Problem Statement:

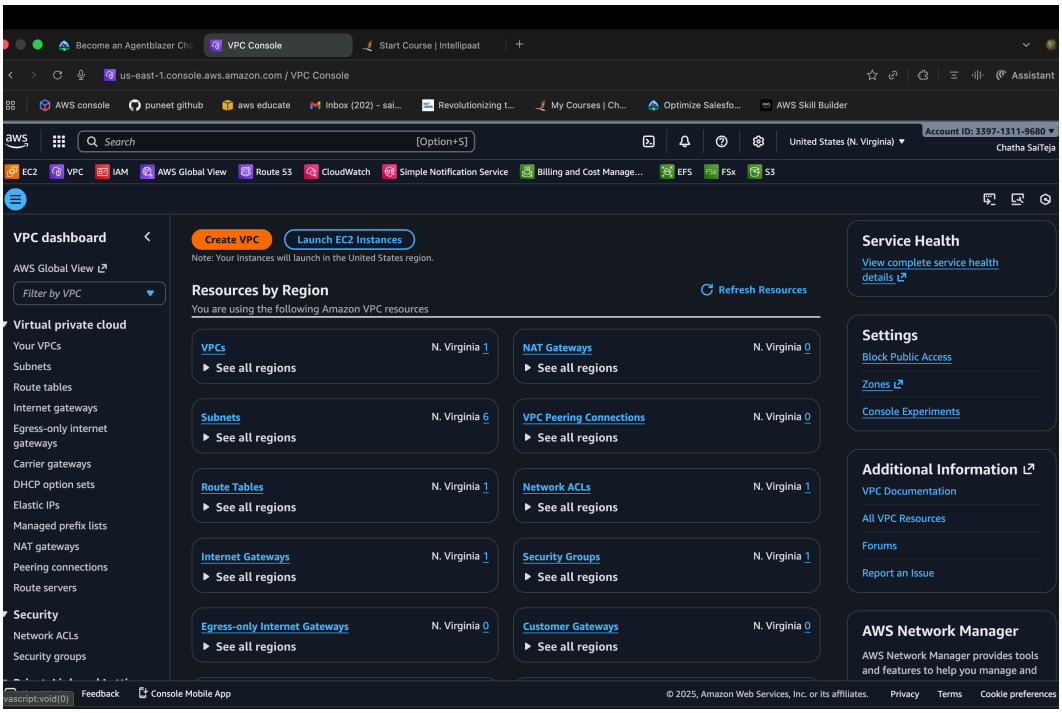
You work for XYZ Corporation and based on the expansion requirements of your corporation you have been asked to create and set up a distinct Amazon VPC for the production and development team. You are expected to perform the following tasks for the respective VPCs

Production Network:

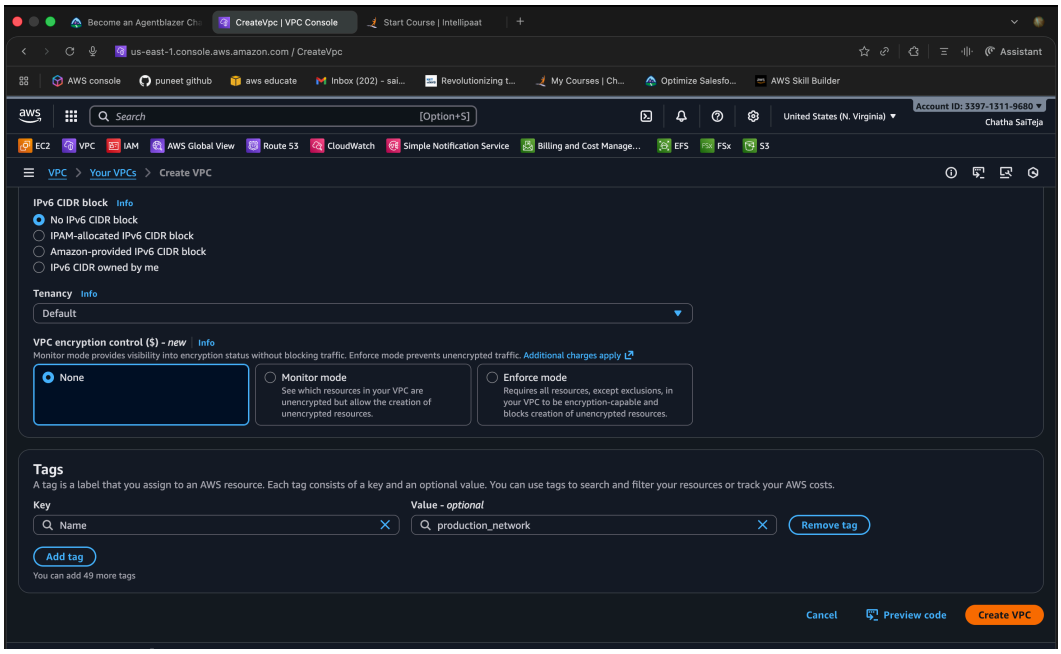
1. Design and build a 4-tier architecture.
2. Create 5 subnets out of which 4 should be private named app1, app2, dbcache and db and one should be public, named web.
3. Launch instances in all subnets and name them as per the subnet that they have been launched in.
4. Allow dbcache instance and app1 subnet to send internet requests.
5. Manage security groups and NACLs.

Step -By- Step Procedure:-

Step 1:-In the AWS VPC console, click Create VPC, assign a CIDR block (e.g., 10.0.0.0/16) to define the network range, name it Production Network, and click Create to establish the VPC.

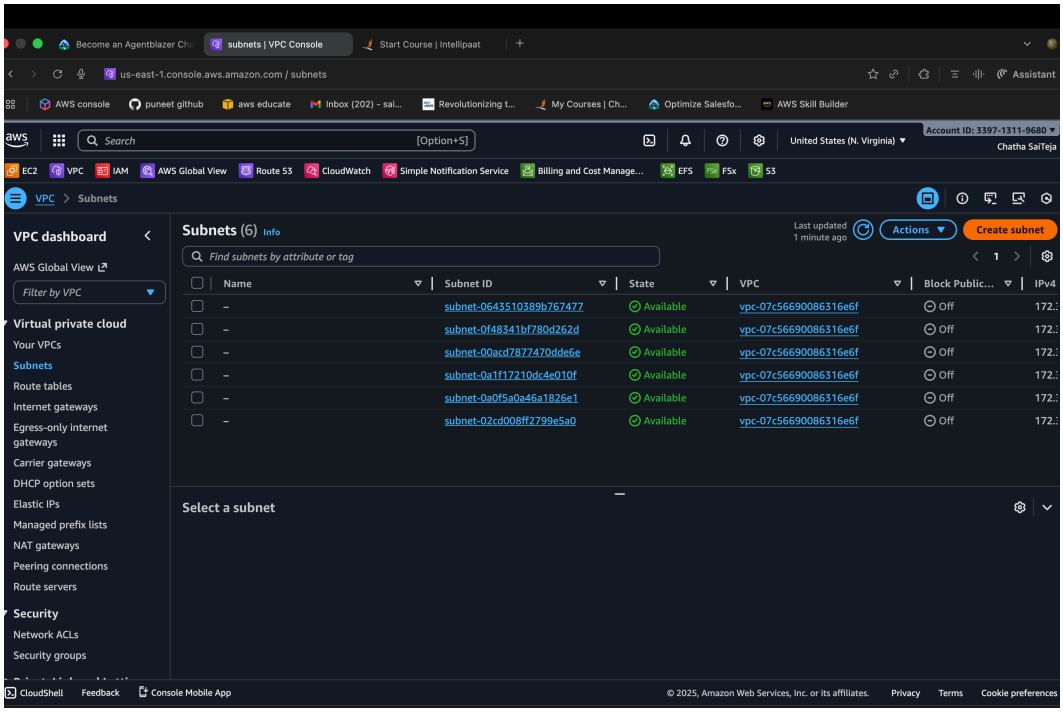


Click on create Vpc

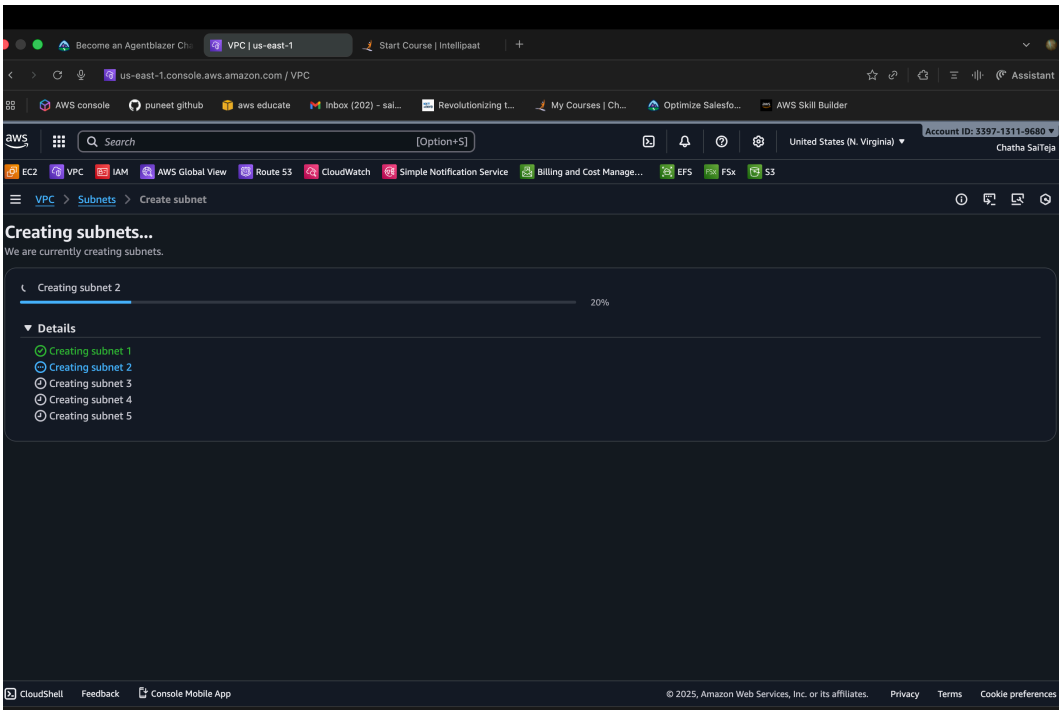


Give network range and specify the name of vpc and click on create

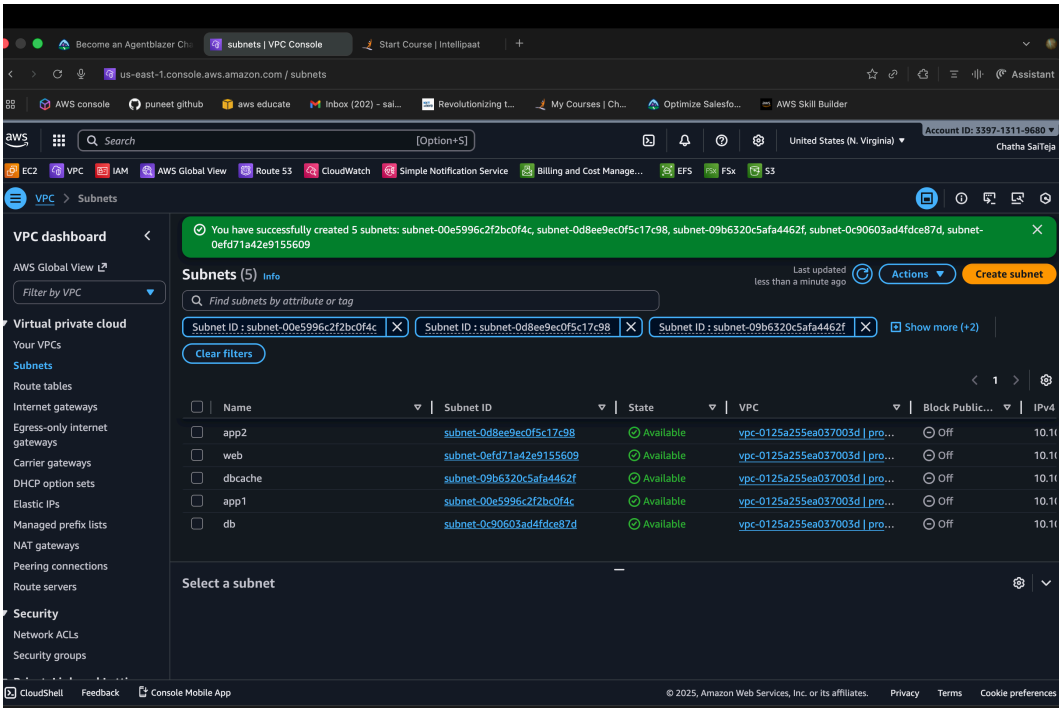
Step 2:- In production network we have to create 5 subnets out of which 4 should be private named app1, app2, dbcache and db and one should be public, named web , To create subnet,go to subnets then give name and network range of subnet which should be under production network range.



