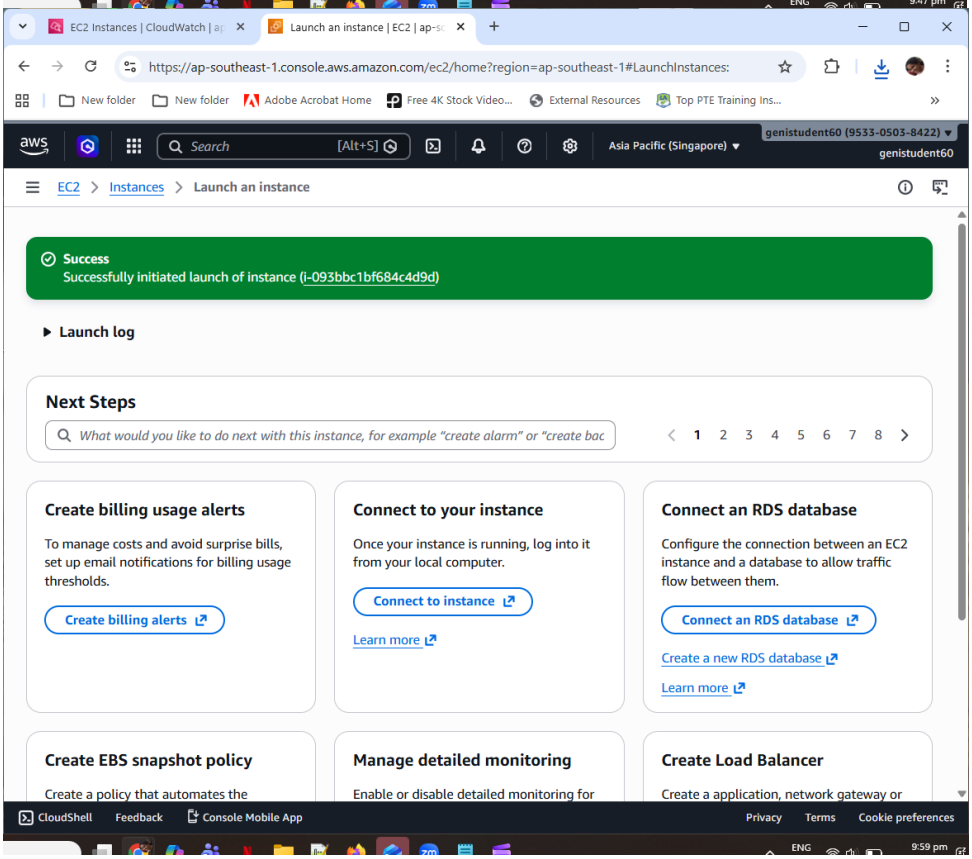
Added routes to allow internet access



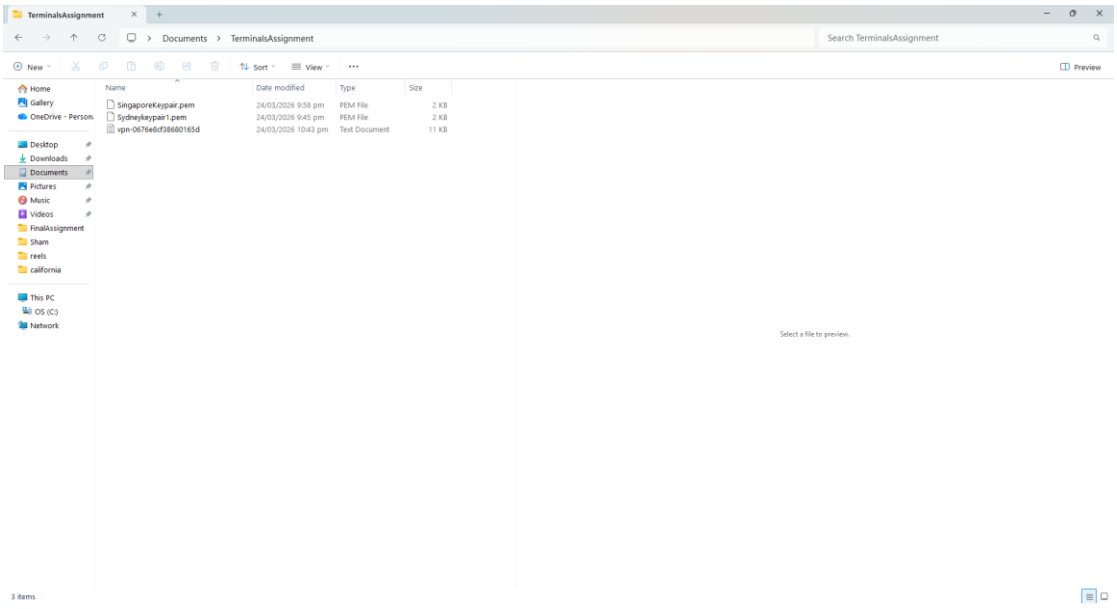


Launched EC2 instances in both regions

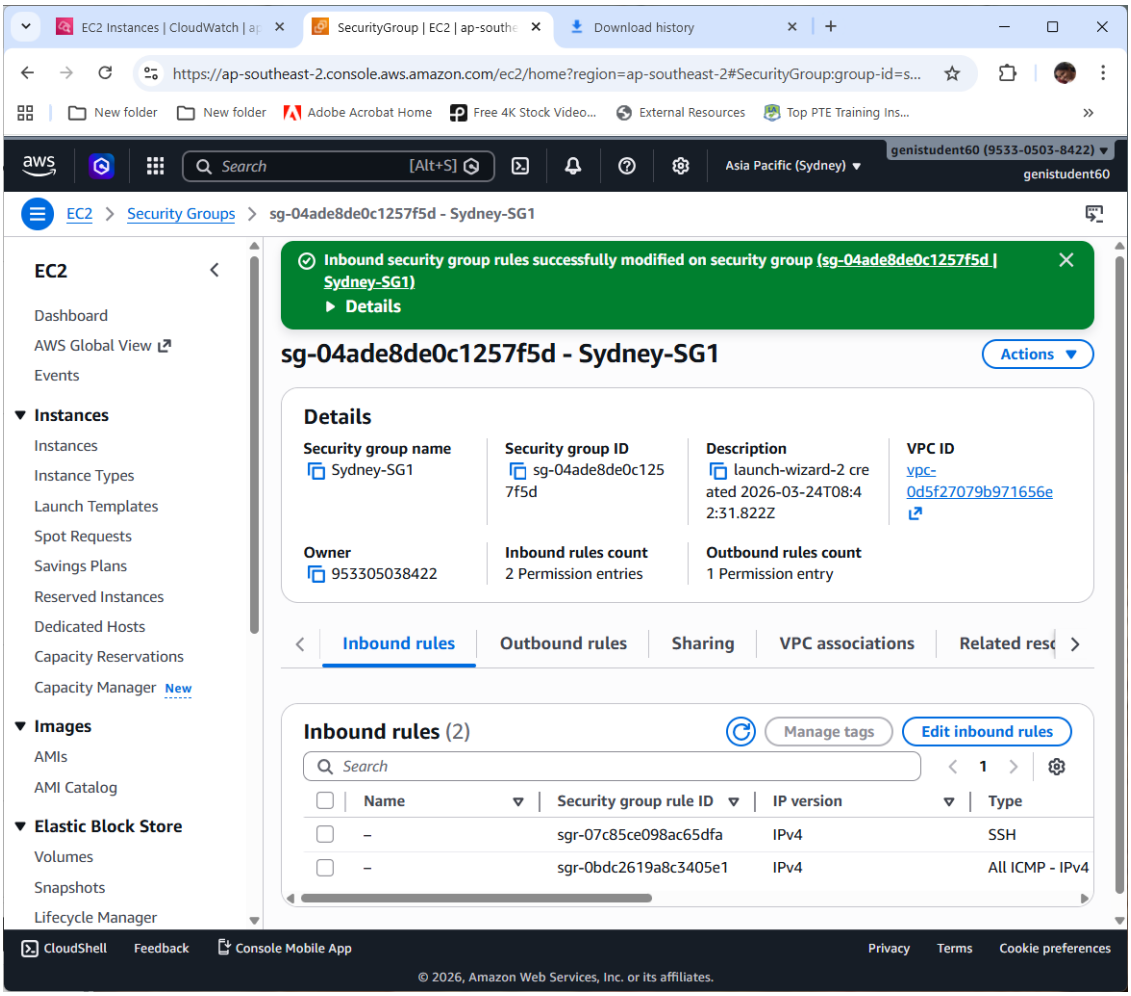




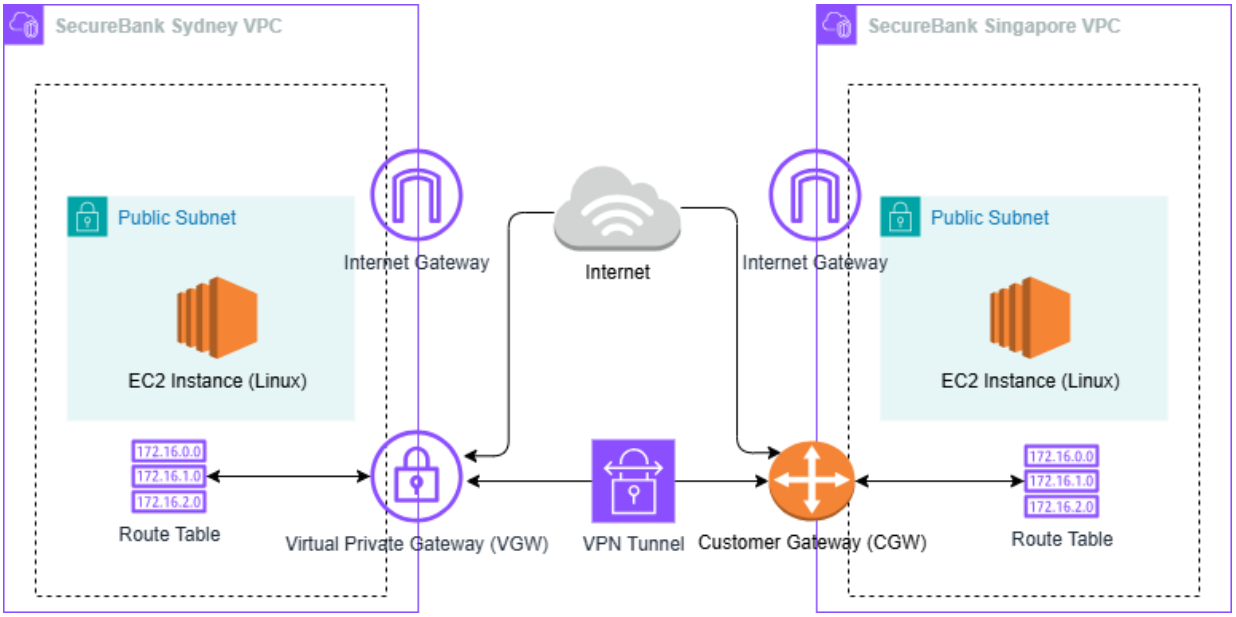
Key pairs created and downloaded



Configured Security Groups



Activity 1.2: Architectural Diagram



How This Design Protects SecureBank Assets

When I started designing the architecture for SecureBank, my main focus was not just making the connection work, but making sure the data stays protected at all times. Since this is a banking environment, even a small mistake in network design can expose sensitive financial information.
To begin with, I created two completely separate VPCs for Sydney and Singapore. I used different CIDR ranges for each VPC to avoid any overlap. This is important because overlapping IP ranges can cause routing issues and may even lead to data being sent to the wrong location. By keeping them separate, I ensured a clean and controlled network structure.
Another important decision I made was to isolate resources using subnets. Even though this is a simple setup, I still followed the idea of separating components logically. This helps in future scaling and also reduces risk if one part of the system is compromised.
