VPC- Challenge1

Use Case: Setting up Transit Gateway and VPC Endpoints for a Multi-VPC Architecture

Scenario:

A large organization is migrating its on-premises infrastructure to the AWS cloud. The organization's architecture involves multiple VPCs for different departments and applications, each requiring secure communication with centralized services and external resources.The IT team needs to design and implement a scalable and efficient network architecture to accommodate the organization's growth and ensure robust connectivity between VPCs and external services.

Objectives:

- Design and deploy a scalable network architecture using AWS Transit Gateway to simplify network connectivity between multiple VPCs.
- Configure VPC endpoints to securely access AWS services without internet gateways or NAT gateways, ensuring data privacy and minimizing exposure to external threats.

VPC- Challenge1

Objective

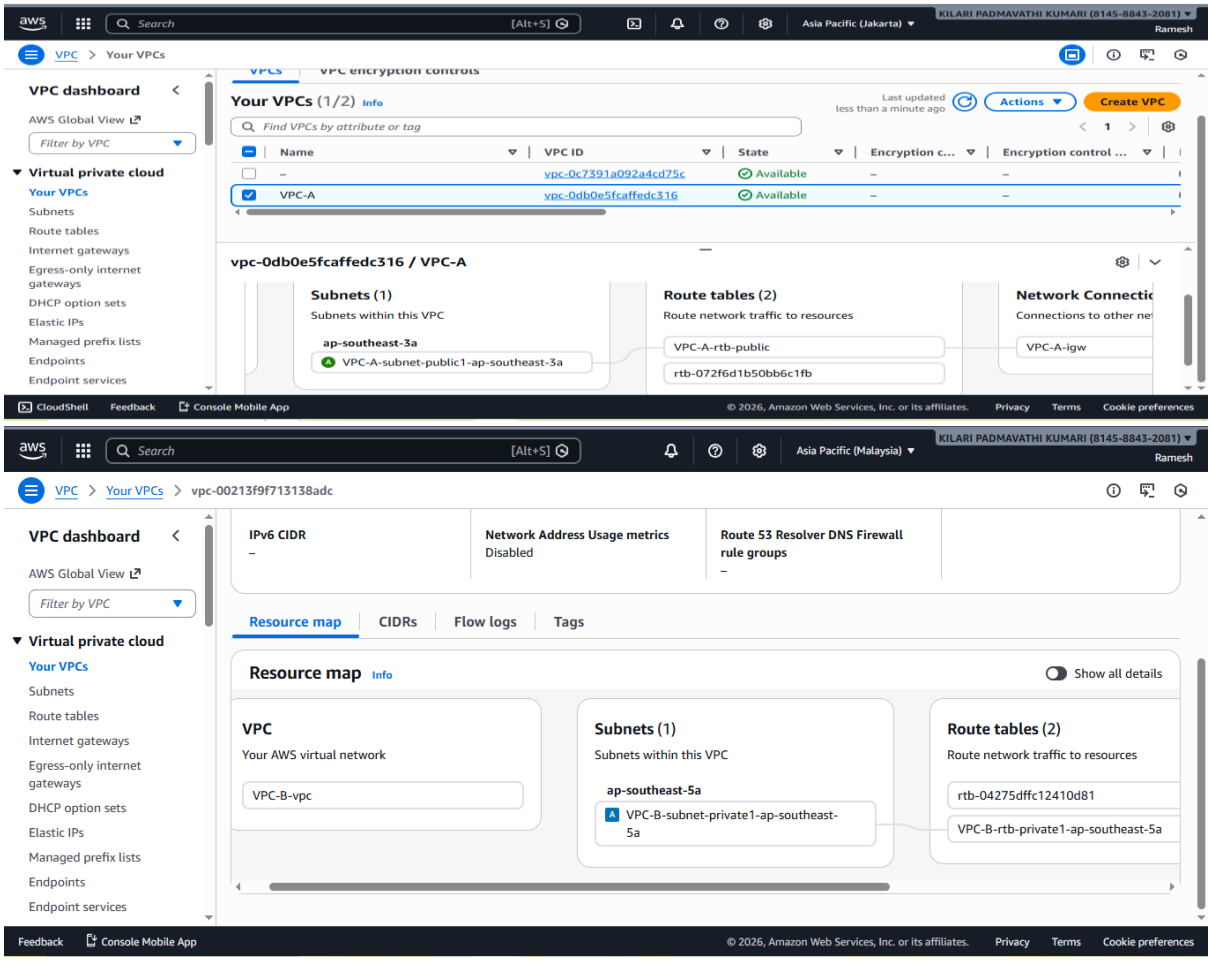
- To design a **centralized, scalable, and secure network architecture** that simplifies connectivity between multiple VPCs using **AWS Transit Gateway**, reducing operational complexity and enabling easy future expansion.

Design Components:

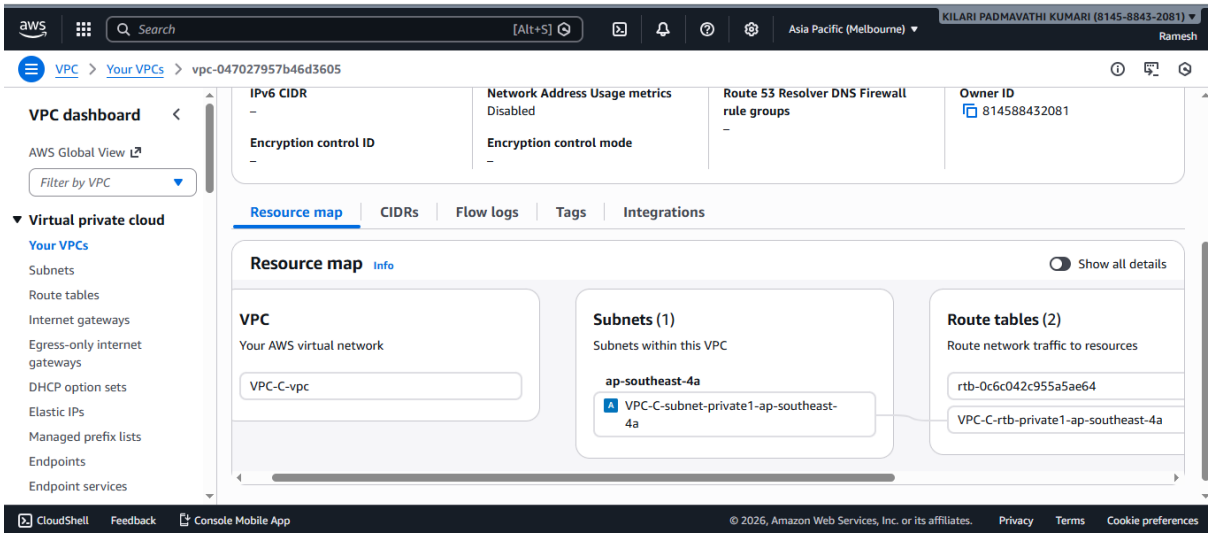
1. Transit Gateway
2. VPC Attachments
3. Transit Gateway Route Tables
4. VPC Route Tables