Click on create

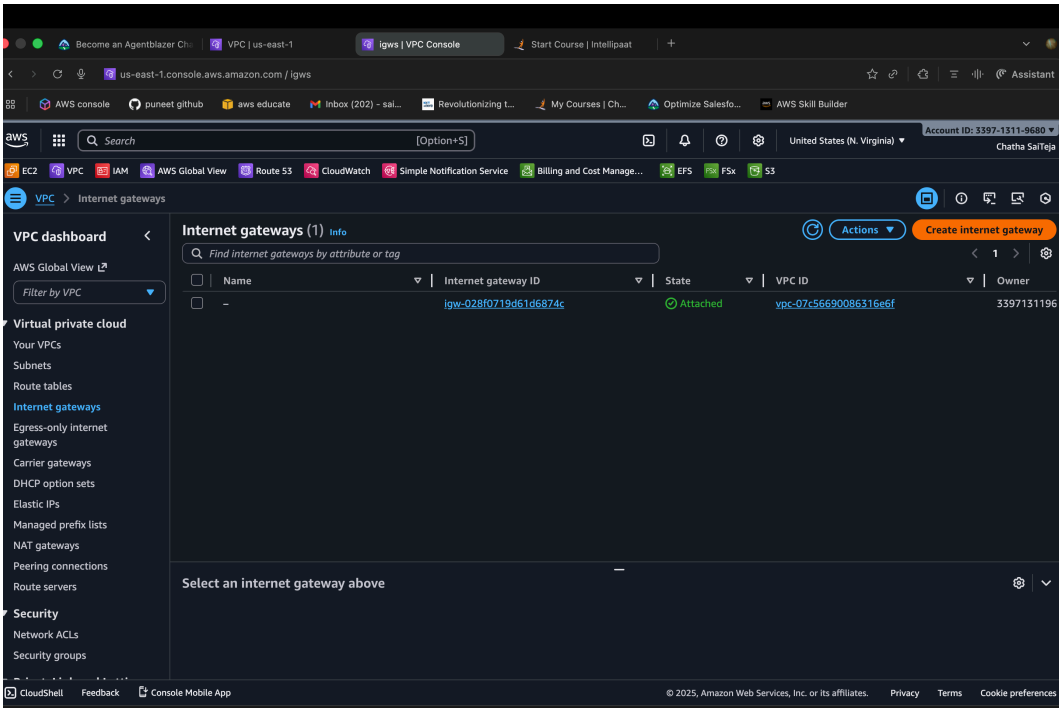


Five subnets are creating

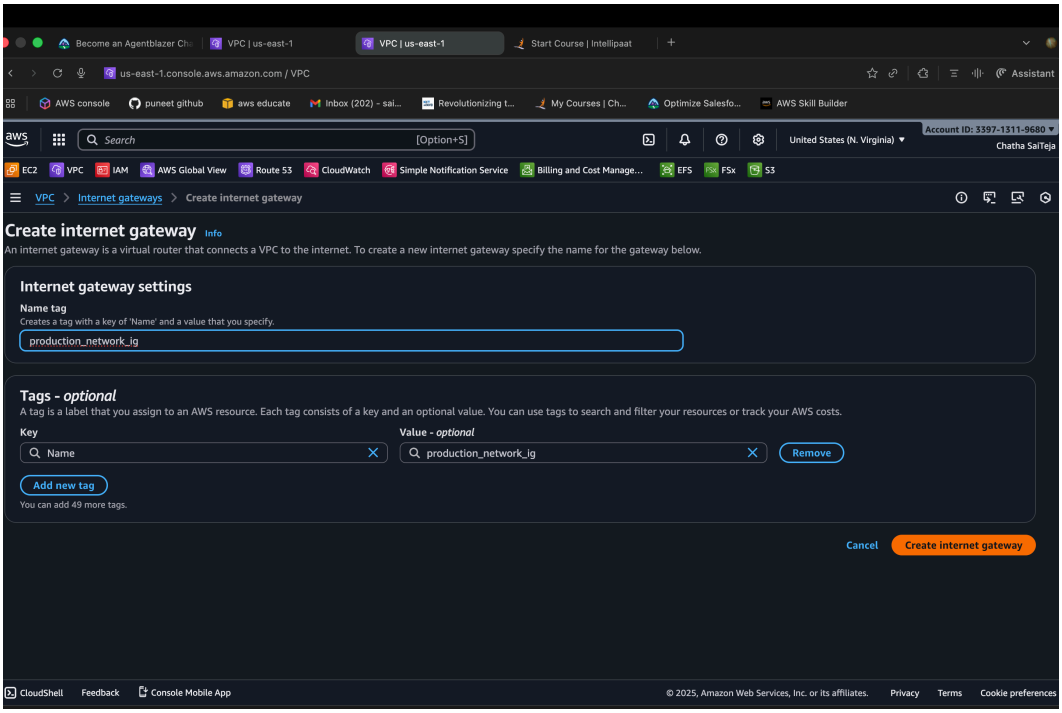


Five subnets were created

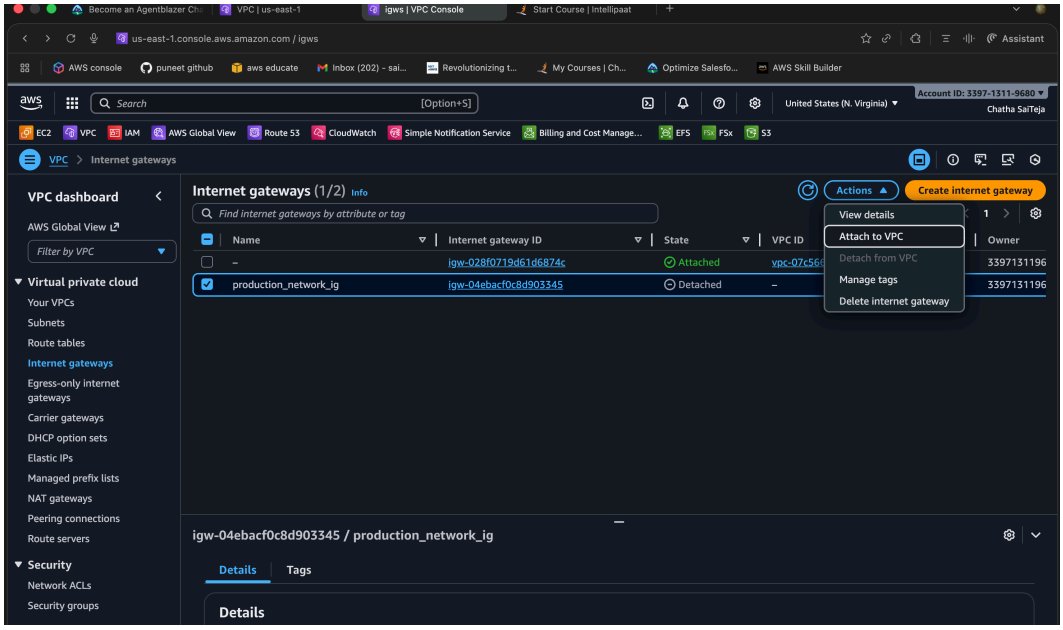
Step 3:- Create an Internet Gateway by clicking Create, naming it appropriately, and clicking Create. Then, go to Actions → Attach to VPC and attach it to the Production Network VPC, enabling internet connectivity for the public subnet



Click on create Internet Gateway

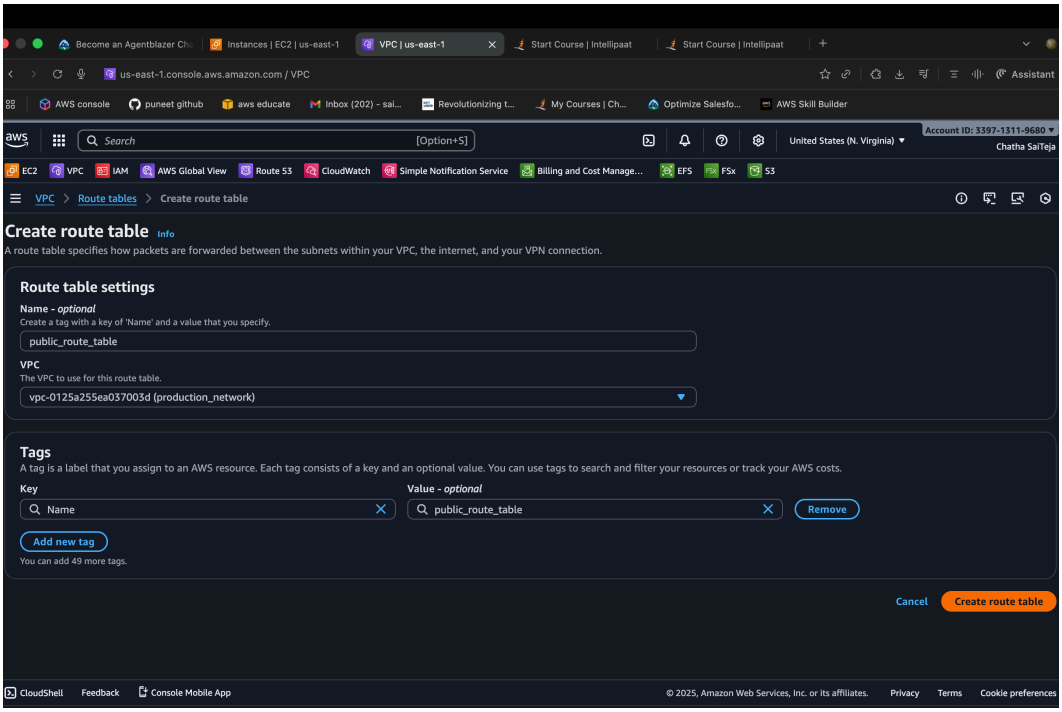


Specify the name (production_network_ig)

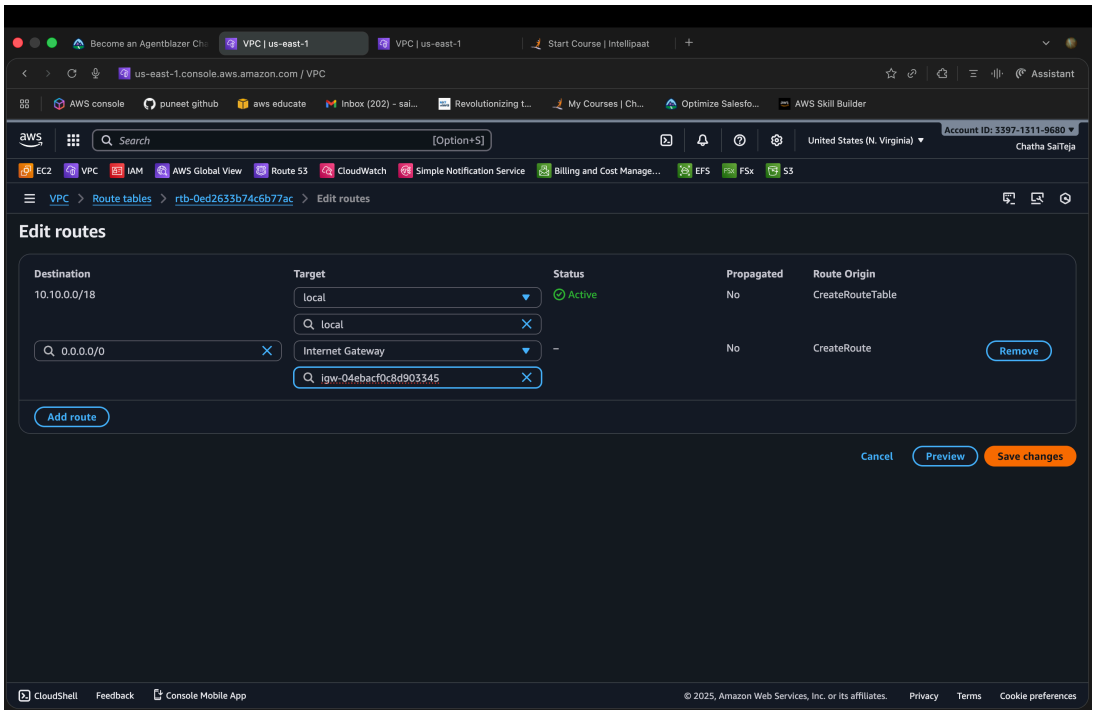


Go to Actions -> click on attach and then attach to production network

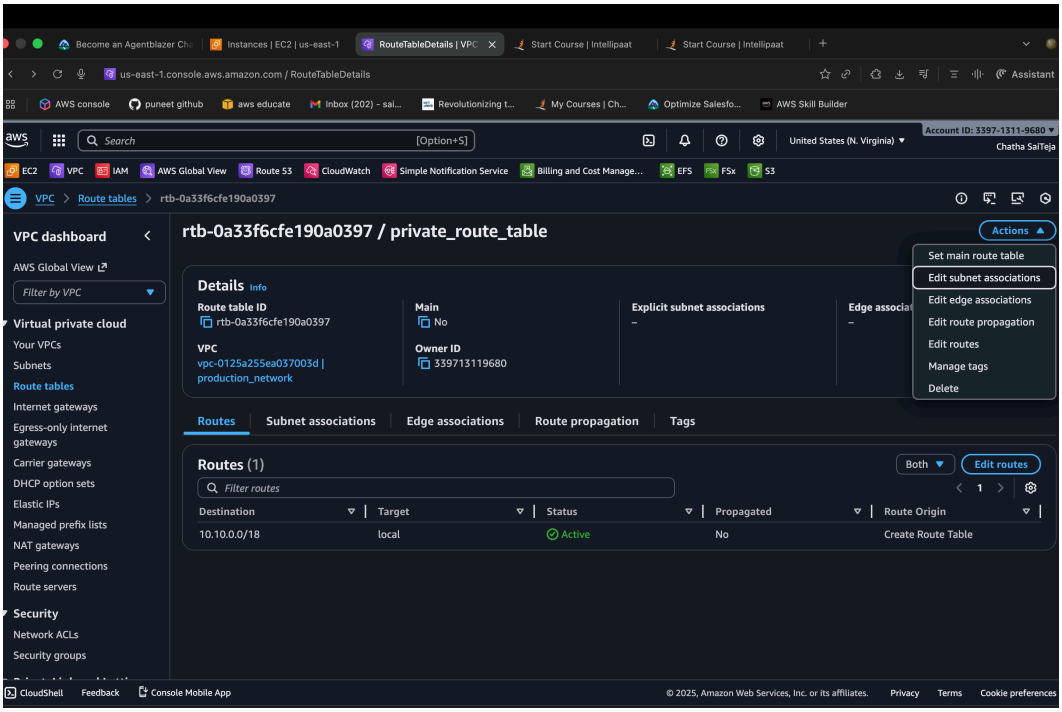
Step 4:- When subnets are created, a route table with only local routes is available by default, but we still need to create separate public and private route tables; each new route table initially contains only local routes, so for the public route table we must add a default route pointing to the Internet Gateway to allow the public subnet to send traffic to the internet.



