TRANSIT GATEWAY

AWS Transit Gateway is a pivotal networking service provided by Amazon Web Services that simplifies the management of network connectivity at scale. It serves as a central hub for connecting Virtual Private Clouds (VPCs), on-premises networks, and other resources within an AWS environment. By consolidating multiple networking connections through a single gateway, AWS Transit Gateway facilitates efficient network management, improves performance, and enhances security.

Key features of AWS Transit Gateway include:

- Centralized Network Management: Simplify network design by consolidating multiple VPCs and VPN connections into a single gateway.
- Scalability: Efficiently handle large-scale networks with high throughput and low latency.
- Segmentation and Isolation: Enable network segmentation and isolation to improve security and compliance.
- Flexible Routing: Utilize advanced routing options and policies to control traffic flow between network resources.
- Cost Efficiency: Reduce operational costs by minimizing the need for complex peering arrangements and extensive network configurations.

Components Explained

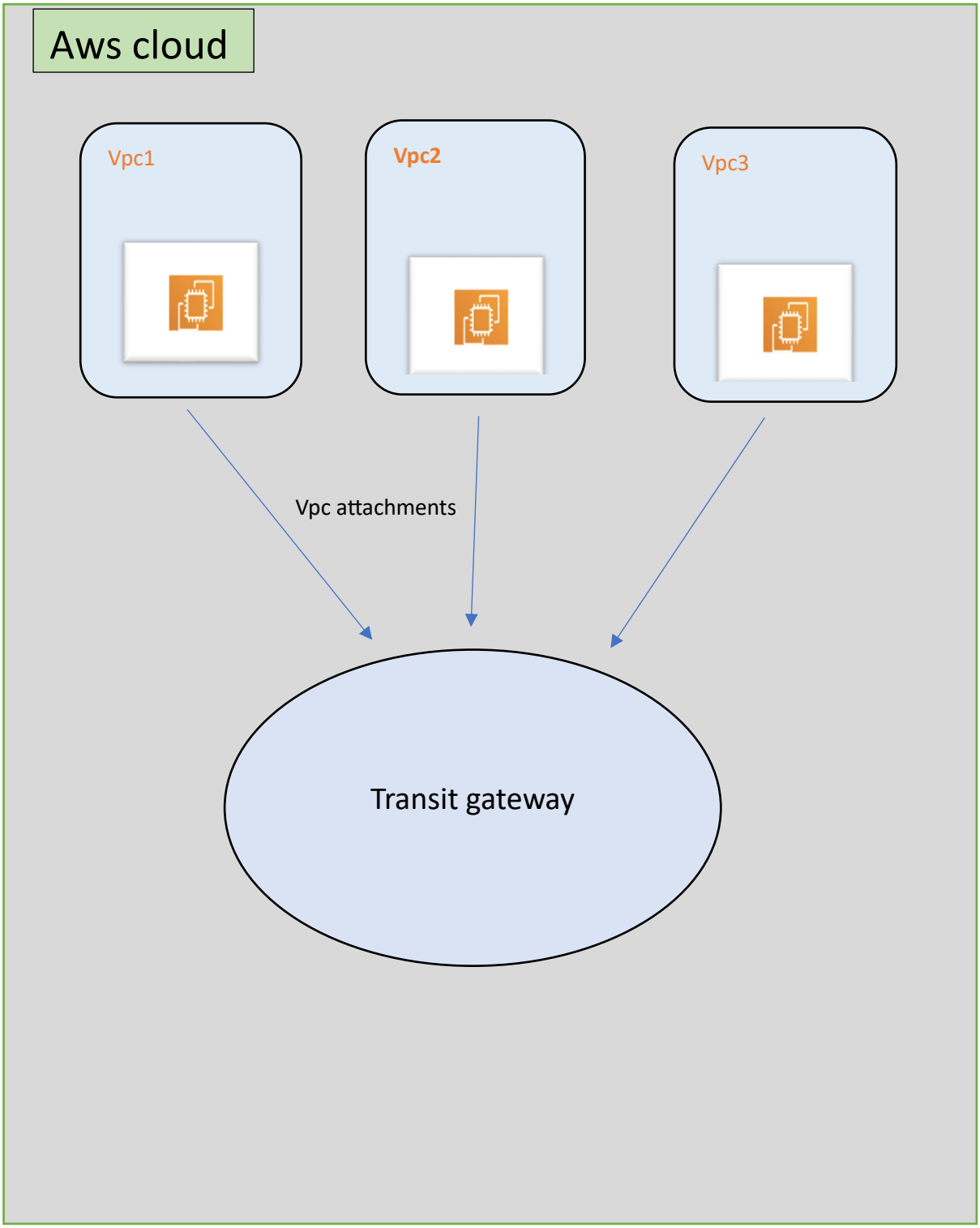
1. **AWS Transit Gateway**: Acts as a centralized router for network traffic between connected VPCs.
2. **VPC 1, VPC 2, VPC 3**: Individual VPCs with their own route tables and subnets. Each VPC connects to the Transit Gateway.
3. **Route Tables**:
    - Each VPC has a route table with routes to the Transit Gateway (TG). This enables traffic to be routed between VPCs via the TG.
    - The Transit Gateway also has its own route table to handle traffic between VPCs and ensure correct routing.

Key Points

- **Connectivity**: VPCs can communicate with each other through the Transit Gateway without needing VPC peering connections.
- **Routing**: Ensure each VPC's route table directs traffic intended for other VPCs to the Transit Gateway.

- **Scalability**: Adding more VPCs is straightforward; just connect them to the existing Transit Gateway.