Deployment Steps

Firstly, create VPC , subnets and Route tables in 3 regions and CIDR should not overlap(Regions:-Jakarta, Malaysia, Melbourne)

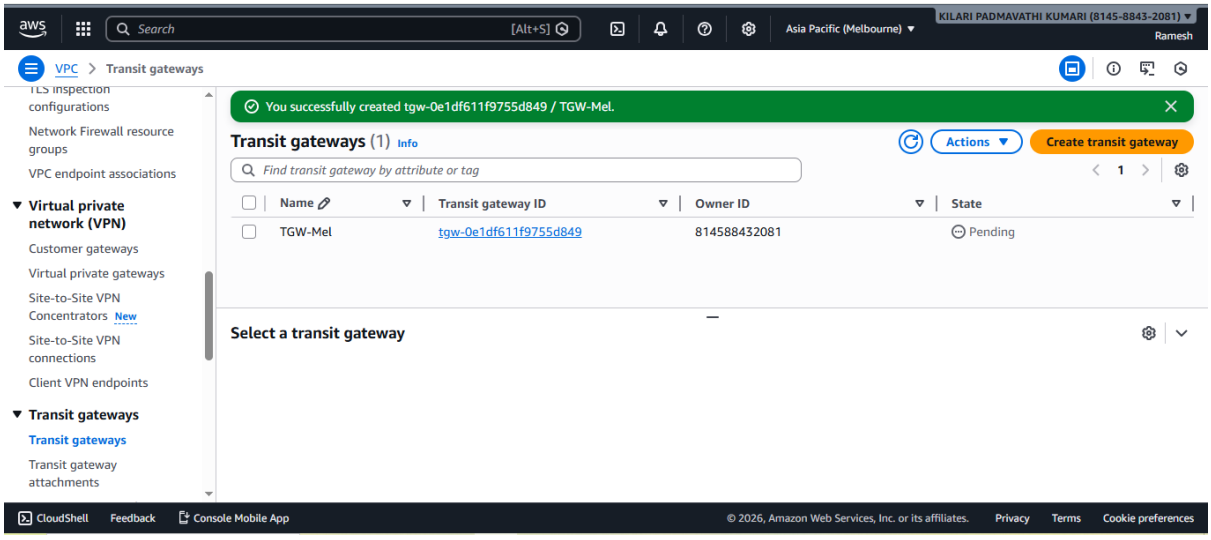


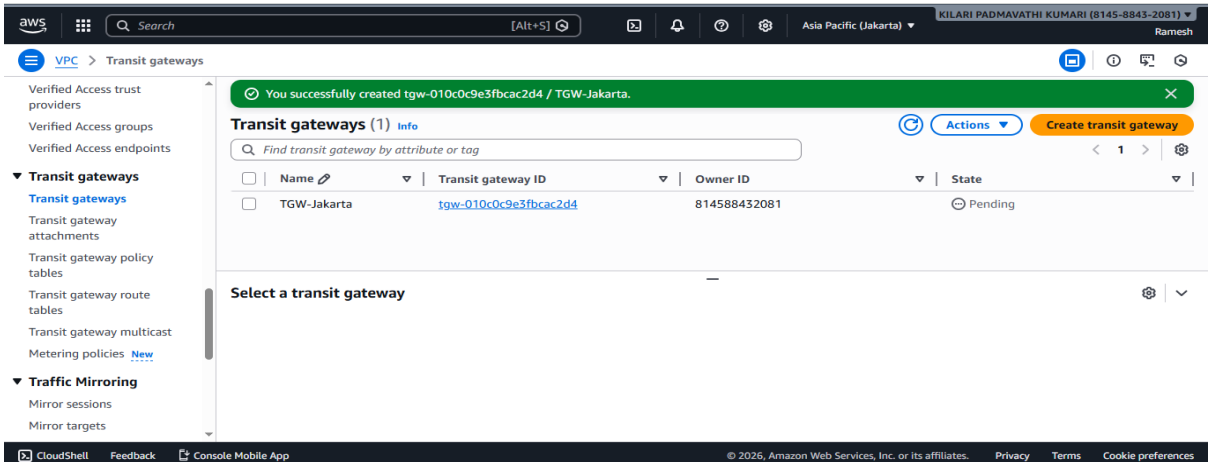
VPC- Challenge1



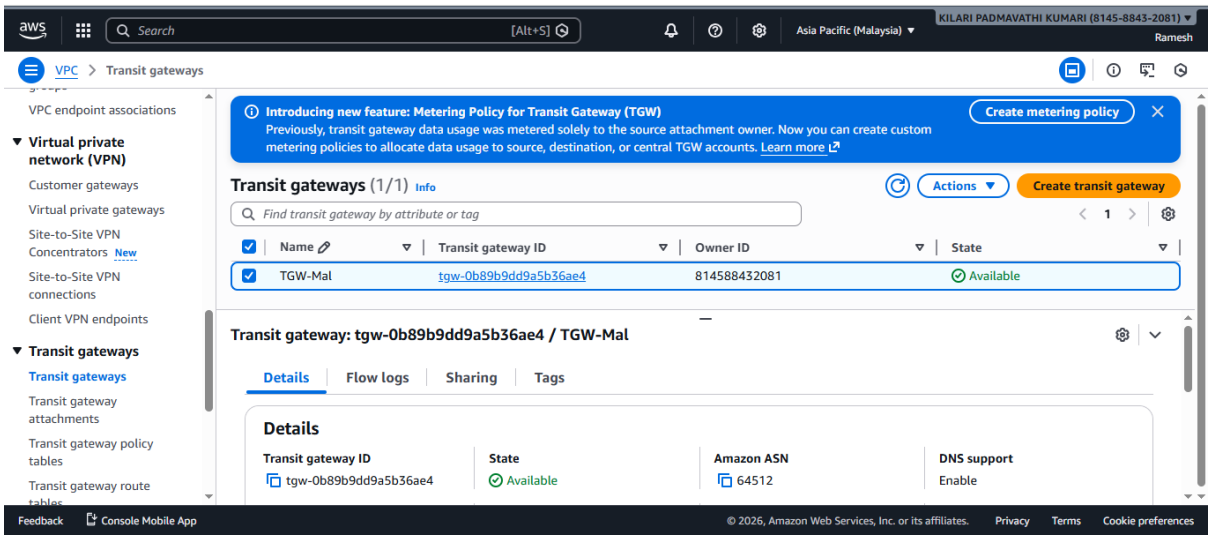
Step 1: Create a Transit Gateway

- Enable DNS support if required
- Disable auto-accept (recommended for security)