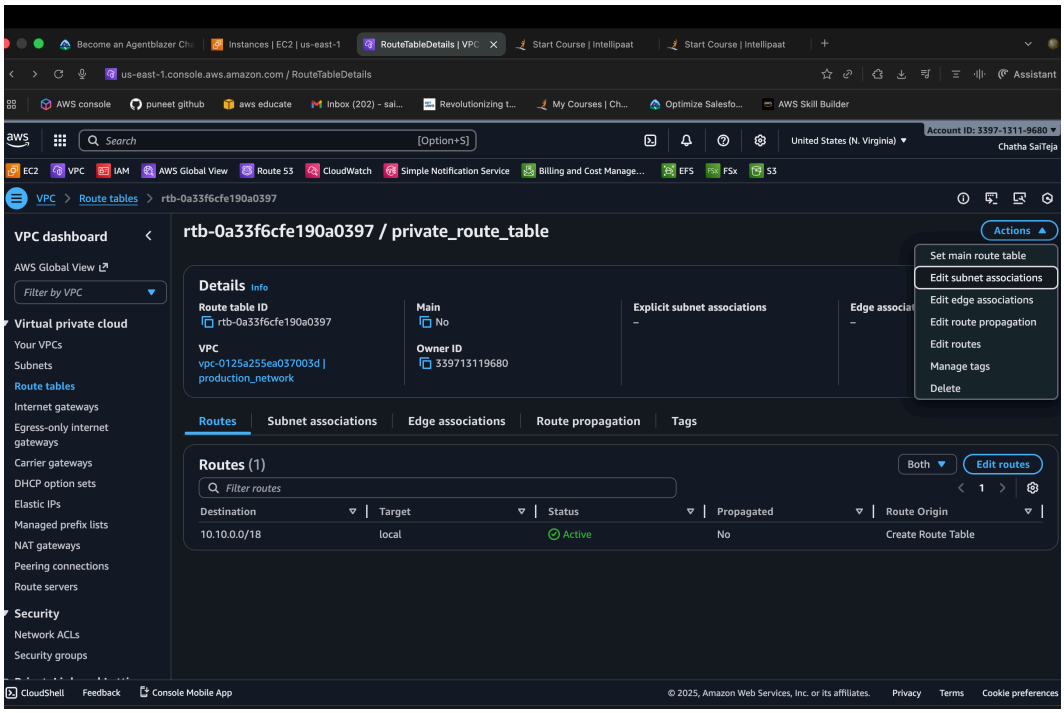
Give name and select production(VPC)



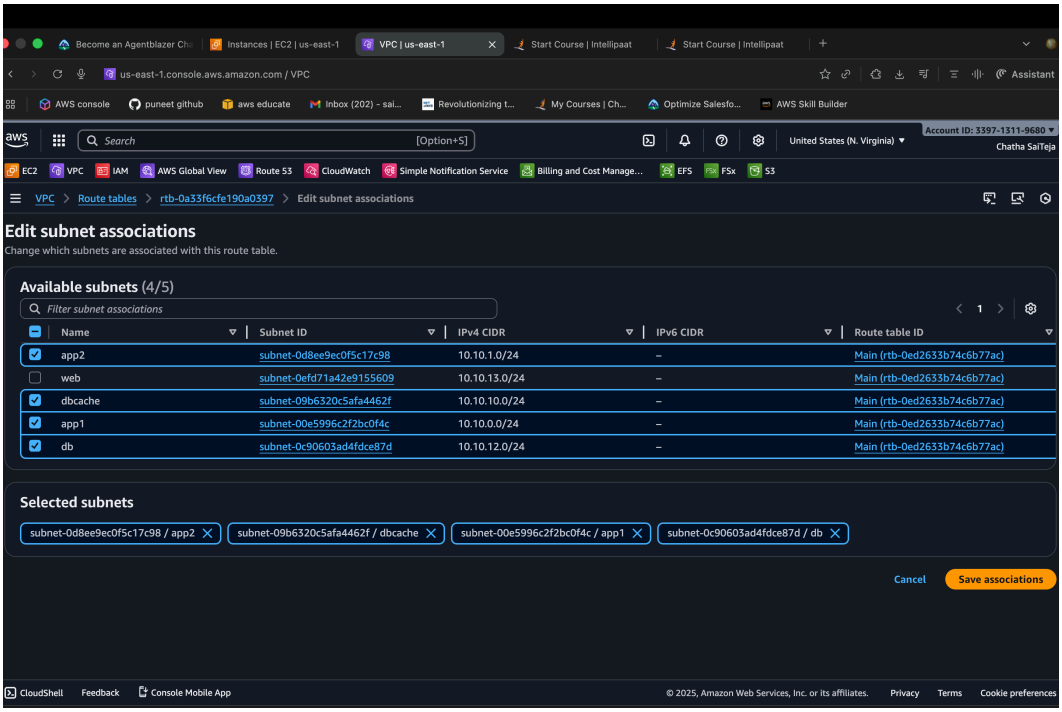
Add internet gateway route to public route table



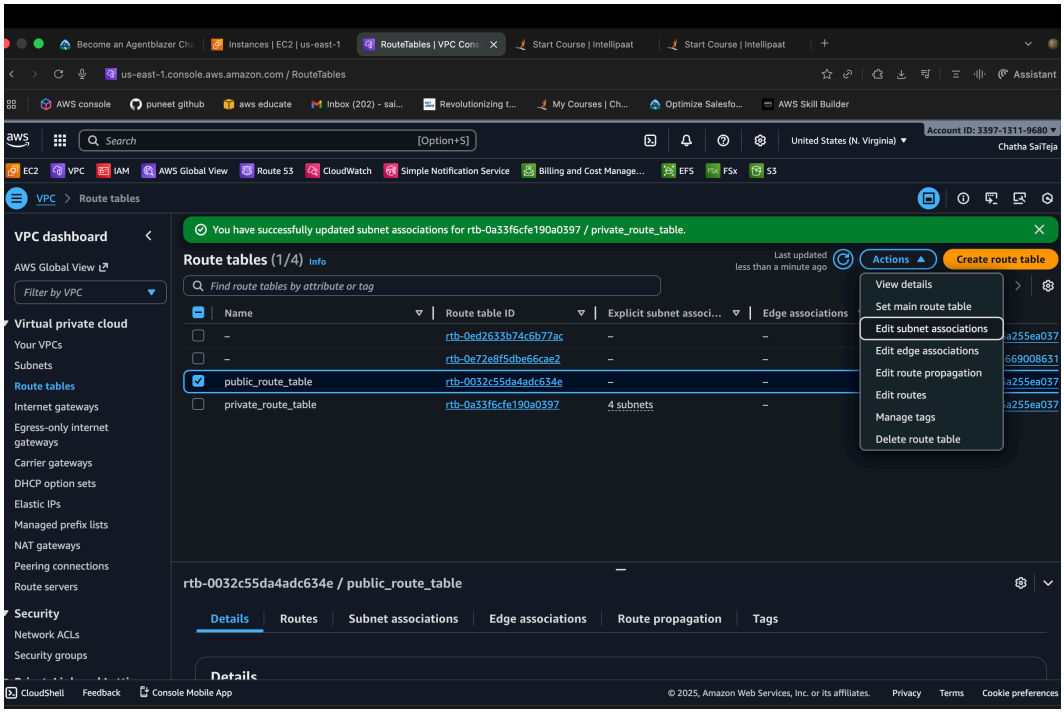
Step 5:-After creating the two route tables (public and private), we need to update the subnet associations so that four private subnets are attached to the private route table and the single public subnet is attached to the public route table.