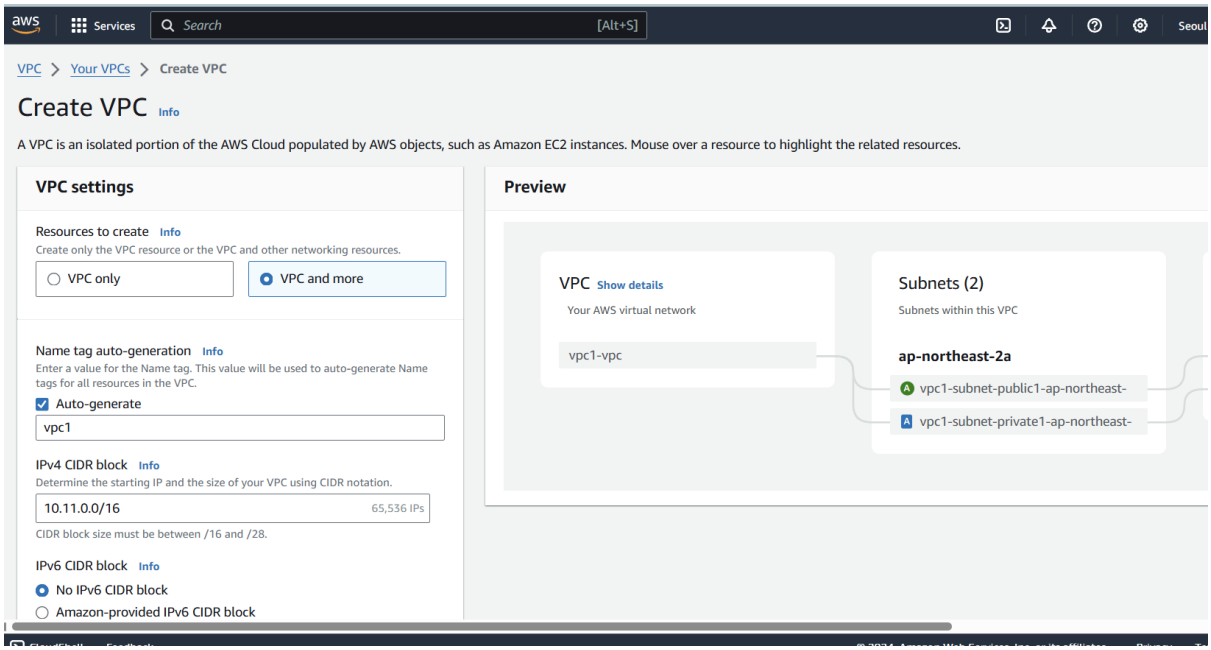
Configuration diagram:



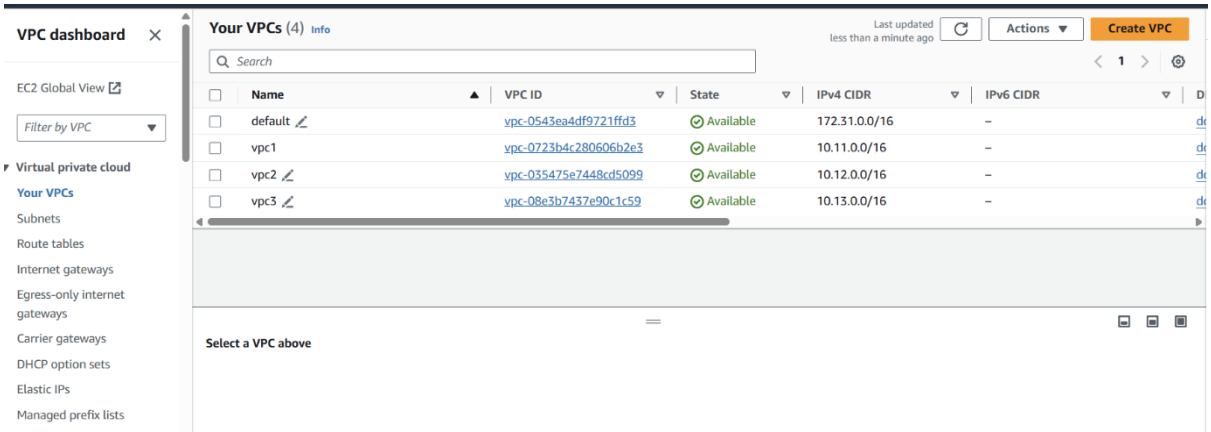
Connecting three VPCs with transit gateway

Open AWS console and select VPC and click on Create.

- I was created VPC in **Asia Pacific (Seoul) Region**.