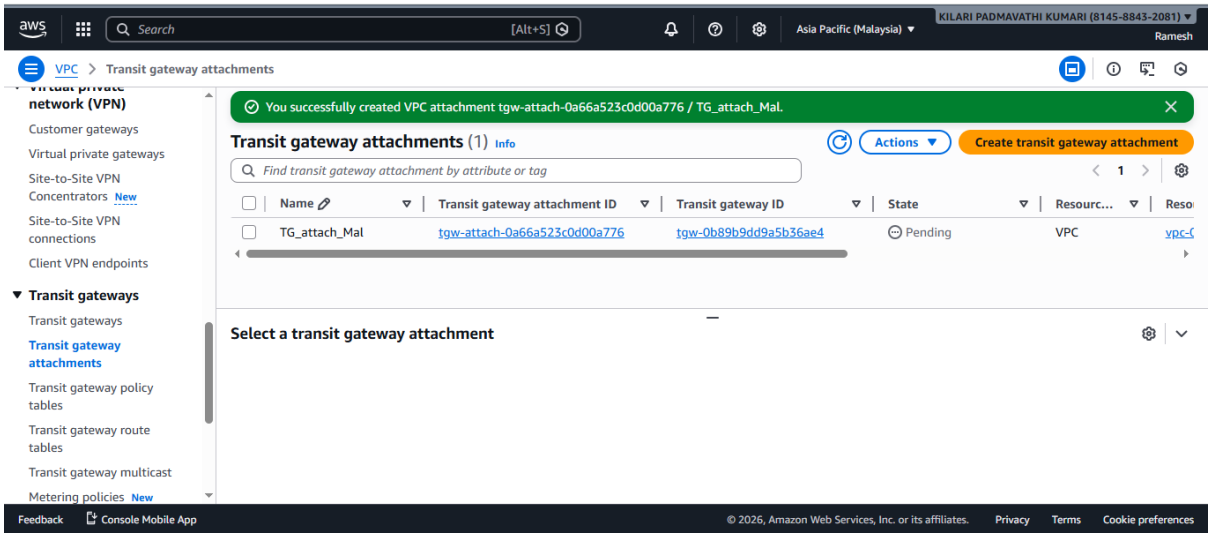


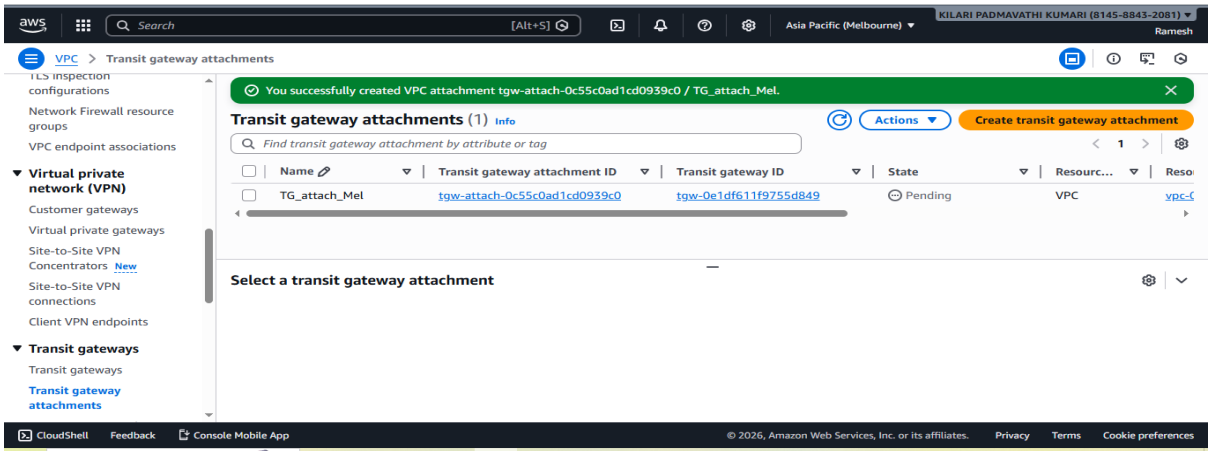
VPC- Challenge1



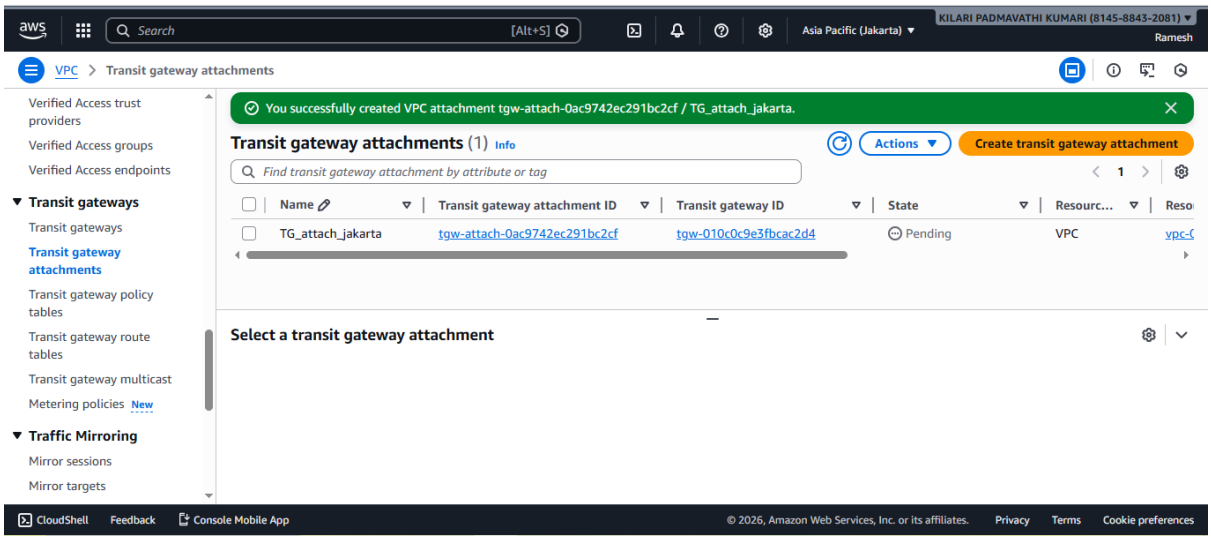
Step 2: Create VPC Attachments

- Attach each VPC to the TGW
- Select private subnets only
- One attachment per VPC