For security, I paid special attention to Security Groups. In many lab setups, it is common to allow access from anywhere just to make things easier. However, I avoided using 0.0.0.0/0 unless absolutely necessary. Instead, I restricted SSH access to only my own IP address. This means that even if someone discovers the public IP of the EC2 instance, they cannot attempt to access it.
The Internet Gateway was attached to both VPCs, but I made sure it is only used where necessary. The main purpose here is to allow controlled communication and support the VPN setup later. Internal communication is not dependent on the public internet, which improves security.
Another important part of the design is that both VPCs are isolated environments. Even if one region is compromised, the attacker cannot directly access the other region. This creates an additional layer of protection for SecureBank.
Overall, this architecture follows a simple but effective security approach:

- Isolate environments using separate VPCs
- Control access using Security Groups

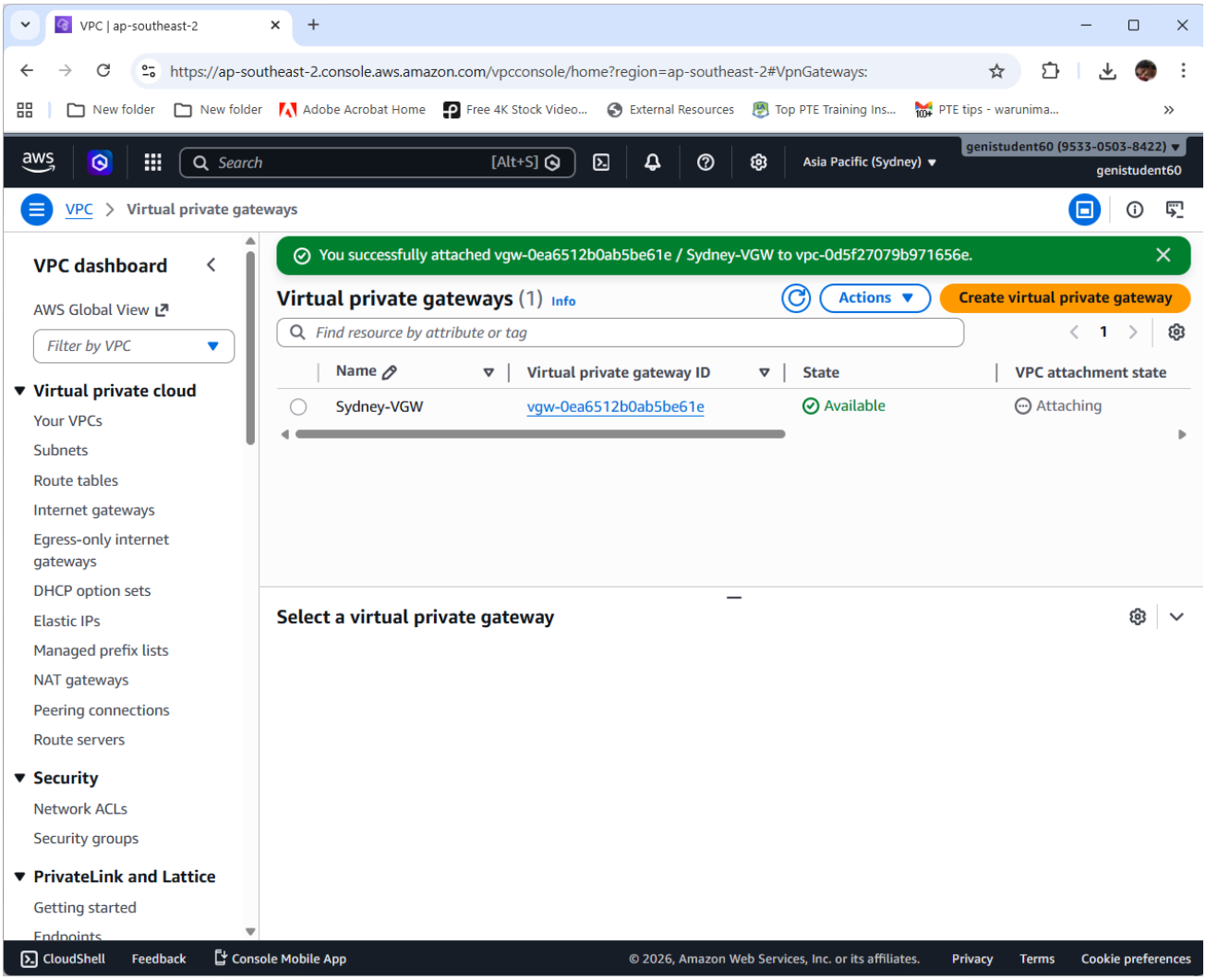
- Avoid unnecessary public exposure
- Prepare the network for secure VPN communication

From this setup, I understood that security is not about adding more services, but about designing the network correctly from the beginning. Even a basic VPC design can provide strong protection if done properly.