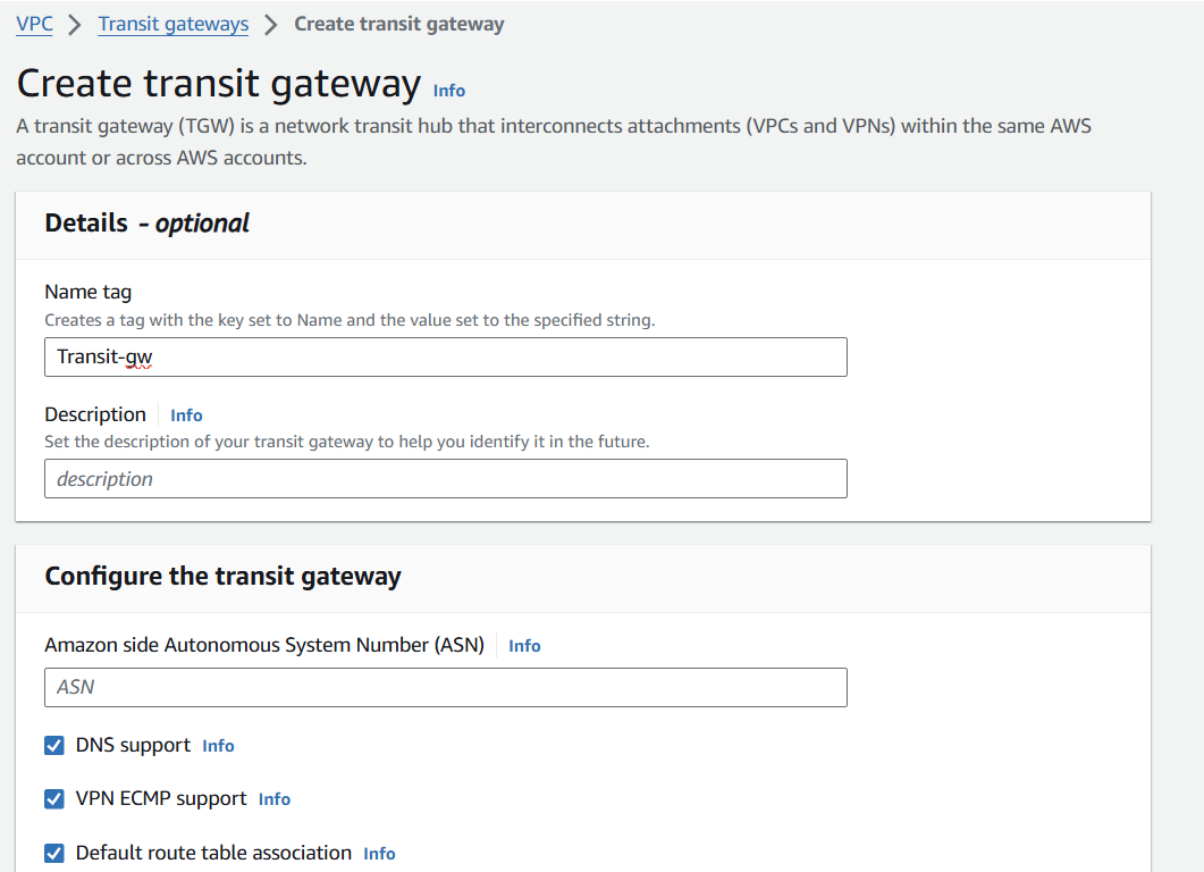


I Was created VPC1 in Zone(ap-northeast-2a) with 2 subnets (1 public, 1 private), 2 route tables (1 public, 1 private) and 1 internet gateway with IP address-10.11.0.0/16. Similarly, I was created another 2 VPCs (VPC2 & VPC3). VPC2 in ap-northeast-2b Zone with IP address as 10.12.0.0/16 and VPC3 in ap-northeast-2c Zone with IP address as 10.13.0.0/16 respectively.

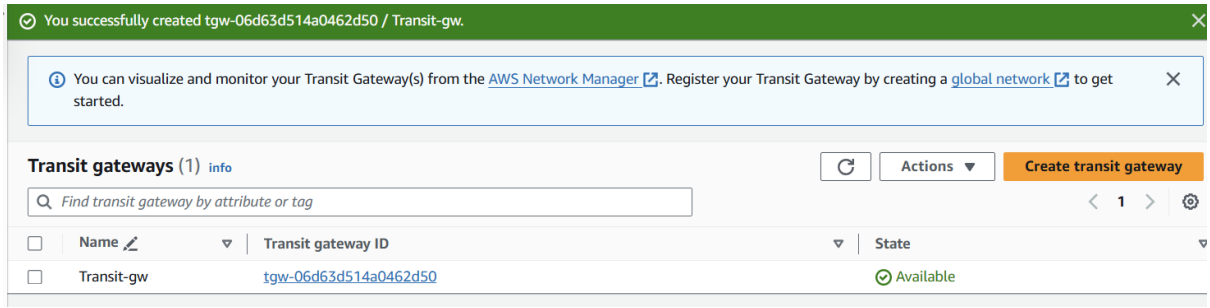


3 VPCs were created in separate zones within the same region.

2.Creating Transit gateway:

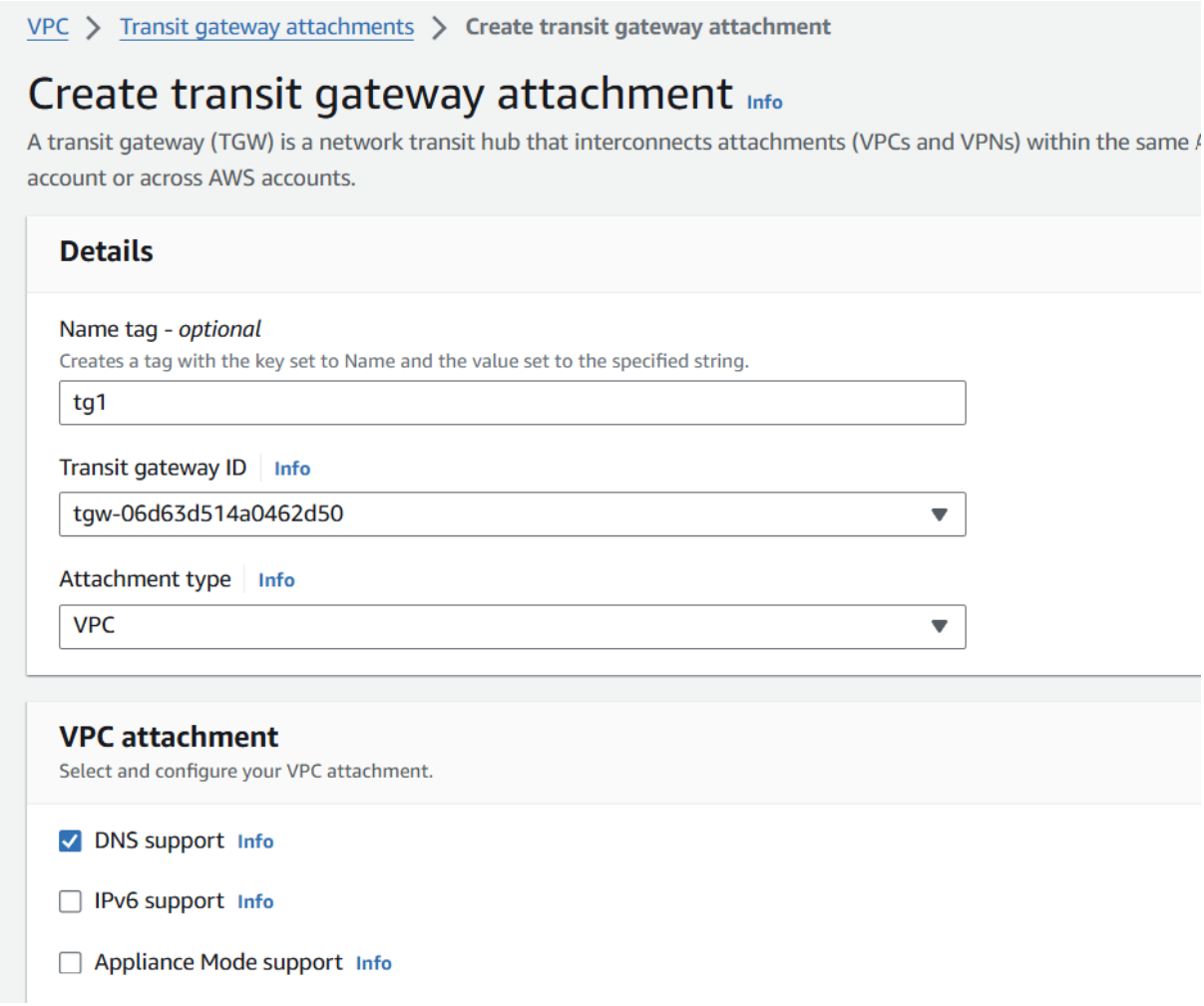


Open Transit gateway and created with name as Transit-GW.

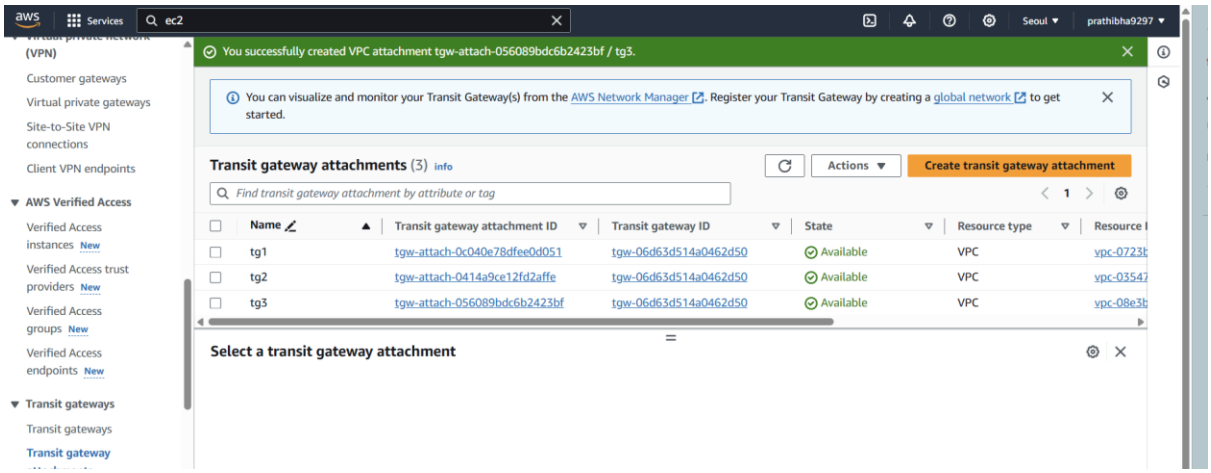


Once transit gateway created go to transit gateway attachments.

3.Transit gateway attachments: Create transit gateway attachments as shown in below.