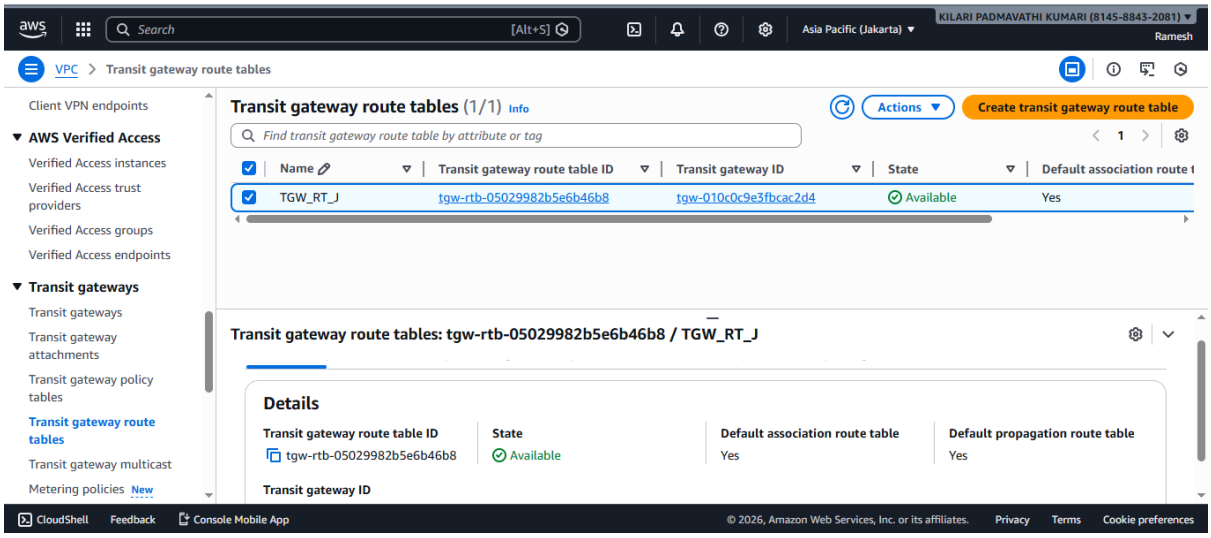


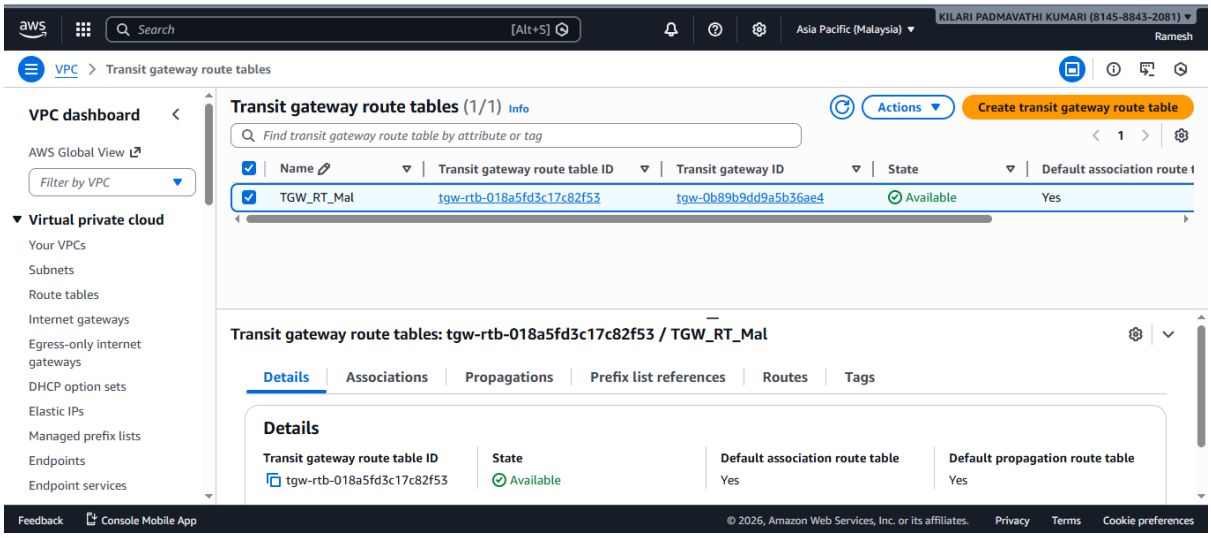


VPC- Challenge1

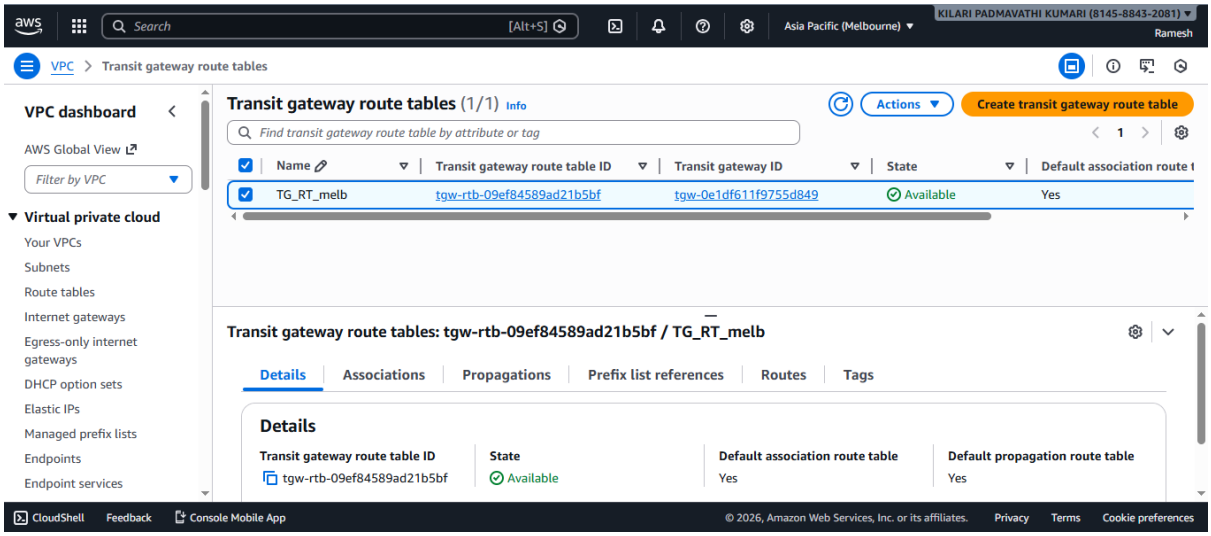


Step 3: Configure Transit Gateway Route Tables





VPC- Challenge1

