# Project-1- Deployment of 3 Tier Applications Using AWS.

## Submitted By – Arya Chaurasia

## Date of Submission – 17/12/2025

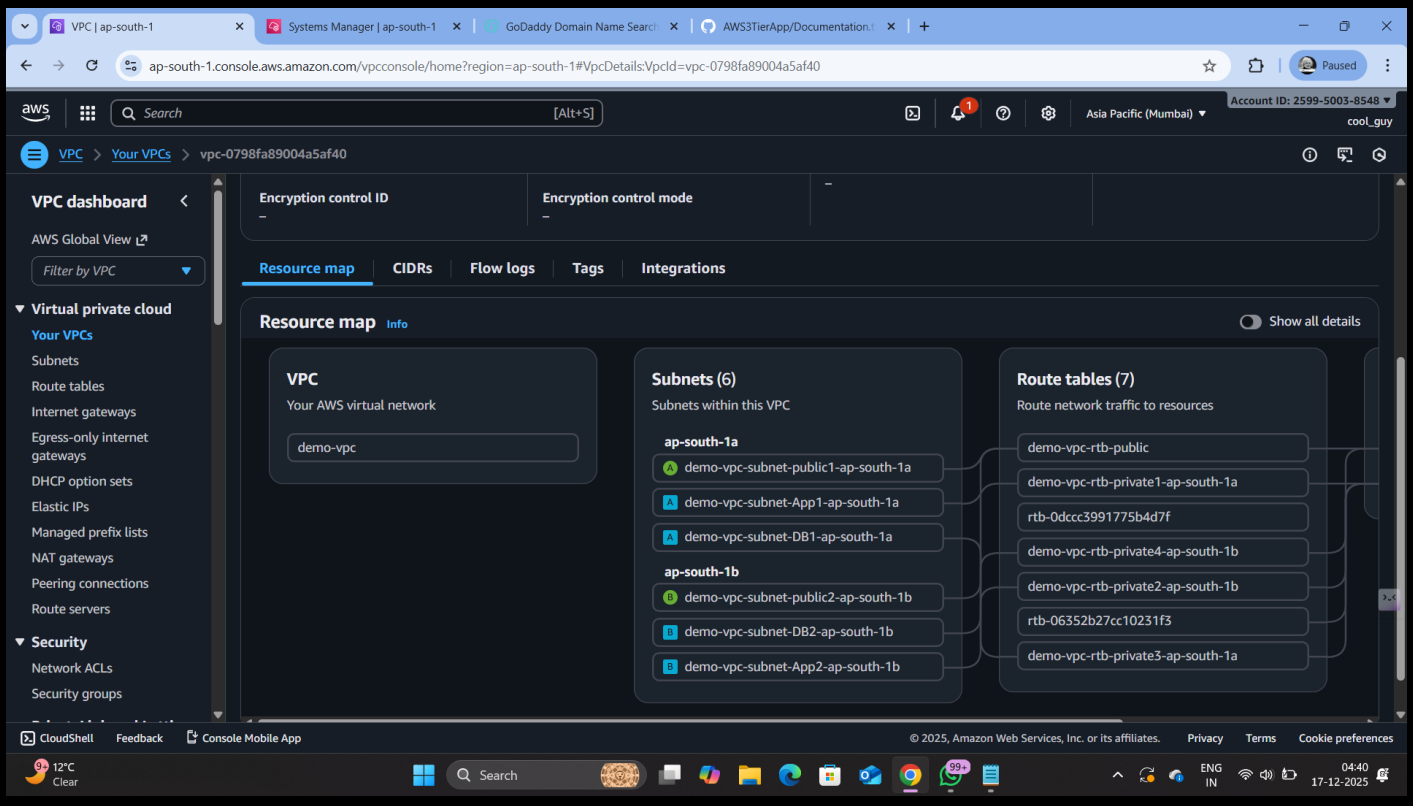
### Project Overview

This project demonstrates a manually deployed 3-tier architecture on AWS for a web application where users can submit data, which is then stored in a backend database. The setup does not use DevOps tools (like Jenkins, Docker, or Kubernetes) and relies on AWS native services for hosting, networking, and storage.

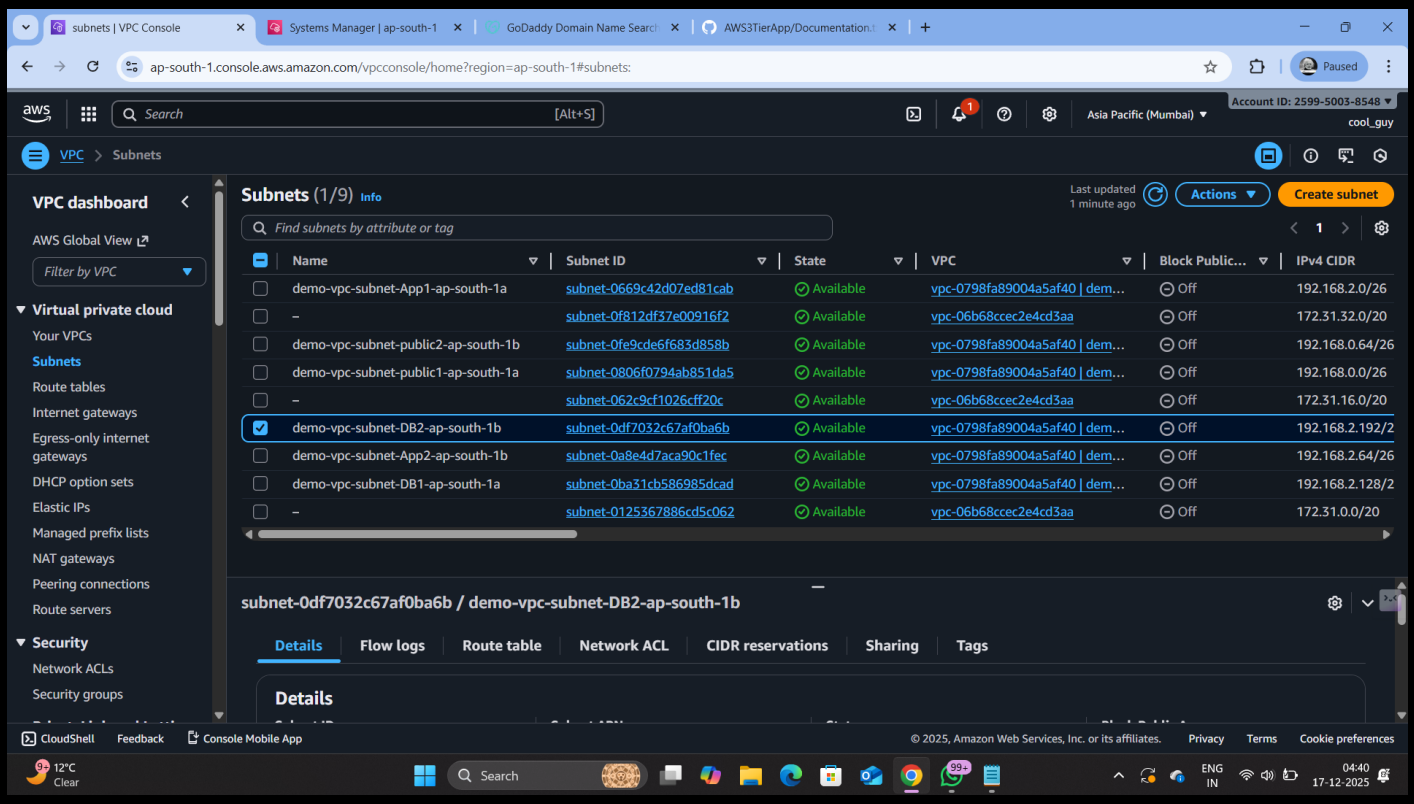
GitHub Link- https://github.com/Aryachaurasia23/AWS3TierApp.git

## Steps to Deploy the Website successfully

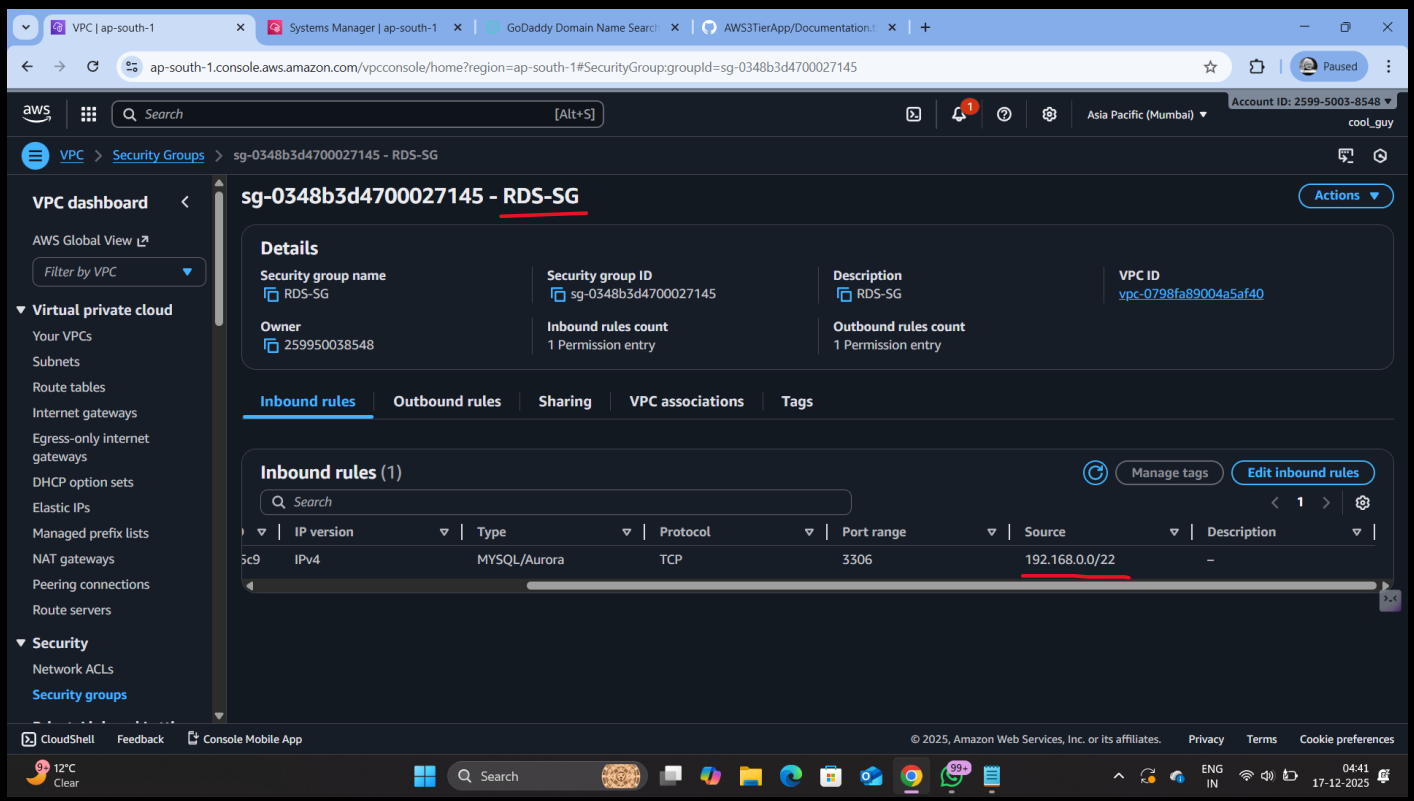
Step-1- First we created the VPC and we will select the option “VPC & more” which we will create the Subnets, Route tables, Network connections.



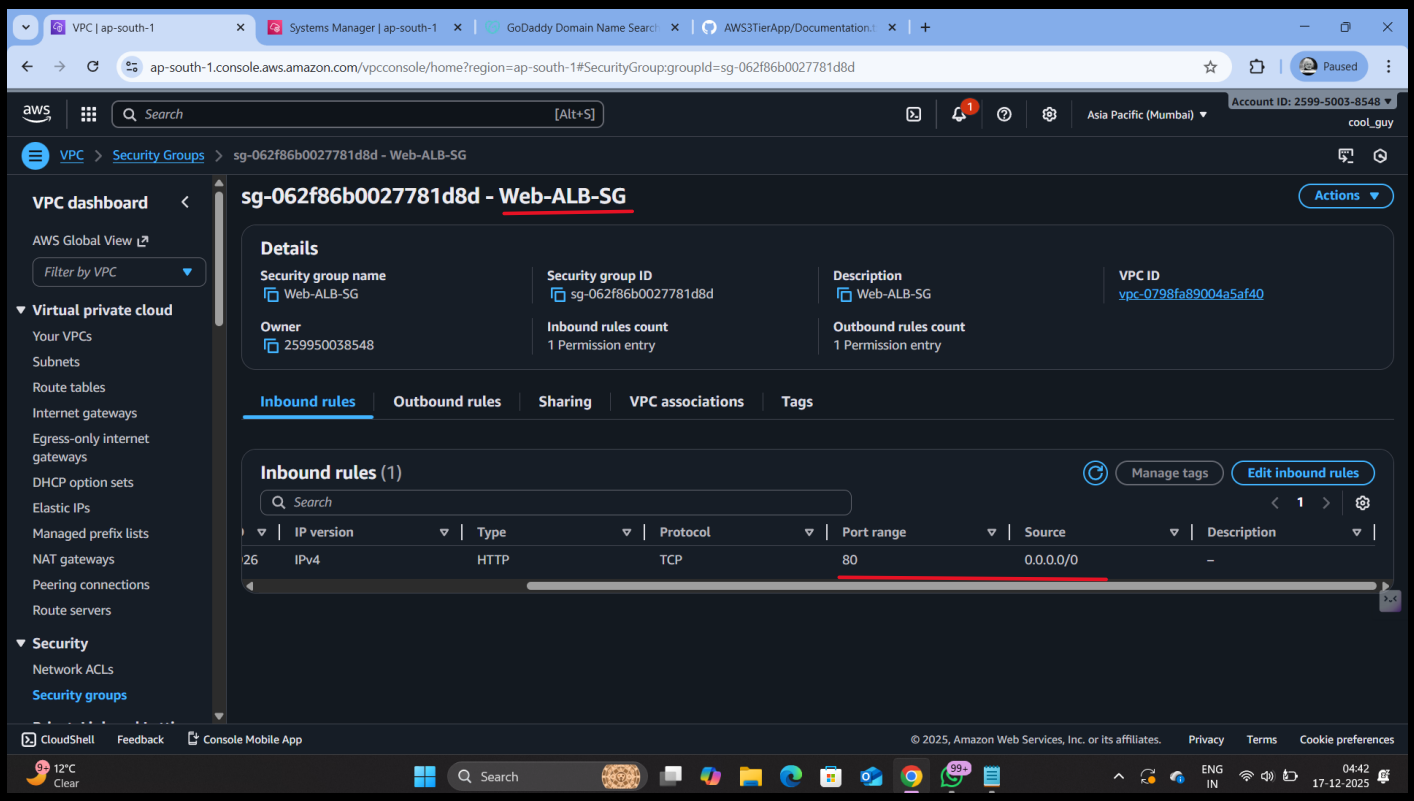
Step-2- Now we name our Subnets Public subnets – {Public-1, Public-2} , Private Subnets {App-1, App-2, DB-1, DB-2}



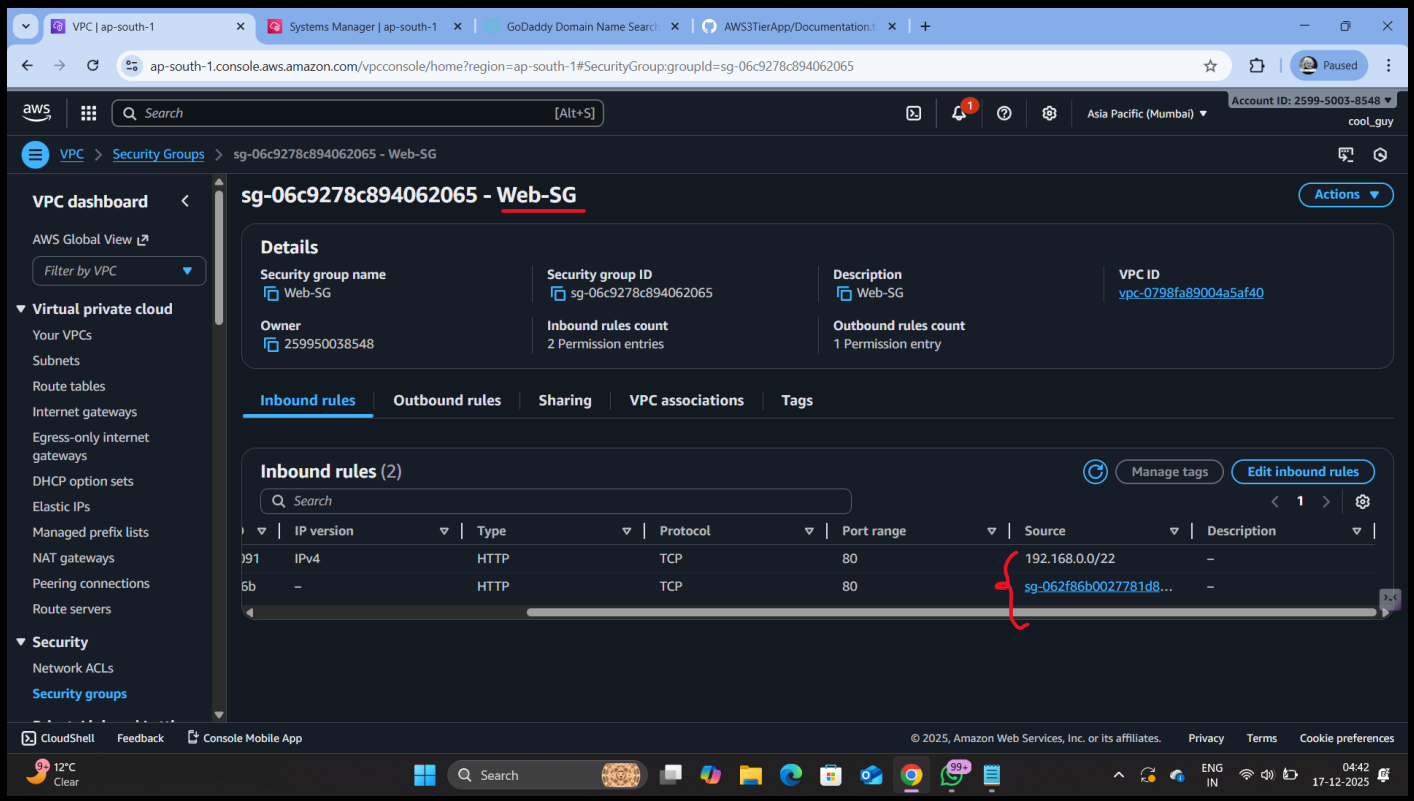

Step-3- RDS-SG has enabled IPv4 CIDR block.



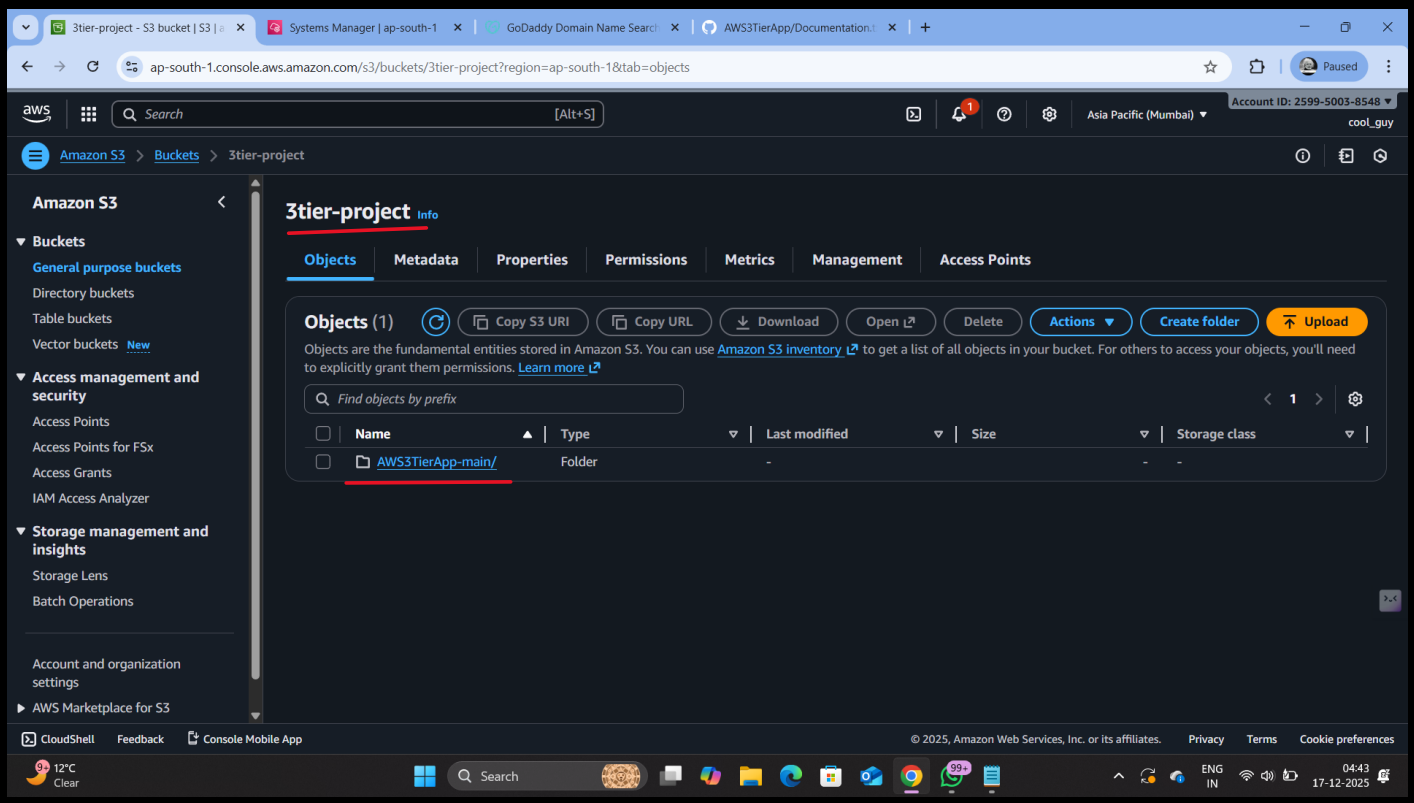
Step-4- Only Web-ALB-SG has all traffic enabled rest all other security group has enabled IPv4 CIDR block.



