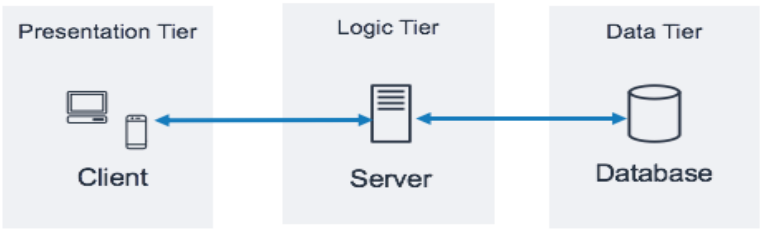
NAME :GEETHANJALI CHELLABOINA
PHONE : 6303766345
MAIL :geethanjalichellaboina09@gmail.com
BATCH:119(7am)

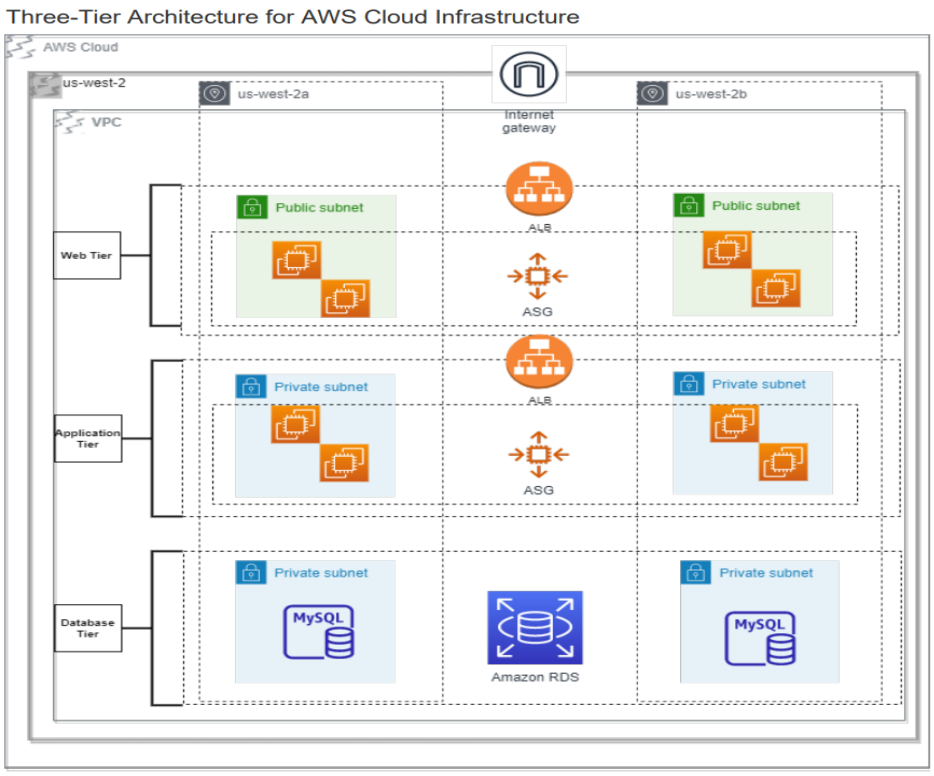
THREE TIER OF ARCHITECTURE

INTRODUCTION:

The three-tier aws architecture consists of three tiers: a presentation tier, logic tier, and data tier. The following diagram illustrates an example of a generic three-tier application.



This type of architecture is found in client-server applications, such as web applications that have a frontend, a backend, and a database. Each tier, or layer, performs a specific task and you can manage it independently of the others. This is actually what we do when we move from the monolithic way of building an application where the frontend, backend, and databases reside in one place.



.