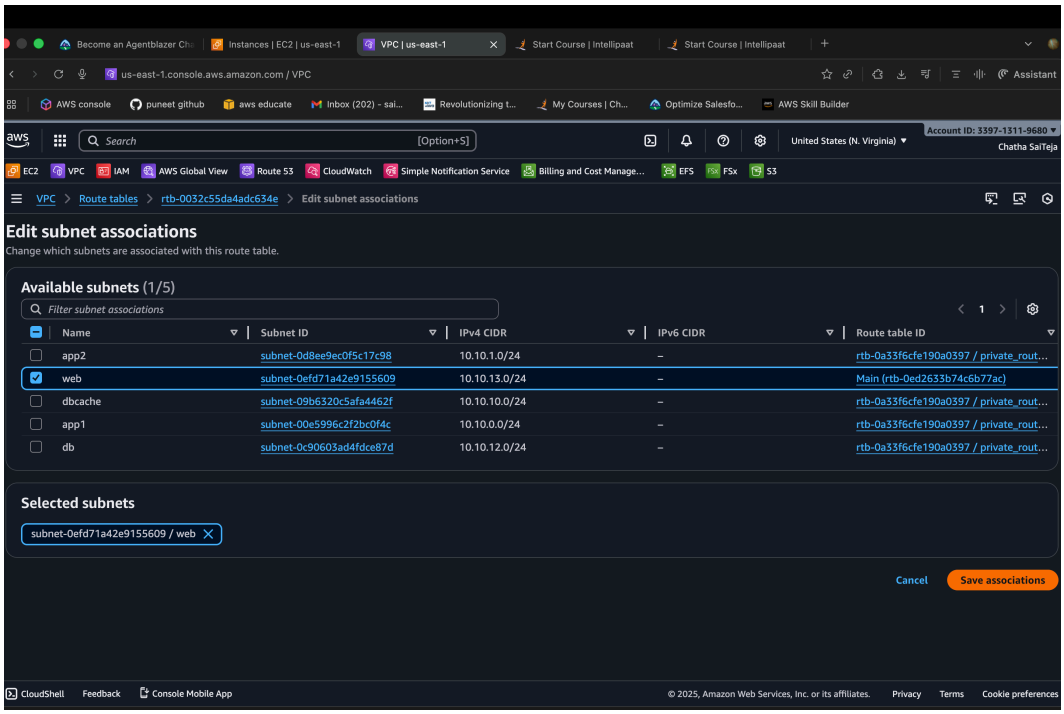
Click on edit subnet associations



Select 4 subnets(app1, app2, dbcache, db) and save associations

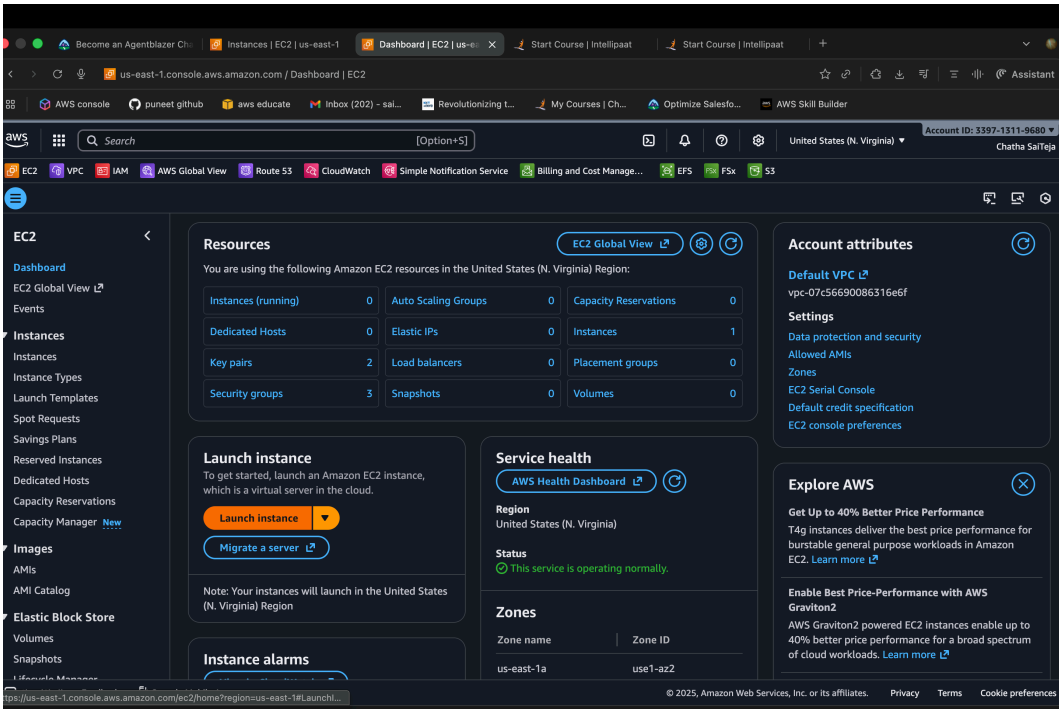


For public route table click on edit subnet

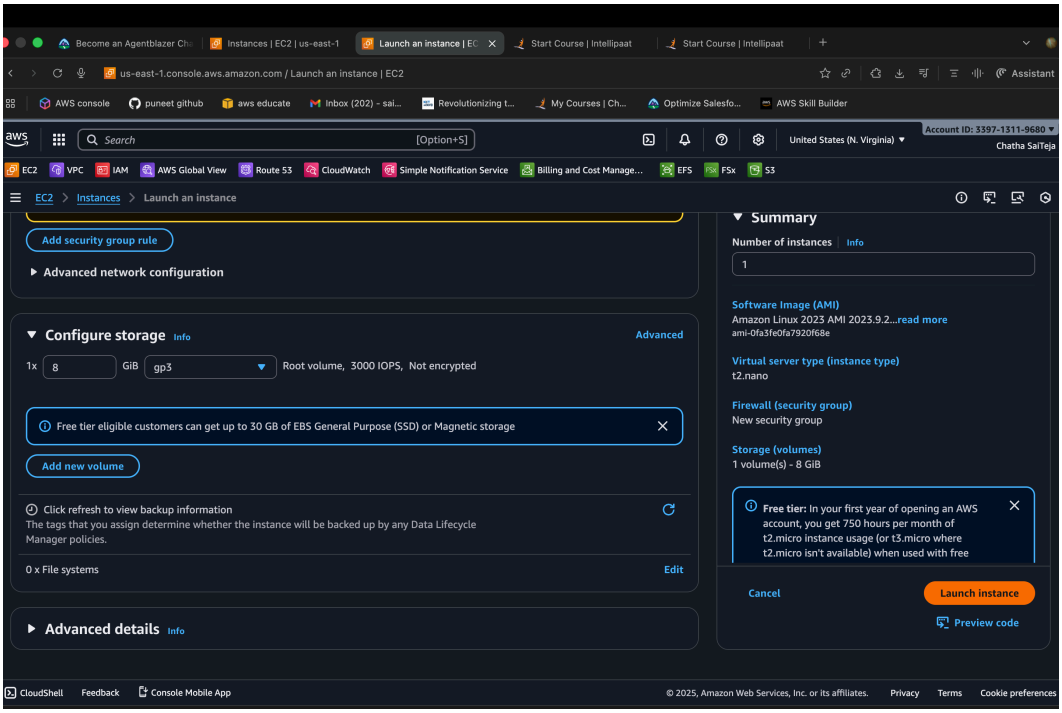


Select public subnet(web) and click on save

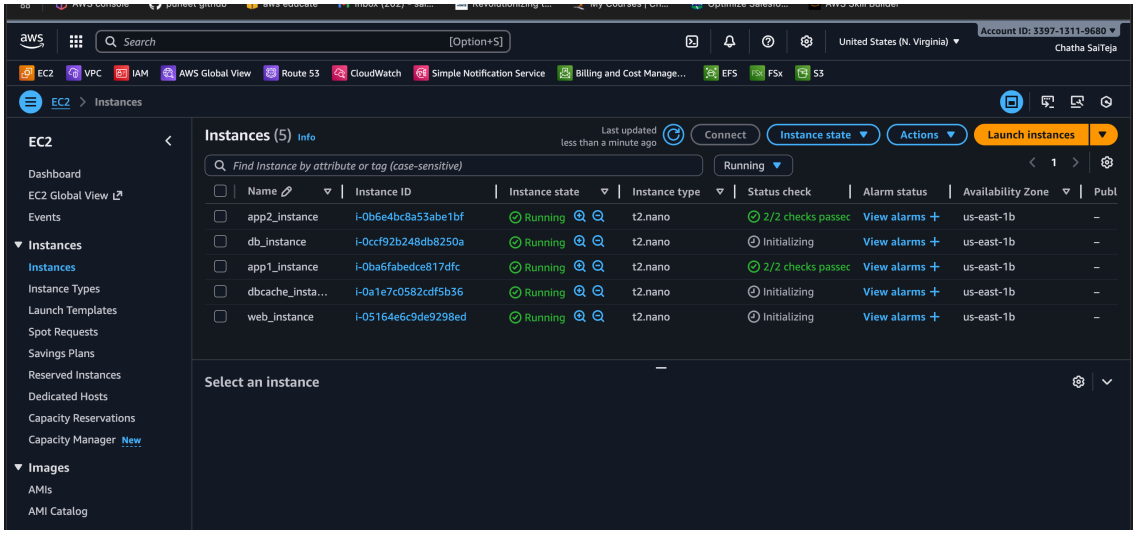
Step 6:-"We have created a production VPC with five subnets (app1, app2, db, dbcache, and web), attached an Internet Gateway, and set up two route tables (public and private); now we need to launch EC2 instances in each subnet.



Click on Launch Instance

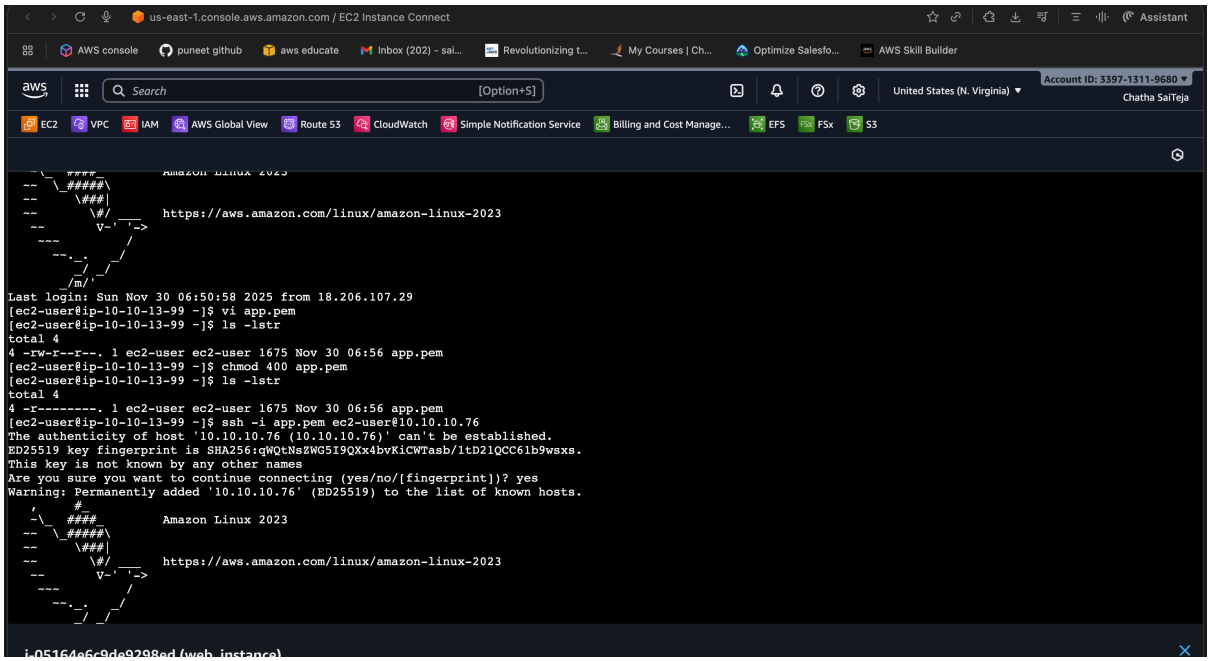


Give instance details and click on launch



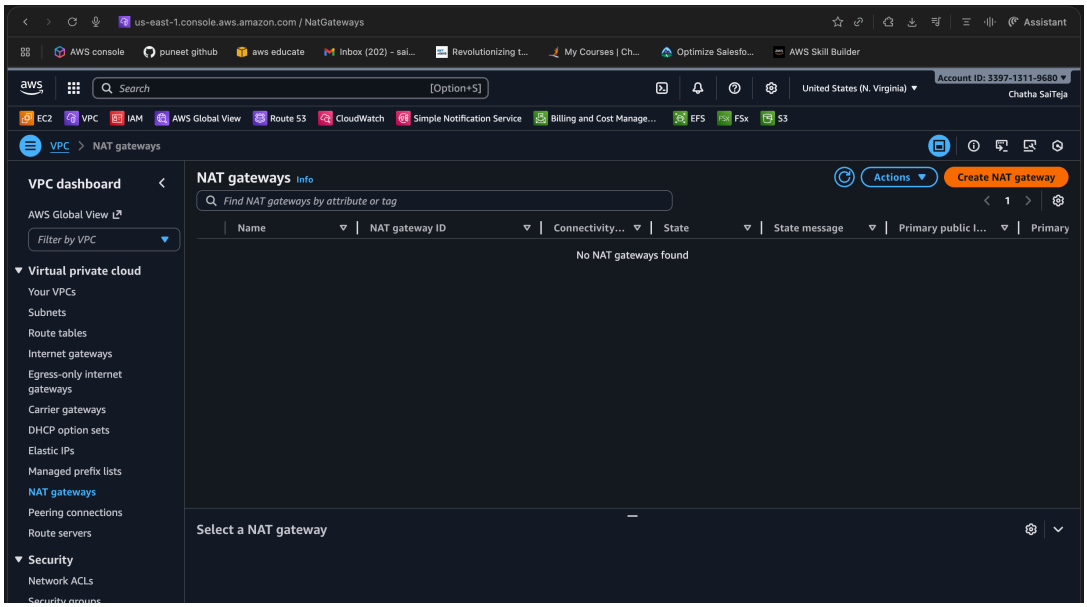
We created five instances in five subnets

Step 7:-To establish SSH connectivity to private instances in the dbcache and app1 subnets through the web instance (which acts as a bastion/jump server), first connect to the web instance in the public subnet, then from within that session, connect to the private instances by executing the SSH command: **ssh -i <privateKey> ec2-user@<privateIp>**, where **<privateKey>** is the path to your EC2 key pair file and **<privateIp>** is the private IP address of the target instance. This multi-hop SSH connection allows instances in private subnets to establish outbound internet connectivity and communicate securely without direct internet exposure.

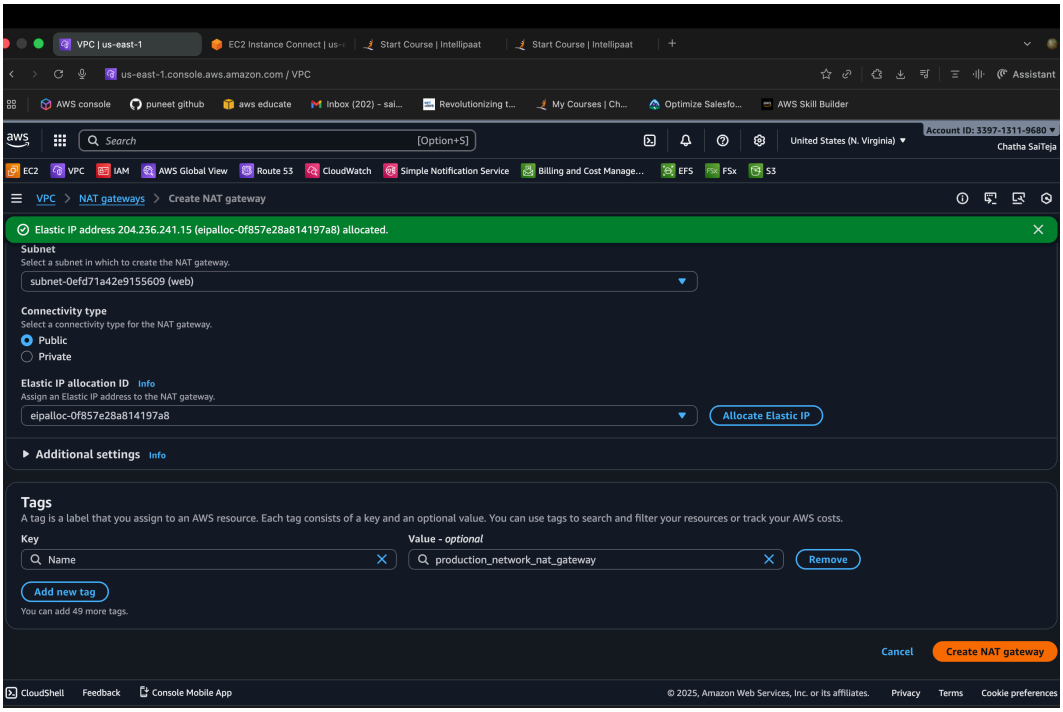


We connected db cache Instance through web by ssh connectivity

Step 8:- To enable outbound internet connectivity from instances in private subnets, a NAT (Network Address Translation) Gateway must be created and placed in the public subnet. During NAT Gateway creation, an Elastic IP address must be allocated and associated with the gateway, which serves as the static public IP address used for all outbound internet communications originating from the private subnet instances. This configuration allows private instances to initiate outbound requests to the internet while remaining isolated from inbound internet traffic