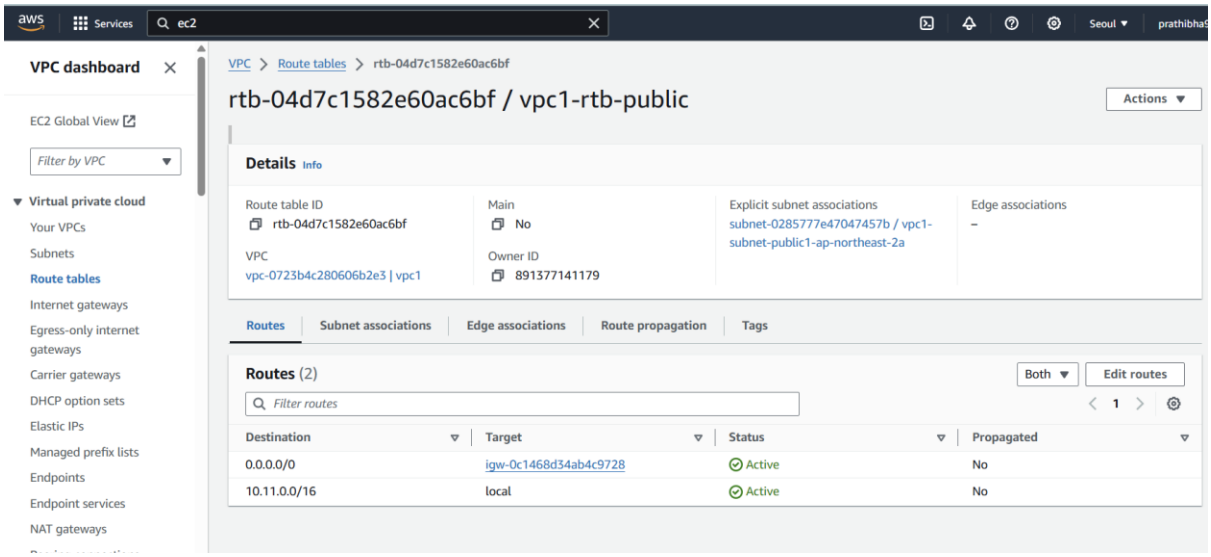


Create Transit gateway attachment for VPC1: Go to Transit gateway attachments and click on create->give name->select Transit gateway ID->VPC ID and then create. Likewise for remaining 2 VPCs need to create transit gateway attachments (tg2 & tg3) as shown in below.

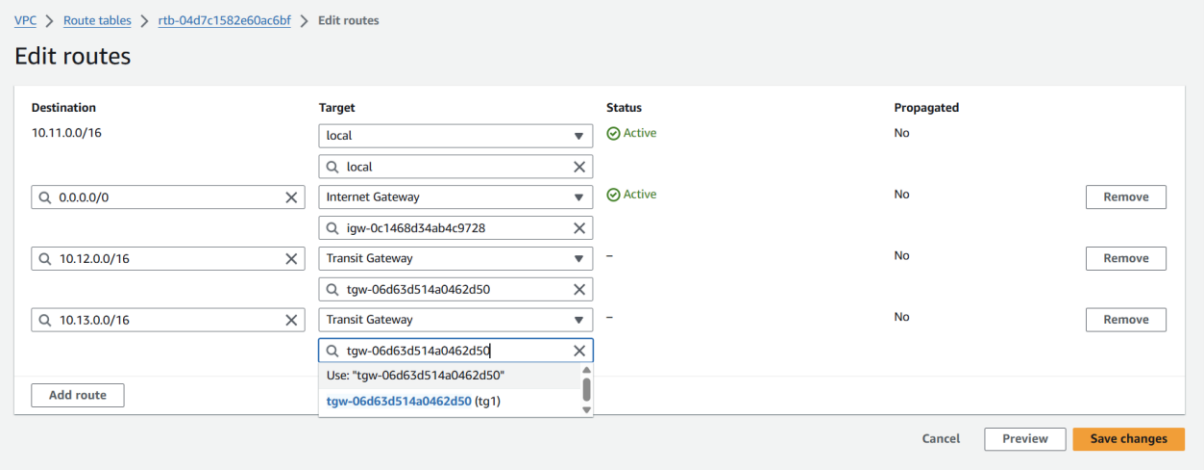


Once transit gateway attachments created go to the route tables.

4.Route tables:



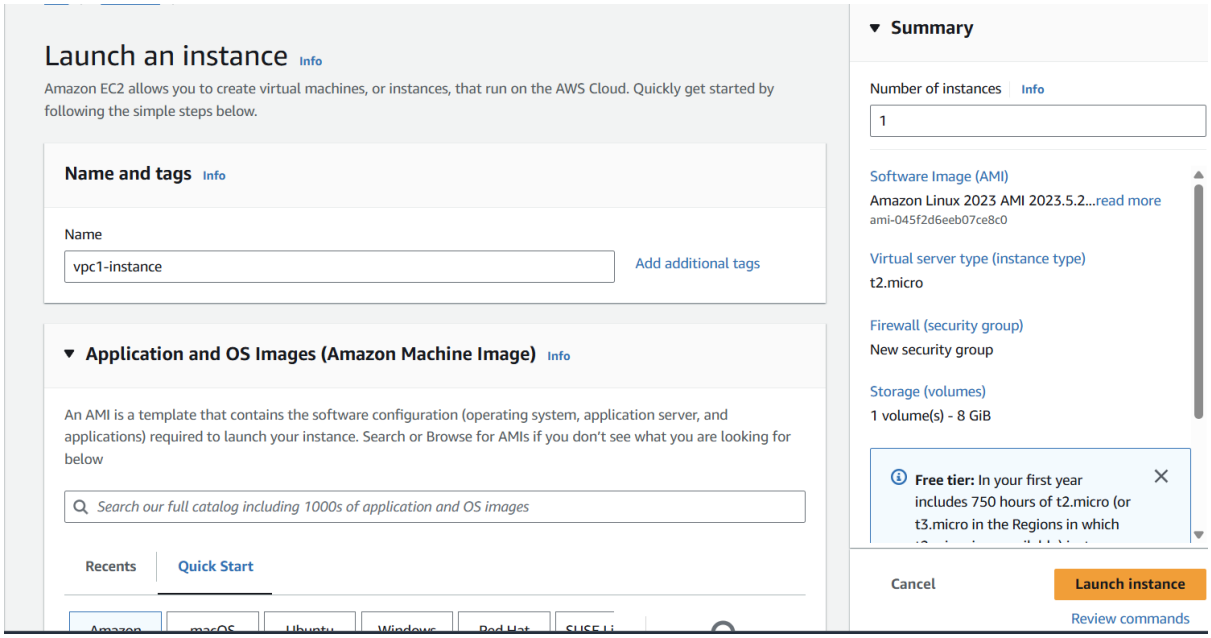
I was taken VPC1 public route table and then go to edit routes. In that we need to add remaining 2 VPCs IP address and transit gateway attachments as shown in below diagram.