BENEFITS OF THREE TIER ARCHITECTURE:

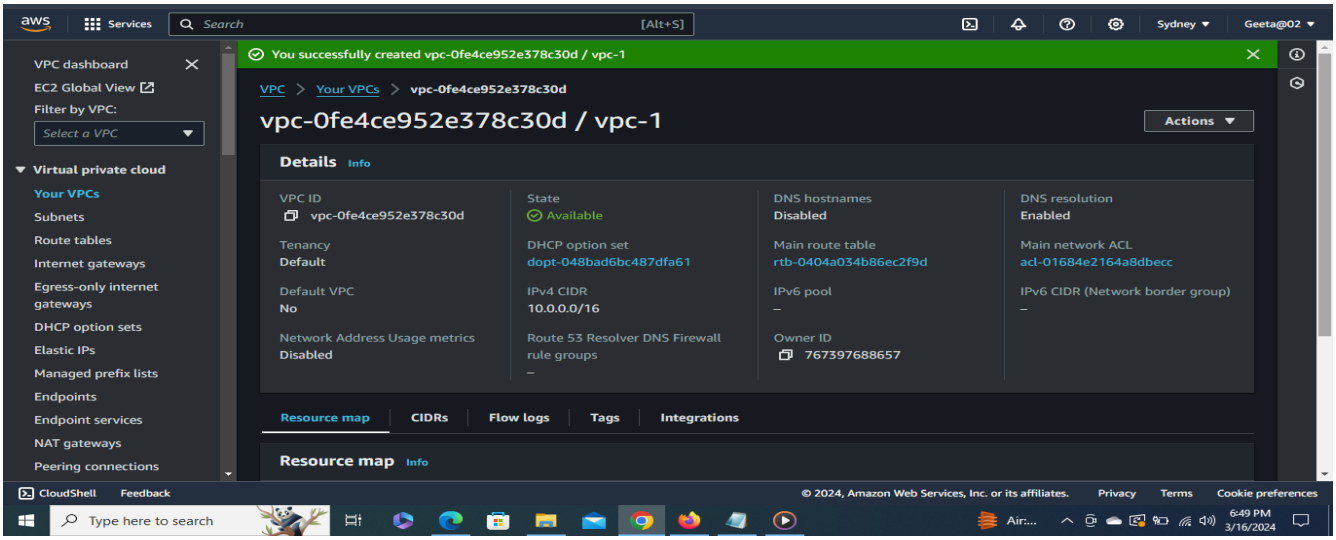
It improves data integrity.

- It offers a great deal of security as your client does not have direct access to the database.
- It is very convenient to maintain and make modifications.
- Offers good performance as the presentation tier does cache operations. This enables better network utilization and also reduces the load on application and data tiers.

PROCEDURE FOR THREE TIER ARCHITECTURE:

STEP-1 CREATE A VPC

We can create VPC in any region here I have taken in Syndney region.



STEP-2:CREATE SUBNETS

Now we need to create subnets. Here we need to create total 6 subnets.

Presentation Tier Subnets:

Subnet-1: 10.0.1.0/24 (ap-southeast-2a)

Subnet-2: 10.0.2.0/24 (ap-southeast-2b)