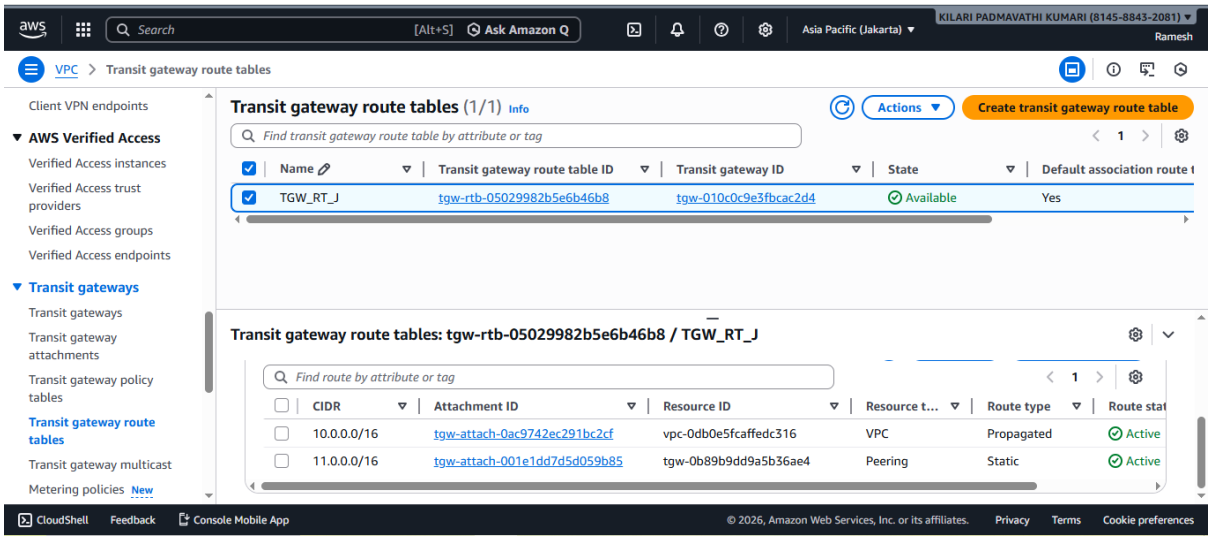


Step 4: Update VPC Route Tables

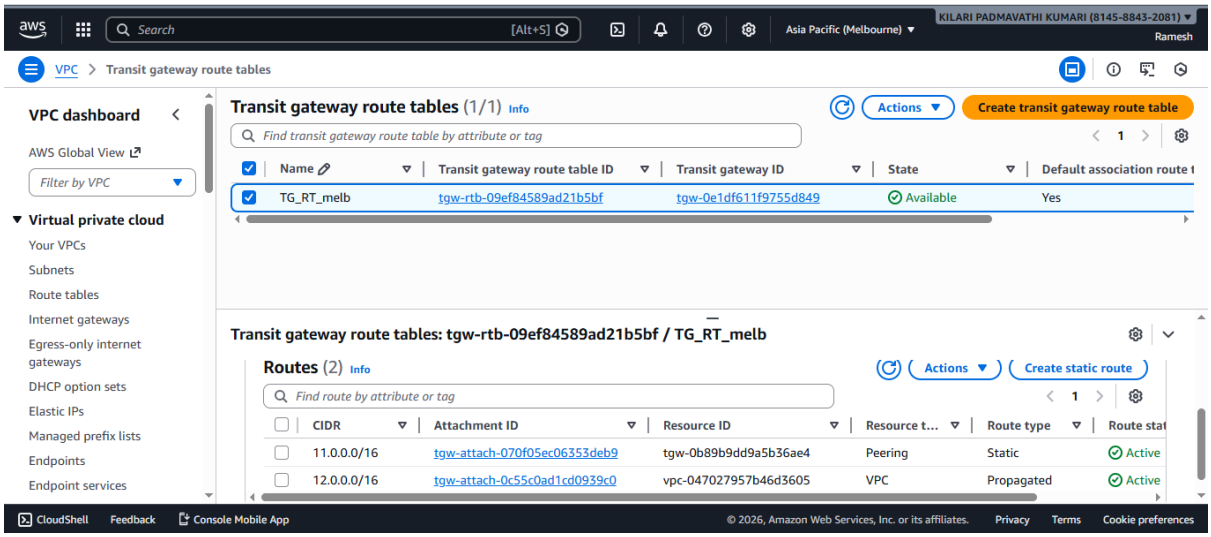
Add routes in private subnet route tables:

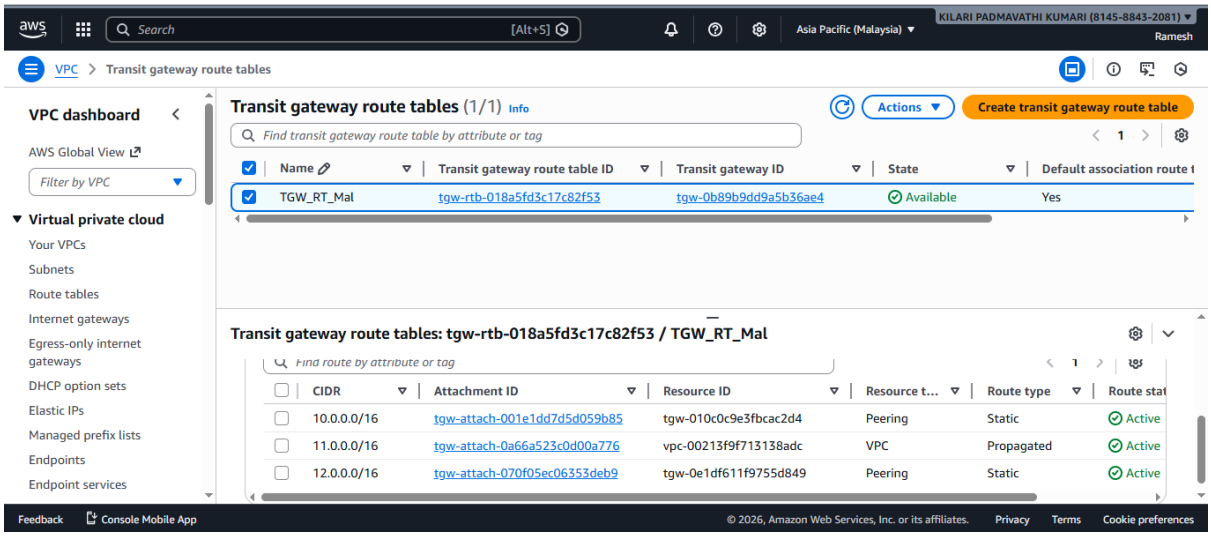
Destination: 10.0.1.0/16 (Remote VPC)