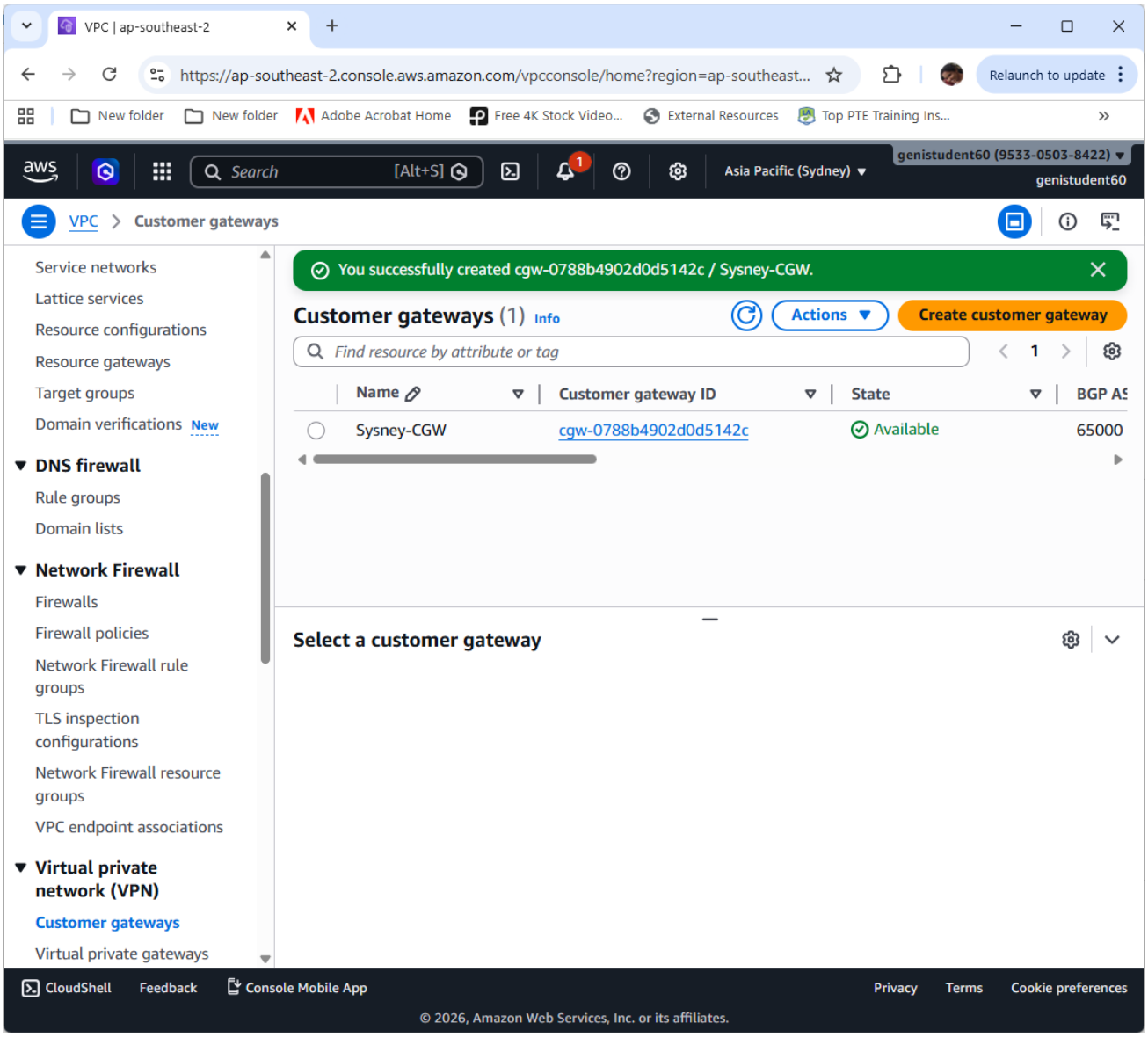
Task 2: VPN Tunnel Configuration

Activity 2.1: VPN Gateway Setup

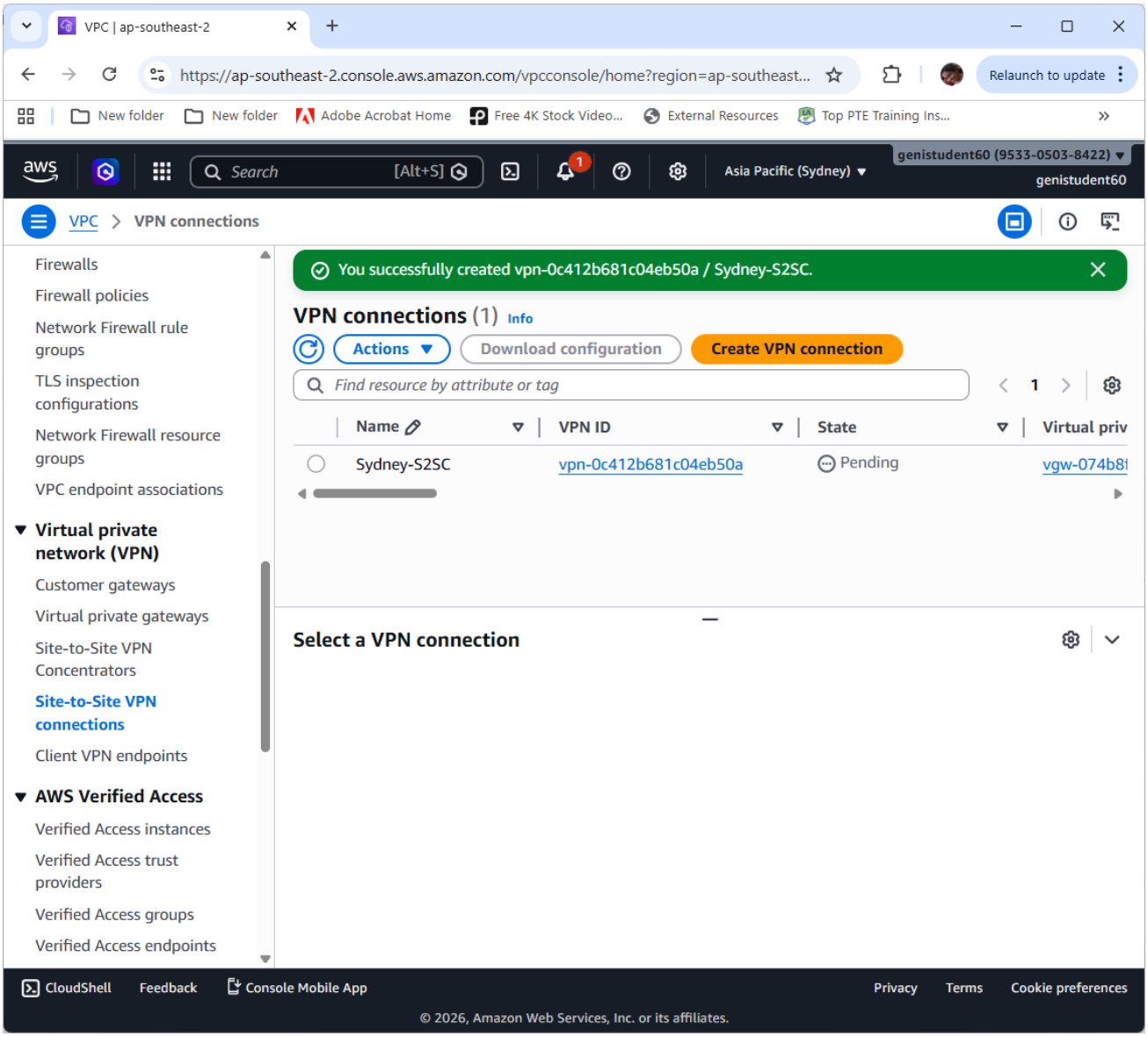
Created Virtual Private Gateway (VGW) in Sydney VPC



Created Customer Gateway (CGW)