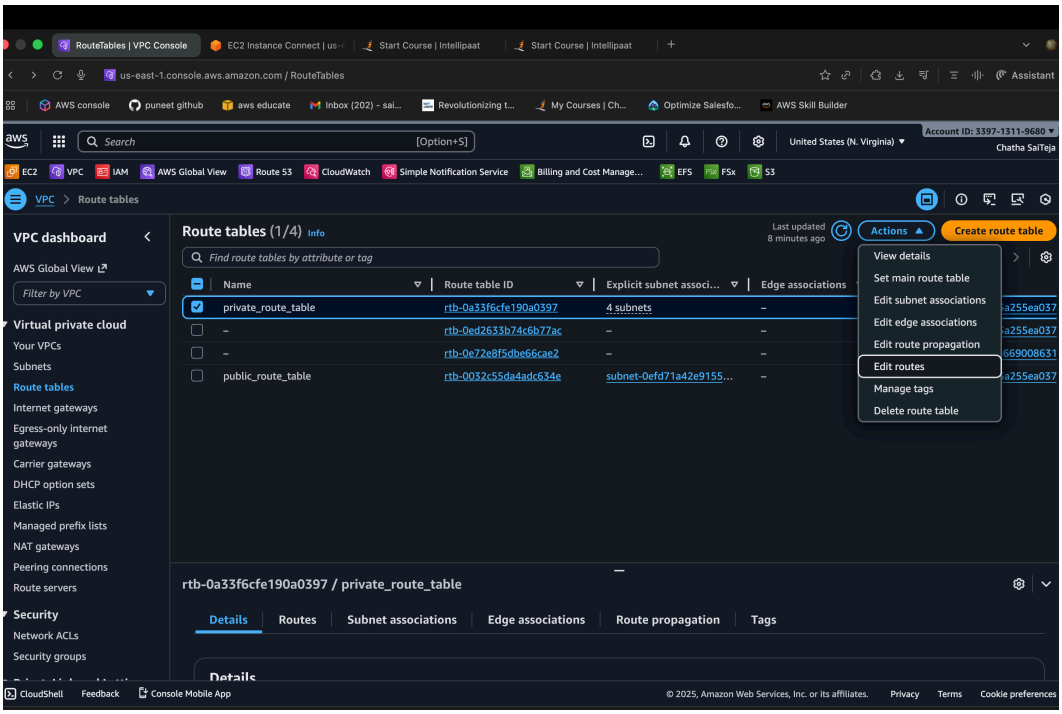


Click on create Nat Gateway

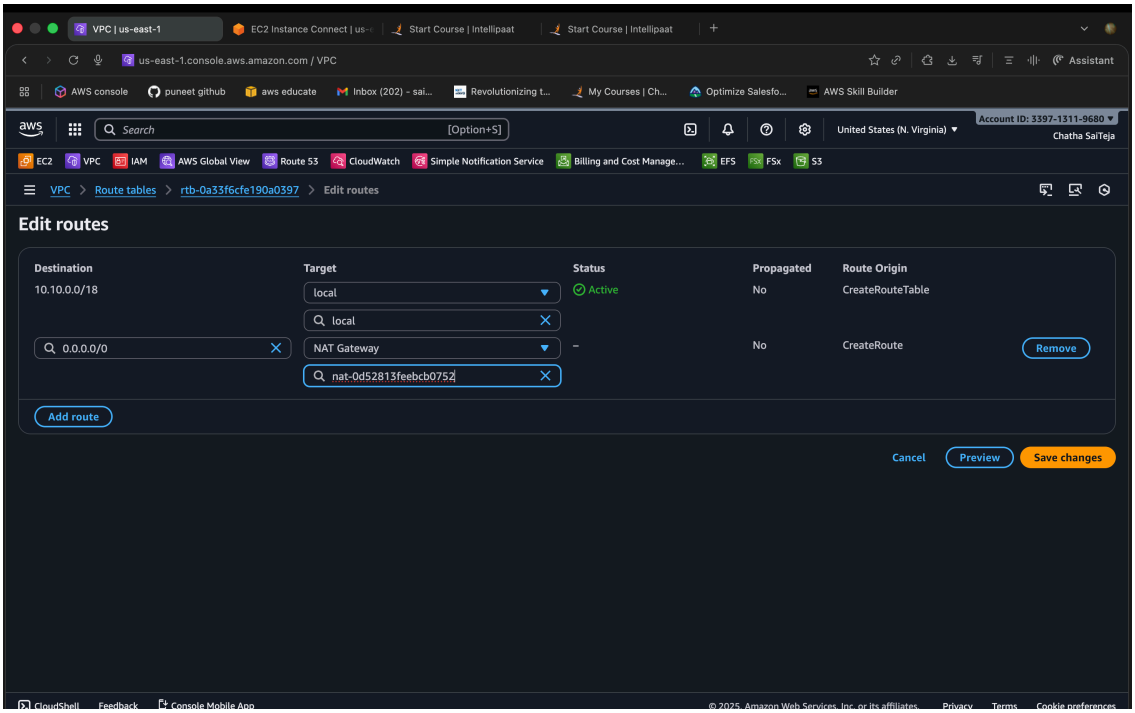


Give name and allocate elastic Ip then click on create

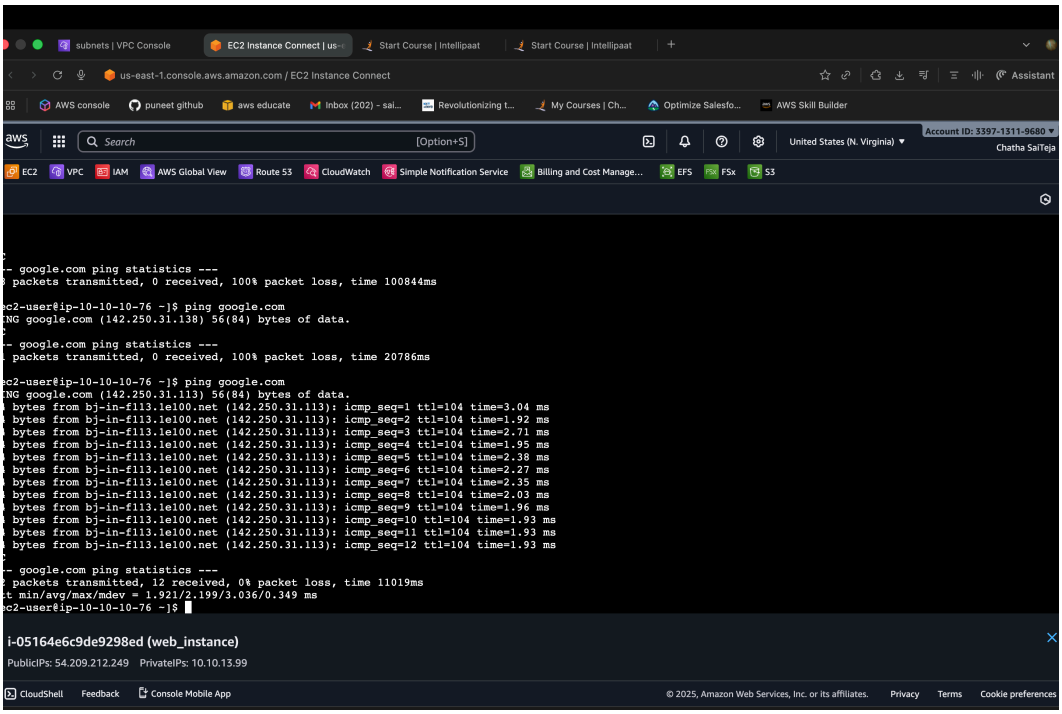
Step 9 :-Now update the private route table by adding a route that points internet-bound traffic to the NAT Gateway. After this, the private-subnet instances can send outbound internet requests. Connect to the instance and test with a command like **ping google.com;** you should see successful packet responses, confirming internet access.



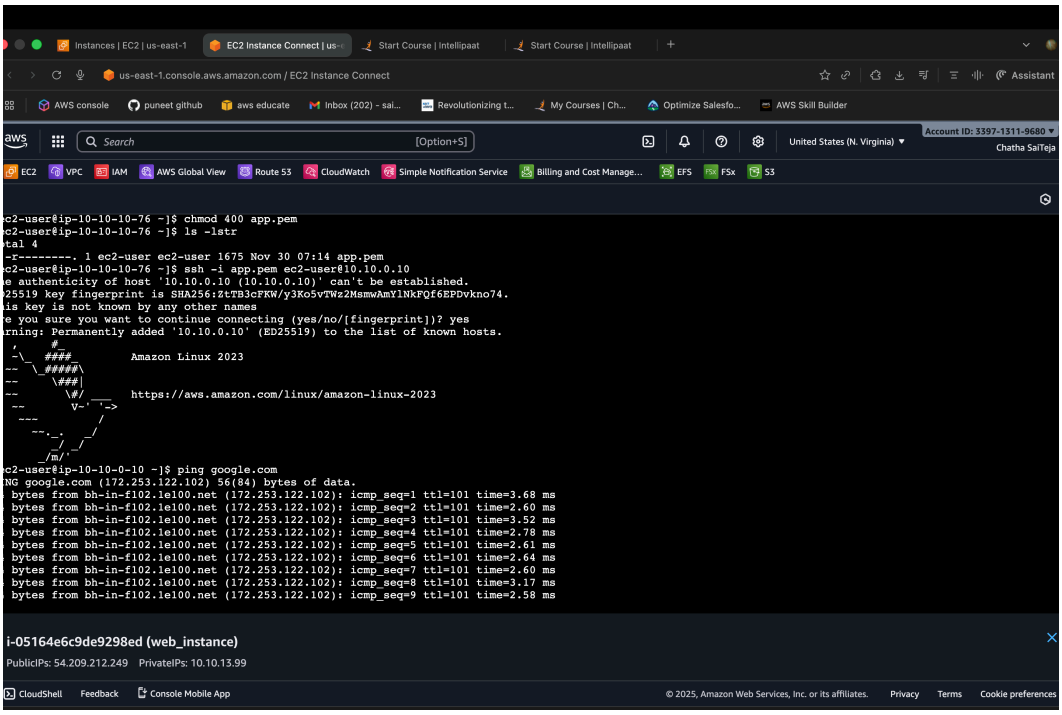
Click on edit routes



Add the nat route and click on save changes



dbcache instance can send internet requests



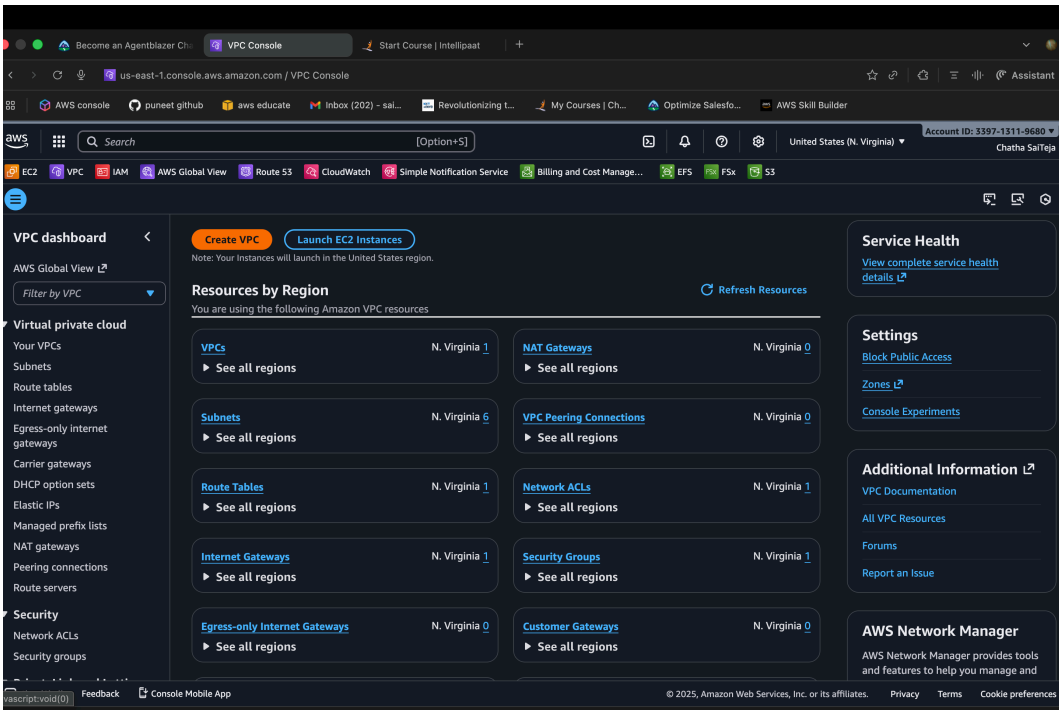
App1 instance can send internet requests

Development Network:

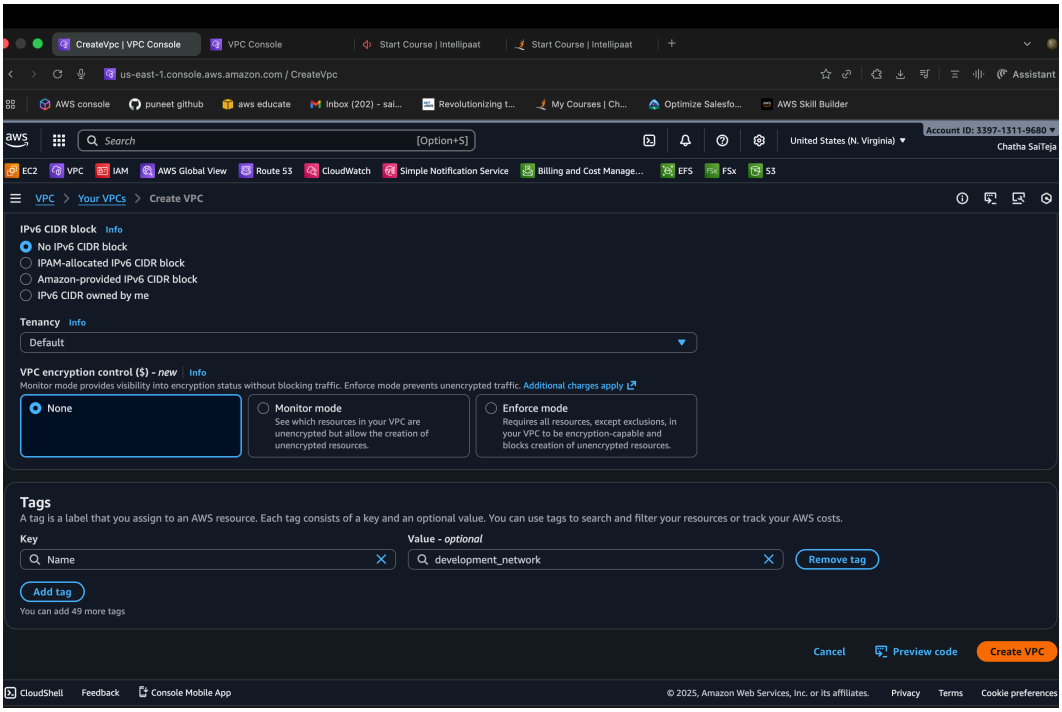
1. Design and build 2-tier architecture with two subnets named web and db and launch instances in both subnets and name them as per the subnet names.
2. Make sure only the web subnet can send internet requests.
3. Create peering connection between production network and development network.
4. Setup connection between db subnets of both production network and development network respectively.

Step -By - Step Procedure:-

Step1 :- In the AWS VPC console, click Create VPC, assign a CIDR block (e.g., 10.0.0.0/16) to define the network range, name it Development Network, and click Create to establish the VPC.