Target: Transit Gateway



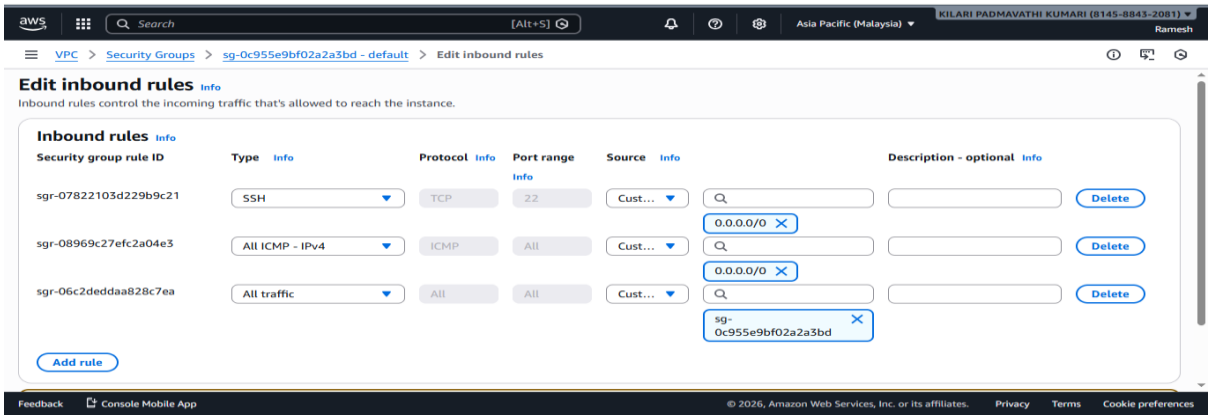
VPC- Challenge1



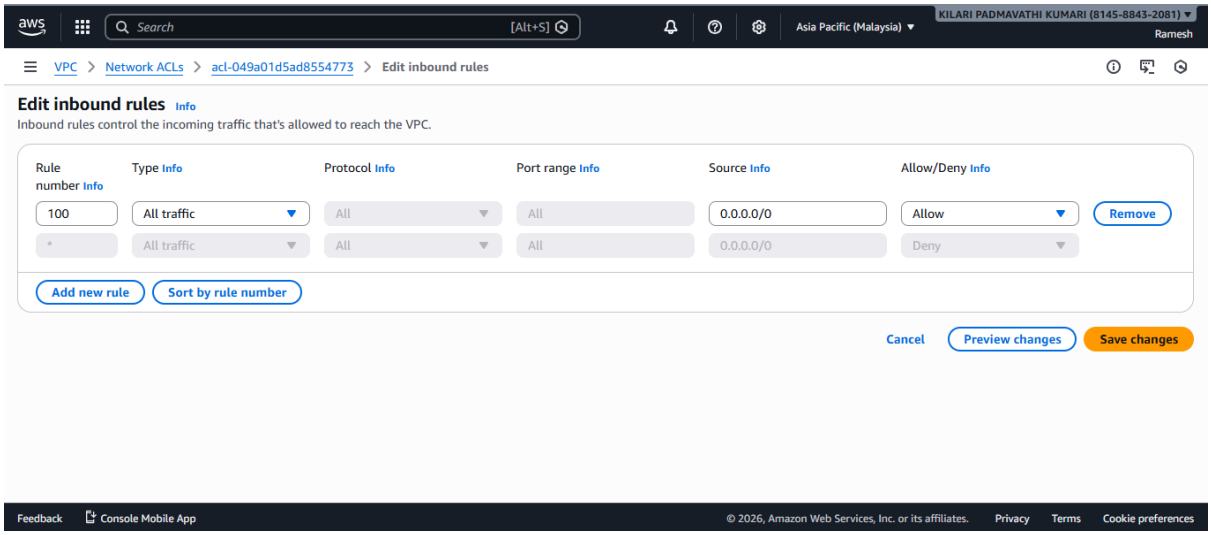


Step 5: Security Configuration

- Use **Security Groups** and **NACLs**(follow these rules for all 3 regions)
- Restrict TGW routes to required CIDRs