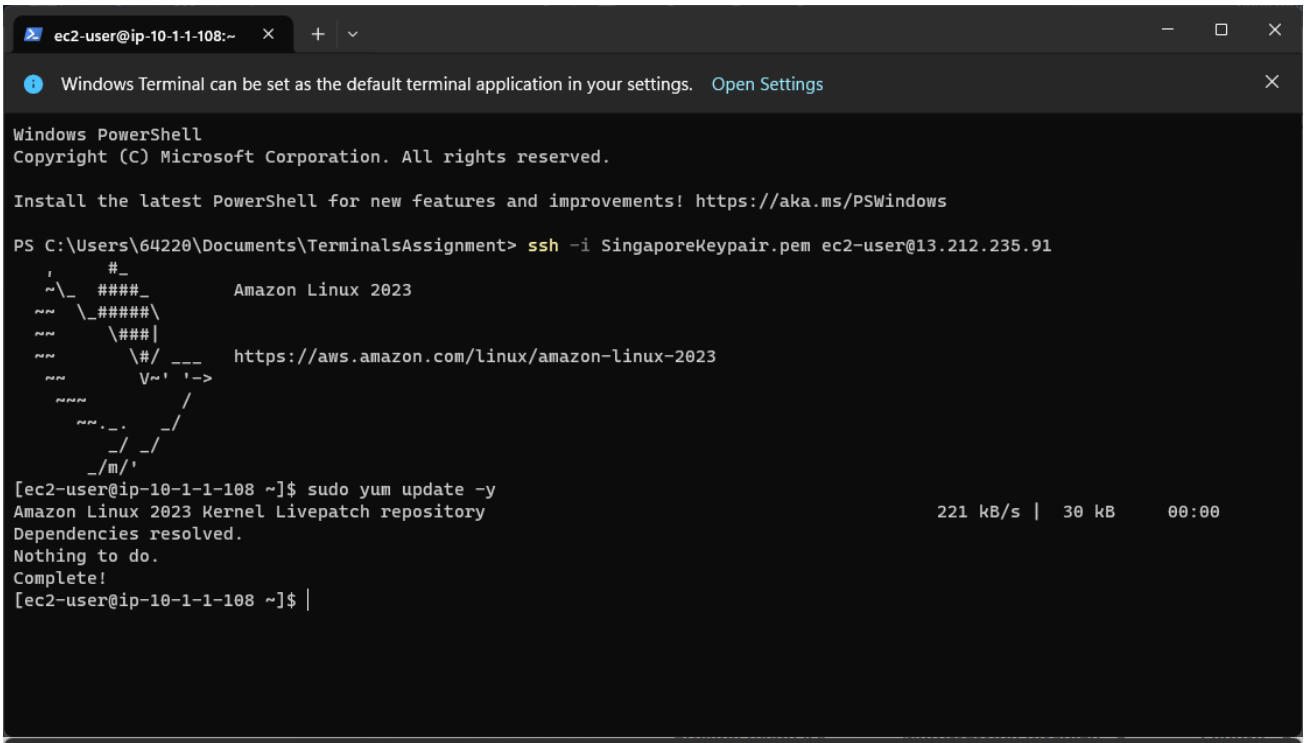


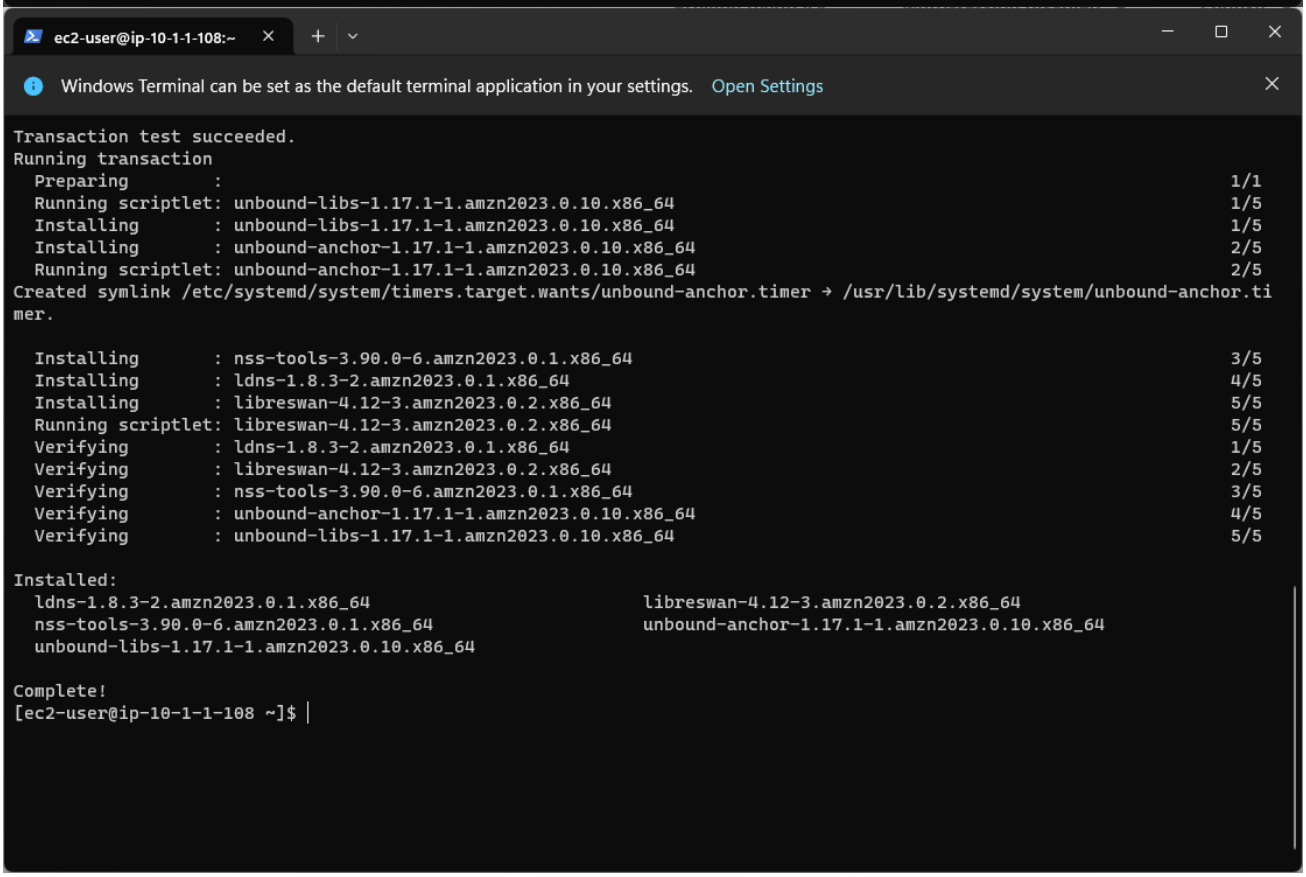
Created Site-to-Site VPN connection



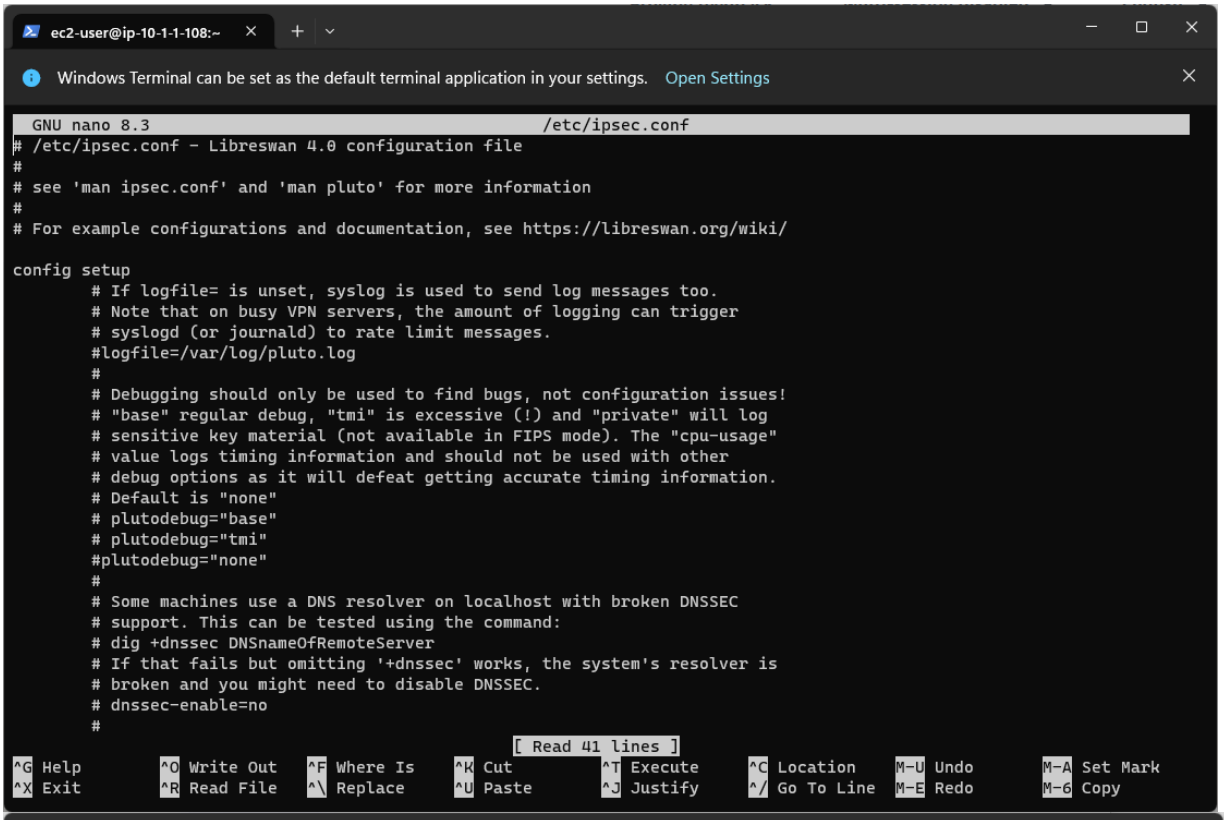
Activity 2.2: VPN Configuration on Singapore EC2