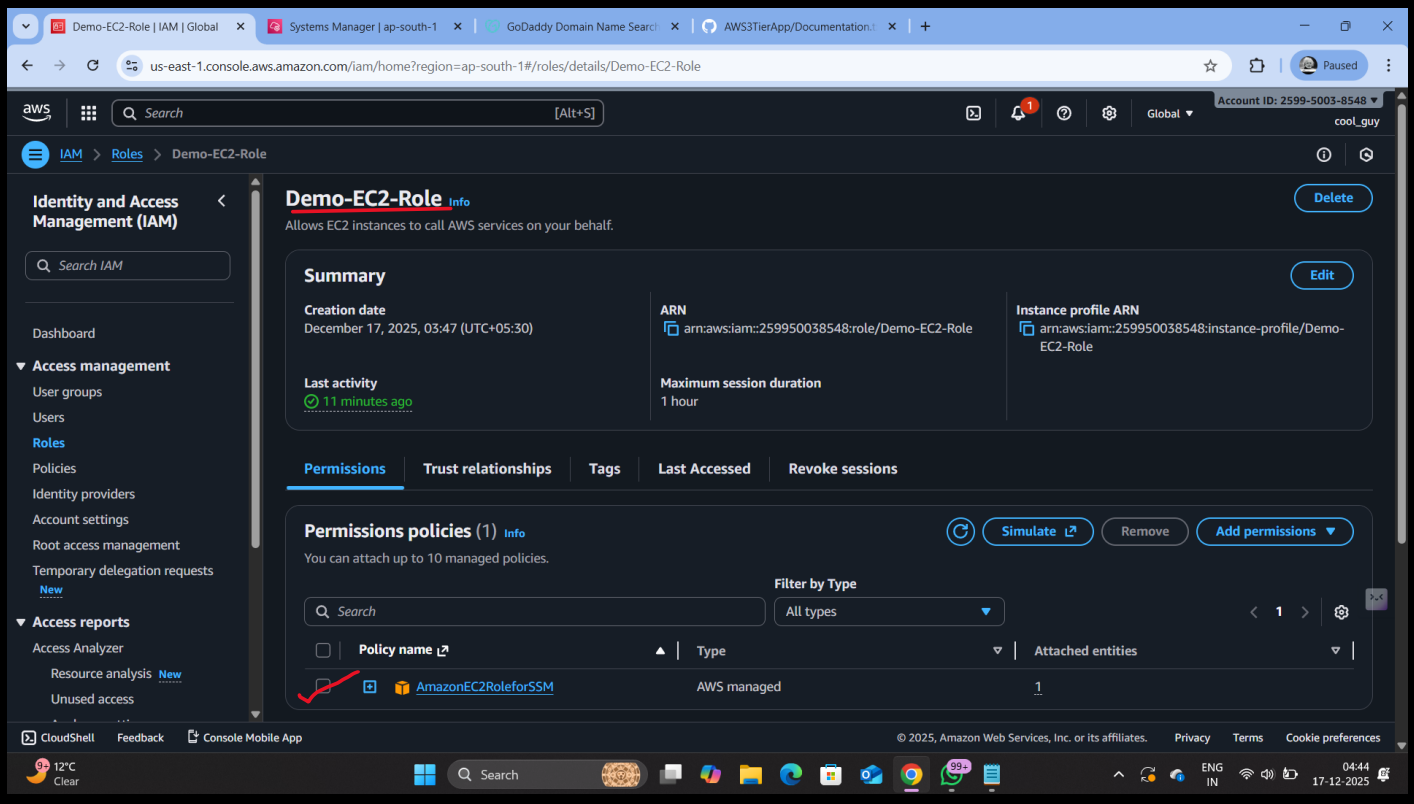

Step-5- In Web-SG we have enabled IPv4 CIDR Block.



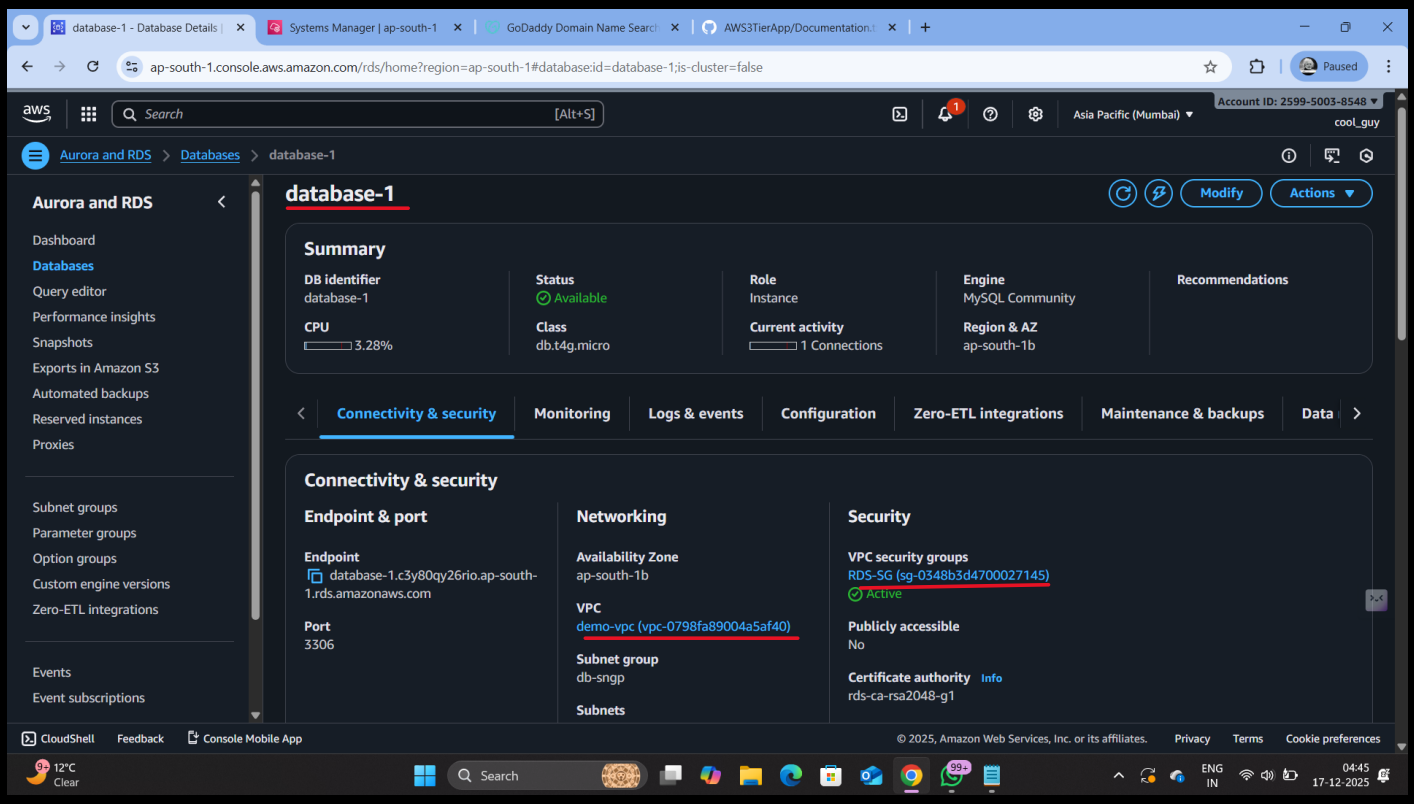
Step-6- Now we made a S3 bucket to upload our application code here.



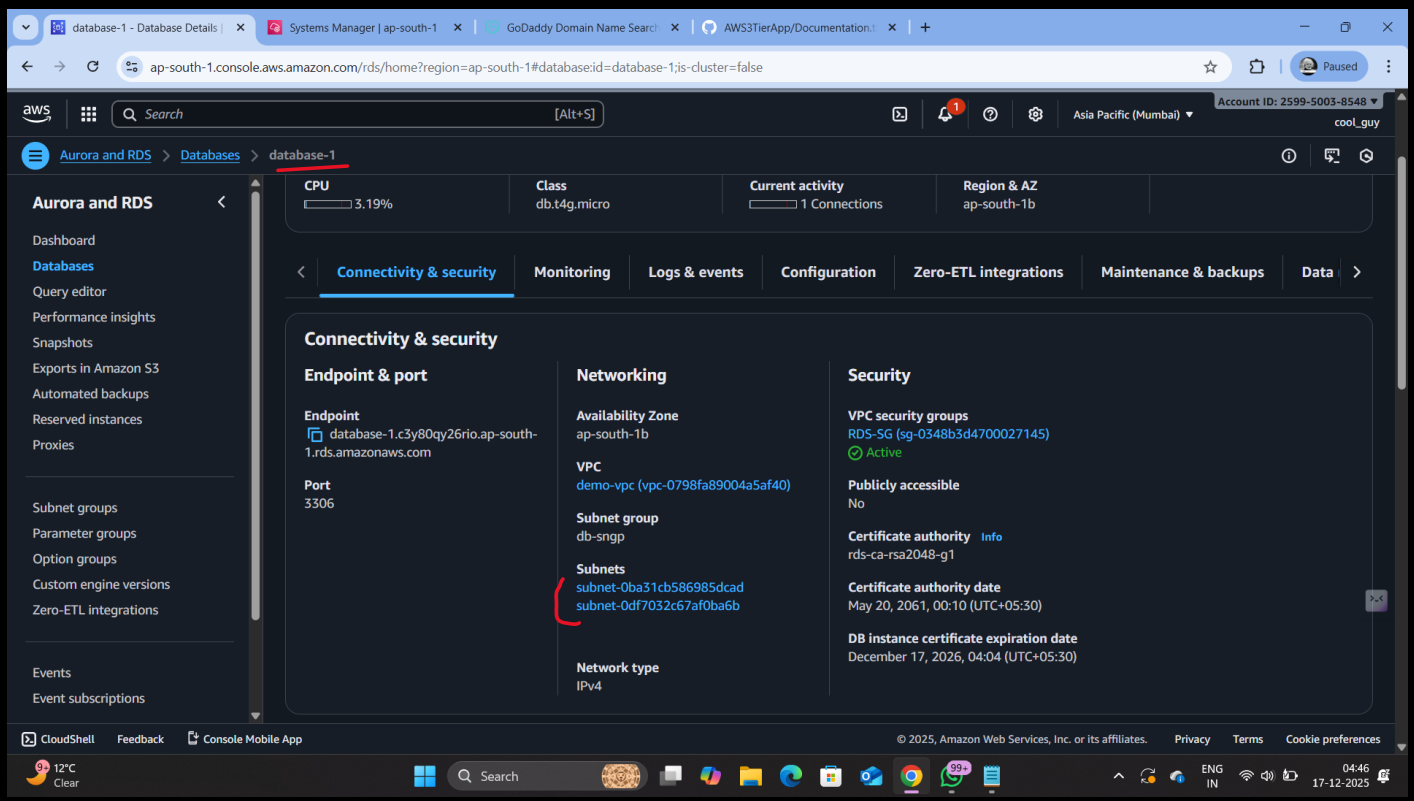

Step-7- We created a role and attached AmazonEC2RoleforSSM.



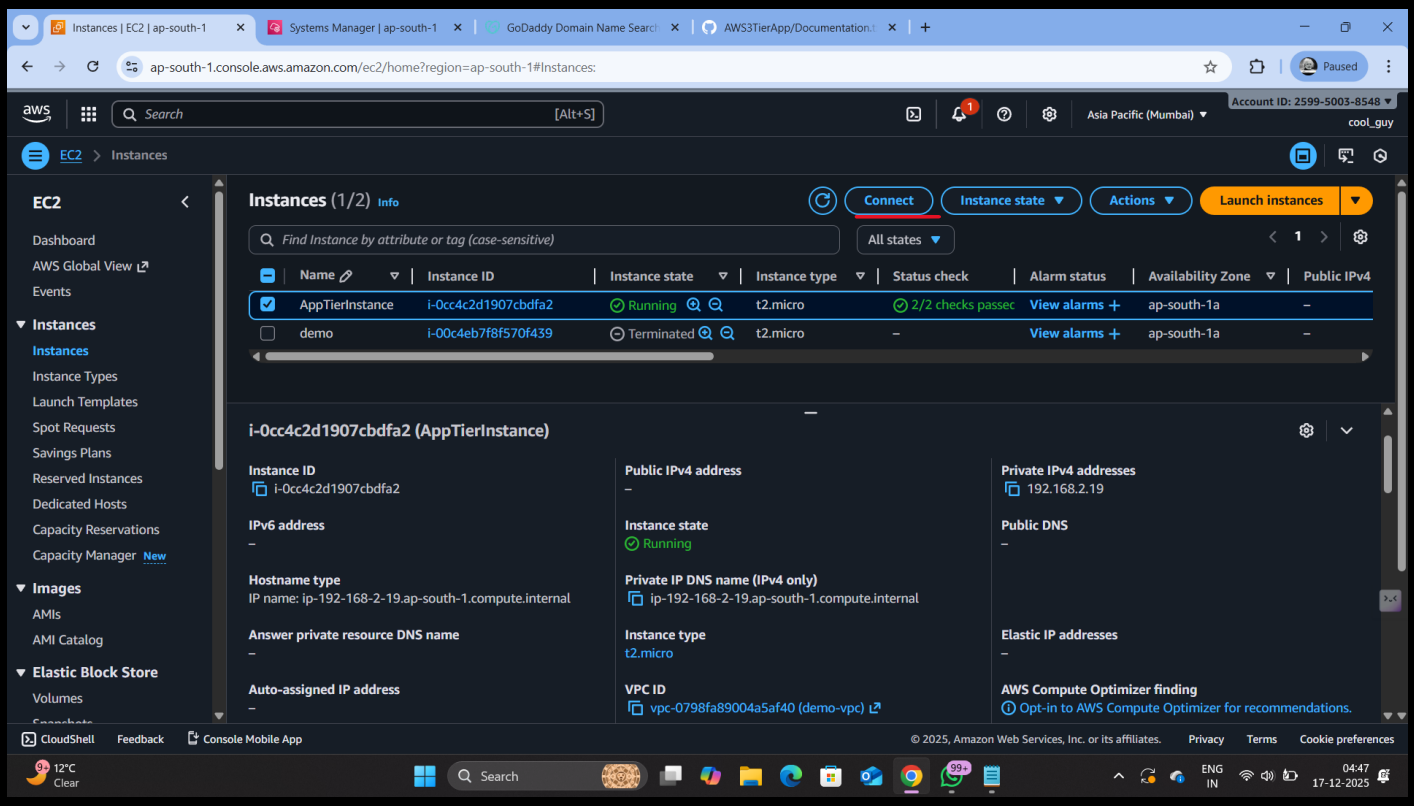
Step-8- We created a RDS database named as database-1 with RDS-SG security group attached and in our private VPC (demo-vpc).



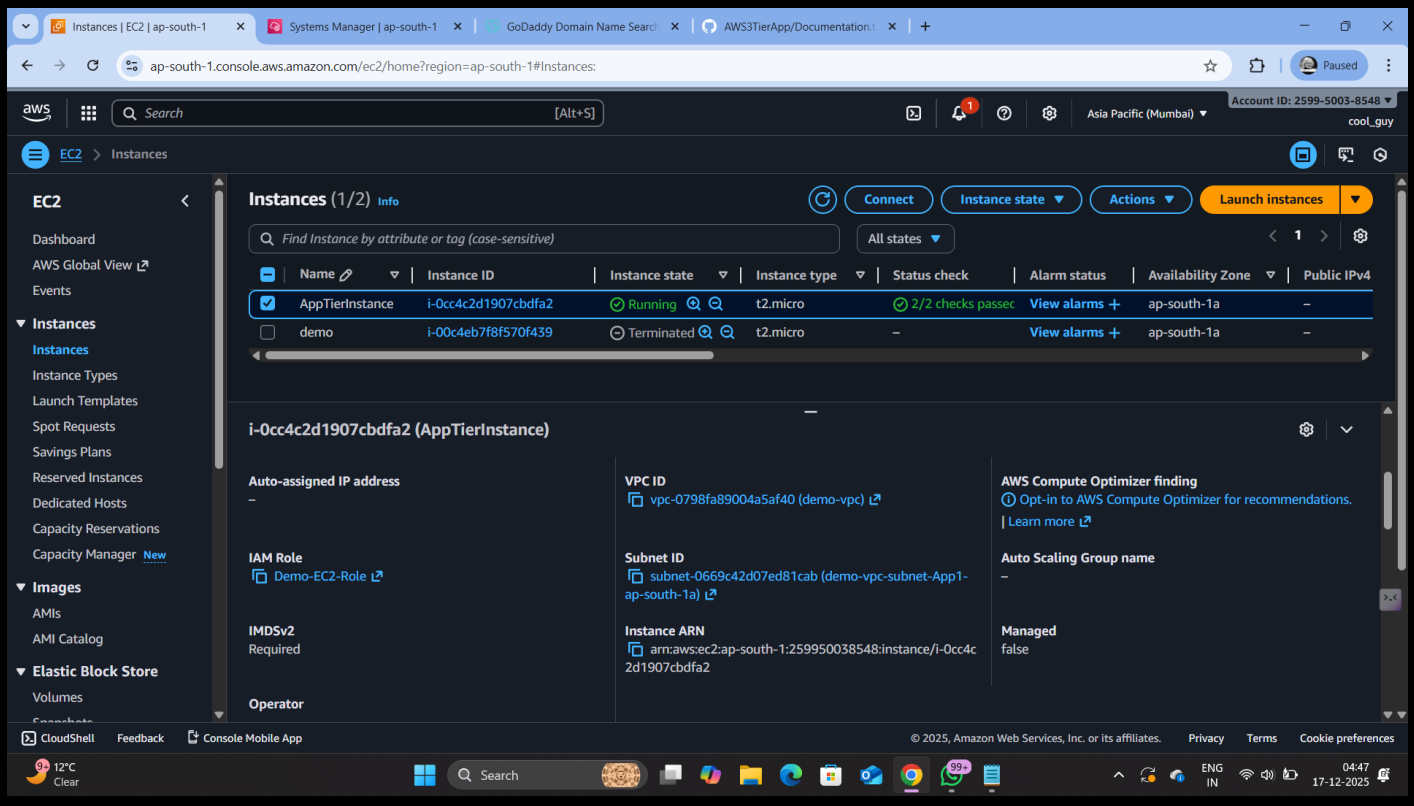

Step-9- This is the configuration pf the database-1



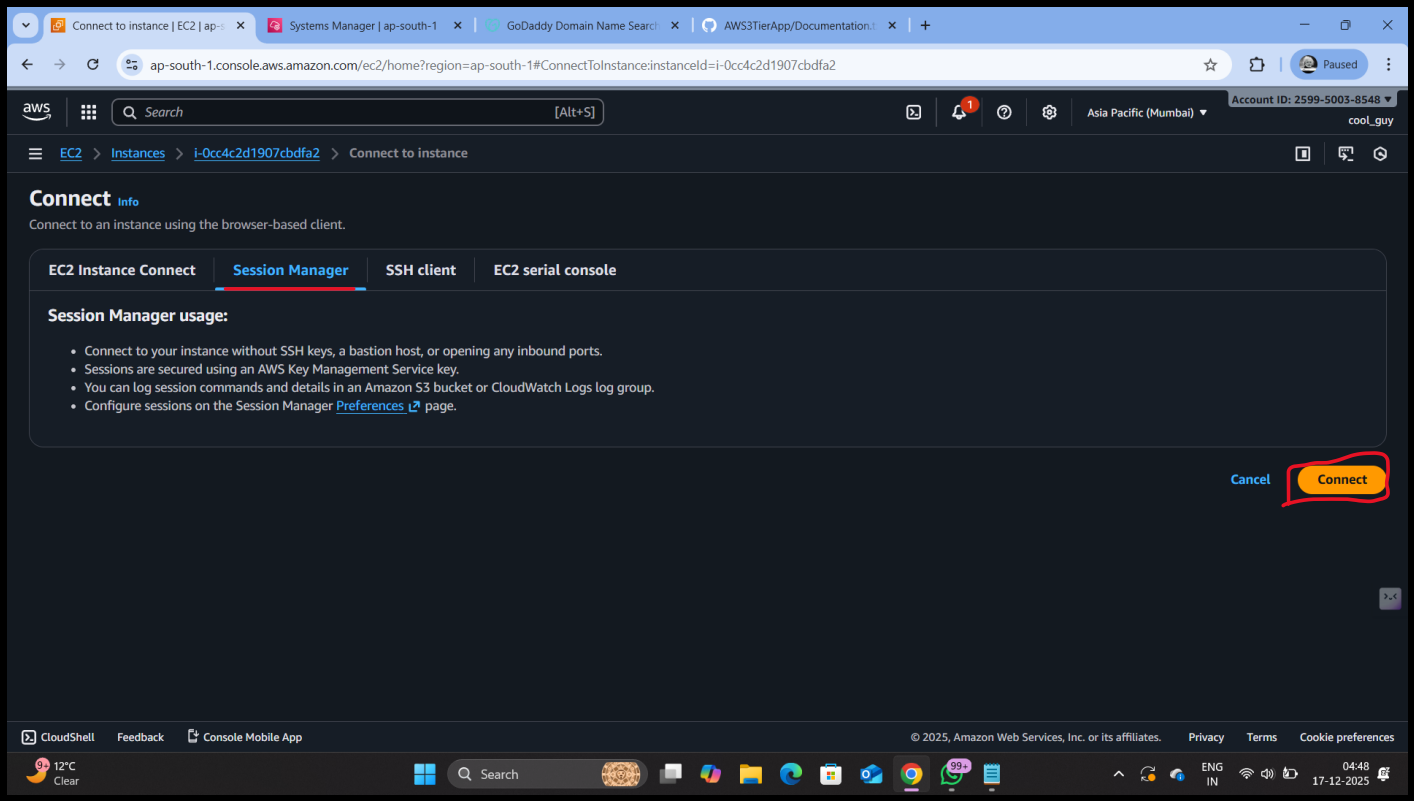
Step-10- Now we created an EC2 Instance (AppTierInstance) and connected with Session manager