- **Application Tier Subnets:**

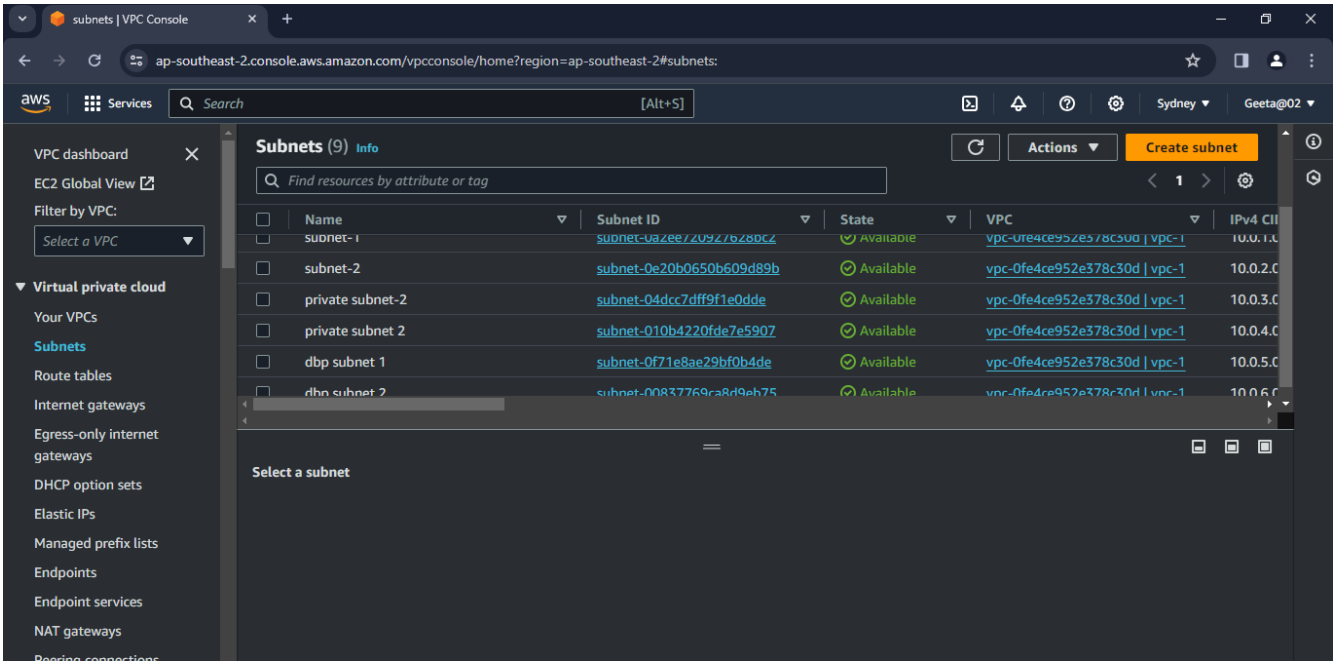
Private Subnet 1: 10.0.3.0/24 (ap-southeast-2a)

Private Subnet 2: 10.0.4.0/24 (ap-southeast-2b)

- **Database Tier Subnets:**

dbp Subnet 1: 10.0.5.0/24 (ap-southeast-2a)

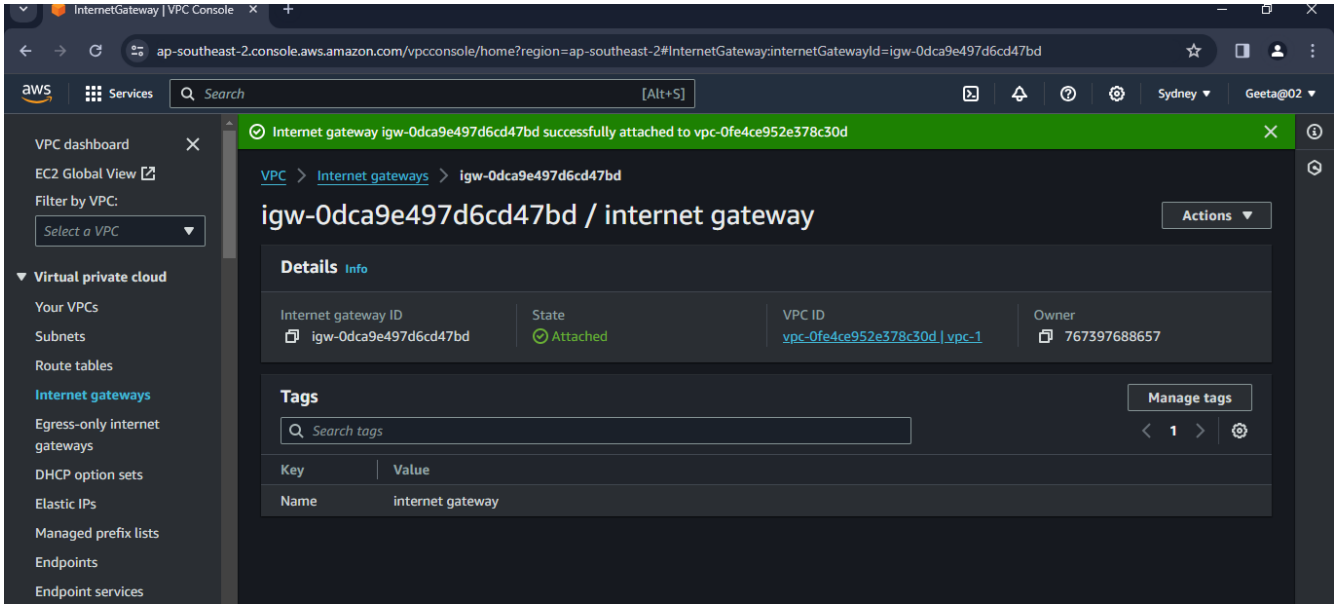
dbp subnet 2: 10.0.6.0/24 (ap-southeast-2b)