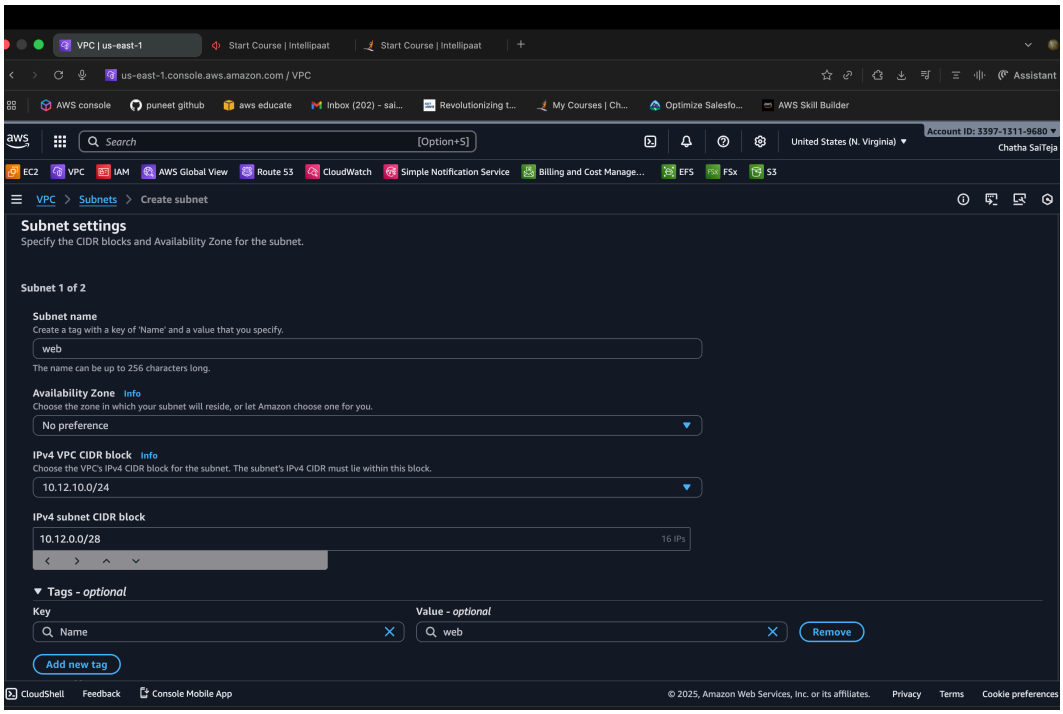


Caption

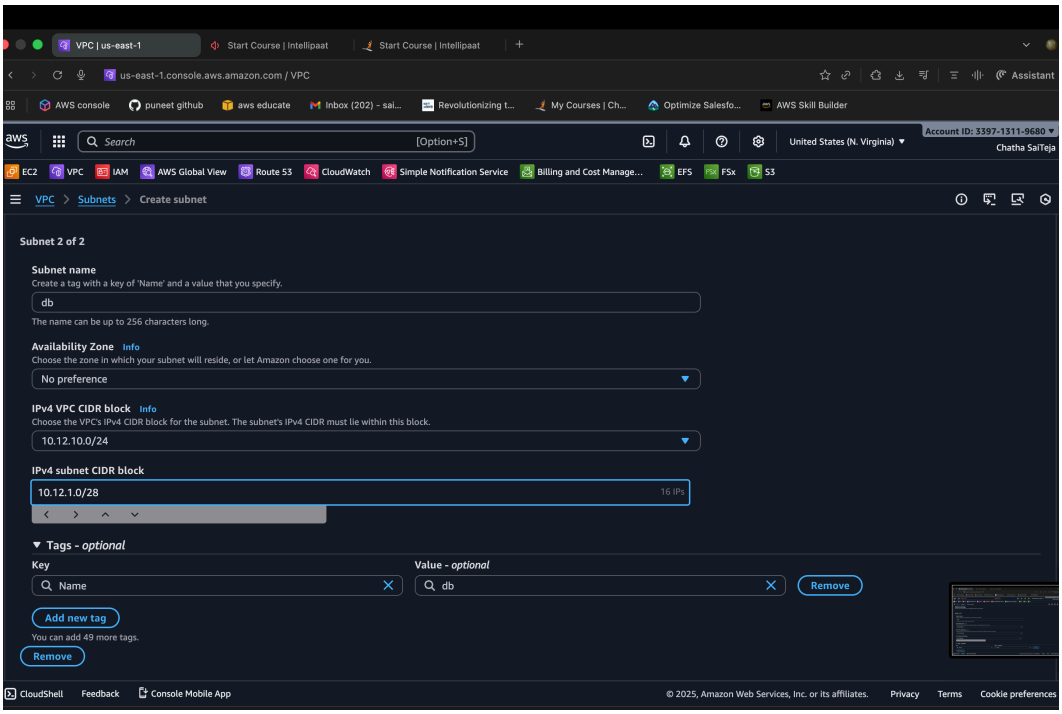


Caption

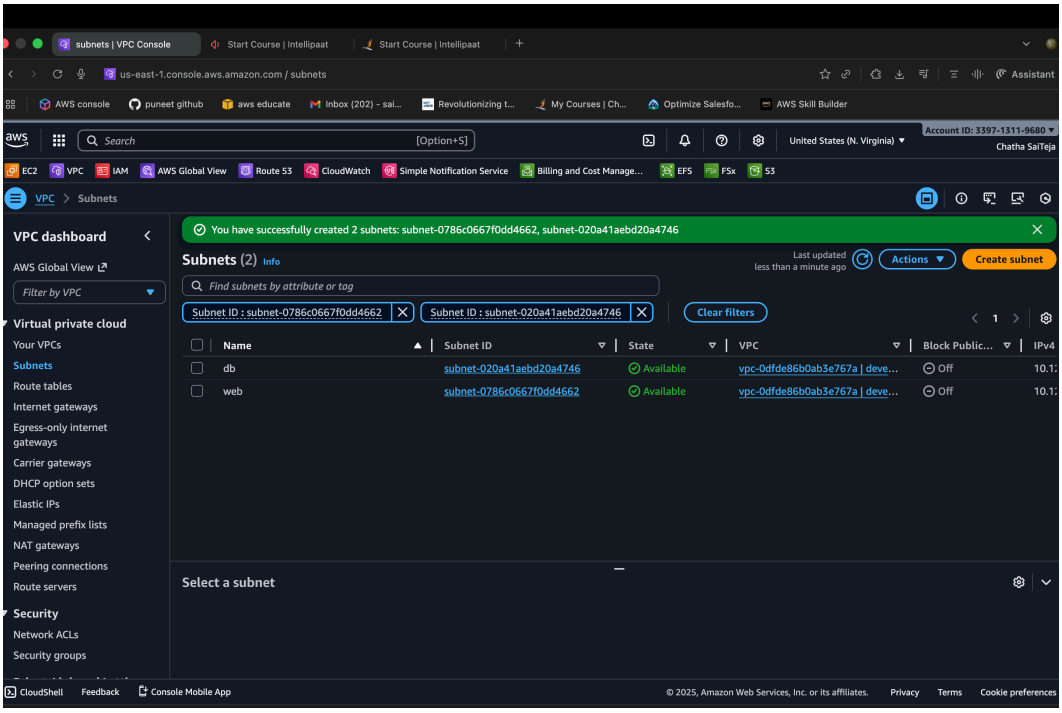
Step 2:-In Development Network we have to create two subnets named web and db and web is public subnet which can send internet requests and db is private.



Creating web subnet

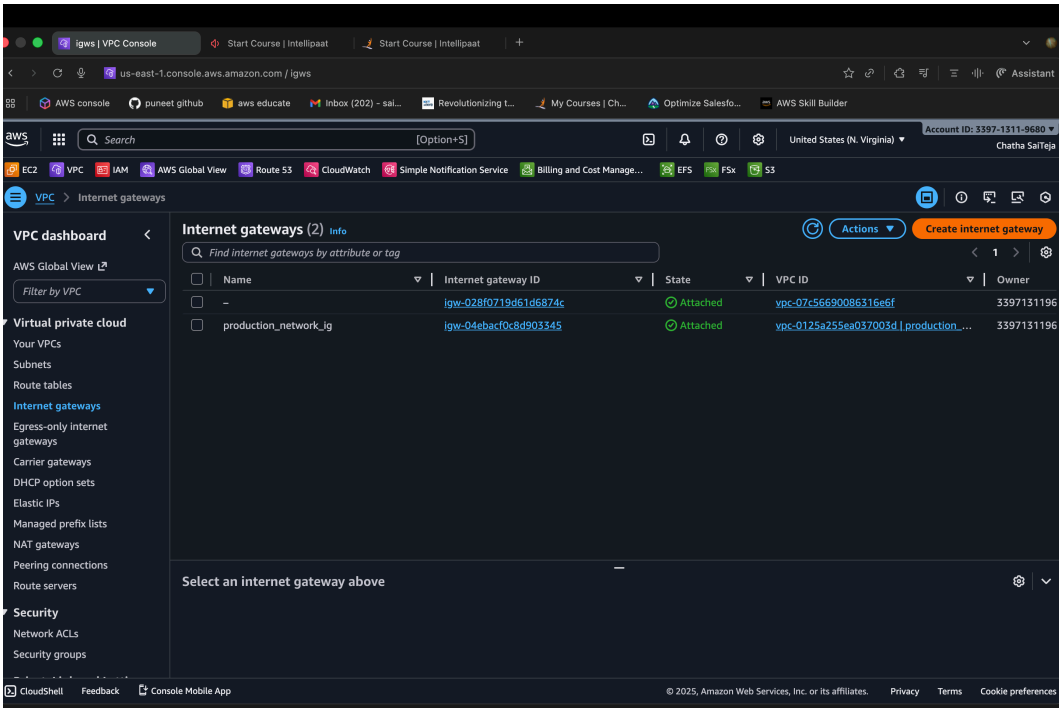


Db subnet

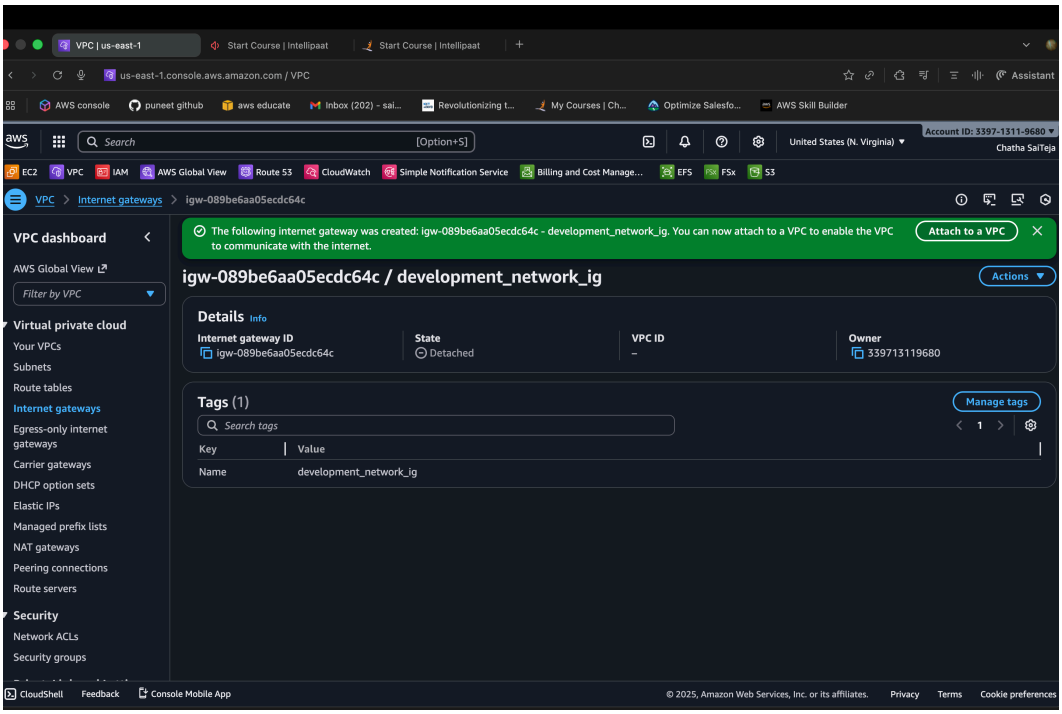


Two subnets are created in development network

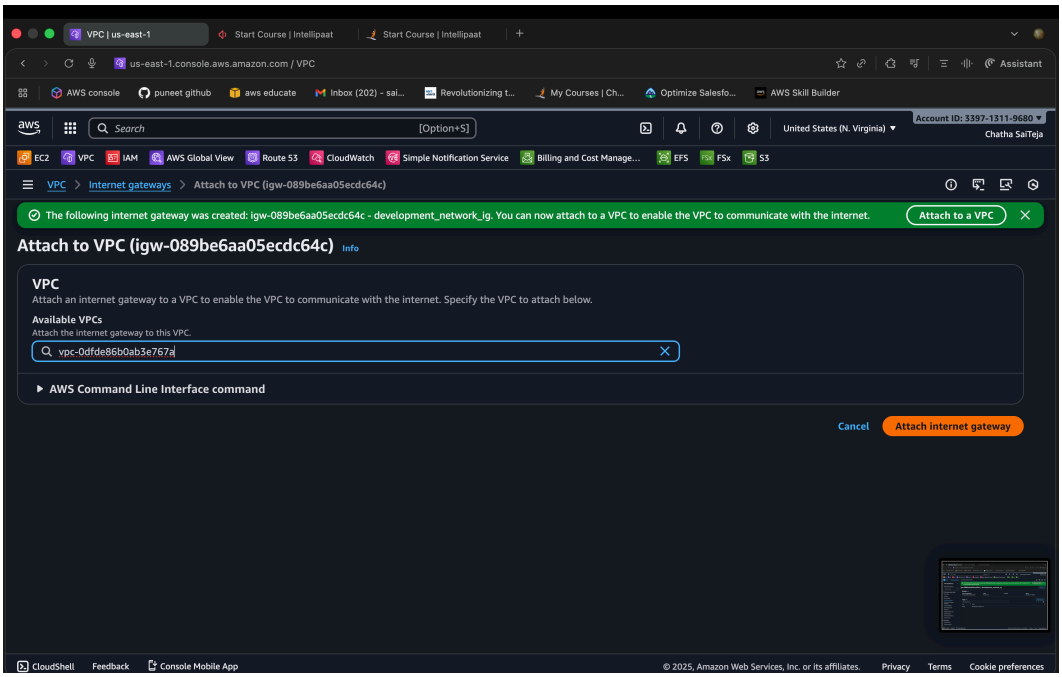
Step 3:- Create an Internet Gateway by clicking Create, naming it appropriately, and clicking Create. Then, go to Actions → Attach to VPC and attach it to the Development Network VPC, enabling internet connectivity for the public subnet