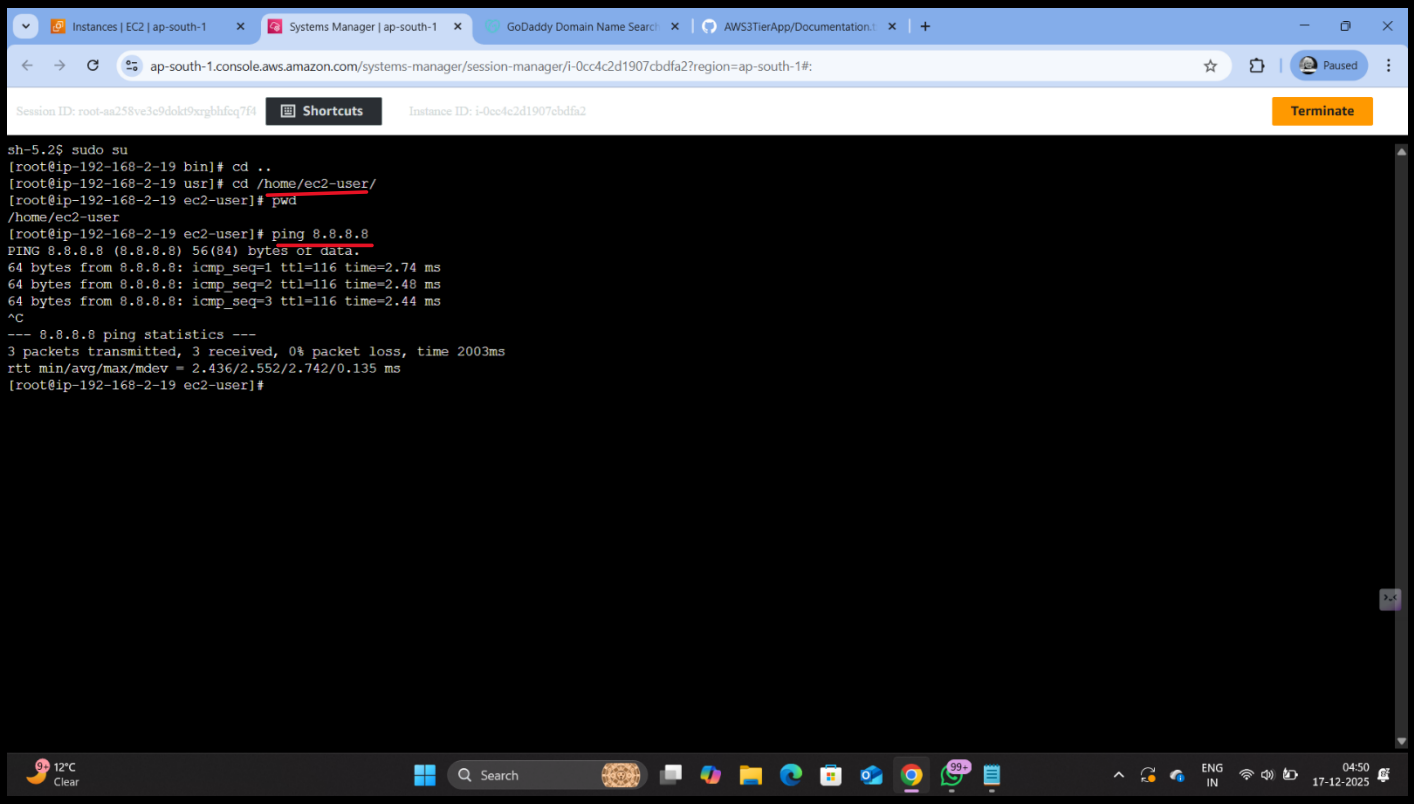

Step-11- Here we clicked on the connect.



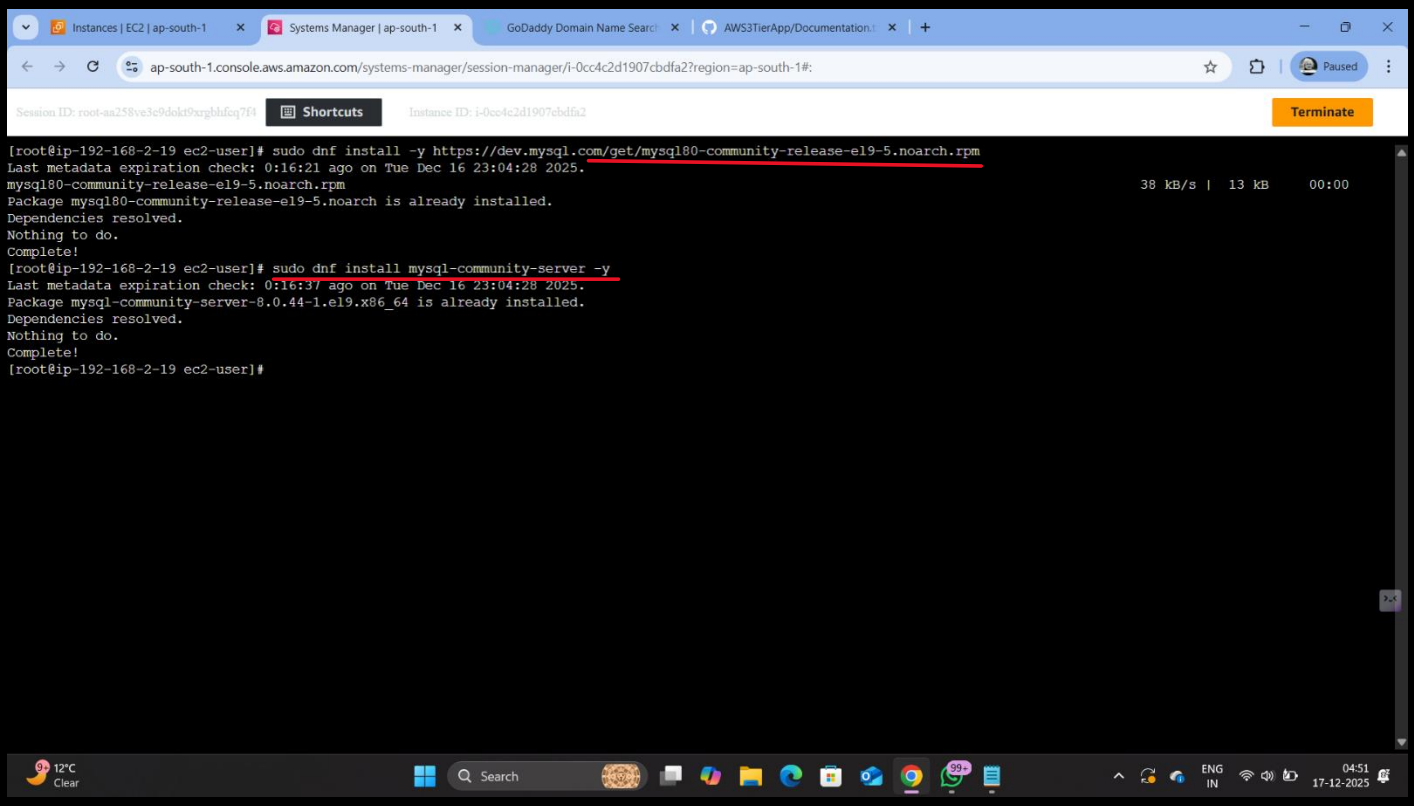
Step-12- Now we will go to the Session Manager and click on Connect.



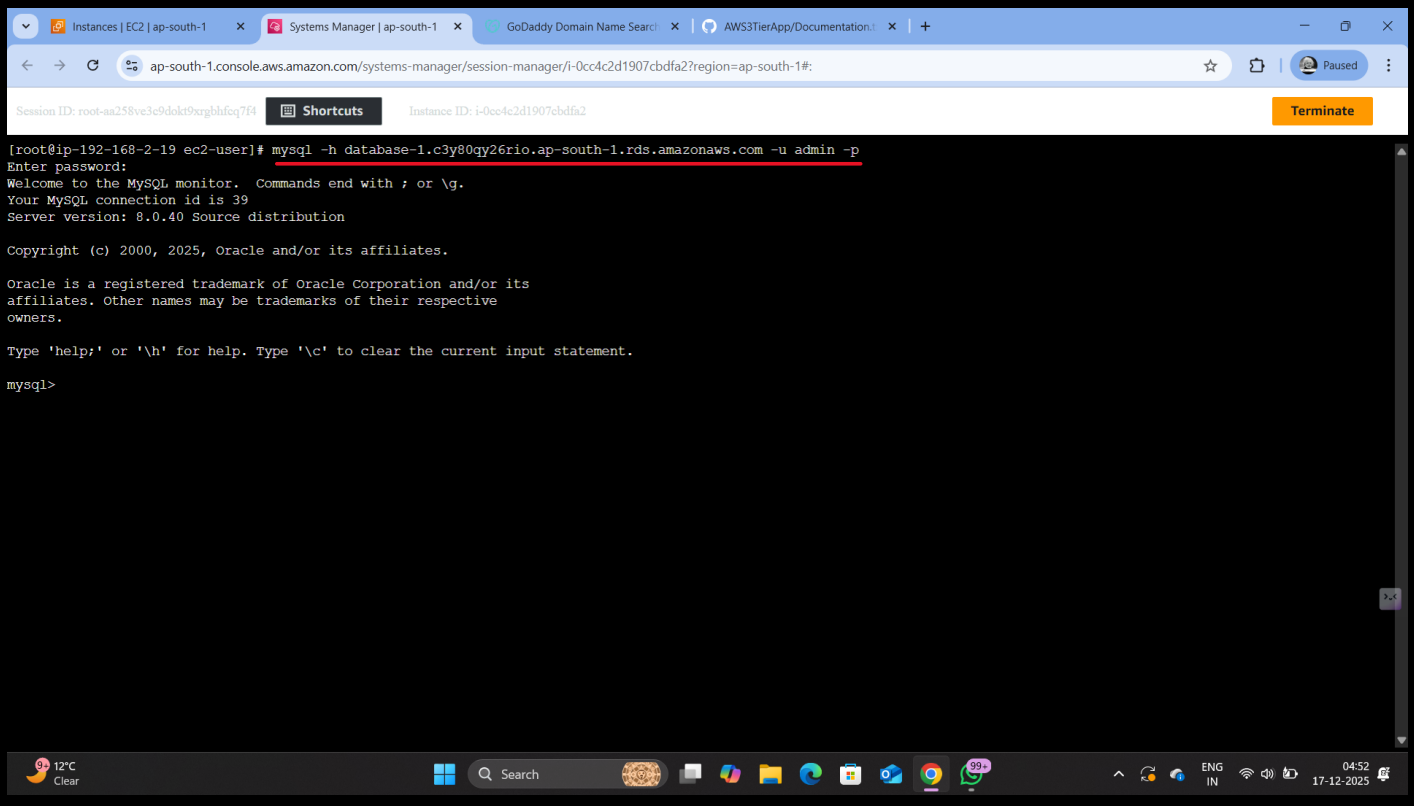

Step-13- We will go to the home directory and check weather the internet is accessible or not.



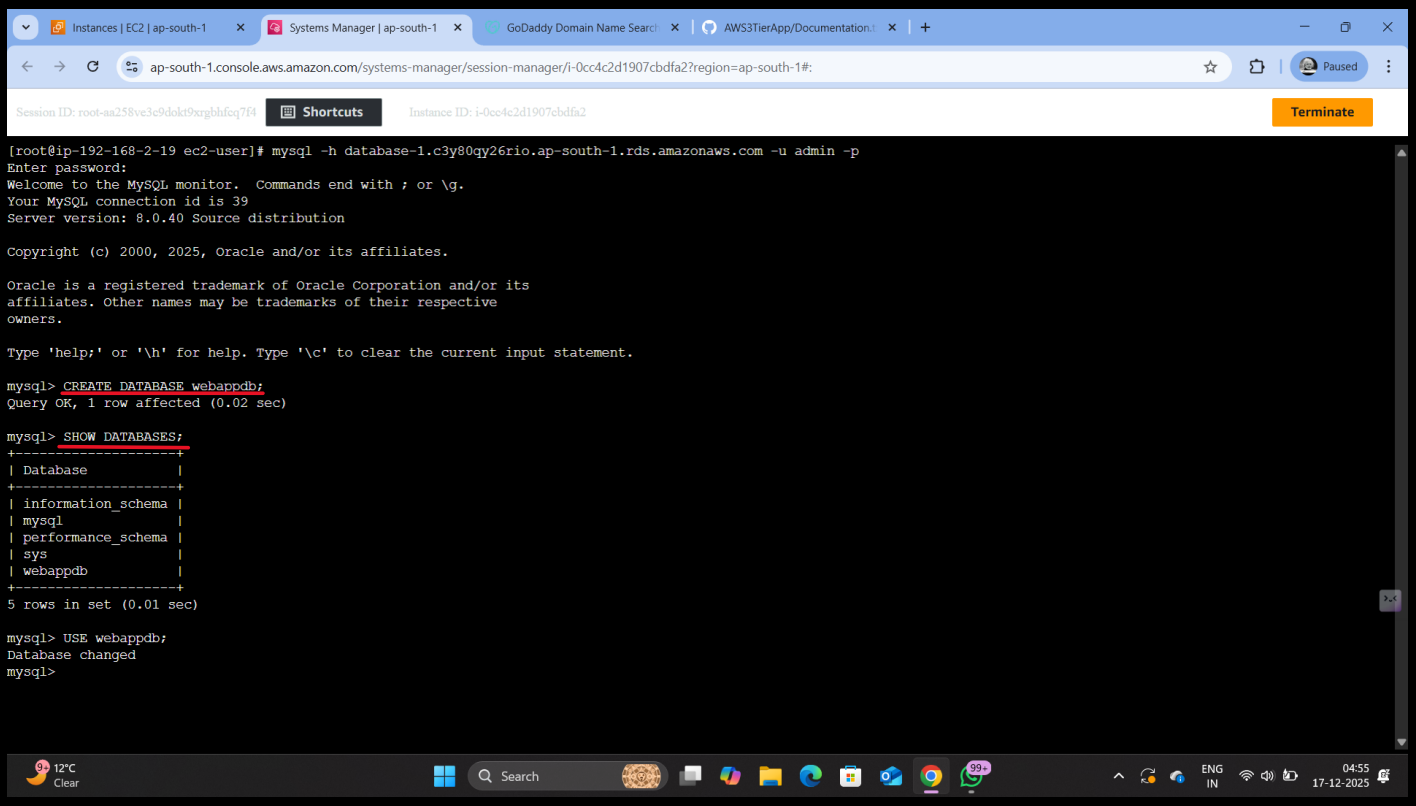
## Step-14- Now we installed MySql in the server.



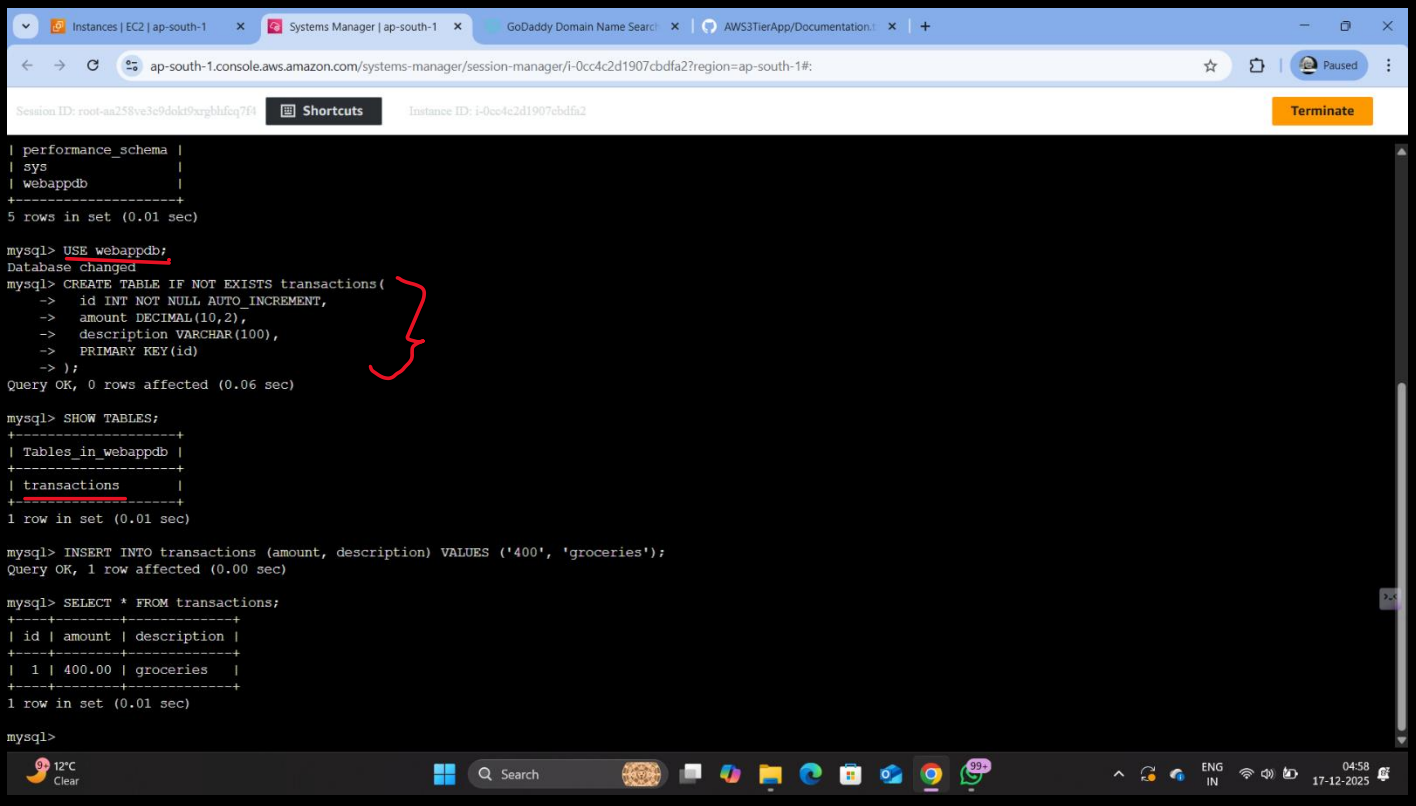

## Step-15- Now we will connect to the database and perform basic configurations.



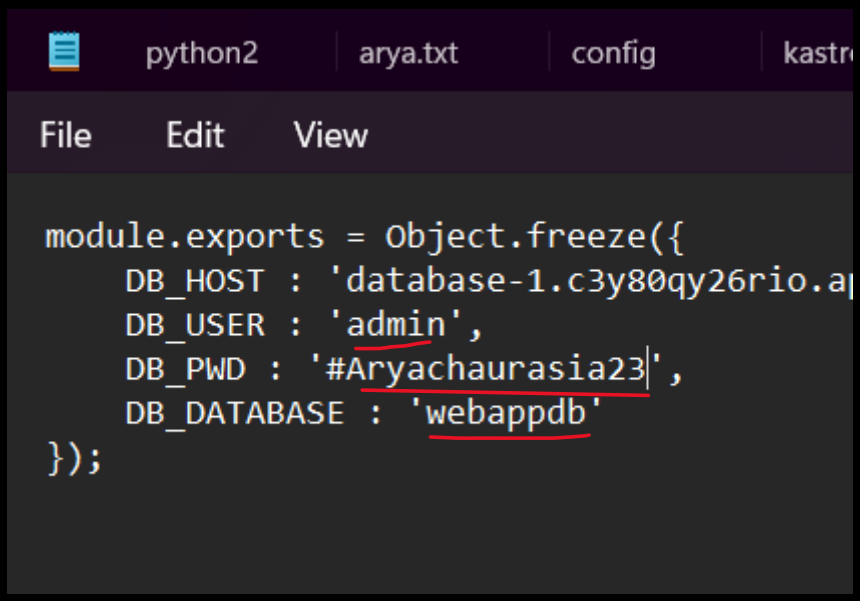
## Step-16- Now we will create a database named webappdb.