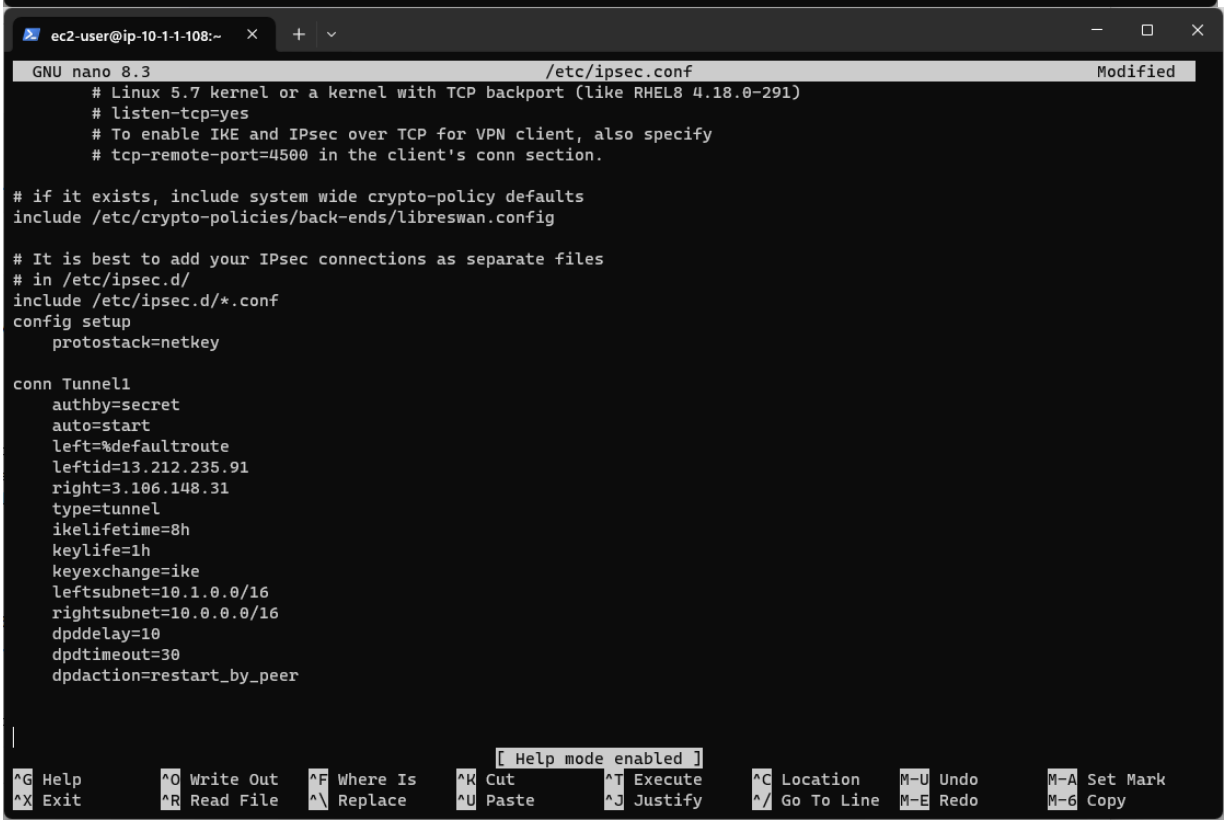
Installed VPN software (Libreswan/StrongSwan)

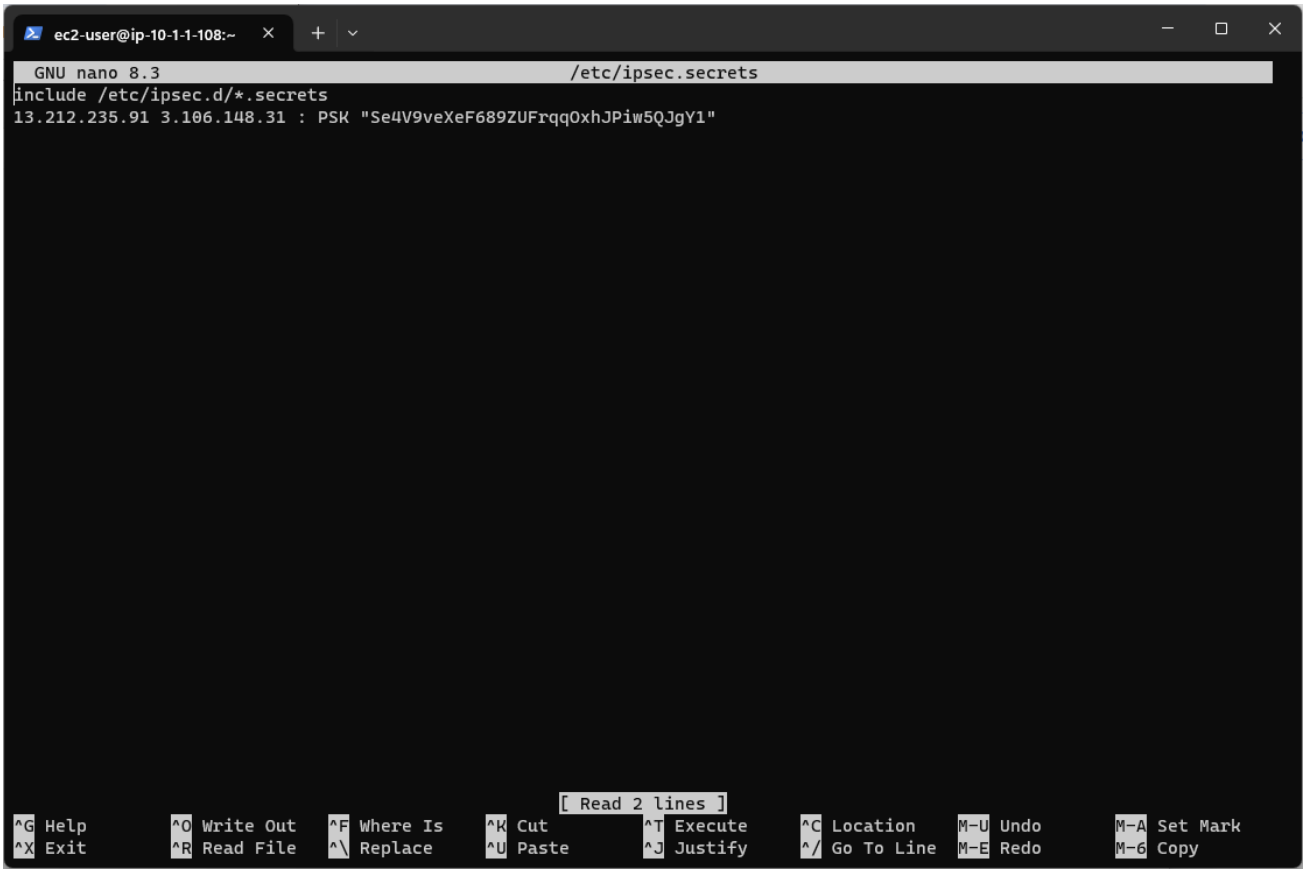




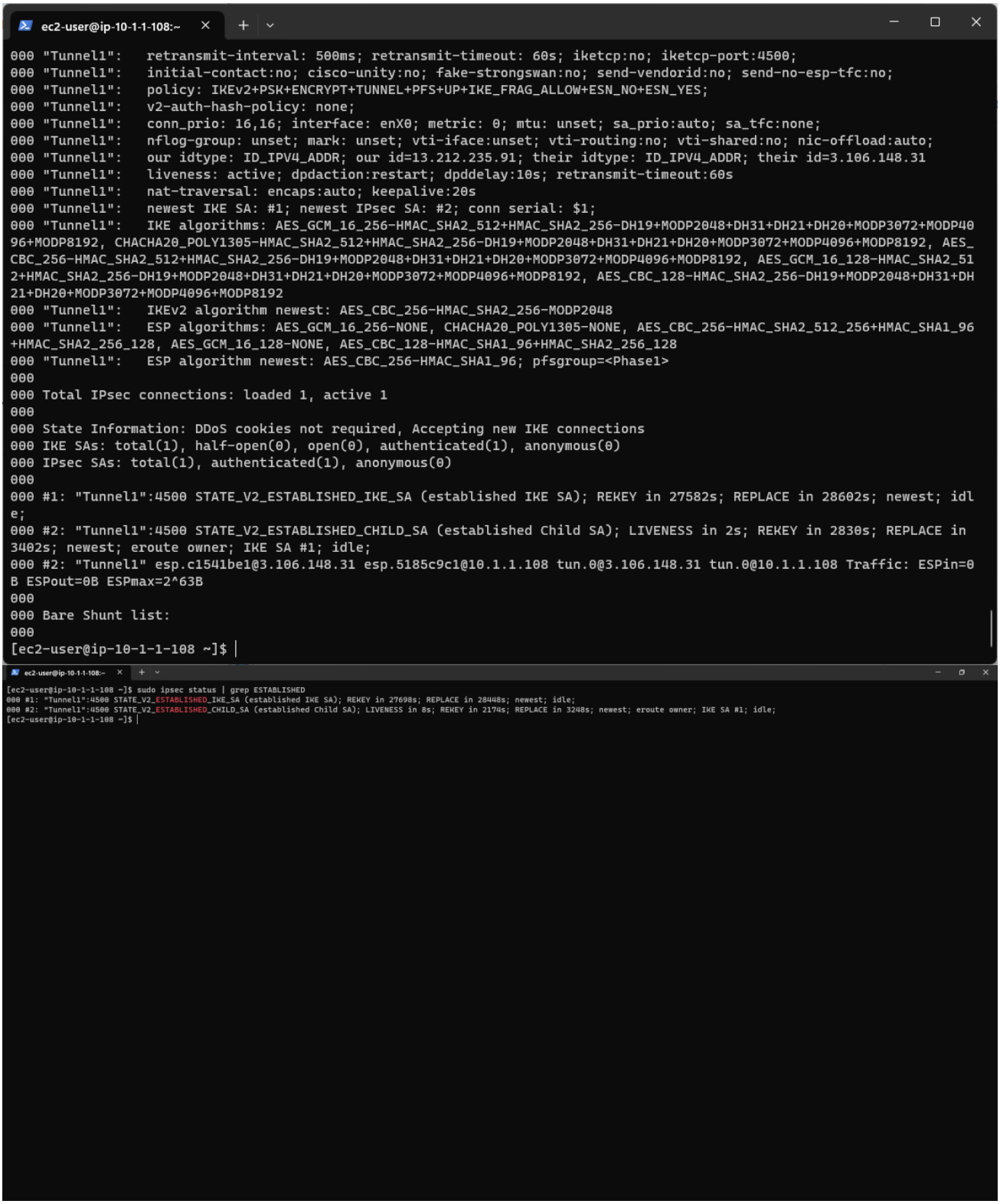
Edited configuration files







Started VPN service and checked status



The VPN tunnel status was shown as “ESTABLISHED” in the EC2 instance, with active IKE and IPsec Security Associations. This confirms that the encrypted tunnel is successfully established.