Click on create

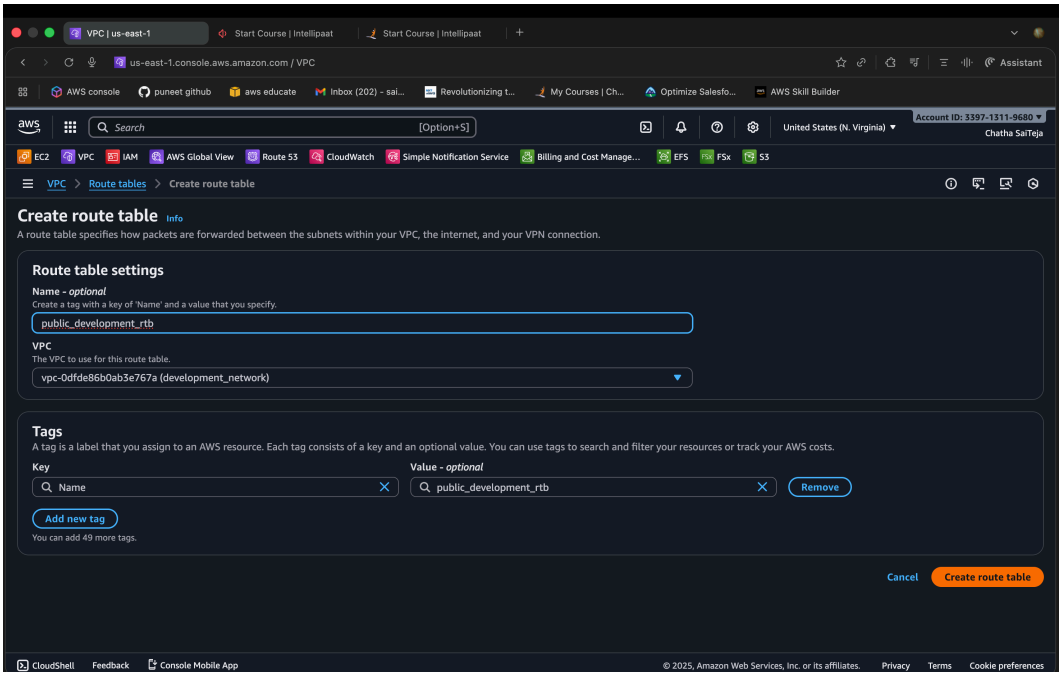


Click on attach to vpc

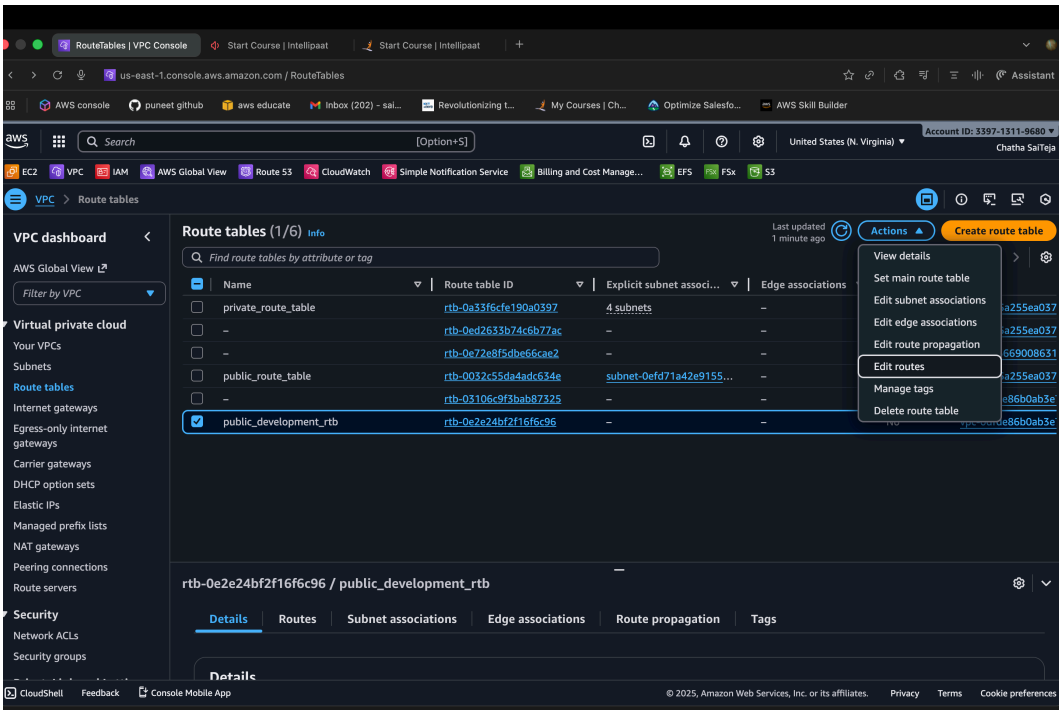


In Attach to Vpc mention -> development Network

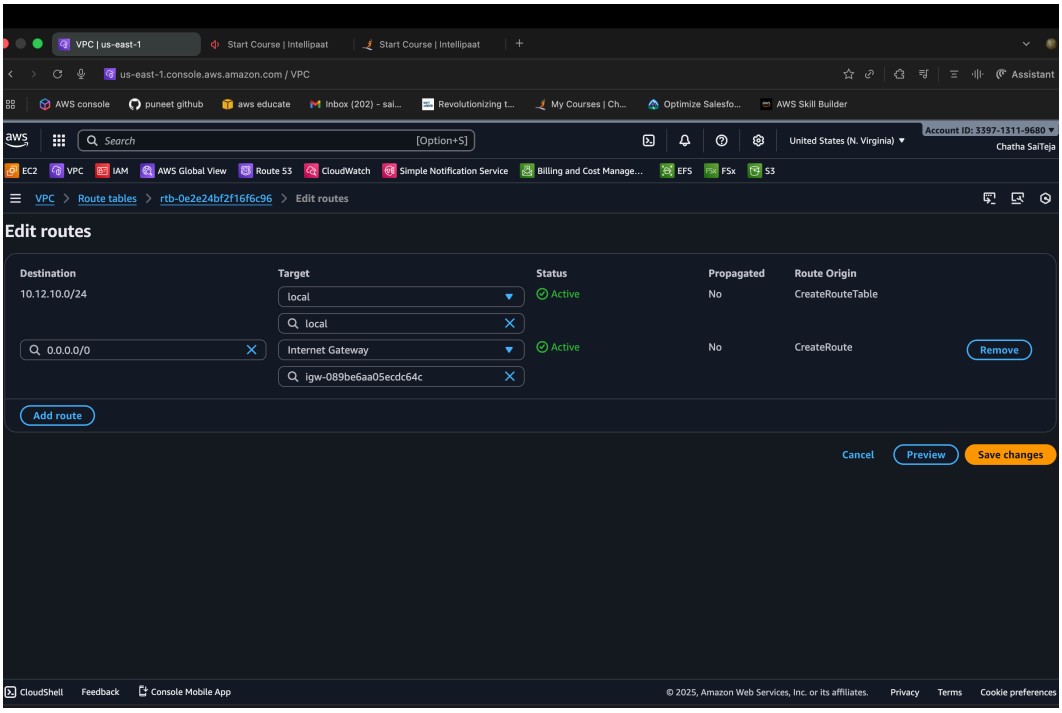
Step 4 :- When subnets are created, a route table with only local routes is available by default, but we still need to create separate public route table; each new route table initially contains only local routes, so for the public route table we must add a default route pointing to the Internet Gateway to allow the public subnet to send traffic to the internet



Create public route table In development network

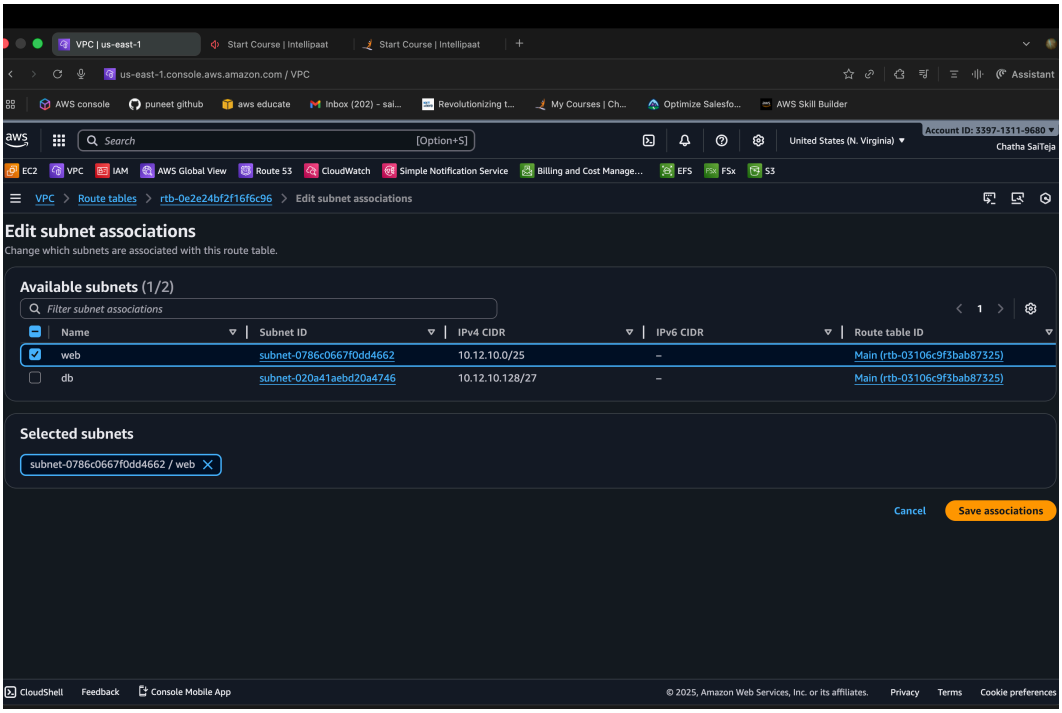


Click on edit routes



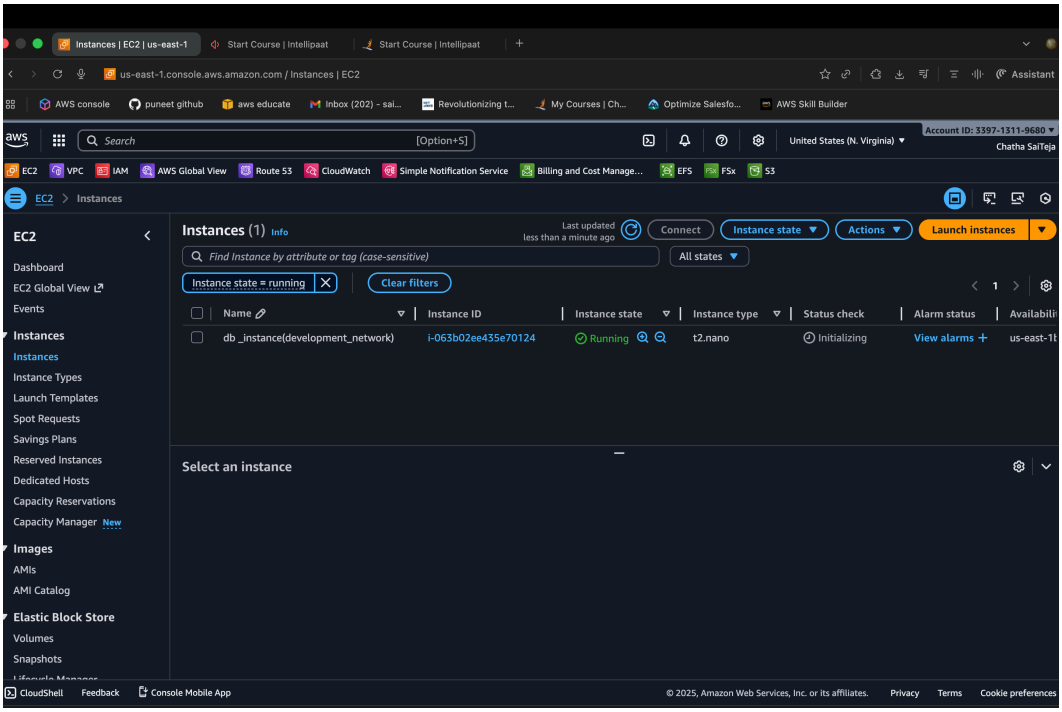
Add internet gateway route to public route table

Step 5:- After creating the route table (public), we need to update the subnet associations .

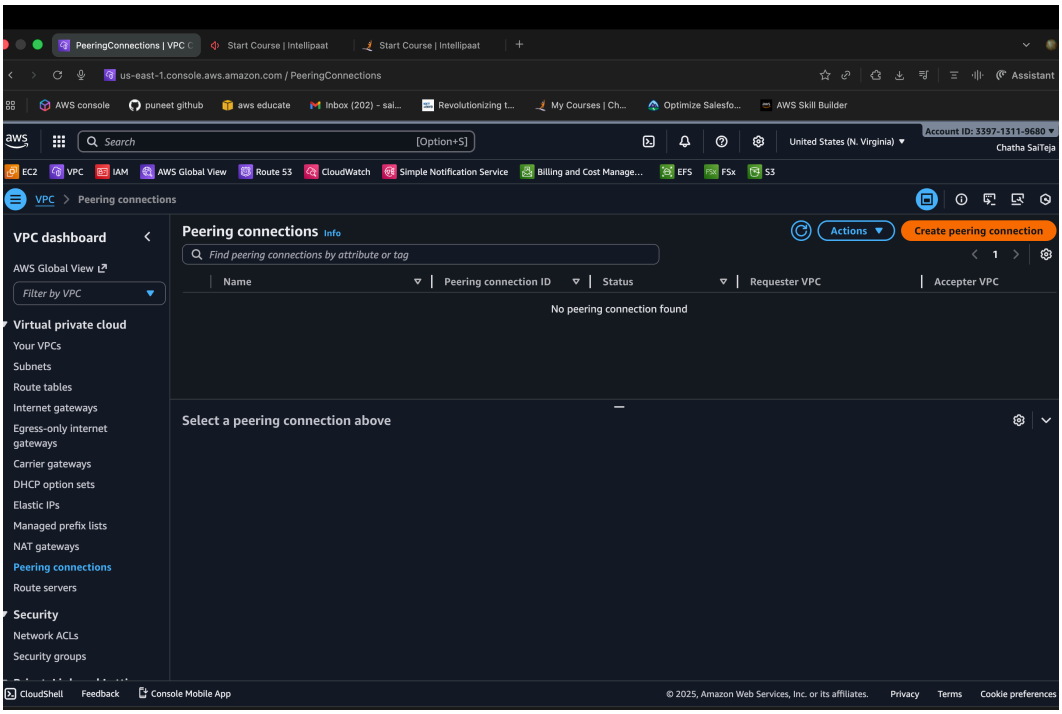


Select web and add save associations

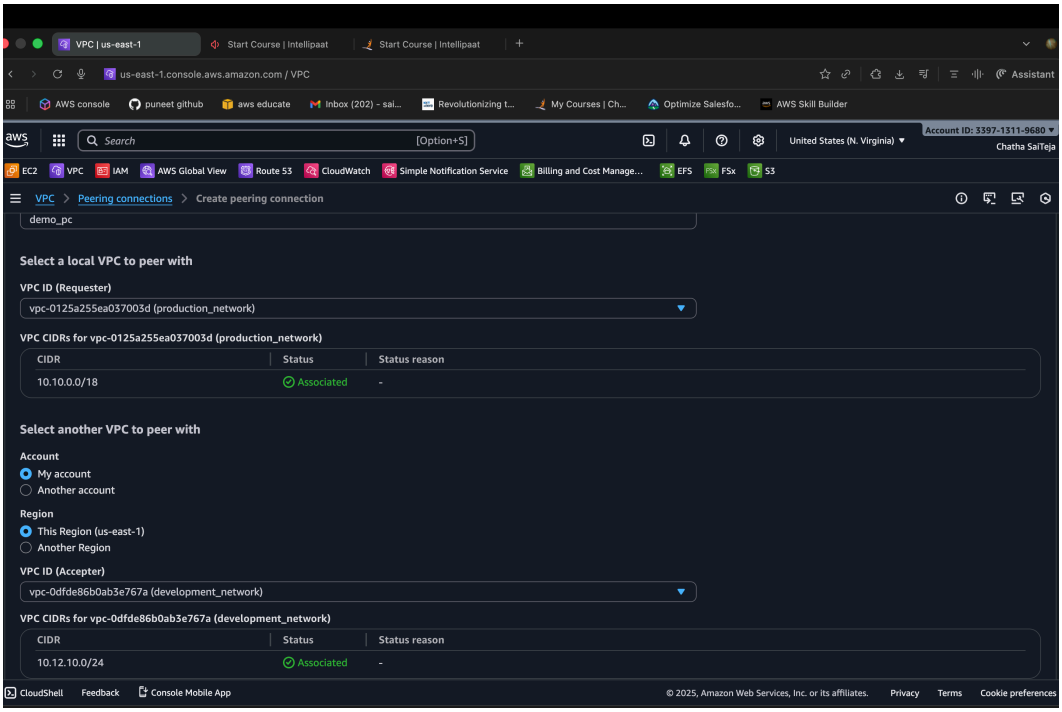
Step 6:-Now we have to launch a instance in development network in db subnet which is a private subnet .