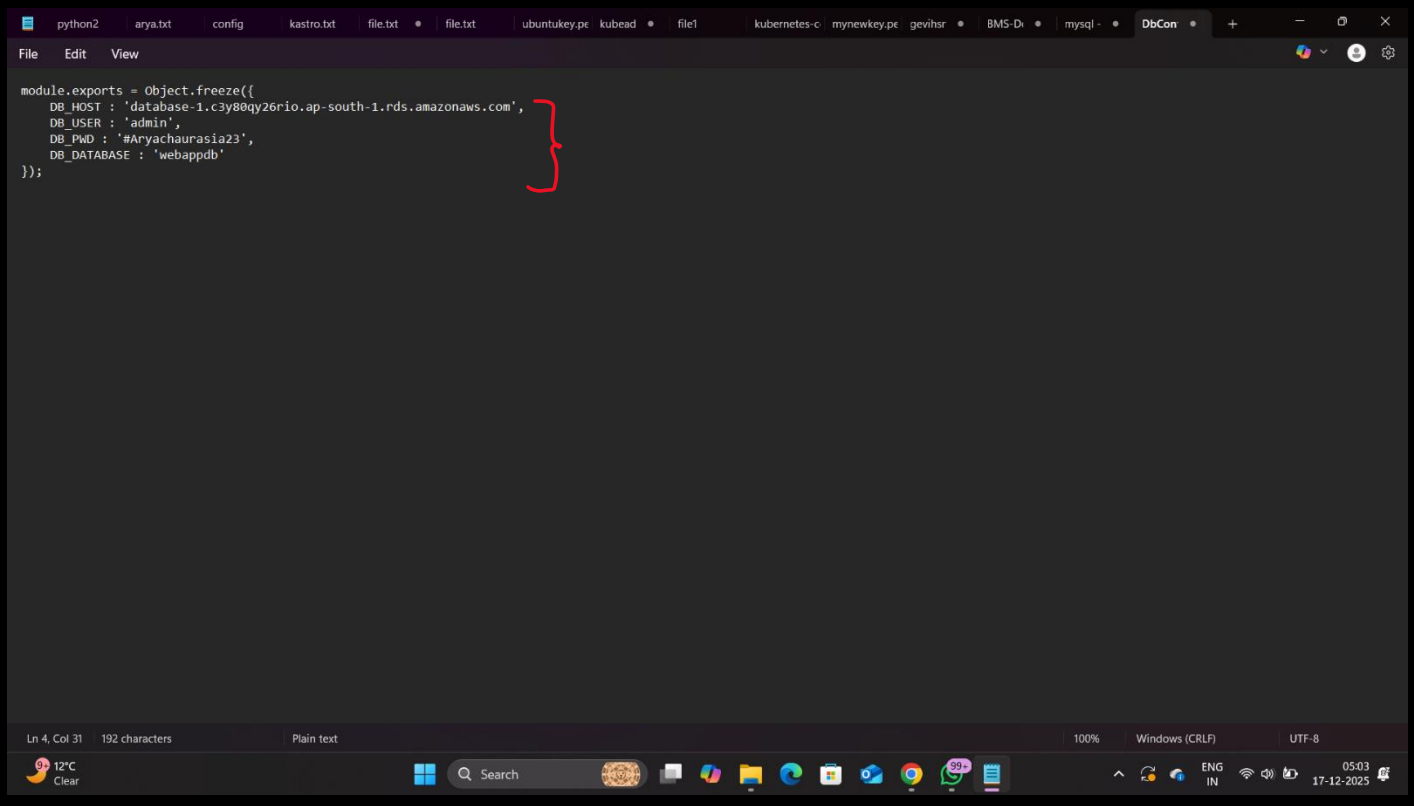

## Step-17- Here we are creating a table named as transaction.



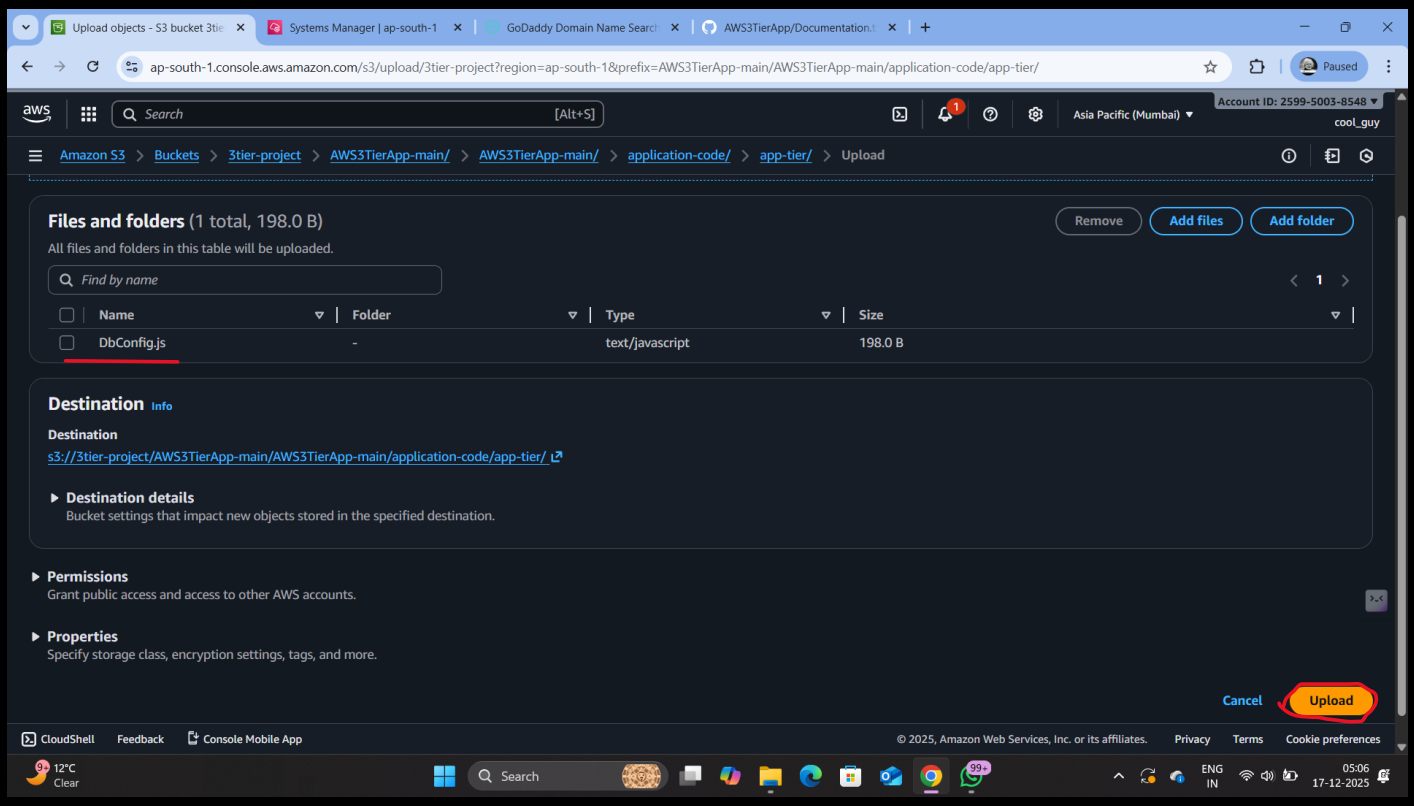
Step-18- Update Application Configuration to with DB information



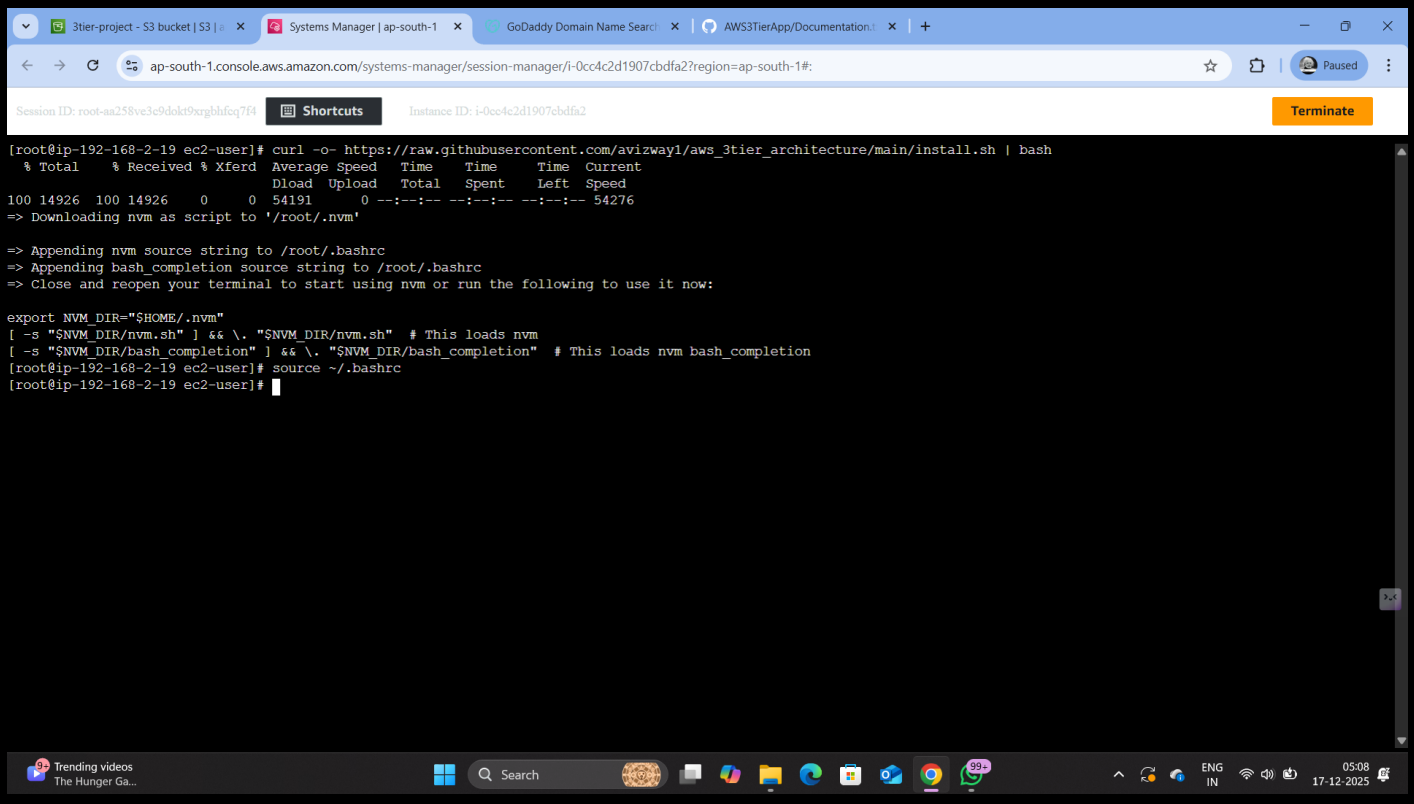

Step-19- Update DbConfig.js file with DB information and we will save it



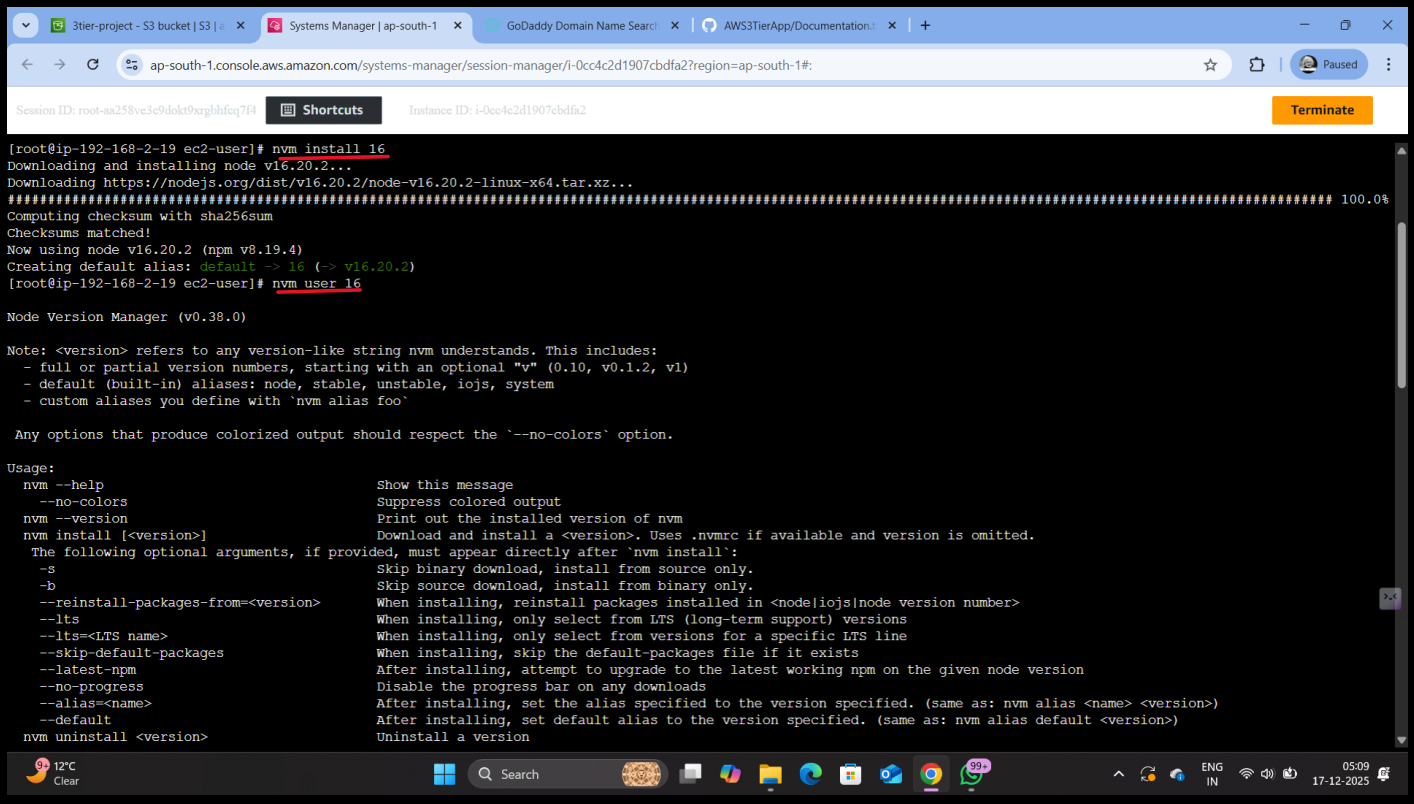
Step-20- Now we will upload the updated file in S3 bucket and the older one will be replaced by the updated one.



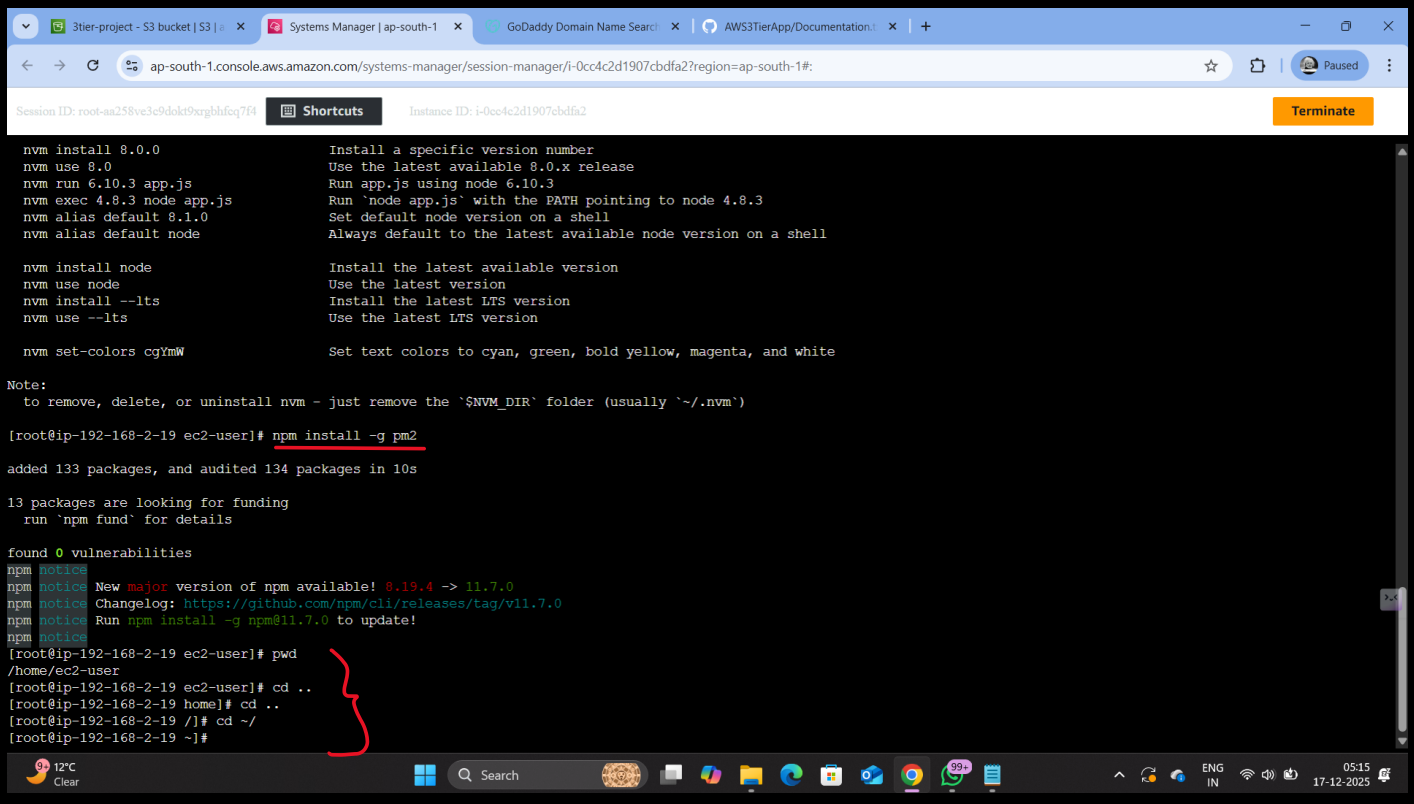

Step-21- Install and Configure Node.js



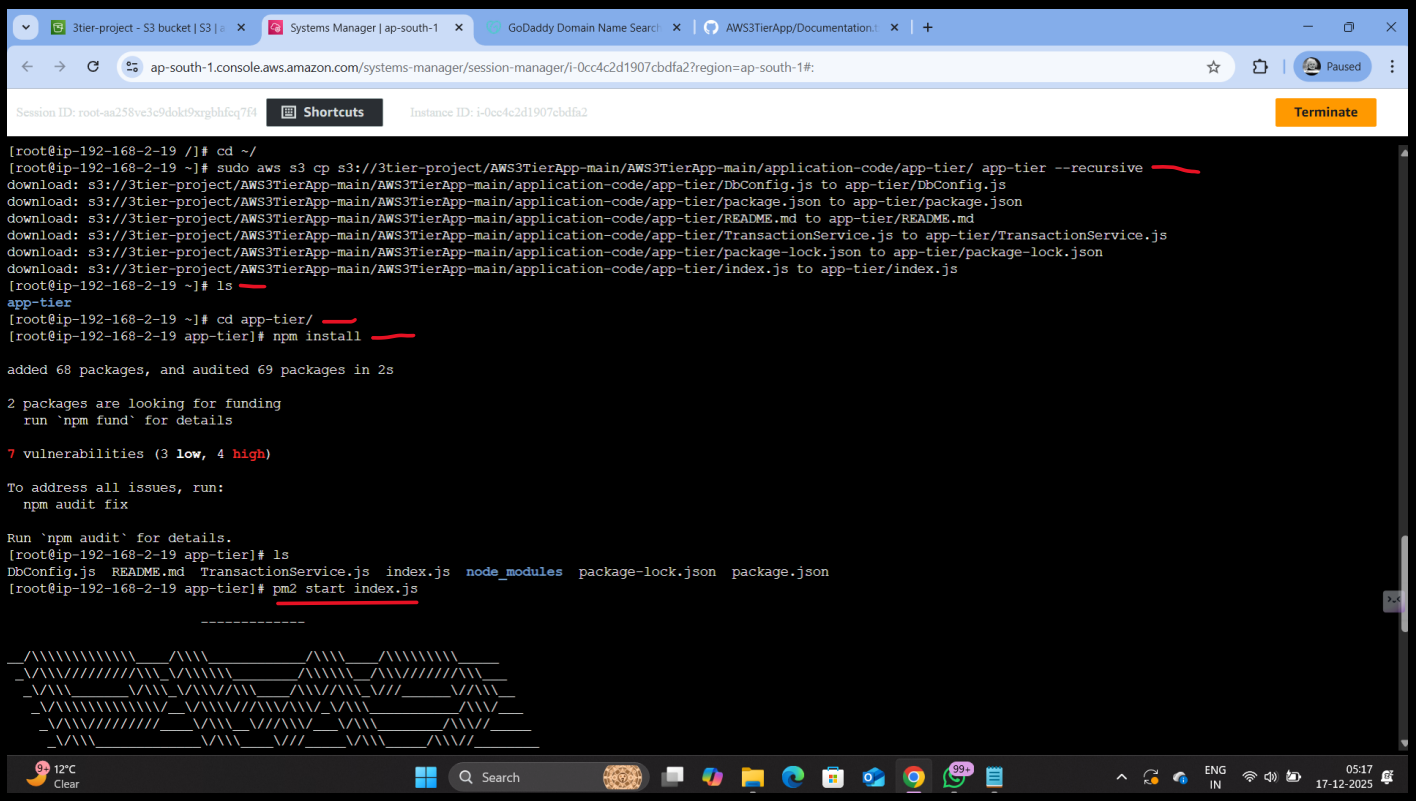
## Step-22- nvm use 16 (You will see 'Now using node v16.20.2)