Encryption, IKE and Best Practices

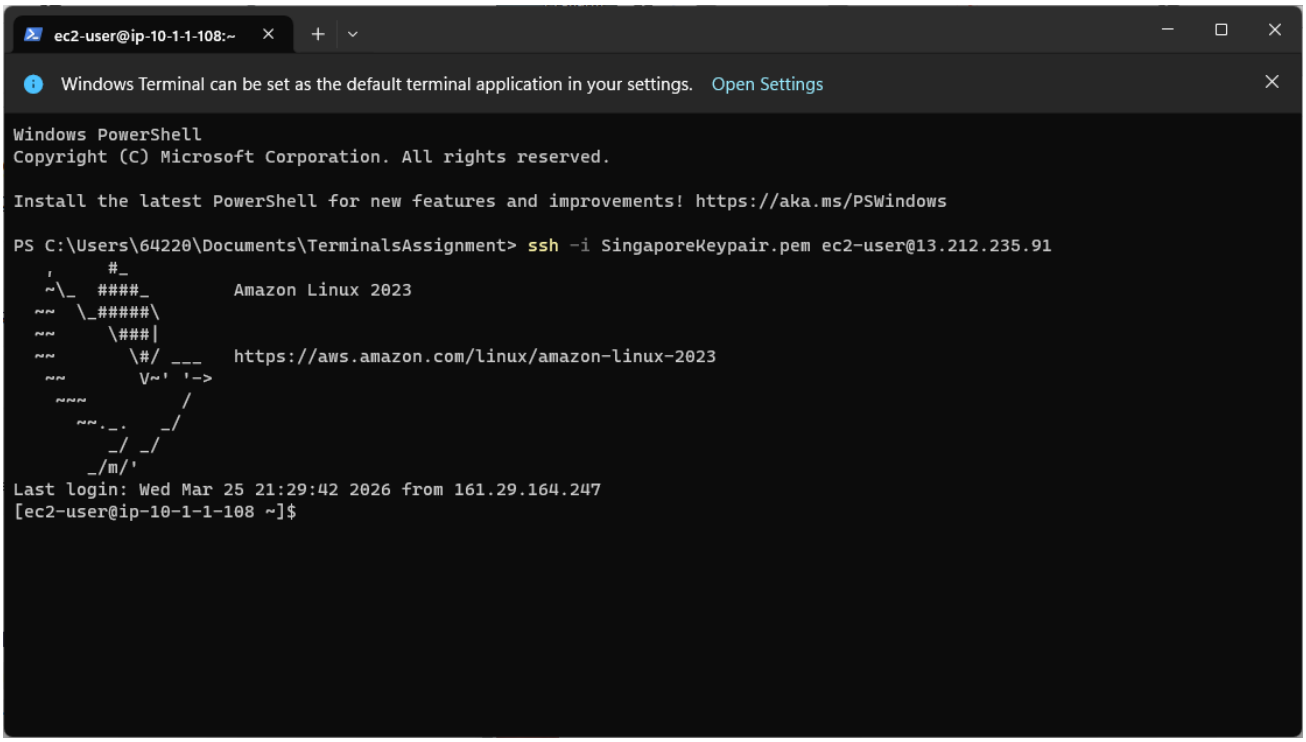
In this task, the main goal was to create a secure Site-to-Site VPN between the Sydney and Singapore environments. This means all communication between the two VPCs should be encrypted and protected from any external access.
To achieve this, I configured Libreswan on the Singapore EC2 instance. This instance acts as the Customer Gateway (CGW), while AWS manages the Virtual Private Gateway (VGW) on the Sydney side. During my implementation, setting this up was a bit challenging because even a small mistake in the configuration files can stop the VPN from working properly
For encryption, I used AES (Advanced Encryption Standard). I chose AES because it is widely used in the industry, especially in banking and financial systems. It provides strong security while still being efficient for handling large amounts of data. Compared to older methods like DES or 3DES, AES is much more secure and is recommended in modern systems.
In addition to encryption, I configured IKE (Internet Key Exchange). IKE is responsible for securely exchanging keys between the two endpoints. Instead of sending the actual encryption key over the network, both sides agree on a shared key using a secure process. This reduces the risk of interception and ensures that only trusted systems can communicate.
During my setup, I faced an issue where the VPN tunnel was not coming up. After checking the configuration, I realised that I had used the private IP instead of the public Elastic IP in the configuration. Once I corrected this, the VPN tunnel status changed to “ESTABLISHED”. This helped me understand how important it is to use the correct identifiers when setting up secure connections.
Another important concept in this setup is multiple tunnelling. AWS automatically provides two VPN tunnels for redundancy. Even though I mainly configured one tunnel for this demonstration, the second tunnel can act as a backup. If one tunnel fails, the other can continue the communication without interruption. This is important in real-world scenarios where downtime is not acceptable.
I also implemented a best practice by configuring the system to automatically restart the VPN if the connection drops. This ensures that the connection remains stable and reduces manual intervention.
Overall, this setup follows key security best practices:

- Using strong encryption (AES)
- Secure key exchange using IKE
- Avoiding outdated encryption methods
- Supporting high availability with multiple tunnels
- Ensuring automatic recovery of the VPN connection

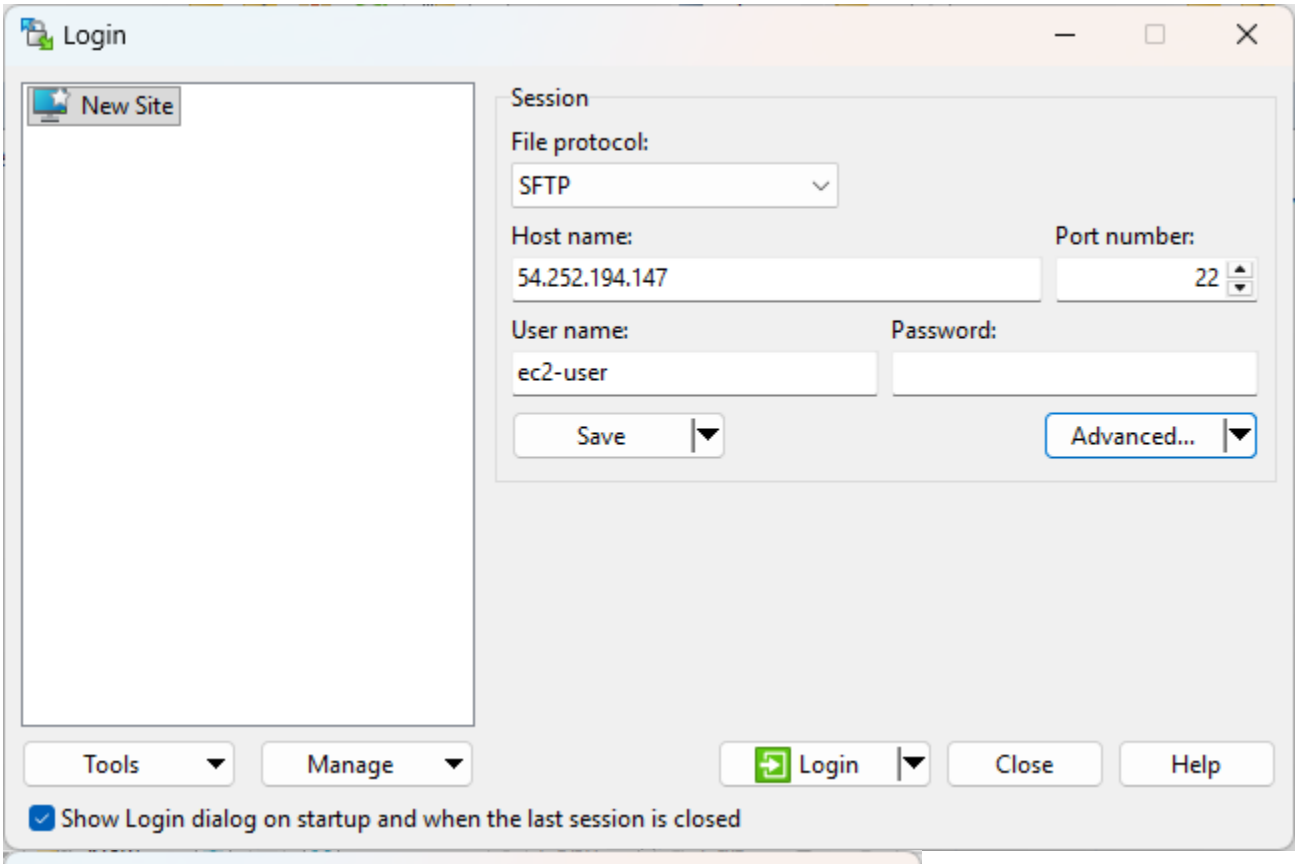
From this task, I learned that setting up a VPN is not just about connectivity. It is mainly about securing the communication and making sure the system remains reliable under different conditions.