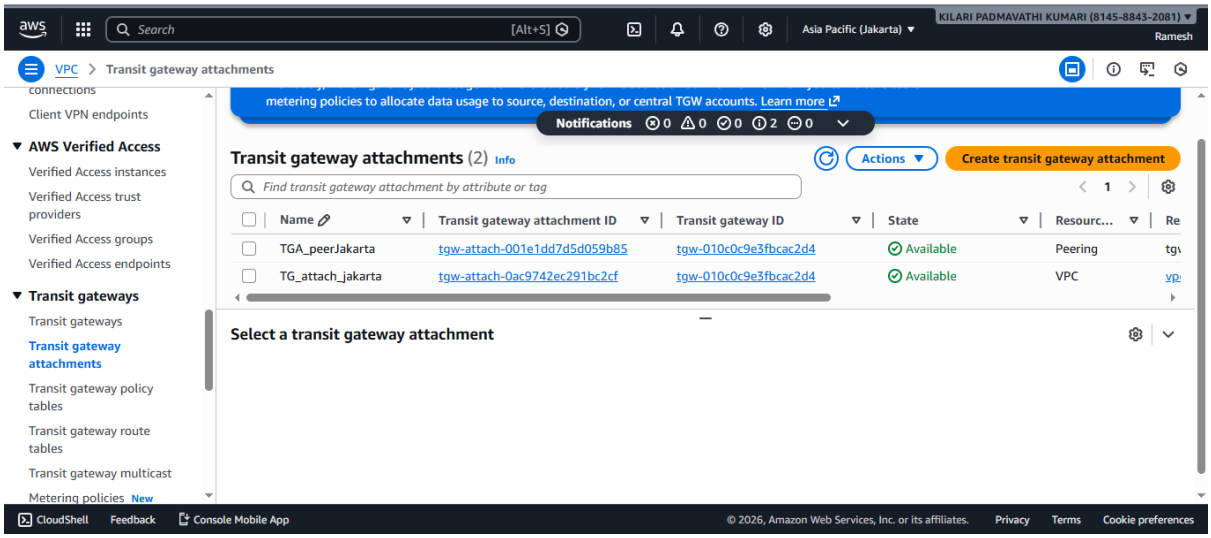


VPC- Challenge1

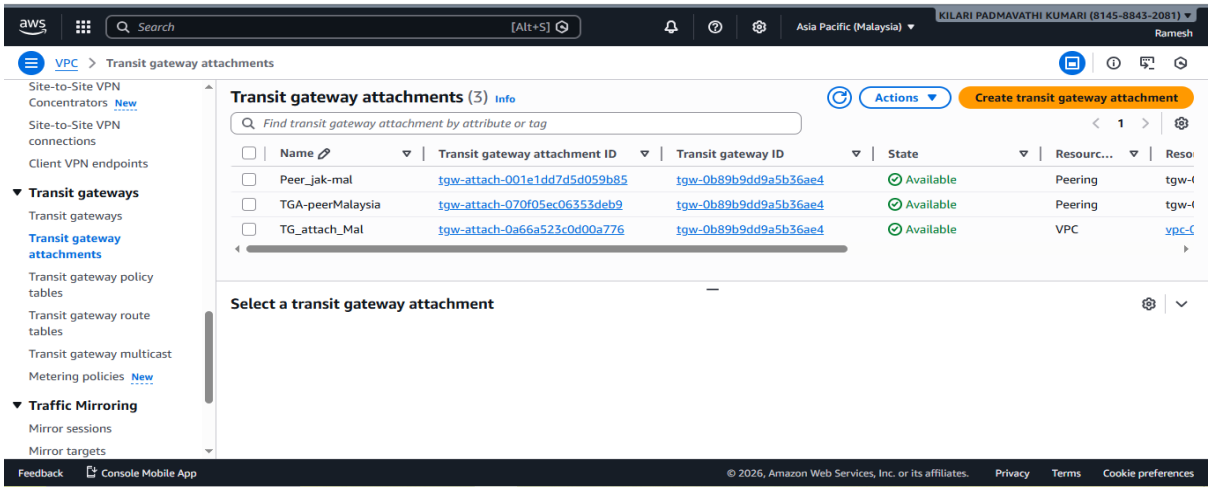


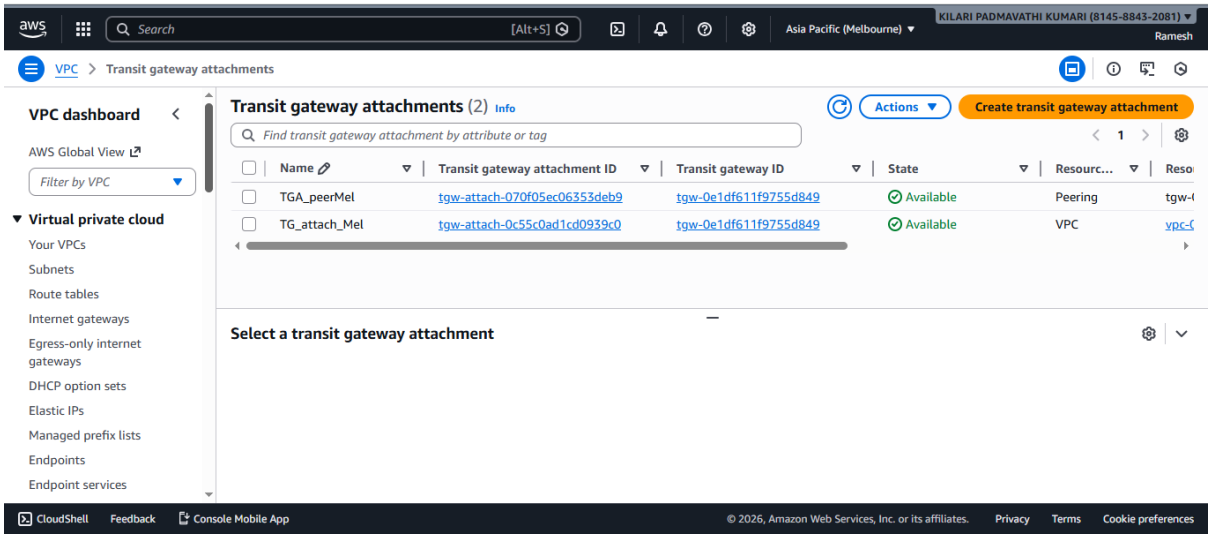
Step 6: High Availability

- TGW is **regionally highly available**
- Use multiple subnets (AZs) per attachment
- For multi-region:
    - Use **TGW Peering**