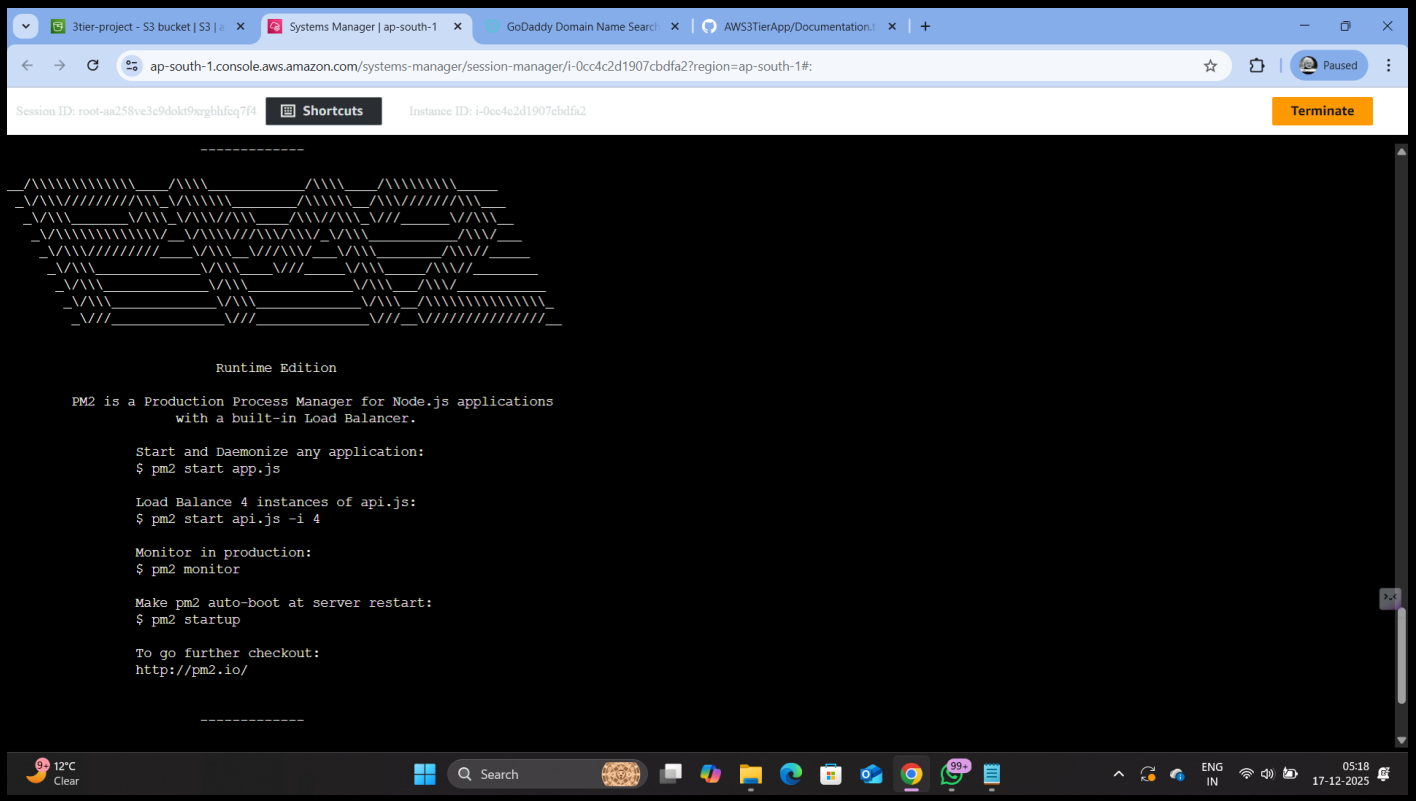

## Step-23- To run node as a service, we will install pm2



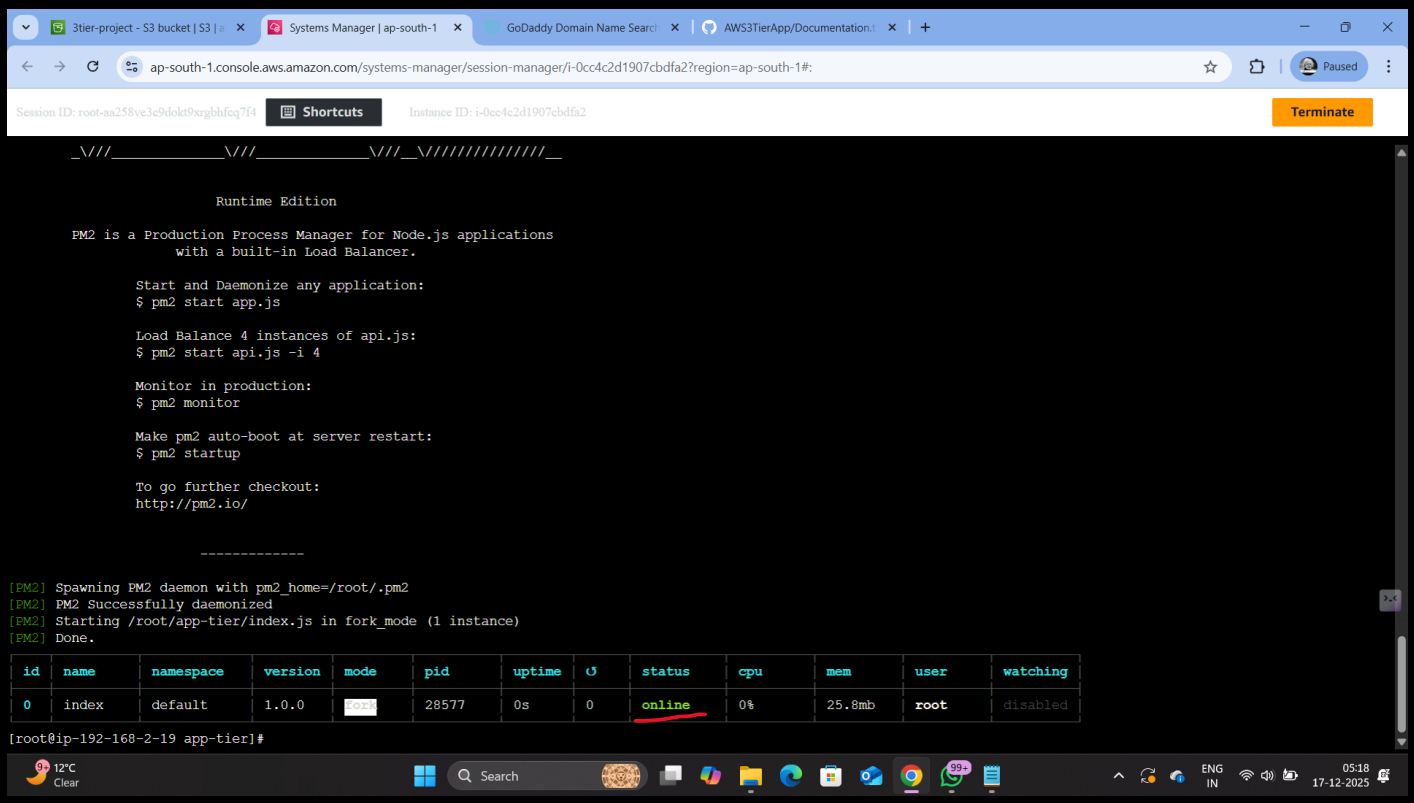
# Step-24- Download application code from S3 and start the application



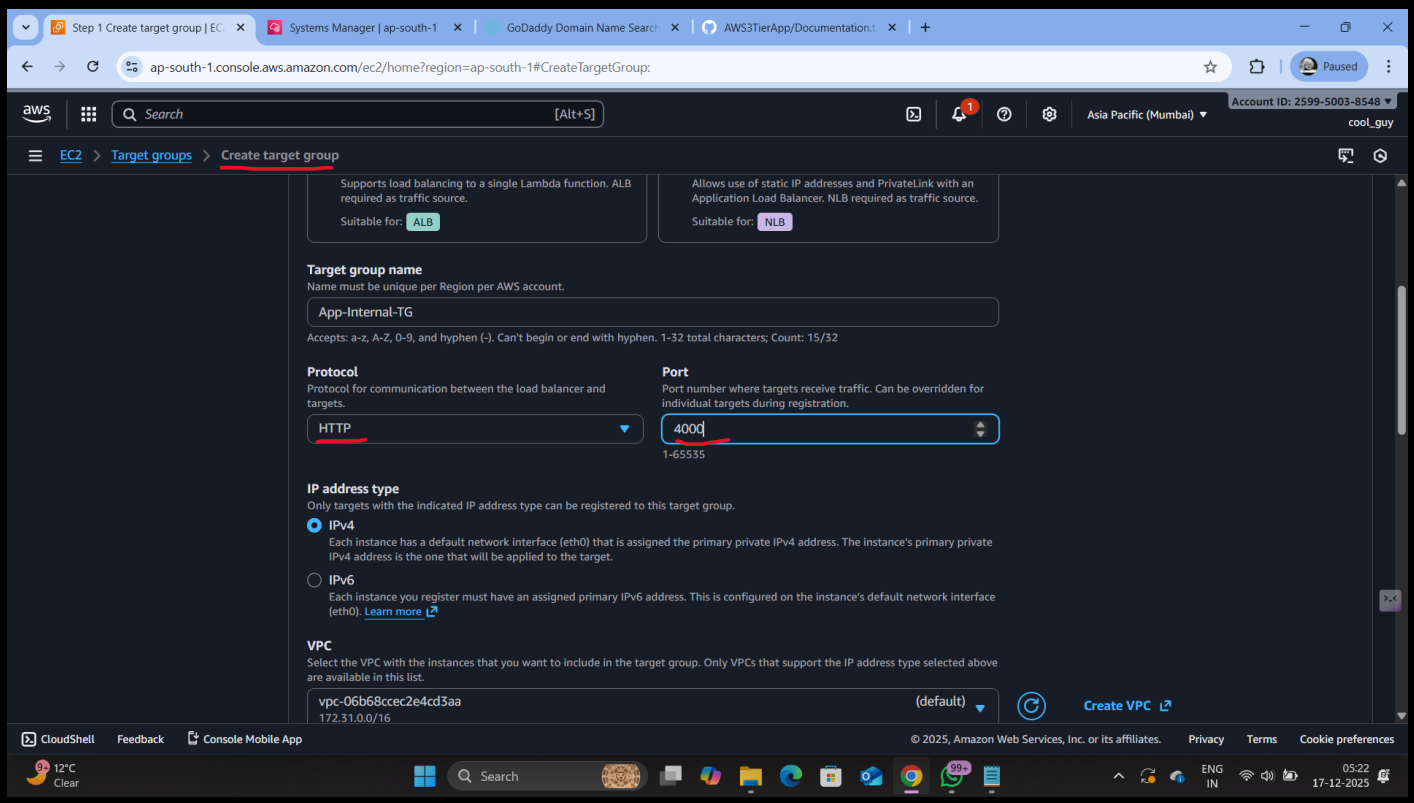

# Step-25- Here we can see that the application is downloading.



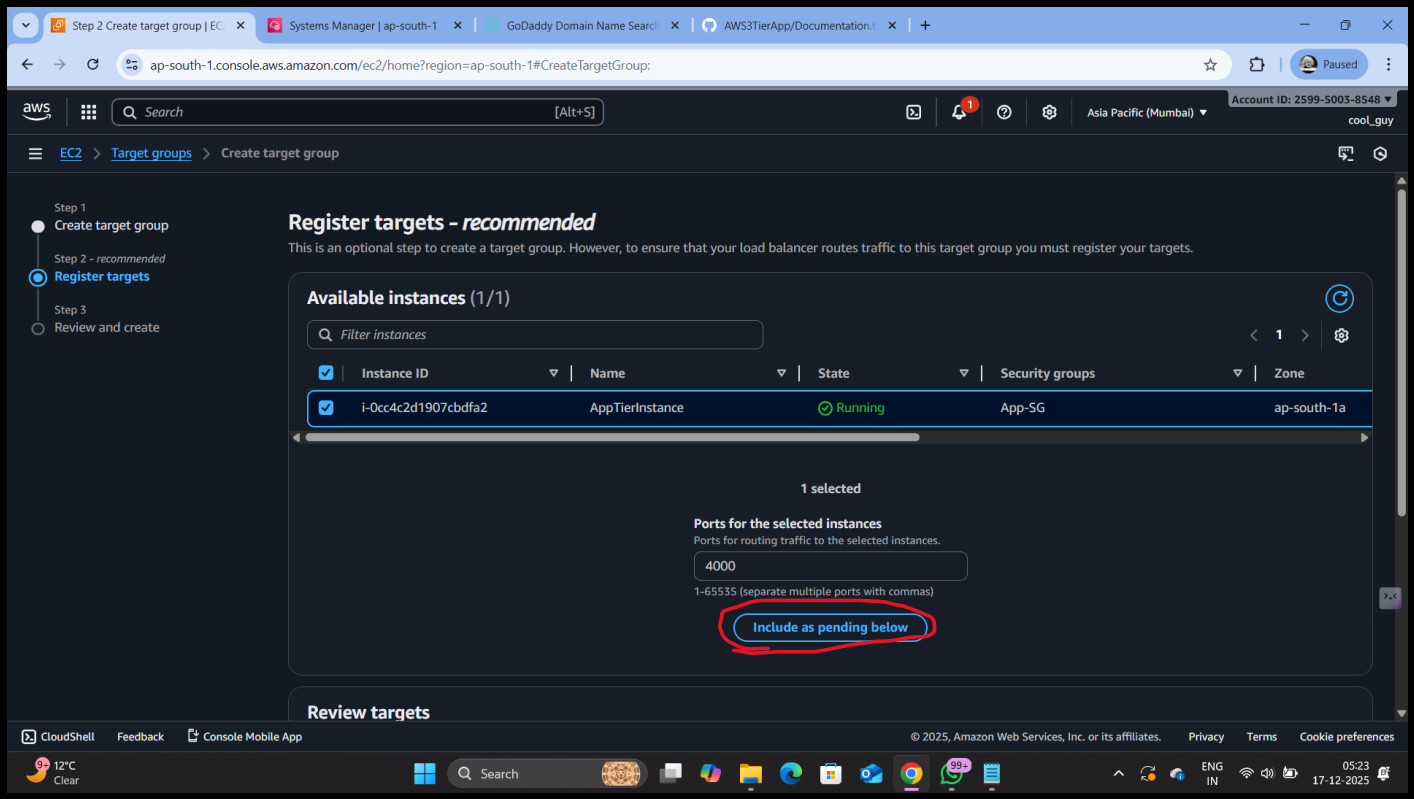
## Step-26- pm2 start index.js (You will see the status as 'online')



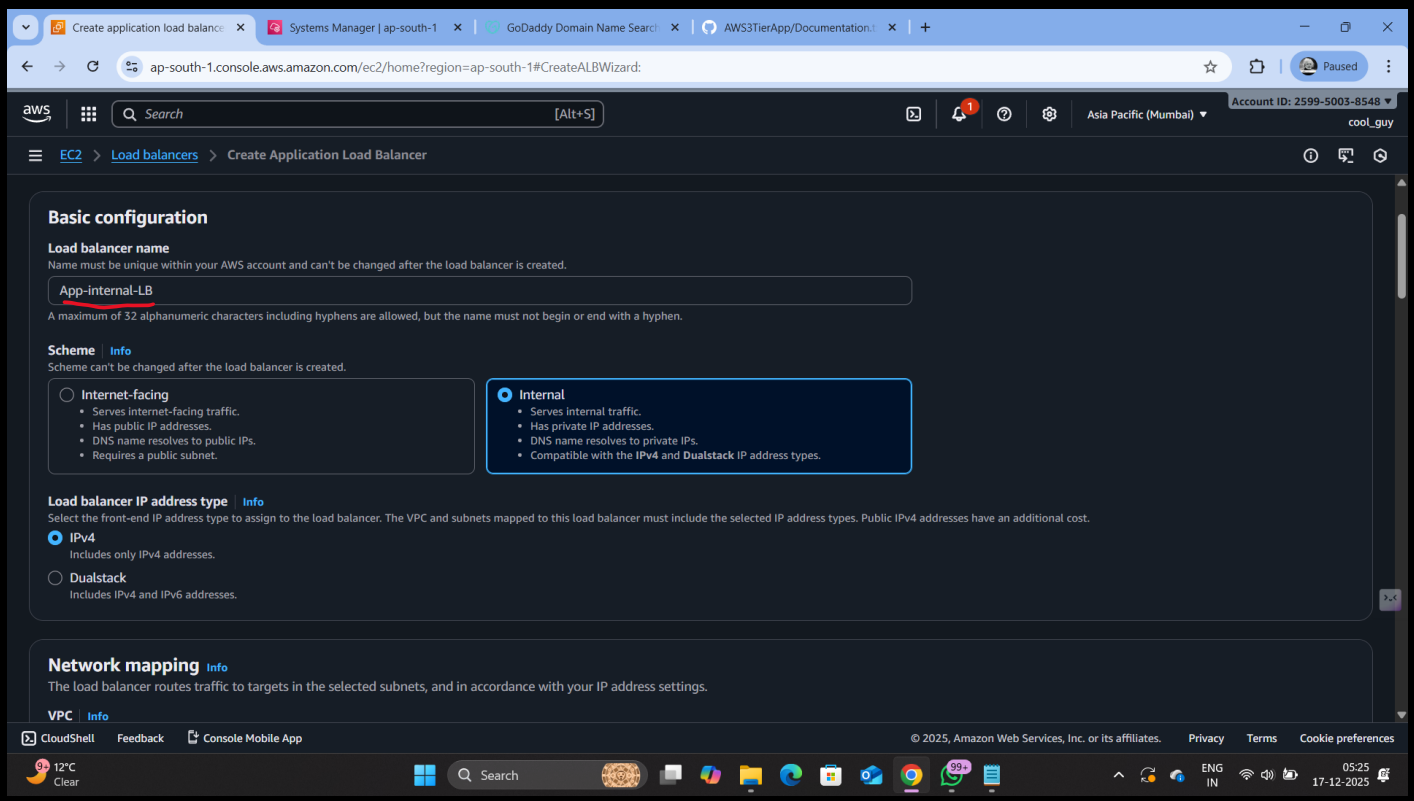

## Step-27- Now we will create the target group.



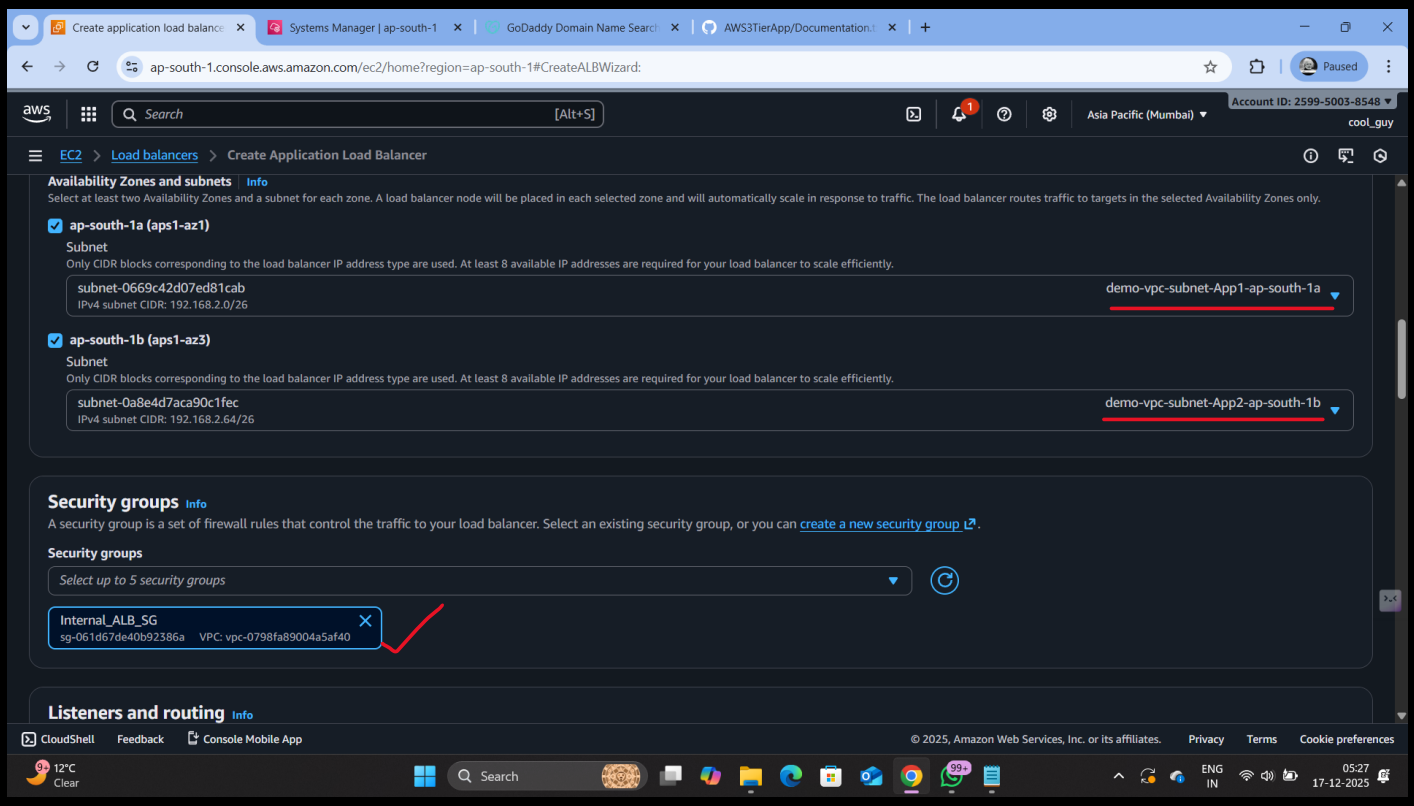
## Step-28- We will click include as pending below and click create.