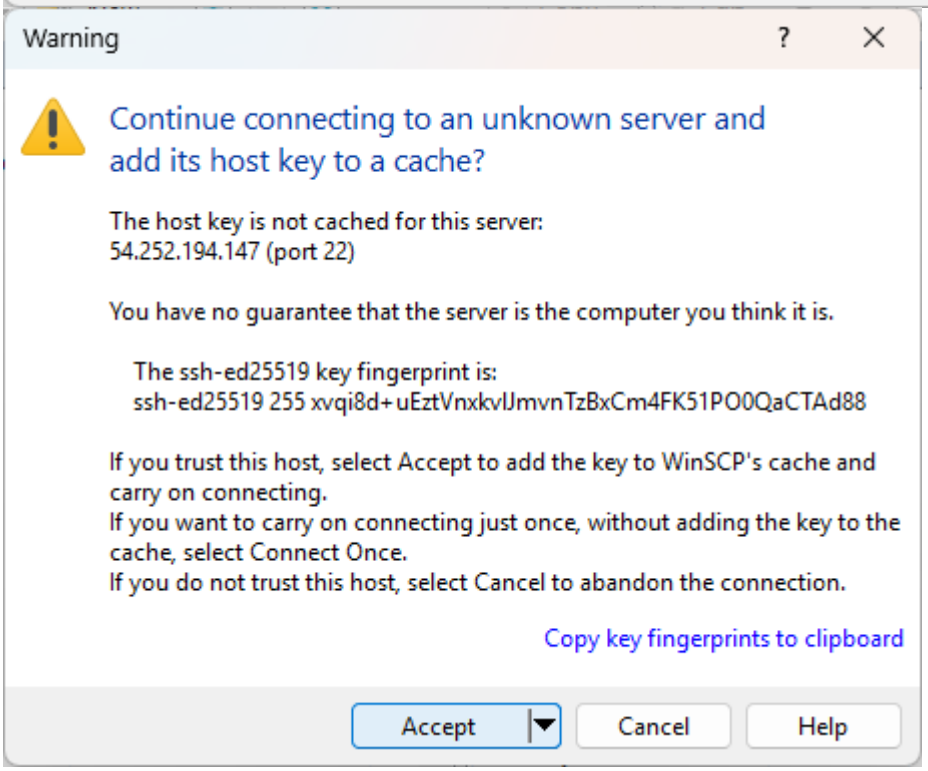
Task 3: Secure Connectivity Testing

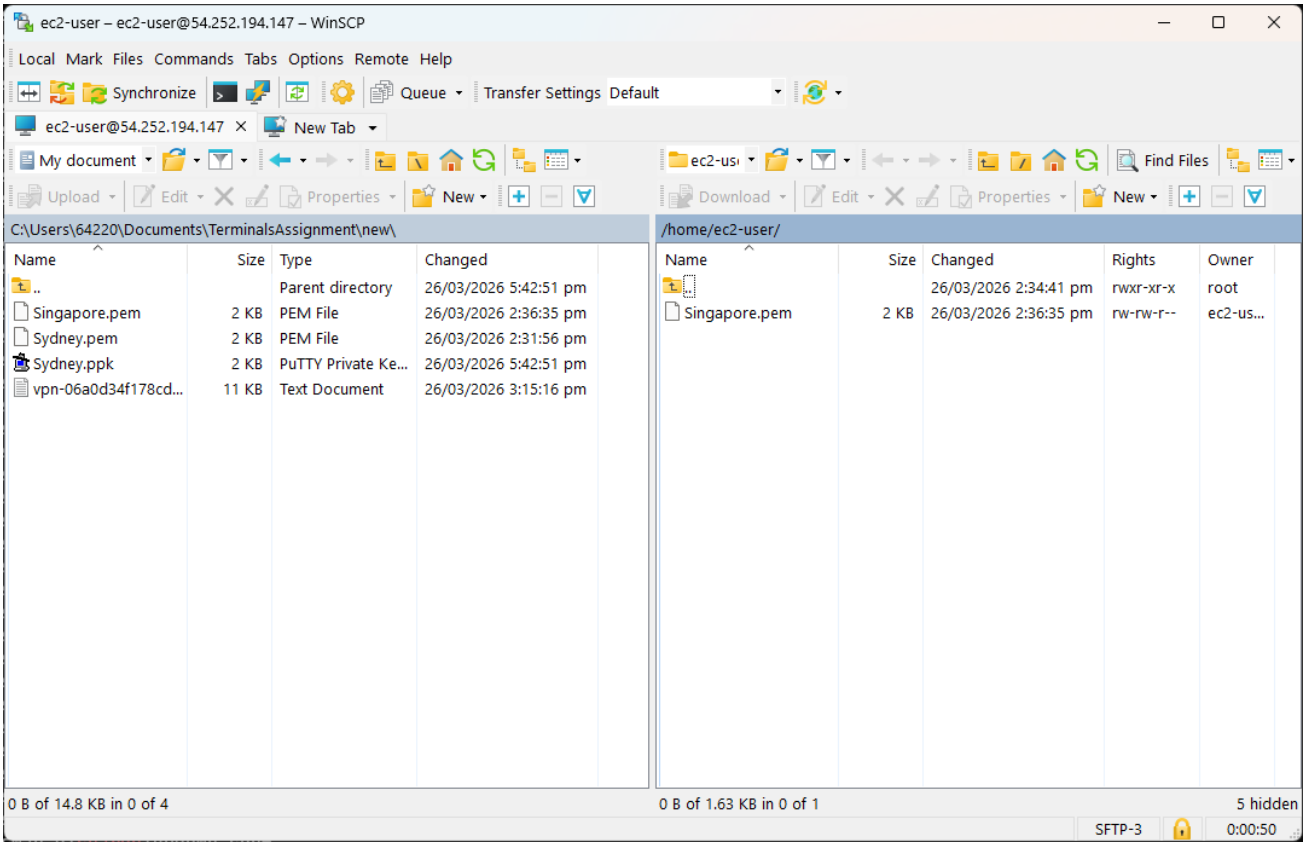
Connected to Sydney EC2 using SSH



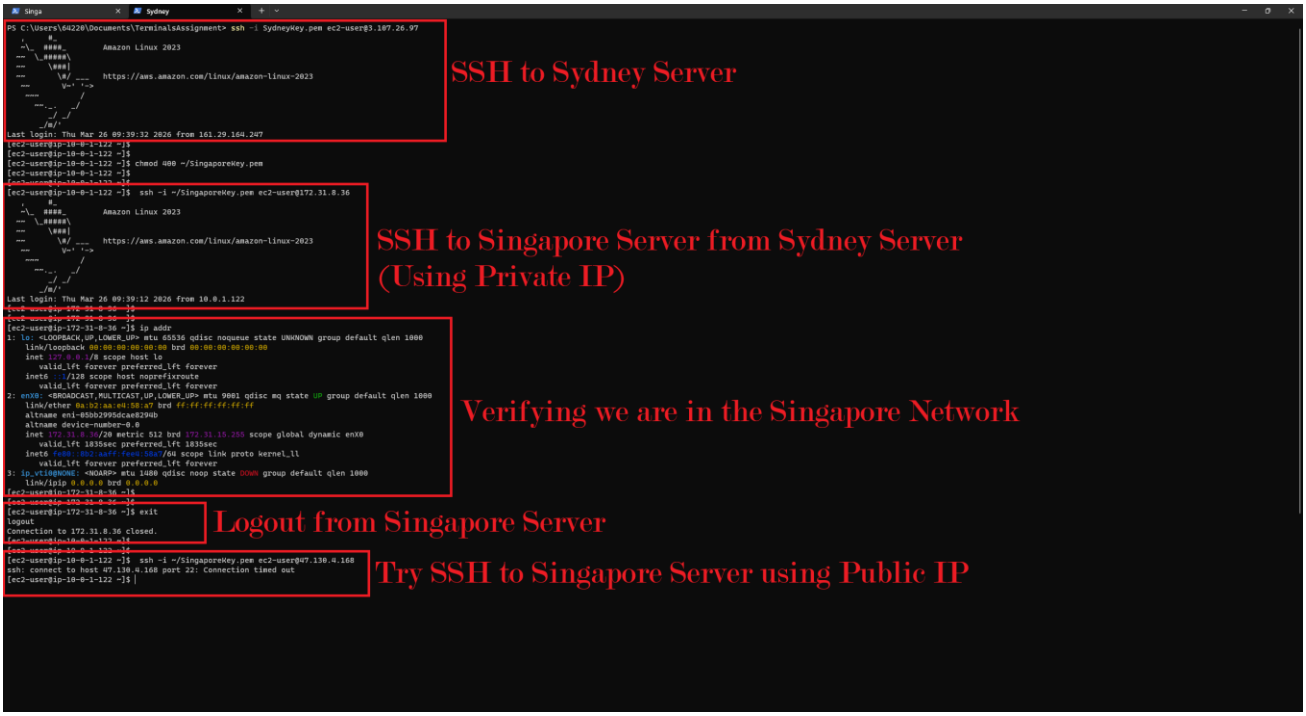
Uploaded Singapore private key securely







Attempted SSH connection using private IP and Public IP



Why Private IP Communication is Used

The main objective of this task was to verify that the communication between the Sydney and Singapore EC2 instances happens securely through the VPN tunnel using private IP addresses.
To test this, I first connected to the Sydney EC2 instance using SSH. From there, I attempted to connect to the Singapore EC2 instance using its private IP address. This connection was successful, which confirmed that the VPN tunnel was working correctly.
The important part of this test was not just making the connection work, but making sure that the connection only works through private IPs. I also tested accessing the Singapore EC2 using its public IP from outside, and this was blocked as expected. This shows that the Security Group rules are correctly configured and not allowing unnecessary public access.
Using private IP communication is very important for security, especially in a banking environment like SecureBank. If public IPs were used for internal communication, the servers would be exposed to the internet. This means they could be targeted by attackers, bots, or automated scanning tools. Even with strong passwords or SSH keys, public exposure always increases the risk.
By using private IPs over the VPN tunnel, the servers are effectively hidden from the outside world. They do not respond to any public traffic, and they are only accessible from within the secure network. This reduces the attack surface significantly.
Another advantage of private communication is better control and monitoring. Since all traffic flows through the VPN, it becomes easier to track and analyse activity. Instead of seeing random external traffic, we only see controlled internal communication between trusted systems.
During this task, I also noticed that the connection using private IP was faster and more stable compared to public access. This is because the traffic stays within the AWS network and does not depend on the public internet.
From a compliance perspective, this approach also supports security standards such as PCI-DSS and GDPR. These frameworks require sensitive data to be protected and not exposed unnecessarily. By using private IP communication, we ensure that data remains within a controlled and encrypted environment.
Overall, this task demonstrated that secure connectivity is not just about connecting two systems, but about controlling how they communicate. Using private IP addresses over a VPN ensures:

- Reduced exposure to external threats
- Secure and encrypted communication
- Better monitoring and control
- Compliance with industry standards

From this, I understood that a properly configured VPN can make two separate environments behave like a single secure internal network, without exposing any resources to the public internet.

Task 4: Risk Mitigation, Best Practices & Report

Risk Identification

In this SecureBank VPN implementation, several potential risks were identified during both the setup and testing phases. Even though the system is functional, these risks can impact security if not properly managed.

Risk 1: Weak or Outdated Encryption Protocols

One of the main risks in any VPN setup is the use of outdated encryption methods such as DES or 3DES. These older algorithms are no longer considered secure because they can be broken using modern computing power. If such encryption methods are used, sensitive banking data could be exposed.

Risk 2: Exposure of Pre-Shared Key (PSK)

The VPN connection relies on a Pre-Shared Key for authentication. If this key is stored insecurely (for example, in plain text without proper file permissions), it could be accessed by unauthorized users. This would allow attackers to potentially establish a connection and access internal systems.

Risk 3: Routing Misconfiguration

Another important risk is incorrect routing configuration. If routes are accidentally pointing to the Internet Gateway instead of the Virtual Private Gateway, traffic may go through the public internet instead of the secure VPN tunnel. This can expose sensitive data and break the intended security design.

Risk 4: Overly Permissive Security Groups

During testing, it is common to temporarily allow open access (e.g., 0.0.0.0/0). If this is not corrected, it can leave EC2 instances exposed to the internet. This increases the risk of brute-force attacks, unauthorized access, and scanning attempts.

Mitigation Recommendations

To reduce the risks identified above, I applied and recommended several mitigation strategies based on best practices.

Mitigation for Encryption Risks