STEP 3: SET UP AN INTERNET GATEWAY:

An Internet Gateway is a highly available, horizontally scaled, yet redundant VPC component. It enables communication between the instances in your VPC and the internet using VPC route tables for internet-routable traffic.

Now we need to create internet gateway and then attach to VPC.



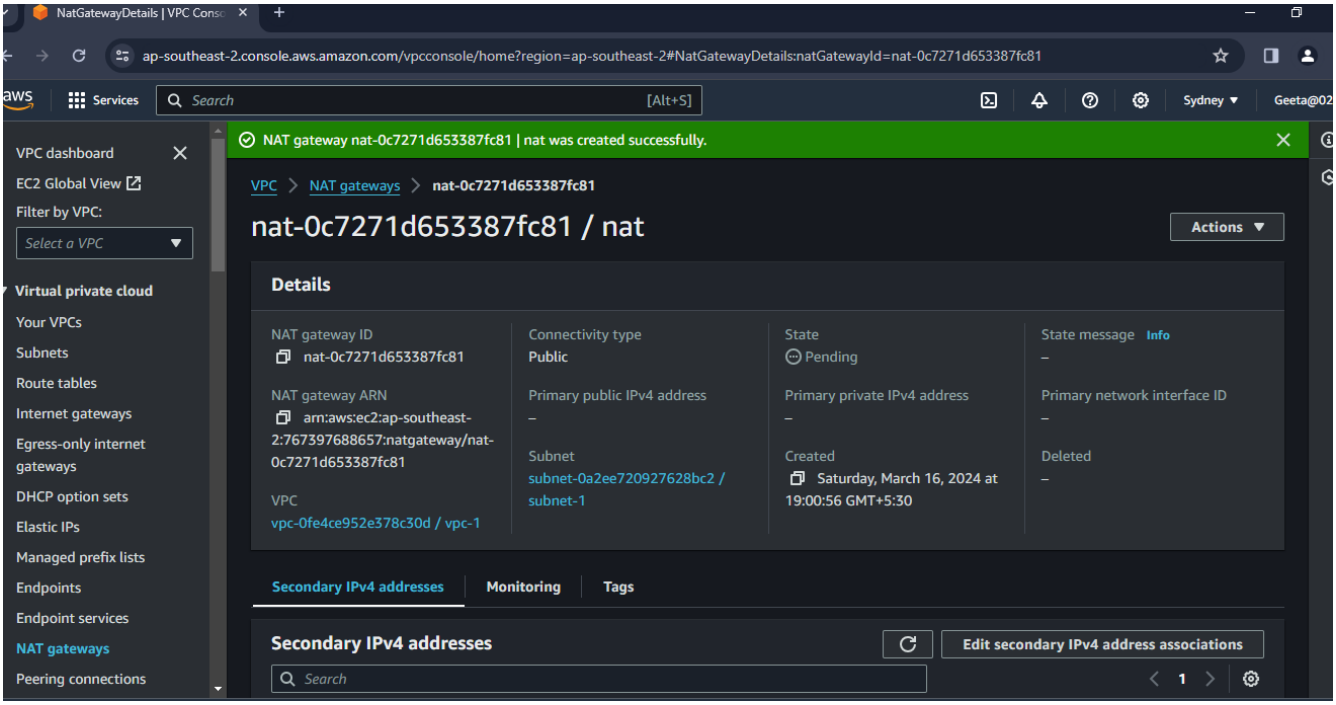
STEP-4: CREATE NAT GATEWAY.