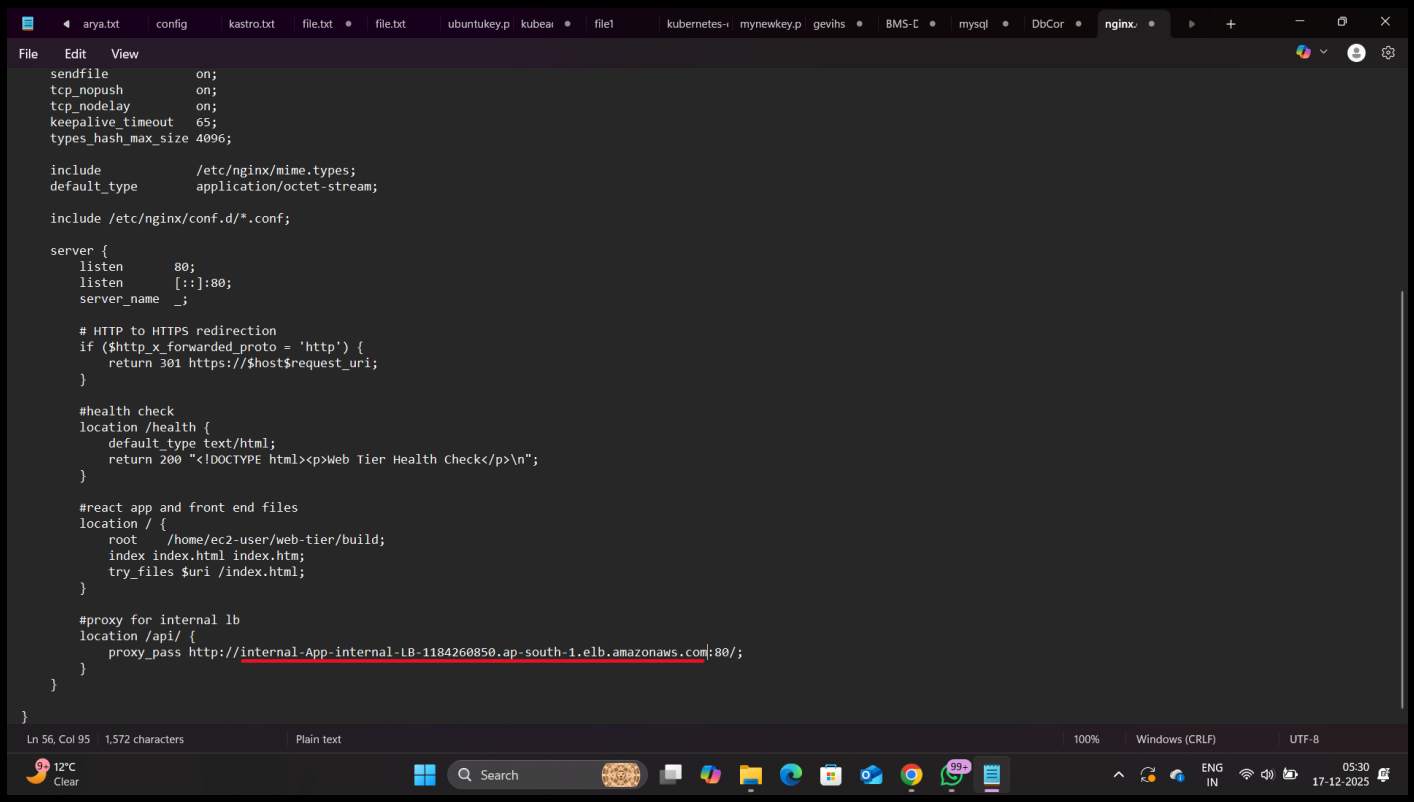

## Step-29- Here we will create the Load-Balancer (App-internal-LB)



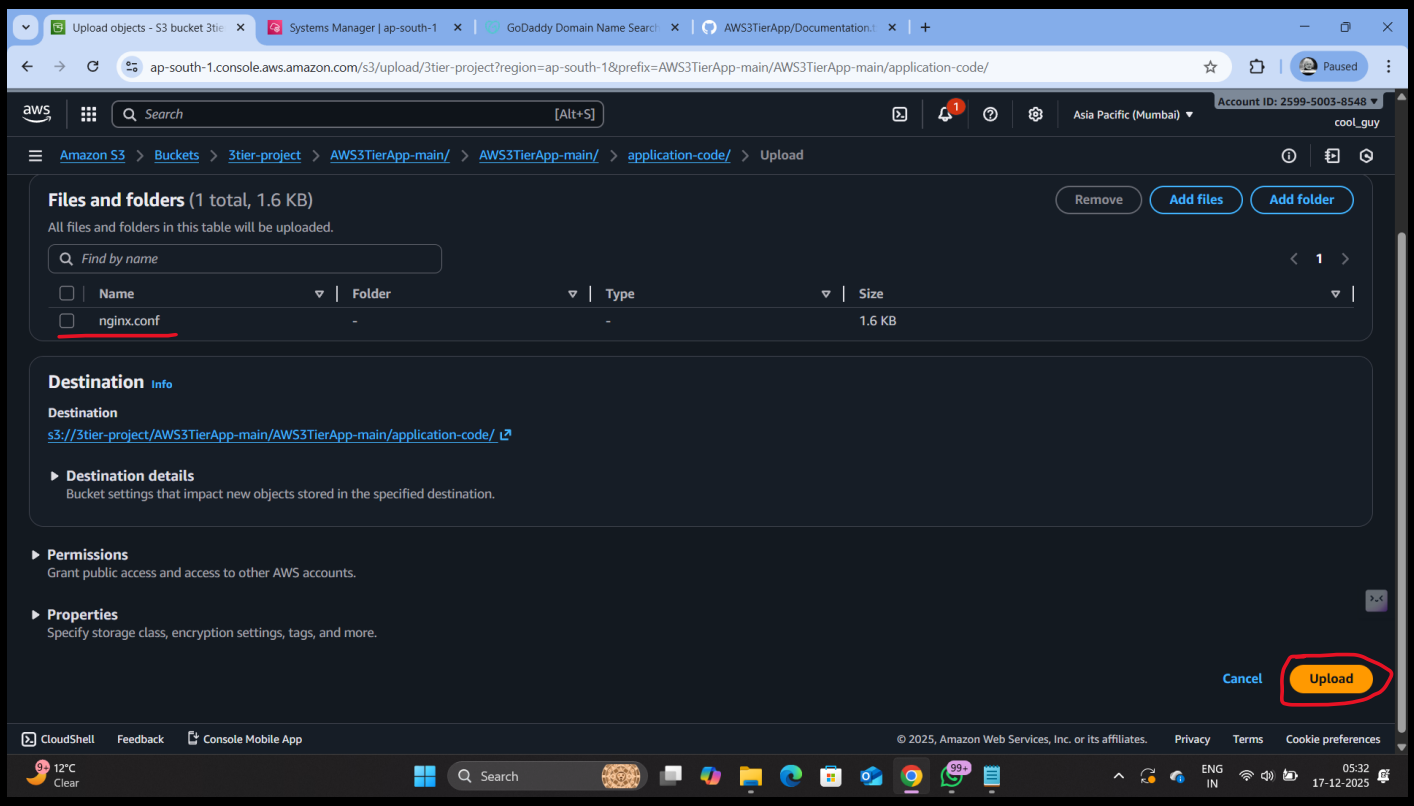
Step-30- Here we will select 2 subnets ( APP1 & APP2 ) and Security group ( Internal_Alb_SG ).



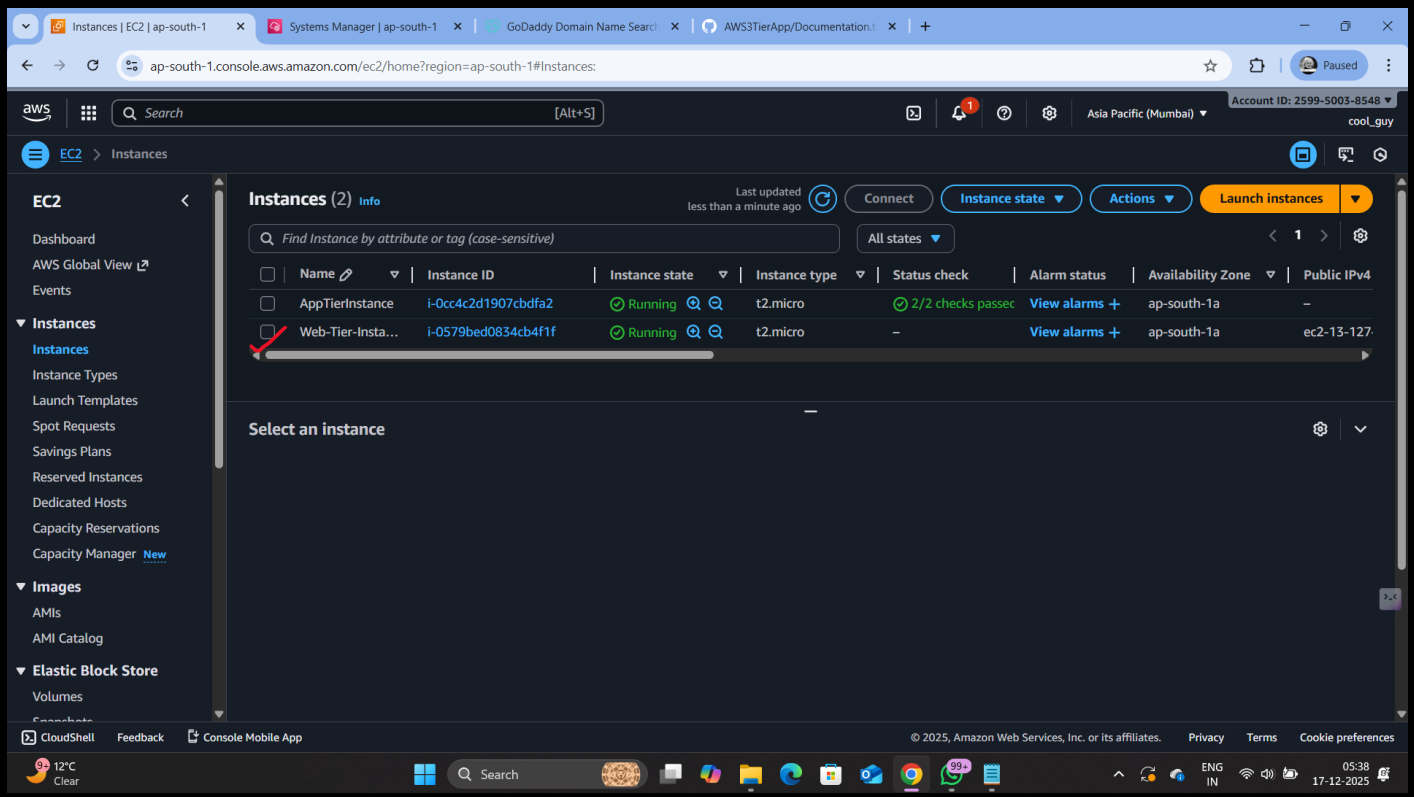

Step-31- Open nginx.conf file and in the end of the file you will see something like below; #proxy for internal lb , REPLACE-IT-WITH-INTERNAL-LB-DNS



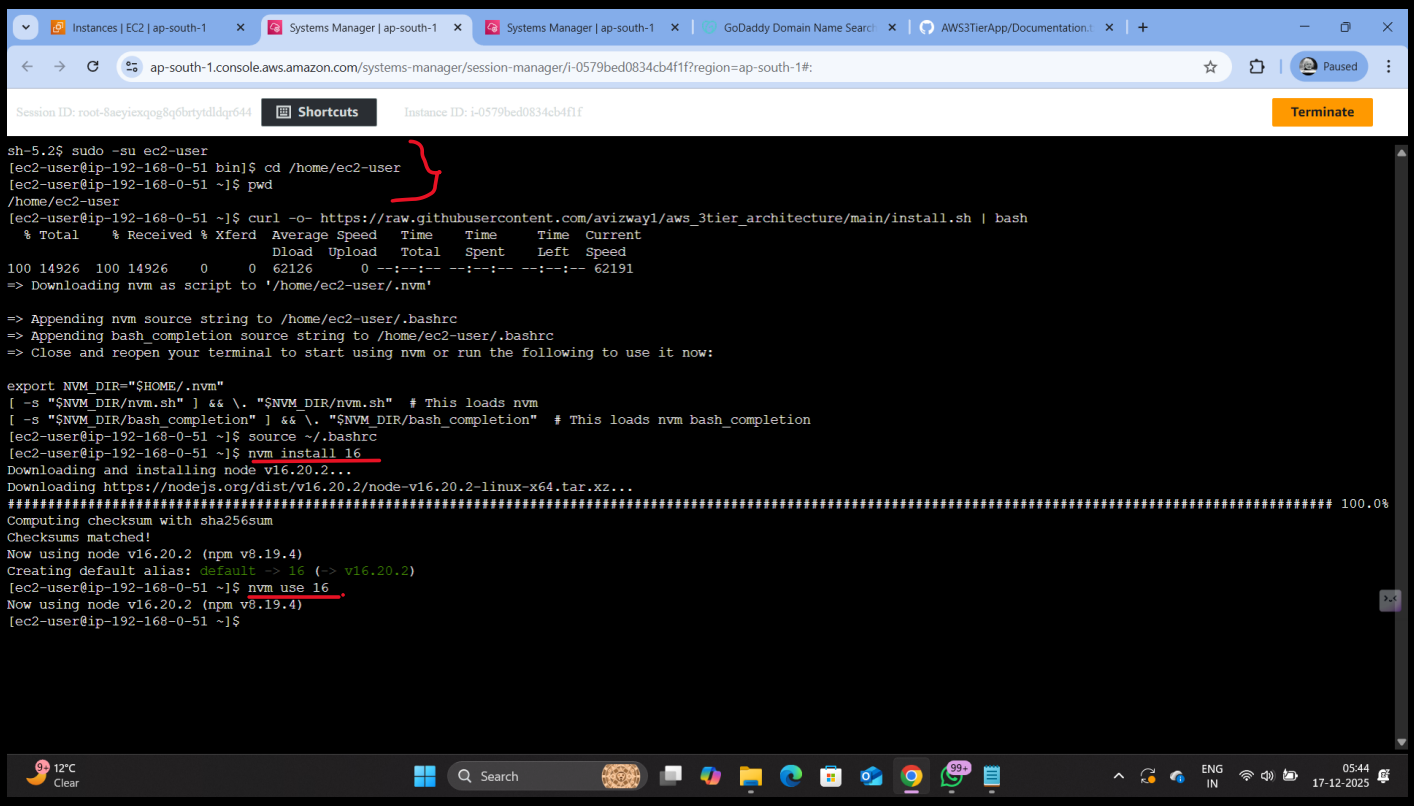
## Step-32- Now upload the updated file to the S3 bucket.



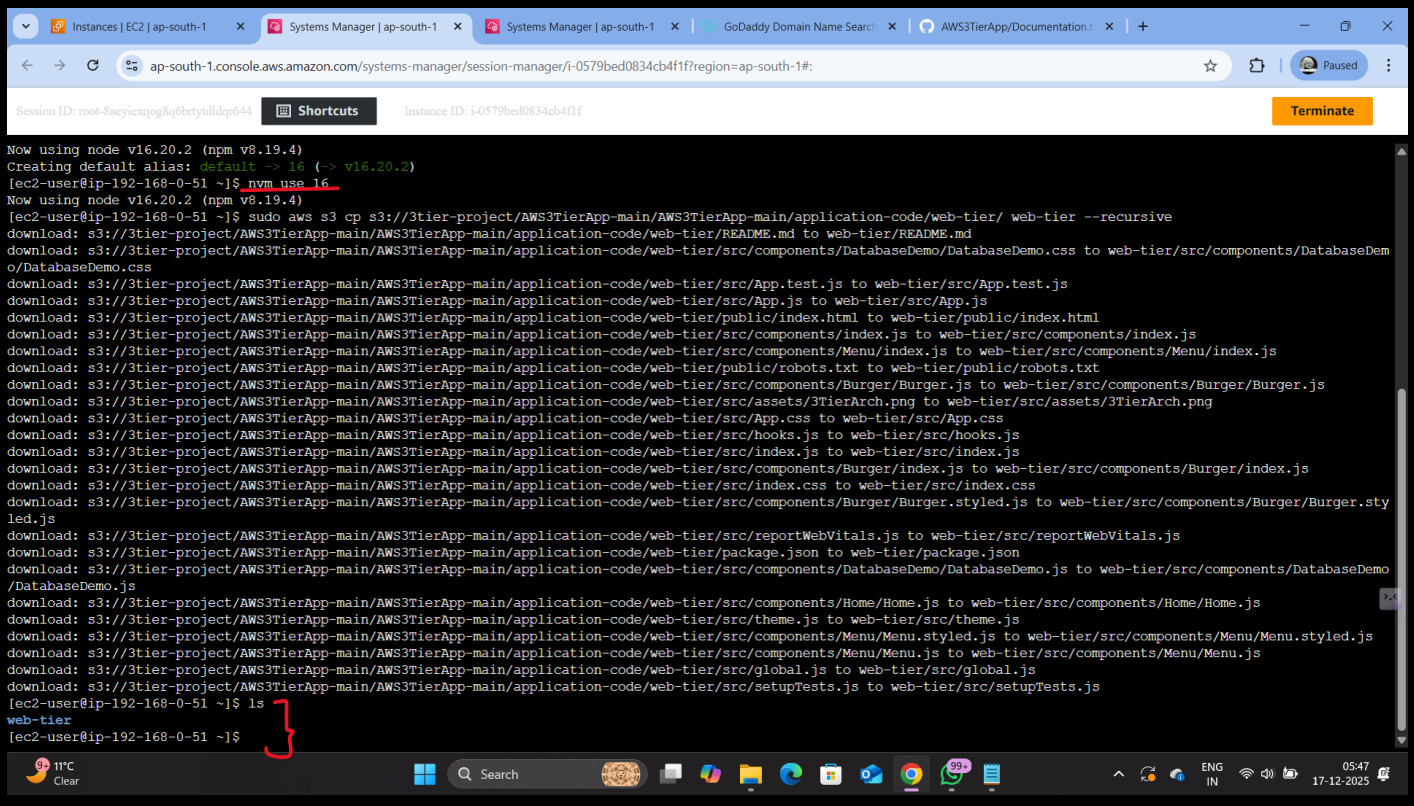

## Step-33- Now we will create the another Instance which is Web-Tier-Instance



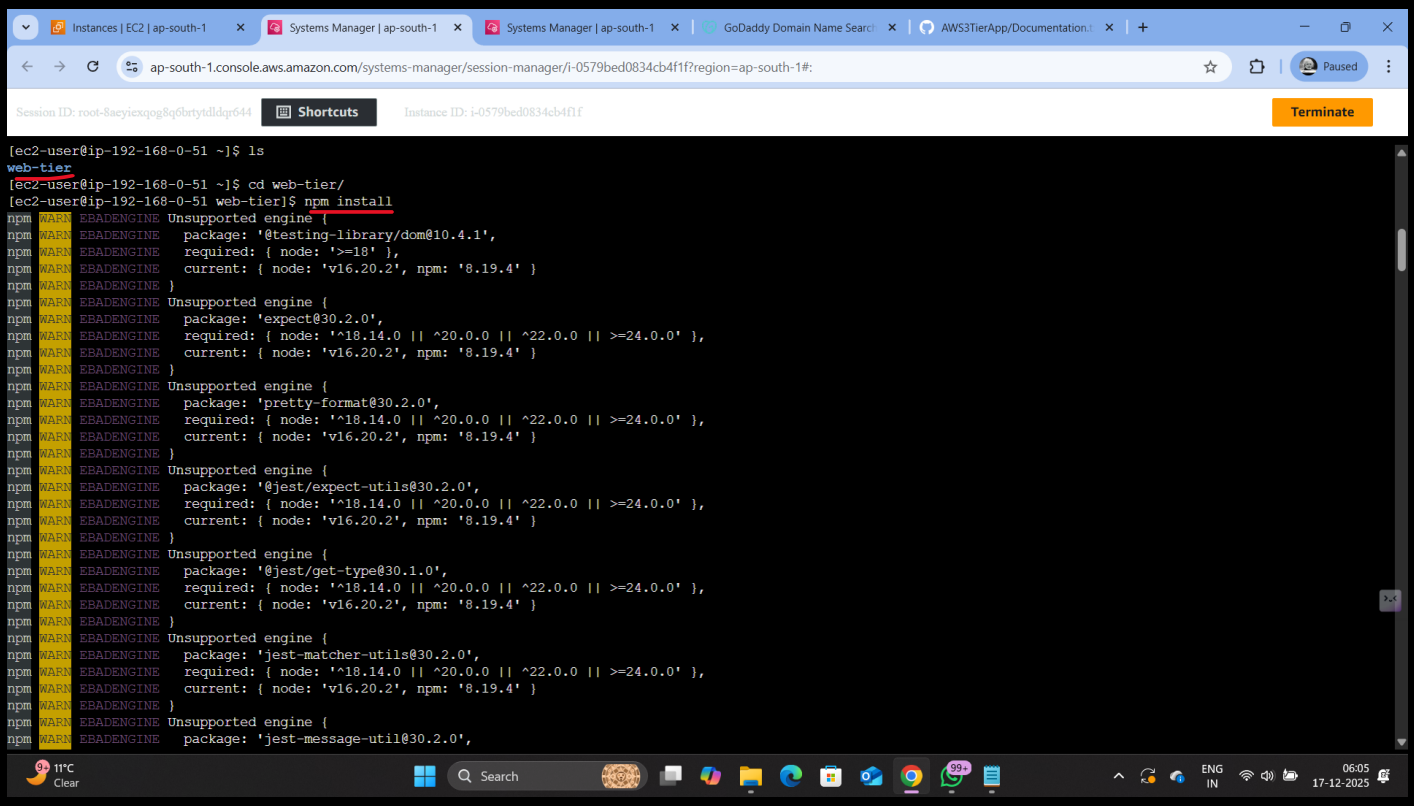
# Step-34- In this server we will install nvm 16