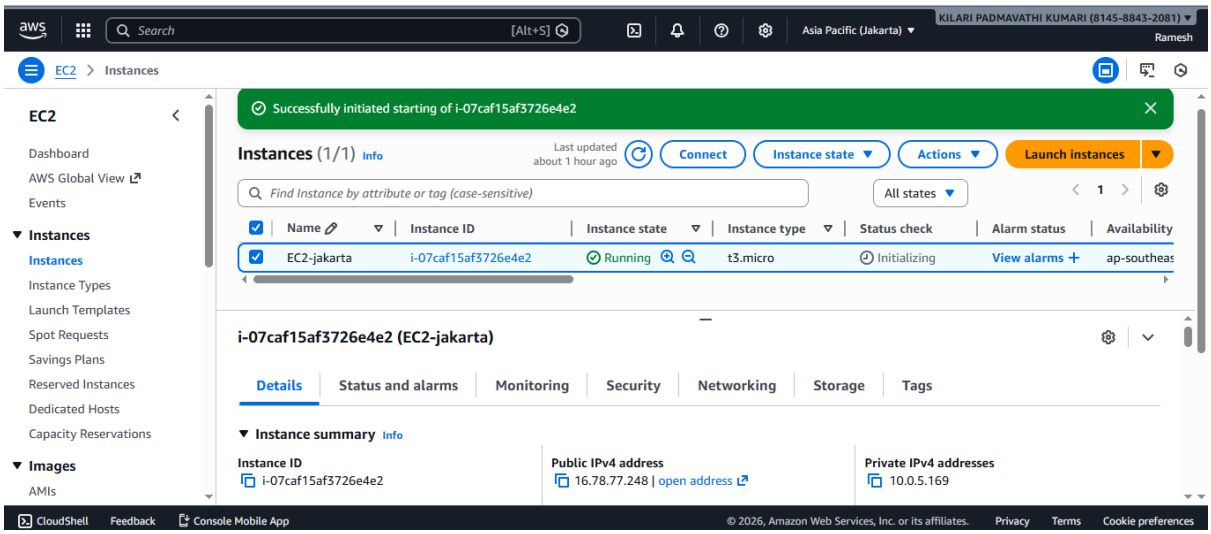


VPC- Challenge1

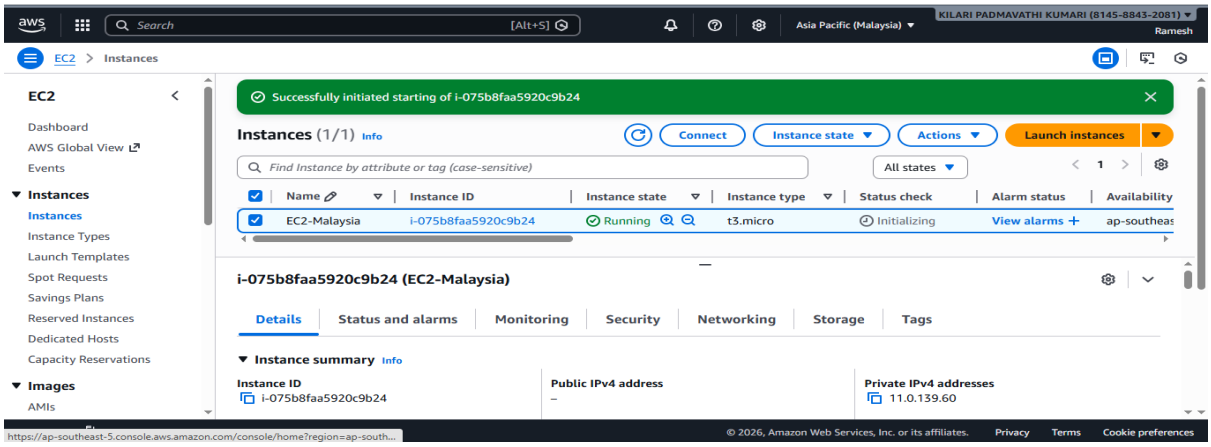


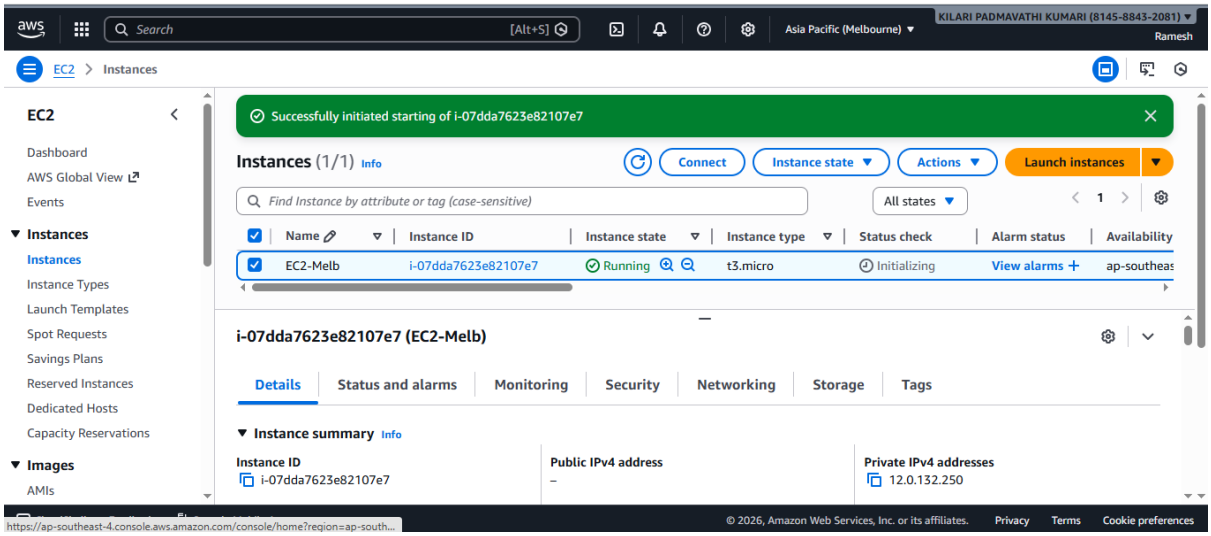


Create EC2 instance for checking connectivity.



VPC- Challenge1





VPC- Challenge1

Connection: To Verify

Ec2 instance for Jakarta region

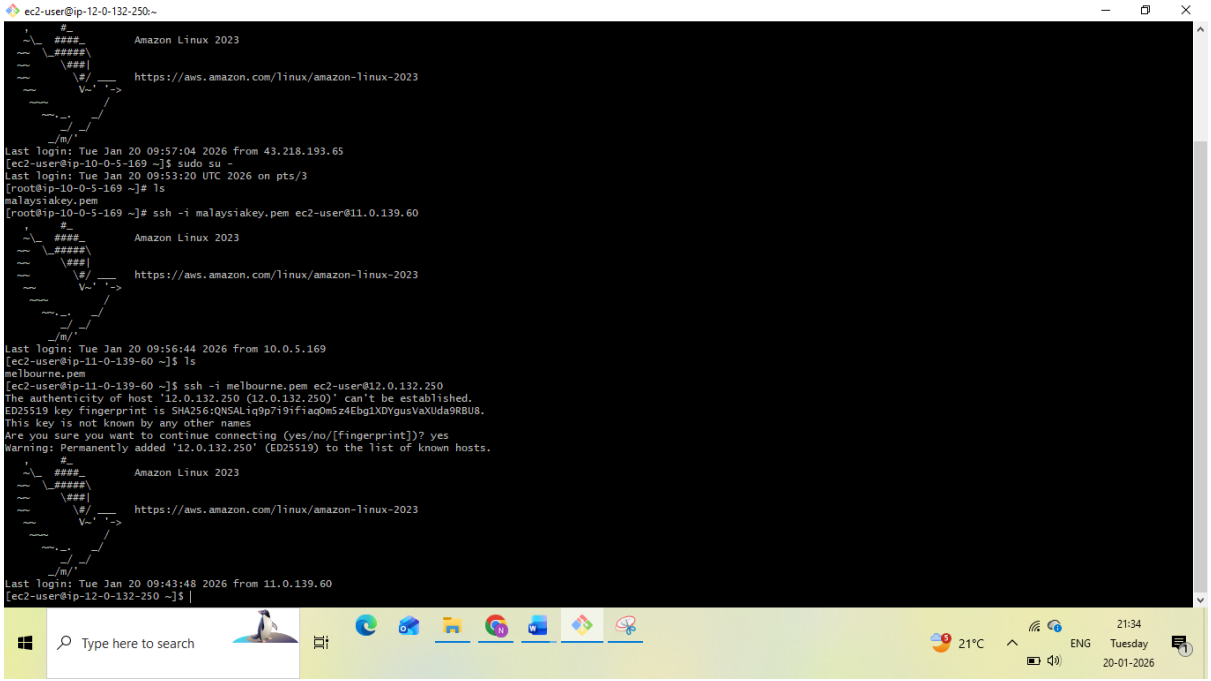
ec2-user@ip-10-0-5-169:~

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "Jakarta_keypair.pem" ec2-user@ec2-16-78-77-248.ap-southeast-3.compute.
amazonaws.com
The authenticity of host 'ec2-16-78-77-248.ap-southeast-3.compute.amazonaws.com
(16.78.77.248)' can't be established.
ED25519 key fingerprint is SHA256:P5EhVyr1bNcgE/VUfRNHXOorqB4sSyt5mRPXll9yr+Y.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-16-78-77-248.ap-southeast-3.compute.amazonaws.co
m' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Jan 20 09:57:04 2026 from 43.218.193.65
[ec2-user@ip-10-0-5-169 ~]$
```

```
[ec2-user@ip-10-0-5-169 ~]$ sudo su -
Last login: Tue Jan 20 09:53:20 UTC 2026 on pts/3
[root@ip-10-0-5-169 ~]# ls
malaysiakey.pem
[root@ip-10-0-5-169 ~]# ssh -i malaysiakey.pem ec2-user@11.0.139.60
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Jan 20 09:56:44 2026 from 10.0.5.169
[ec2-user@ip-11-0-139-60 ~]$
```

VPC- Challenge1

```
[ec2-user@ip-11-0-139-60 ~]$ ls
melbourne.pem
[ec2-user@ip-11-0-139-60 ~]$ ssh -i melbourne.pem ec2-user@12.0.132.250
The authenticity of host '12.0.132.250 (12.0.132.250)' can't be established.
ED25519 key fingerprint is SHA256:QNSALiq9p7i9ifiaqOm5z4Ebg1XDYgusVaXUda9RBU8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '12.0.132.250' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Jan 20 09:43:48 2026 from 11.0.139.60
[ec2-user@ip-12-0-132-250 ~]$
```



Conclusion

- By using **AWS Transit Gateway**, we can design a **scalable, secure, and centralized network architecture** that simplifies VPC connectivity, reduces operational overhead, and supports future growth without re-architecting the network.

VPC- Challenge1

- Configure VPC endpoints to securely access AWS services without internet gateways or NAT gateways, ensuring data privacy and minimizing exposure to external threats.

  Objective:

  - Implemented **VPC Gateway and Interface Endpoints (PrivateLink)** to enable secure, private access to AWS services without Internet or NAT Gateways.
  - Ensured **data privacy and reduced attack surface** by keeping all service traffic on the AWS private backbone using Private DNS and endpoint policies.
  - Optimized **security and cost** by eliminating public IP dependencies and enforcing least-privilege access controls.