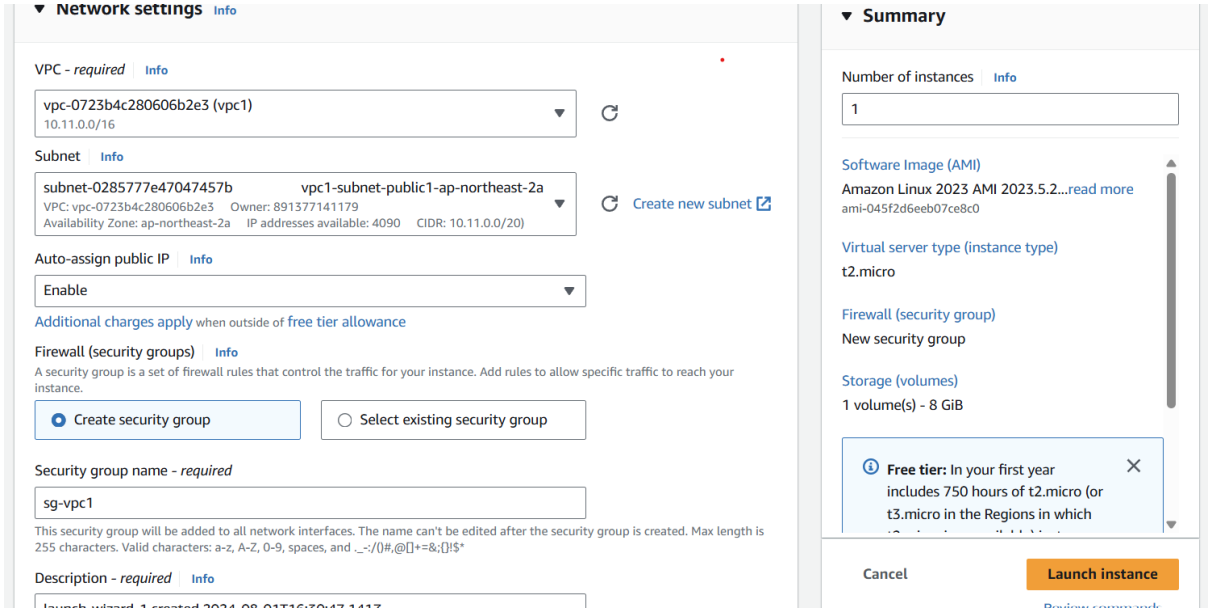
Like wise we need to add for remaining 2 route tables of VPC2 & VPC3. Once route table configuration done we need to create one EC2 instance for each VPC [VPC1, VPC2 & VPC3].

5.EC2 instance:

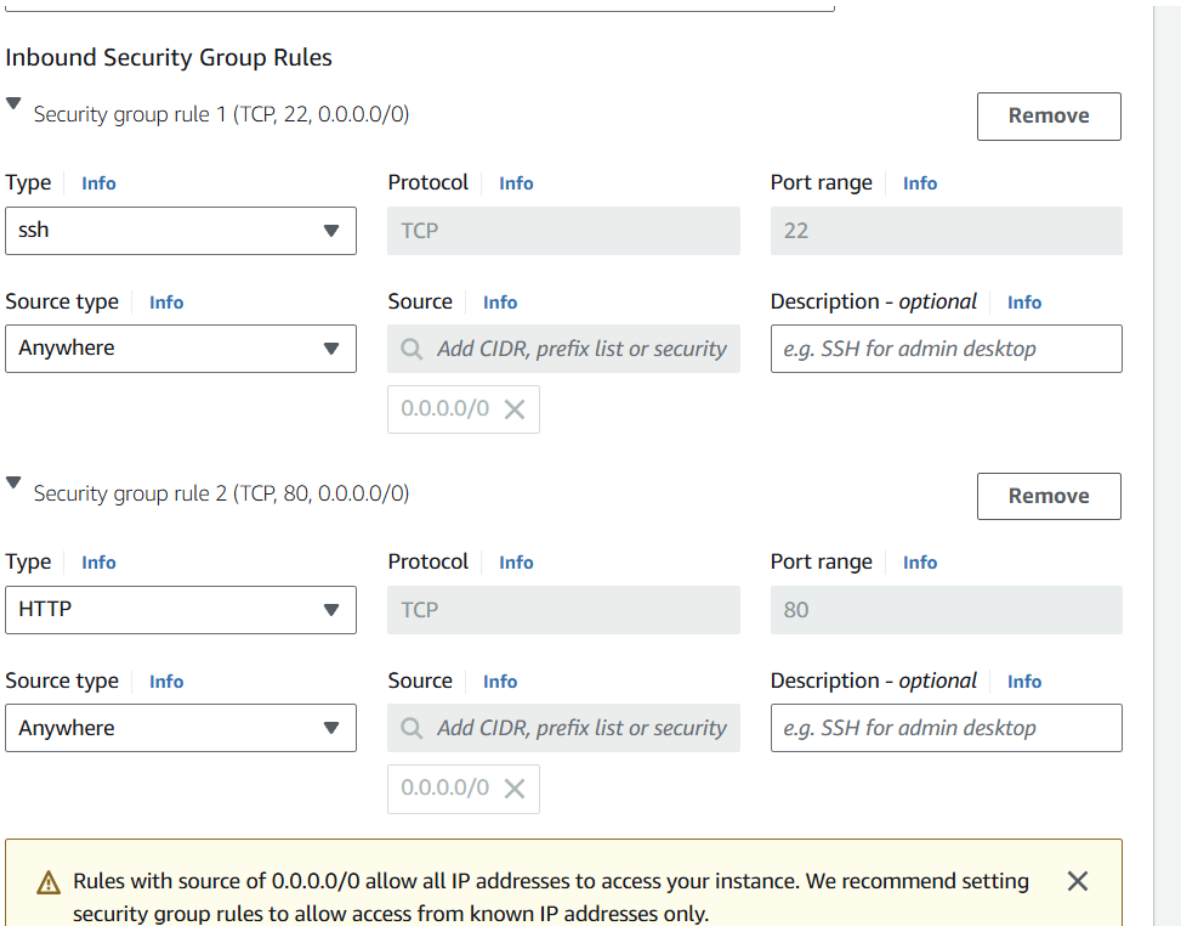
Go EC2 instance and click on launch instance. In that we need to give name for instance and then click on aws or ubuntu or etc -> create key pair as shown in below.