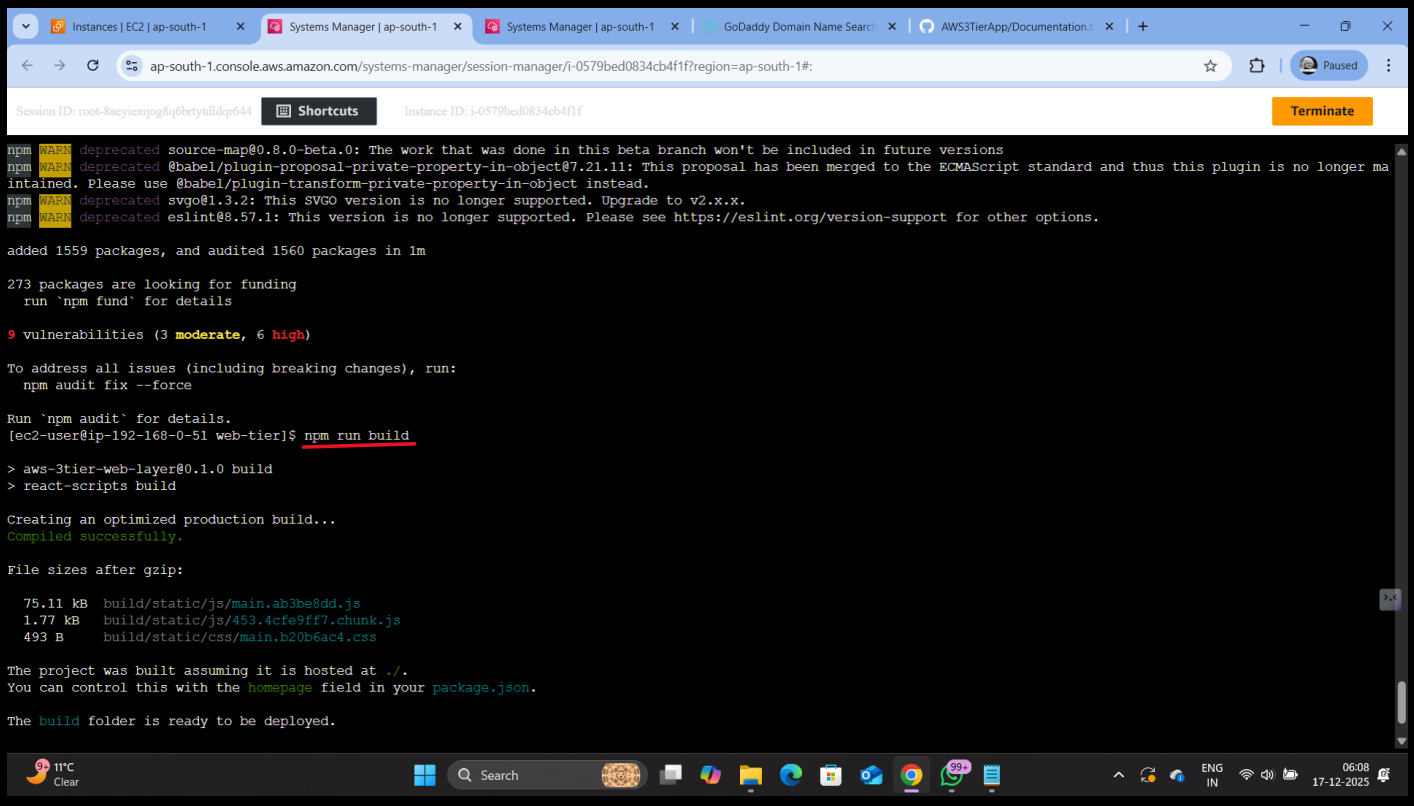

# Step-35 - Now we will download the web-tier folder of our code.



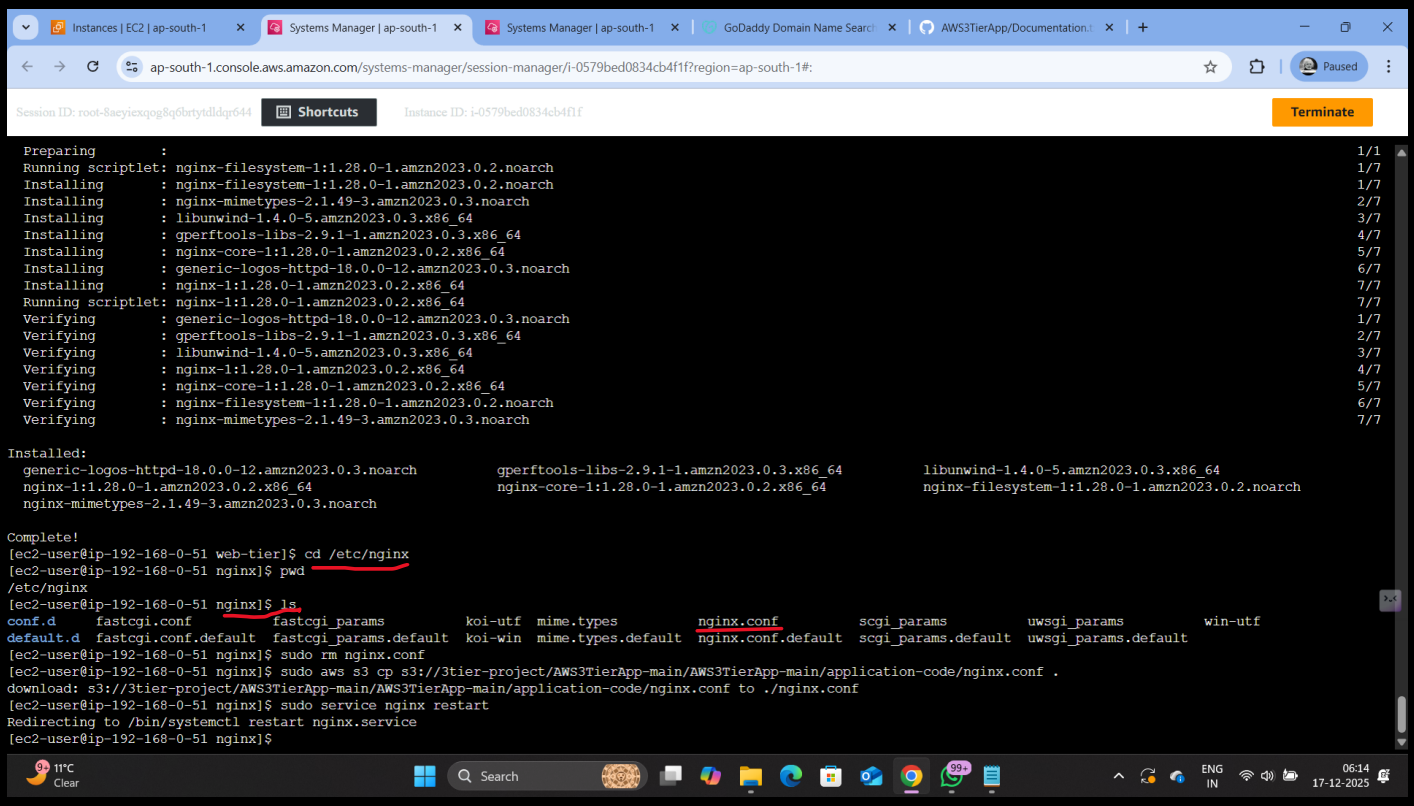
## Step-36- Now we will go to the web-tier directory and install npm



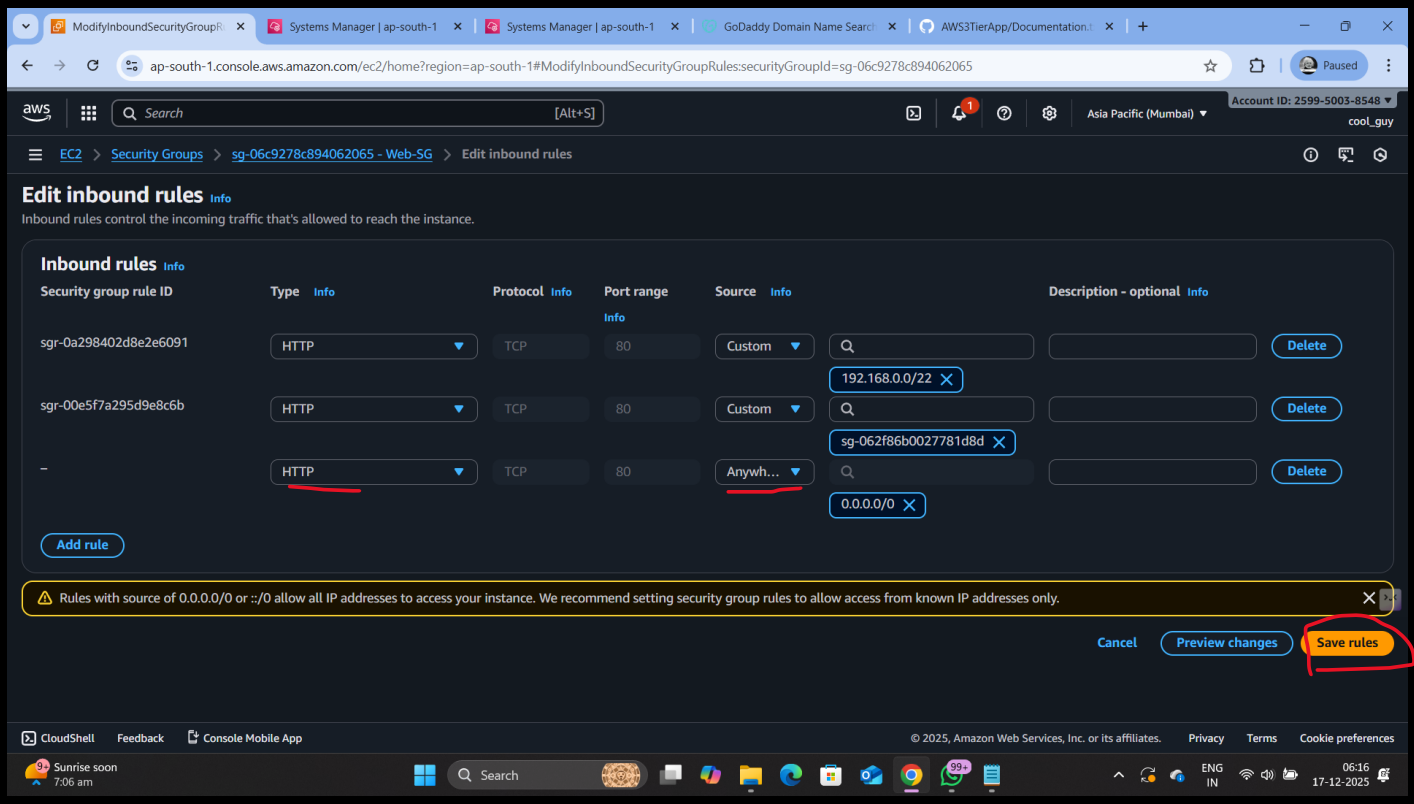

## Step-37- We will use command npm run build and the build folder is ready to be deployed.



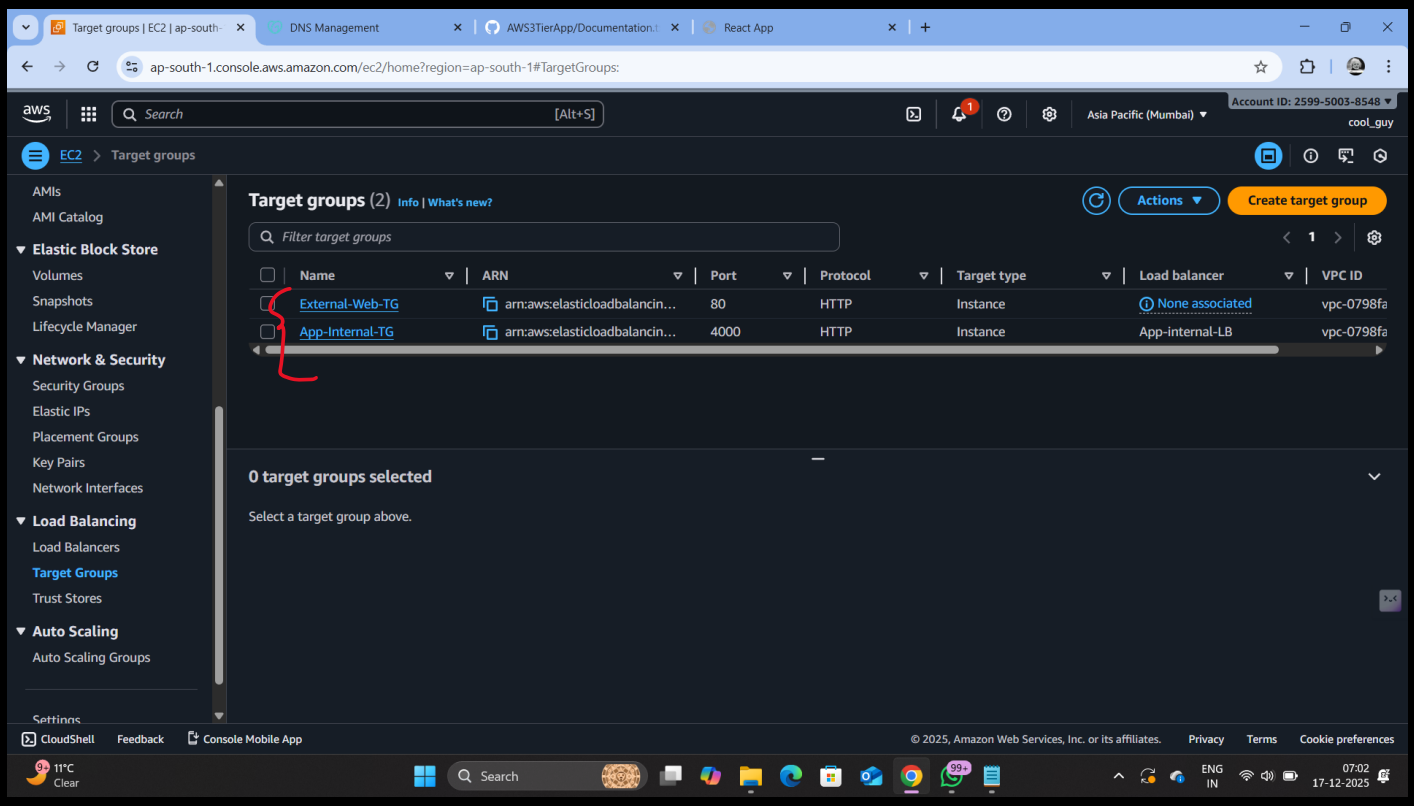
## Step-38- Update Nginx configuration and we will see nginx.conf file in the /etc/nginx directory.



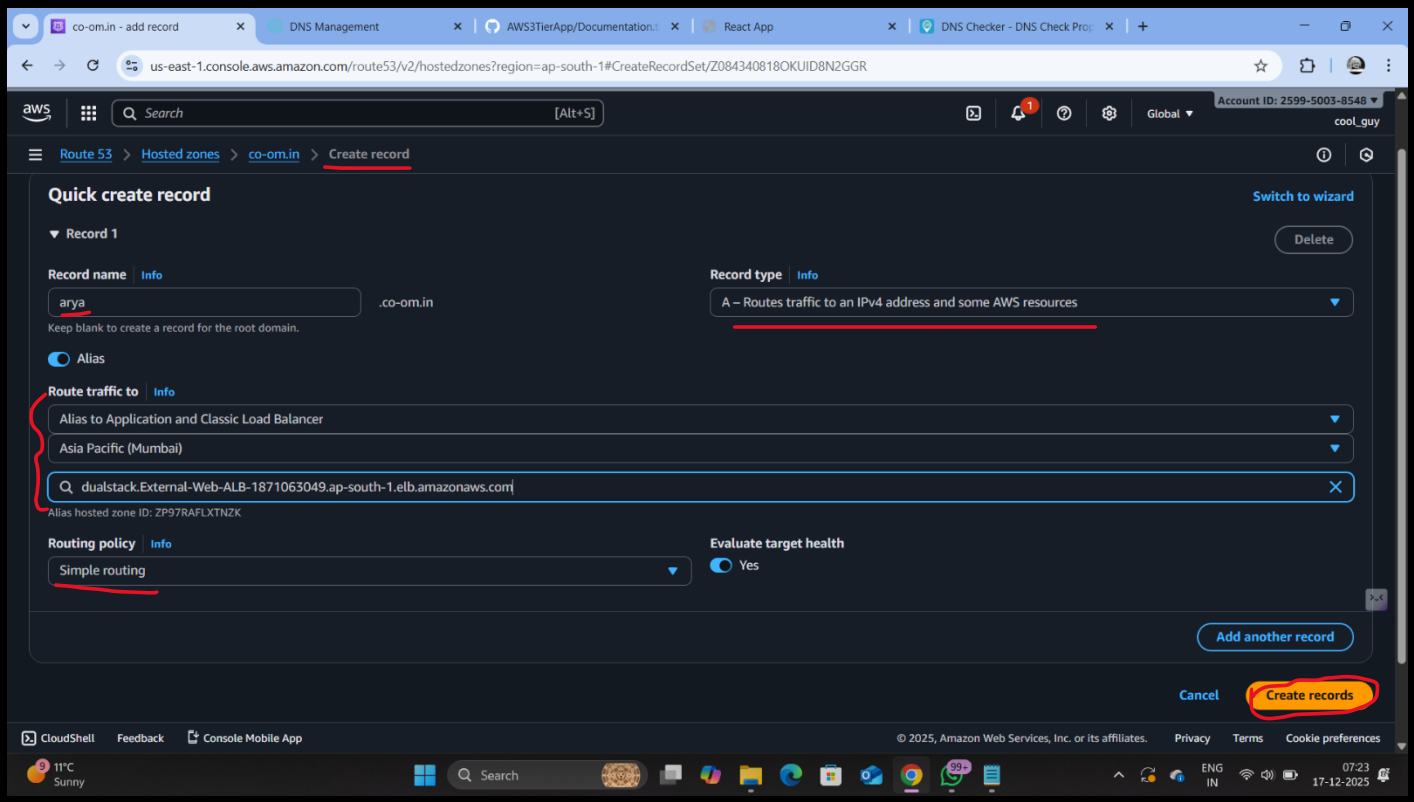

## Step-39- We will edit the inbound rule for the security group Web-SG.



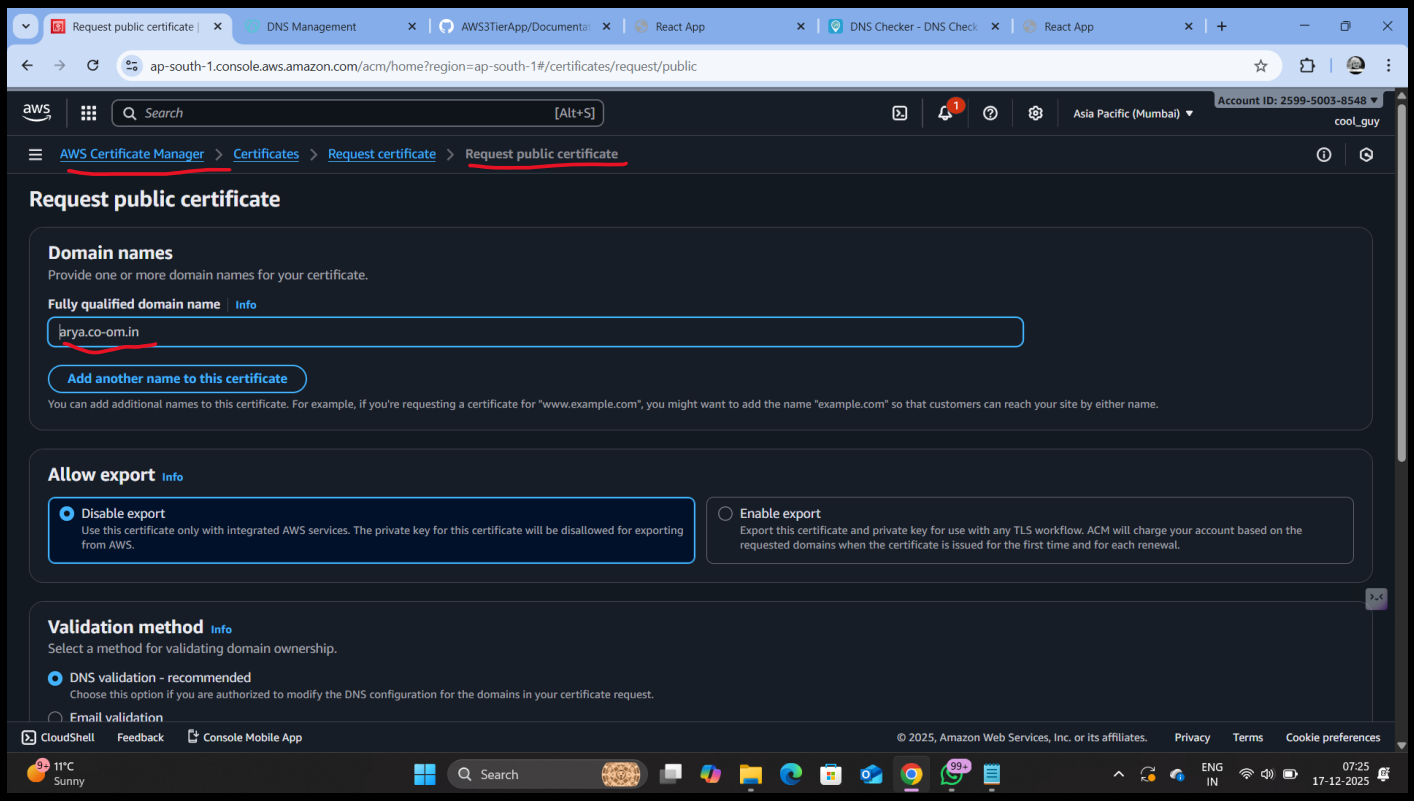
Step-40- Here we can see both the target group.