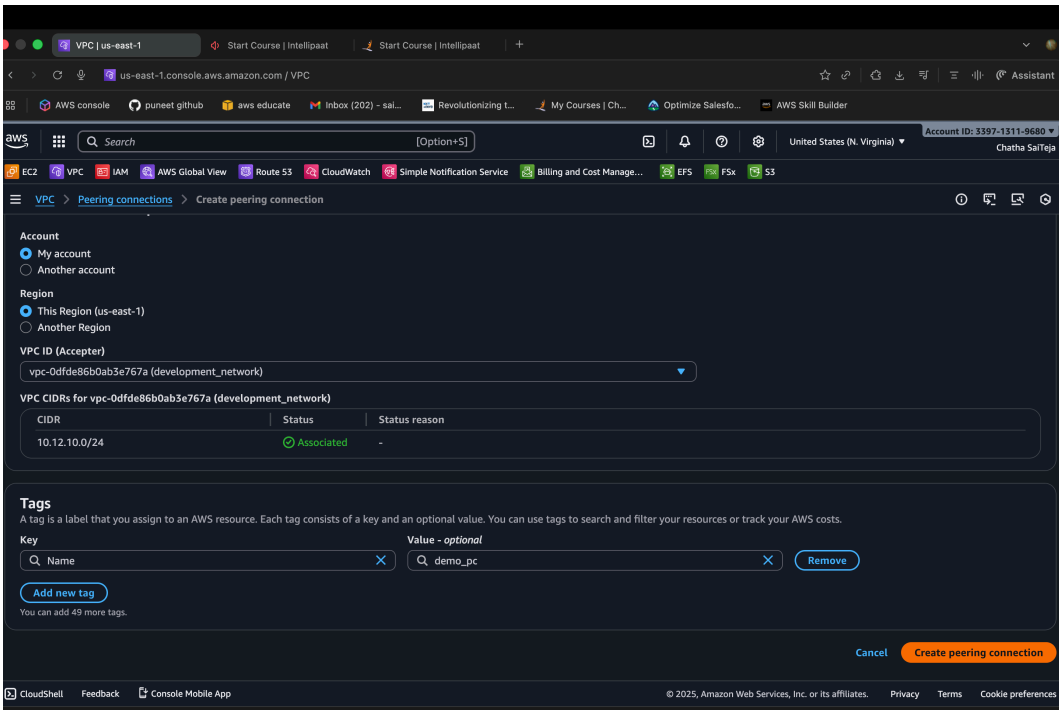
Step 7 :-To connect the Production and Development networks, create a VPC peering connection by specifying the requester and accepter VPCs, whether they're in the same region or another, and whether they belong to the same account or different accounts. After the peering connection is created, go to the accepter VPC and approve the request by selecting **Accept Request**.



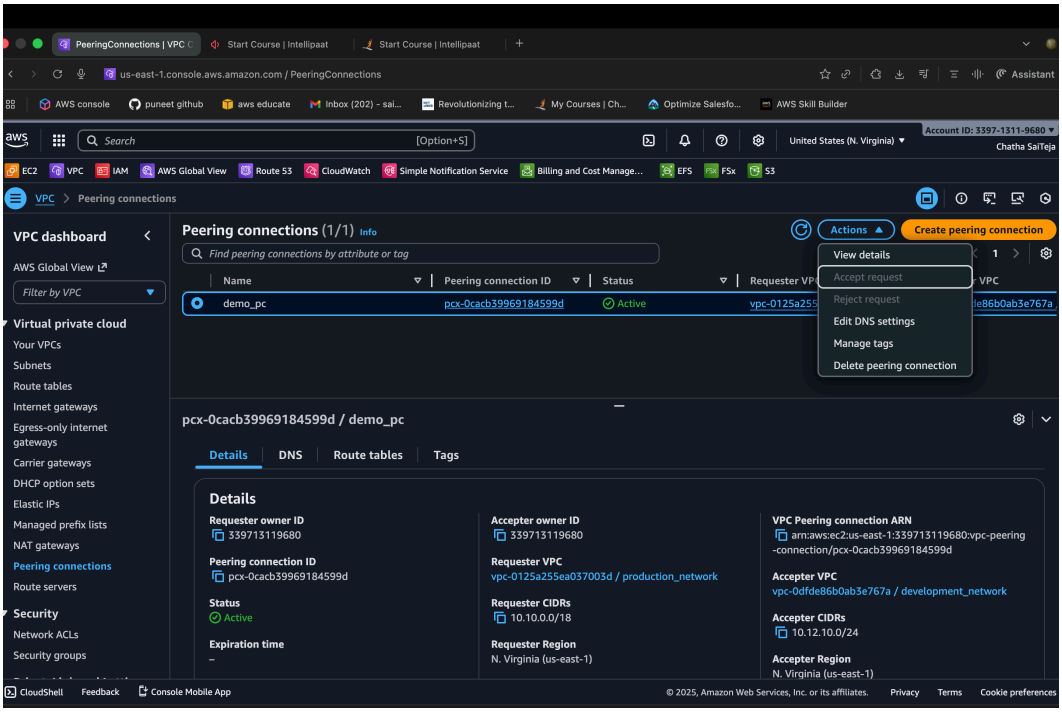
Click on create peering connection



Specify the details

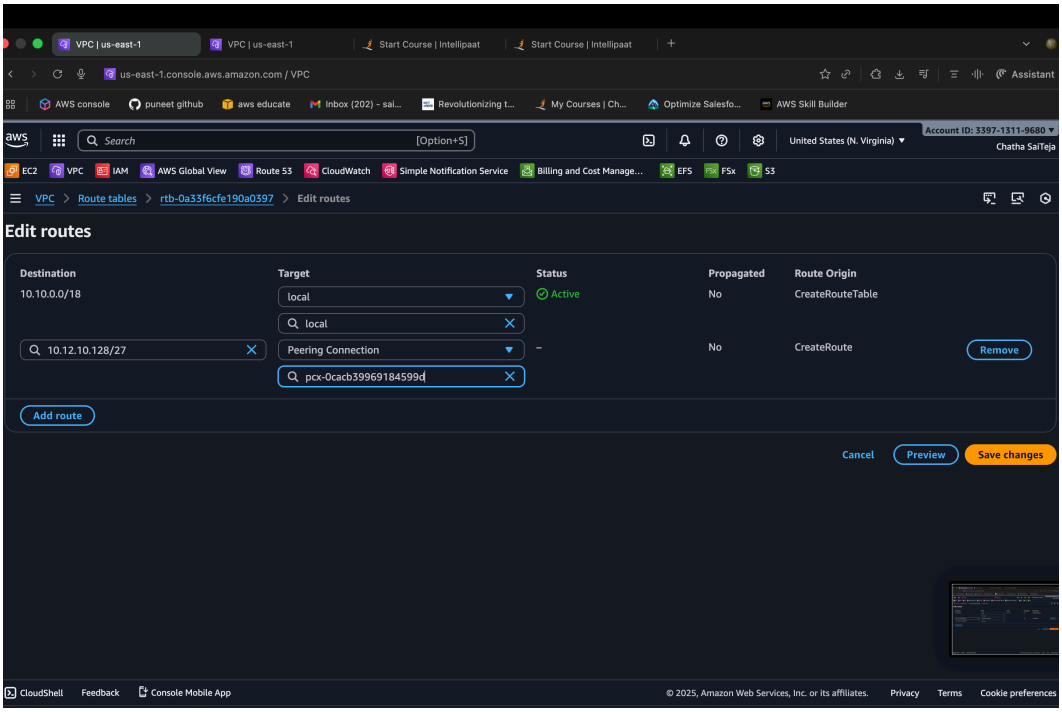


Click on create



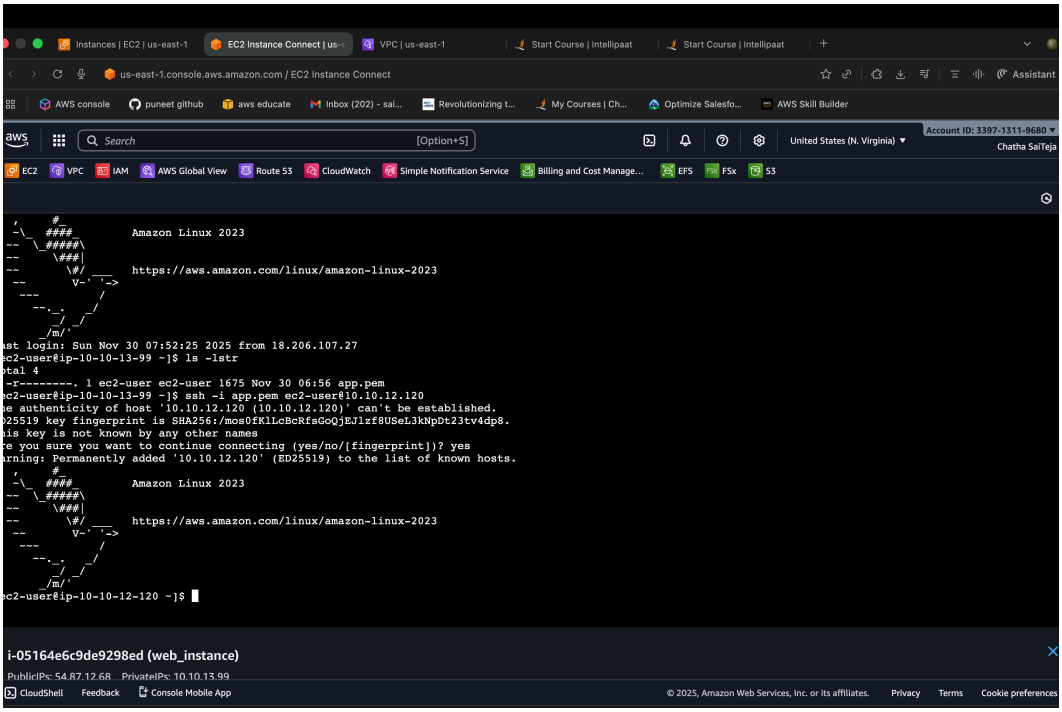
Click on Accept request

Step 8:- Now we have to edit route tables (private) of both production and Development network so that db subnet in production network can communicate with db subnet in development network.

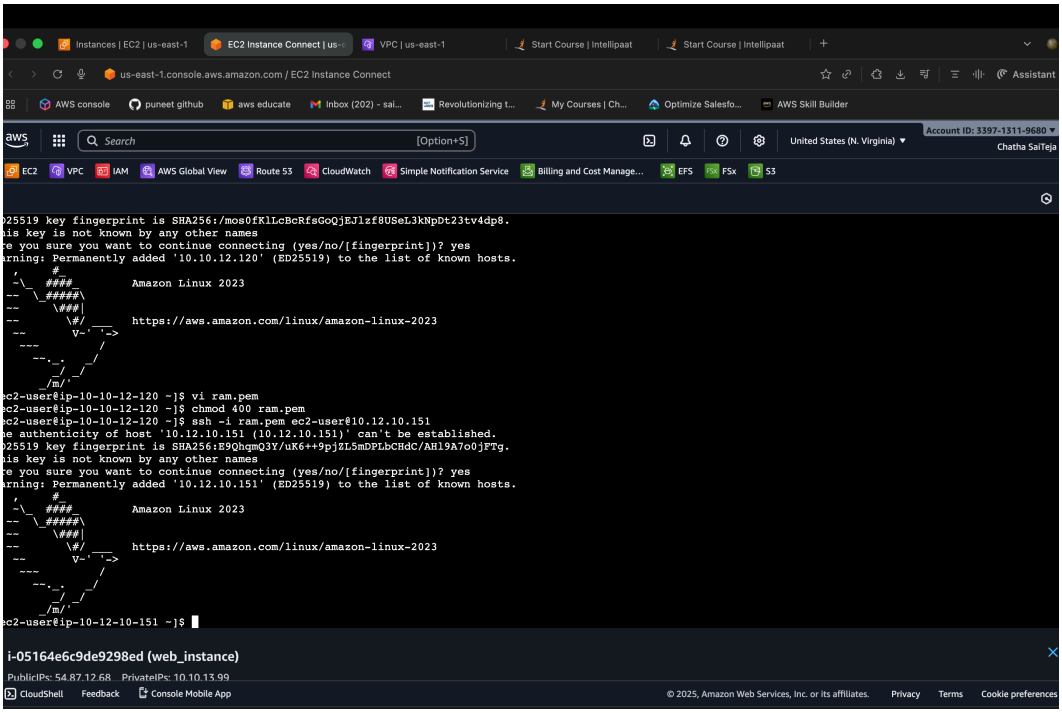


Add pcx connection and give destination of production network subnet it could be done for development network subnet also.

Step 9:- Now it can connect with subnet in other network so that first we connect the web instance in production networks and perform ssh connectivity to connect a private instance (db instance) from that instance to db instance in development network .



Connected db instance (production network) through web Instance



dbinstance (production network) connected to db instance in development network by peering connection between two vpc's