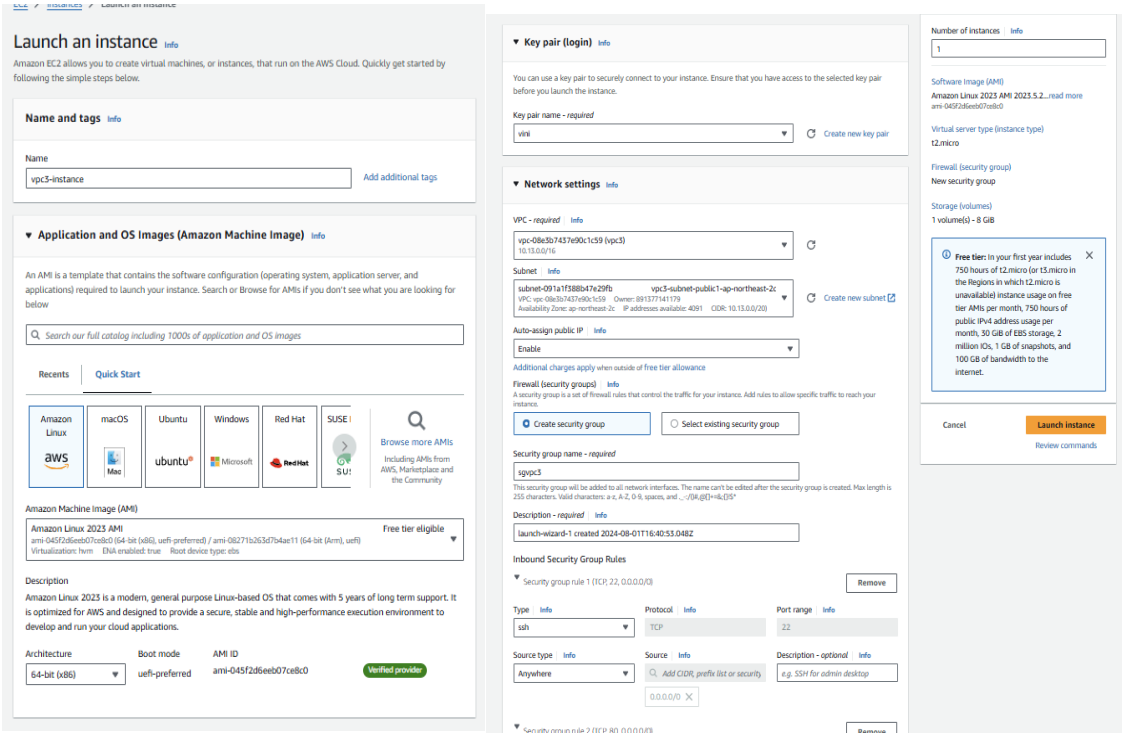
Configure Network Settings as below for VPC1-EC2 instance

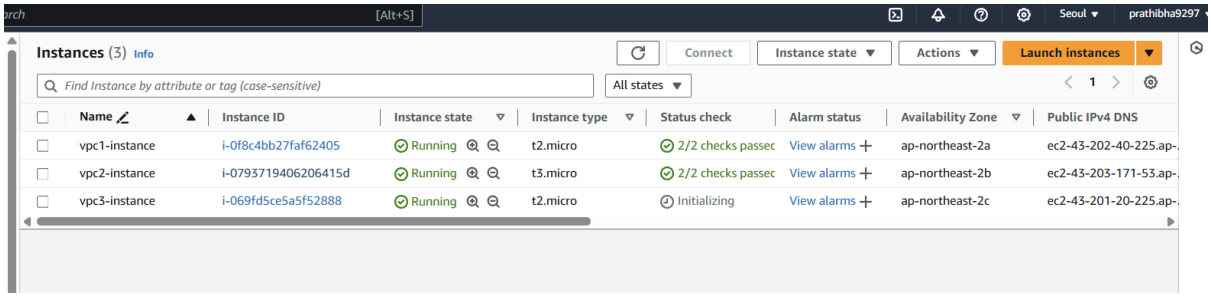


Create the security group and allow the port numbers 22 for ssh and 80 for http as shown in below