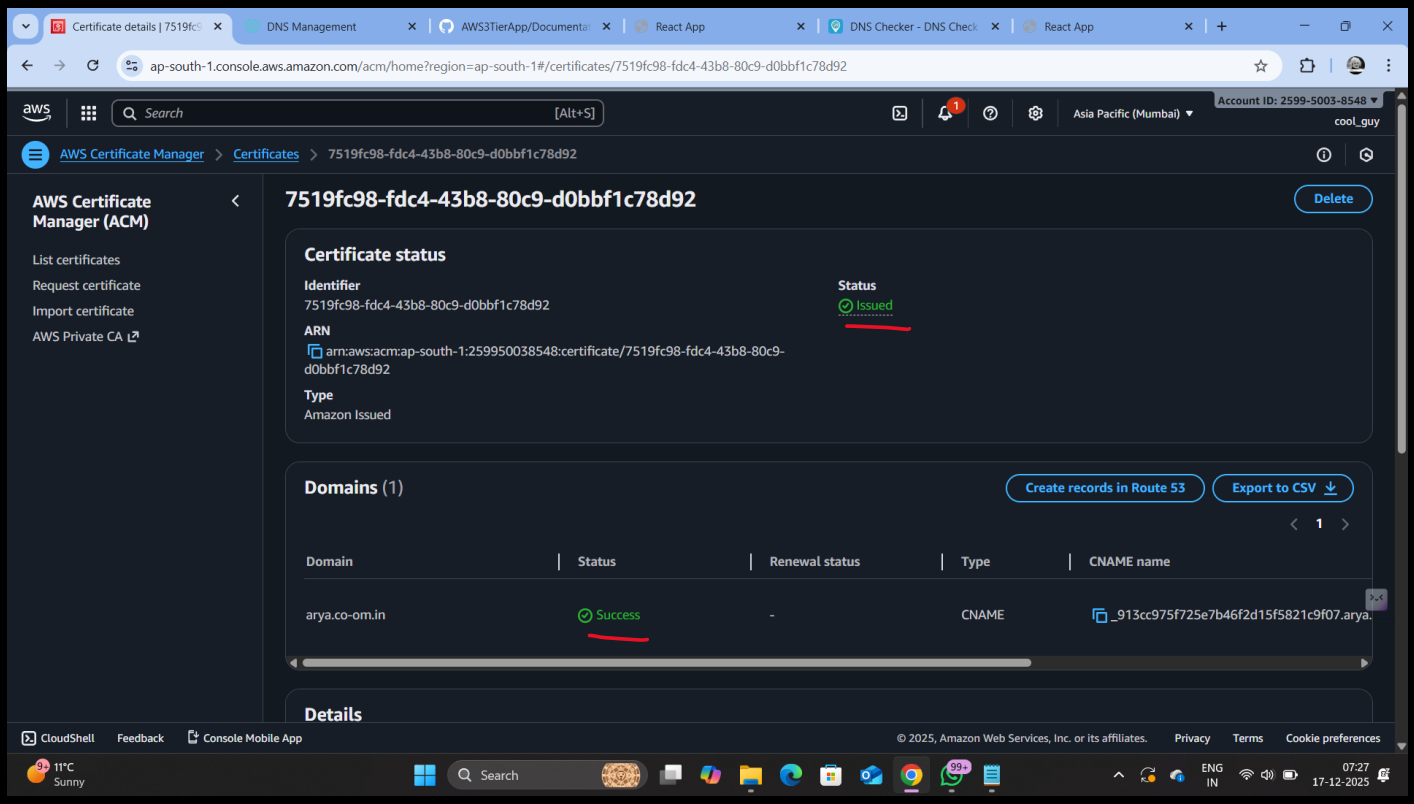

Step-41- Now we had our domain which is co-om.in and we will create a record to use this domain as we can't give the IP address to the public. So our domain name will start from arya i.e arya.co-om.in



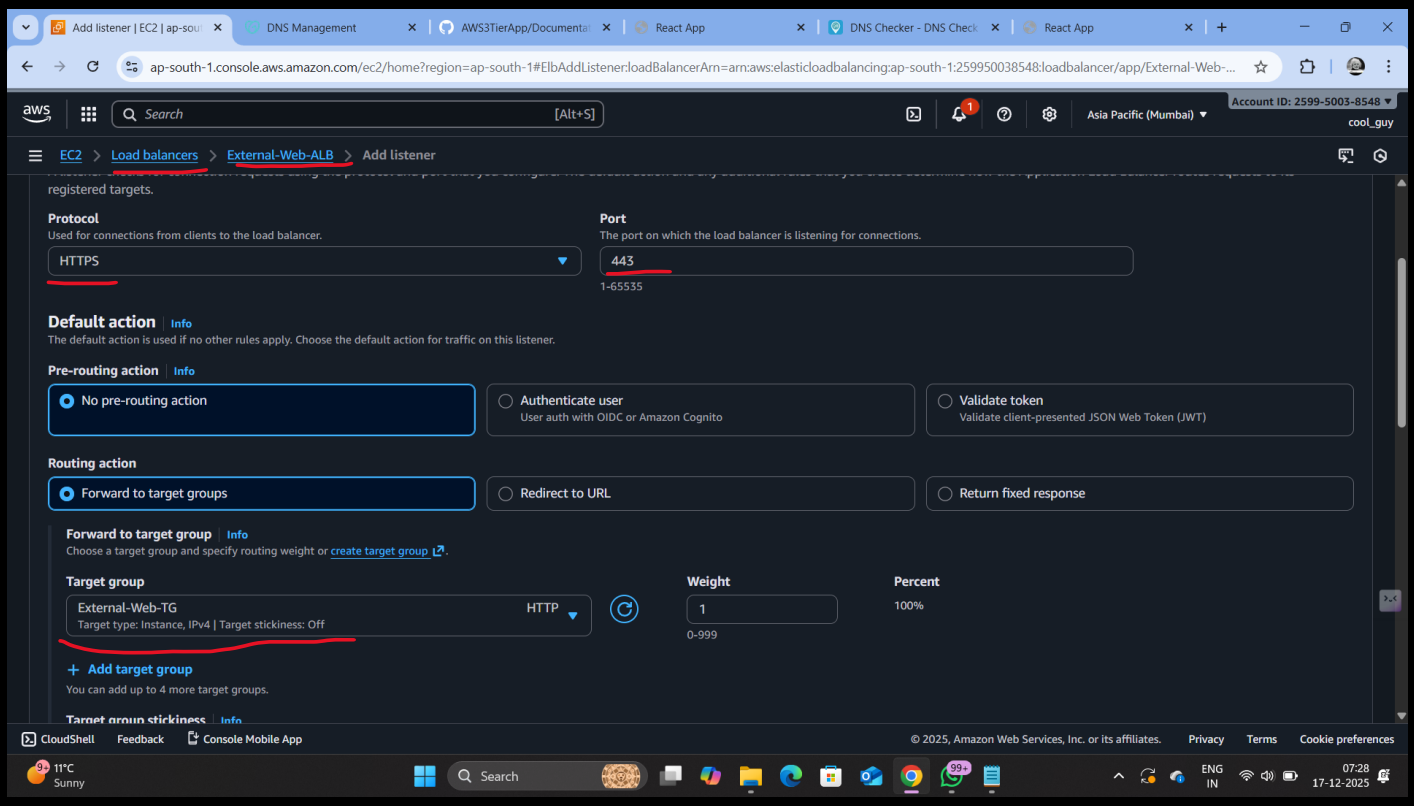
Step-42- Now we will register in the certificate manager make our domain Certified



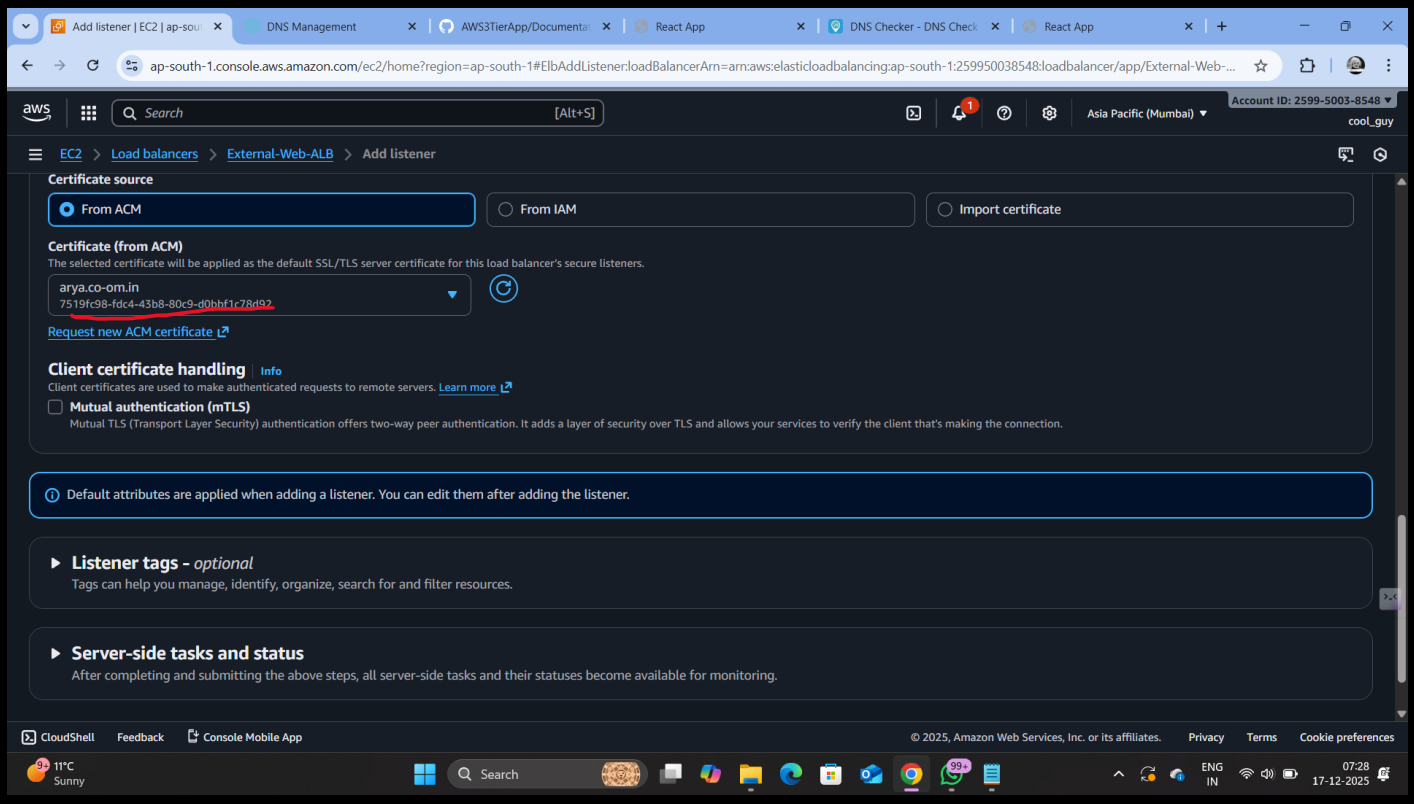

Step-43- We can see that our status of certificate is successful.



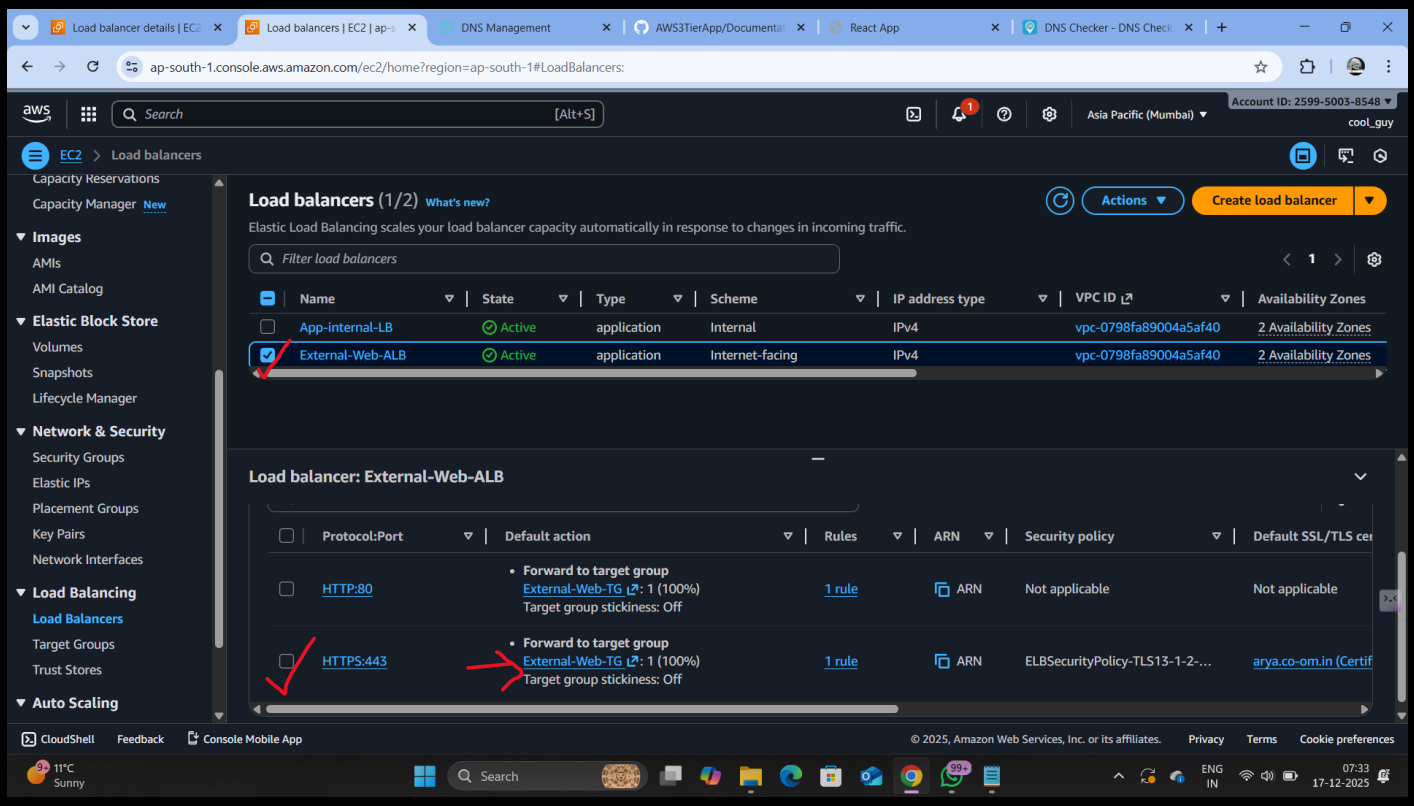
## Step-44- We will add a listener to our External-Web load balancer.



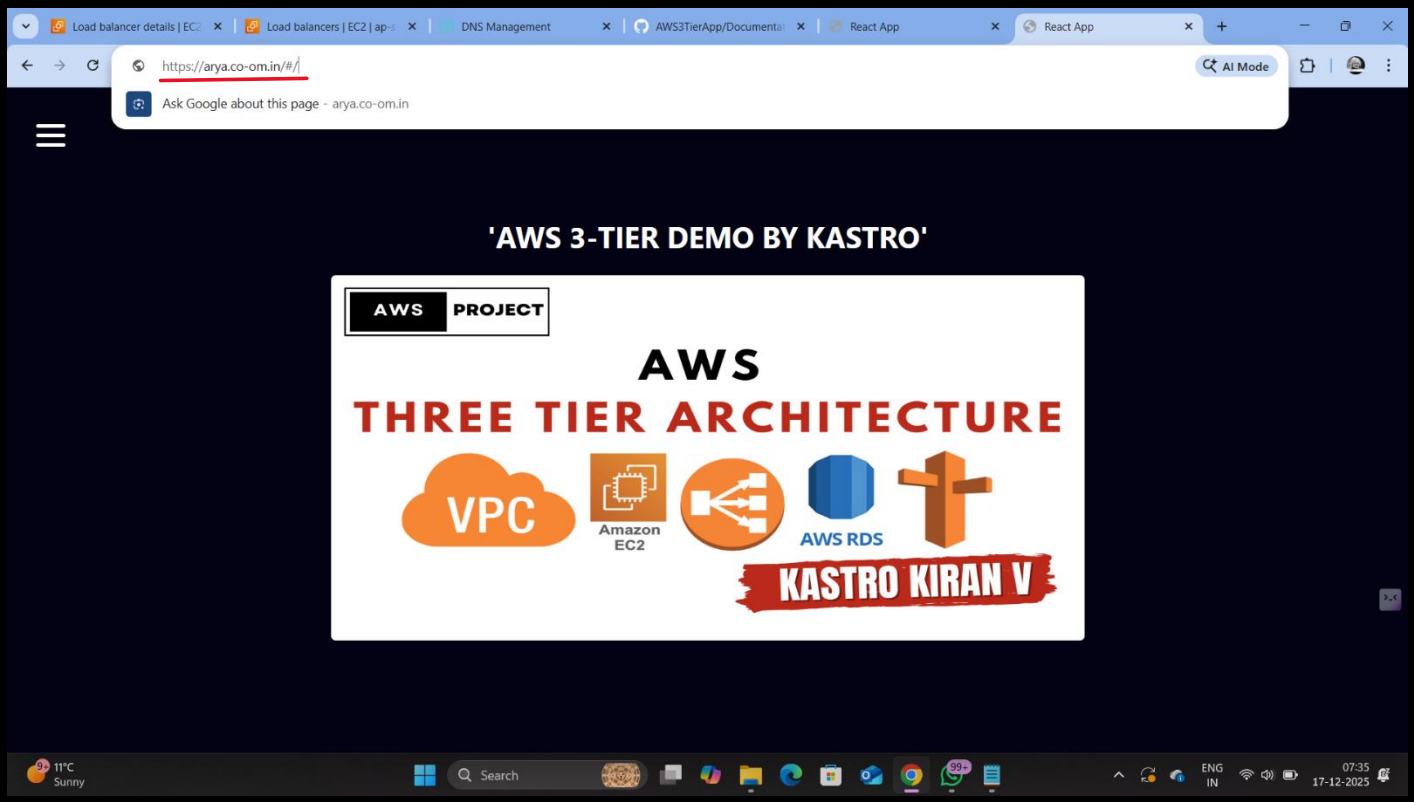

## Step-45- We will select our certificate from ACM and click on OK.



Step-46- In the security group we enabled the port 443 as it was not enabled earlier and now we can access our website.




Step-47- On browser we used our domain arya.co-om.in and now we can see that our website is running.



Step-48- We can add some data also like an user for example I added Hindi and amount is 700, so it will be oploaded in the database also like an ordinary website.