To address weak encryption risks, I used AES (Advanced Encryption Standard) for VPN configuration. AES is currently the industry standard and is widely used in financial systems. It is important to explicitly disable outdated algorithms such as DES and 3DES in the configuration files to prevent downgrade attacks.

Mitigation for PSK Exposure

To secure the Pre-Shared Key, I ensured that the ipsec.secrets file has strict permissions. Only the root user should be able to read this file (e.g., chmod 600). This prevents other users on the system from accessing sensitive credentials.

Mitigation for Routing Issues

To avoid routing mistakes, I recommend using route propagation in AWS. This allows the Virtual Private Gateway to automatically manage routes, reducing the chance of manual errors. Regular review of route tables is also important to confirm that all traffic is flowing through the VPN.

Mitigation for Security Groups

Security Groups should always follow the least-privilege principle. In my setup, I restricted SSH access to only my own IP address instead of allowing access from anywhere. This significantly reduces the attack surface and prevents unauthorized access attempts.

Monitoring and Alerts

Although not fully implemented in this lab, I recommend using AWS CloudWatch to monitor VPN status and network activity. Alerts can be configured to notify if the VPN goes down or if unusual traffic patterns are detected. This helps in early detection of issues.

Compliance Alignment

This VPN setup aligns with several well-known security frameworks and industry standards.

AWS Well-Architected Security Pillar

This implementation follows the “Protect Data in Transit” principle. By using a Site-to-Site VPN, all data between Sydney and Singapore is encrypted before being transmitted. This ensures that even if traffic is intercepted, it cannot be read.

CIS Benchmarks

The setup also follows CIS recommendations by minimizing exposure. Security Groups are configured to allow only necessary traffic, and public access is restricted. This reduces the overall attack surface.

NIST Cybersecurity Framework

This project aligns with the “Protect” function of the NIST framework. Specifically, it focuses on securing data communication and controlling access to systems. The use of encryption and restricted access helps protect sensitive information.

PCI-DSS Compliance

Since SecureBank is a financial organization, PCI-DSS requirements are important. This setup supports PCI-DSS by ensuring:

- Encrypted data transmission
- Restricted access to systems
- Use of secure network design

By using private IP communication over VPN, sensitive data is not exposed to the public internet, which is a key requirement in PCI-DSS.

Summary of AWS Services Utilized

This project used several AWS services to build a secure and reliable VPN architecture.

Amazon VPC

VPC provides the isolated network environment for both Sydney and Singapore. It allows full control over IP ranges, subnets, and routing.

Subnets

Subnets were used to logically separate resources within each VPC. This helps improve organisation and security.

EC2 Instances

EC2 instances act as the main compute resources. The Singapore EC2 instance also acts as the Customer Gateway for VPN configuration.

Virtual Private Gateway (VGW)

VGW is attached to the Sydney VPC and is responsible for handling VPN connections on the AWS side.

Customer Gateway (CGW)

CGW represents the external VPN endpoint. In this setup, it is configured using the Singapore EC2 instance.

Site-to-Site VPN

This service connects the two VPCs securely. It ensures all communication between regions is encrypted.

Security Groups

Security Groups act as firewalls controlling inbound and outbound traffic. They enforce least-privilege access.

IAM (Recommended)

IAM can be used to control access to AWS resources and ensure that only authorized users can make changes.

CloudWatch (Recommended)

CloudWatch can be used to monitor VPN health, detect failures, and trigger alerts.

Architectural Diagram Refinement

The architectural diagram created in Task 1 represents the complete SecureBank VPN setup. It clearly shows both VPCs, subnets, EC2 instances, and the VPN connection between Sydney and Singapore.
In this design:

- Each VPC is isolated with its own CIDR range
- EC2 instances are placed inside secure subnets
- The VPN tunnel connects the two environments securely
- The Virtual Private Gateway and Customer Gateway act as controlled entry points

From a security perspective, the diagram highlights that:

- There is no direct public communication between systems
- All traffic is routed through the VPN tunnel
- Security Groups control access at the instance level

This visual representation helps to understand how all components work together to provide a secure and private network.
From this overall implementation, I understood that security in cloud environments is not only about configuring services, but also about continuously monitoring and improving the setup. Even small configuration mistakes can create risks, so following best practices is very important.