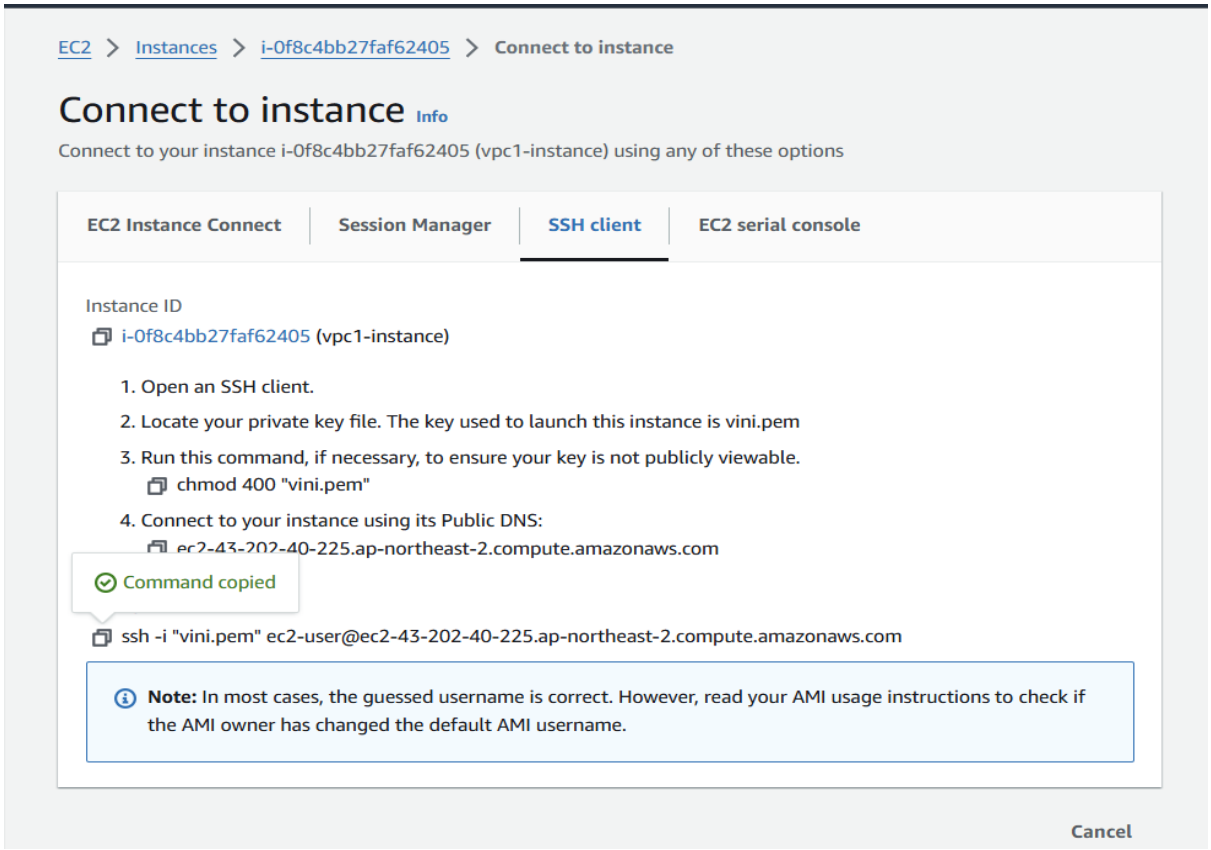


Likewise create another 2 instances for VPC2 and VPC3. The overall configuration as shown in below diagrams for creating EC2 instance.





Once instances came to running or available state then click on connect and copy the ssh command in ssh client field as shown in below diagram.



After that go to git bash and connect to the server. Once connected install the nginx and create the html file. Likewise connect remaining 2 instances [VPC2-instance & VPC3 - instance] and create files in it. Once done try to connect from VPC1-instance server to remaining 2 servers. It will get connect because of transit gateway. The output was shown in below pictures.

Output from VPC1-instance:

```
[root@ip-10-11-0-46 html]# curl 10.11.0.46:80
Hi this is from vpc1
[root@ip-10-11-0-46 html]# curl 10.12.11.144:80
hi this is from vpc2
[root@ip-10-11-0-46 html]# curl 10.13.12.252:80
hi this is from vpc3
[root@ip-10-11-0-46 html]#
```

Output from VPC2-instance:

```
[root@ip-10-12-11-144 html]# curl 10.12.11.144:80
hi this is from vpc2
[root@ip-10-12-11-144 html]# curl 10.11.0.46:80
Hi this is from vpc1
[root@ip-10-12-11-144 html]# curl 10.13.12.252:80
hi this is from vpc3
[root@ip-10-12-11-144 html]#
```

Output from VPC3-instance:

```
[root@ip-10-13-12-252 html]# curl 10.13.12.252:80
hi this is from vpc3
[root@ip-10-13-12-252 html]# curl 10.12.11.144:80
hi this is from vpc2
[root@ip-10-13-12-252 html]# curl 10.11.0.46:80
Hi this is from vpc1
[root@ip-10-13-12-252 html]#
```

Conclusion:

utilizing an AWS Transit Gateway to connect three Virtual Private Clouds (VPCs) offers a highly efficient and scalable networking solution. The Transit Gateway simplifies the network architecture by providing a central hub for interconnecting multiple VPCs, thereby reducing the complexity and number of peering connections needed. This setup not only enhances network management and monitoring but also improves security and performance through centralized traffic routing.