PROCESS:

Choose Create NAT gateways do the following:

1. (Optional) Specify a name for the NAT gateway. This creates a tag where the key is the Name and the value is the name that you specify.
2. Choose the subnet in which you want to create the NAT gateway.
3. For the Connectivity type, select Private to create a private NAT gateway or Public (the default) to create a public NAT gateway.
4. (Public NAT gateway only) For Elastic IP allocation ID, select an Elastic IP address to associate with the NAT gateway.
5. (Optional) For each tag, choose to Add a new tag and enter the key name and value.
6. Choose create Nat gateways.

After creating your screen will be as follows:



STEP-5 CREATE ROUTE TABLES.

A route table carries a set of rules, called routes, that you use to ascertain the network traffic's direction from your gateway or subnet.

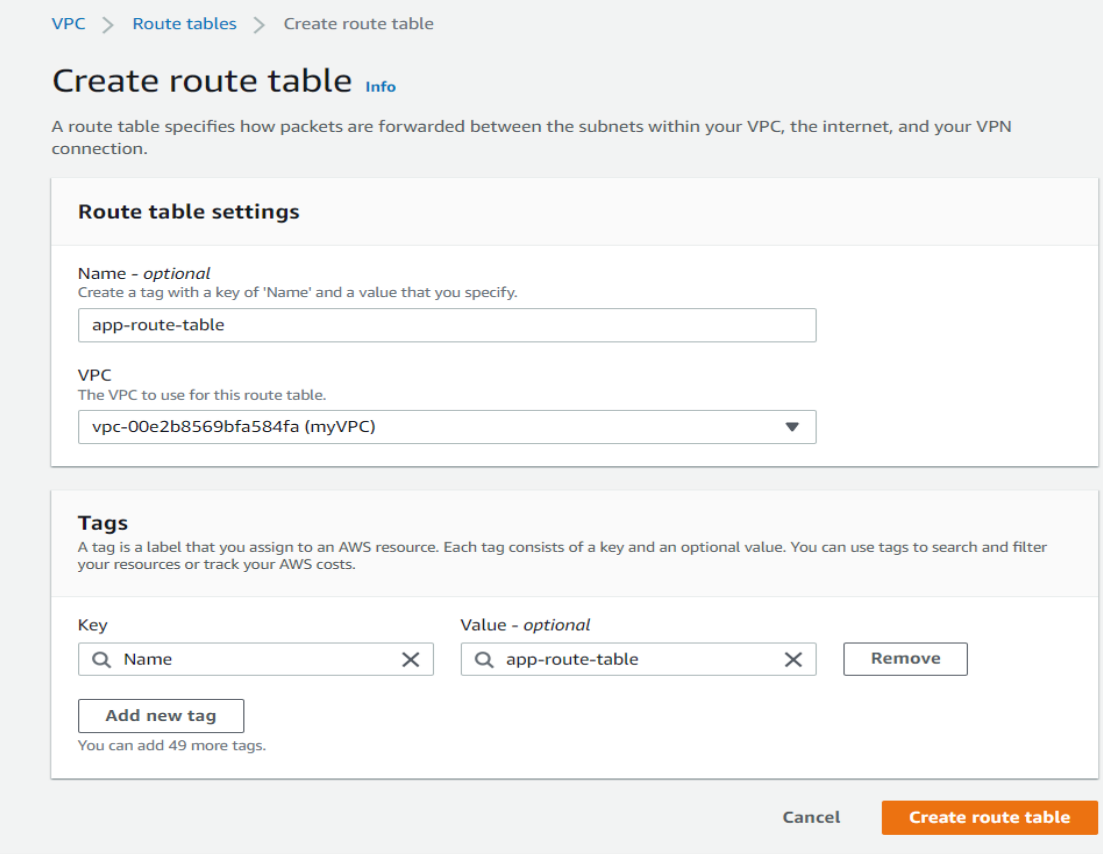
Although your subnets should be linked to the main route table by default, you have to ensure each tier has its routing table.

For the Subnets of Application and Database Tiers, you will create two Private Route Tables for each Subnet. Once you are done creating them, you will have to make a public route table for your two Presentation (Web) Subnets.

So for 2 public subnets we create one route table

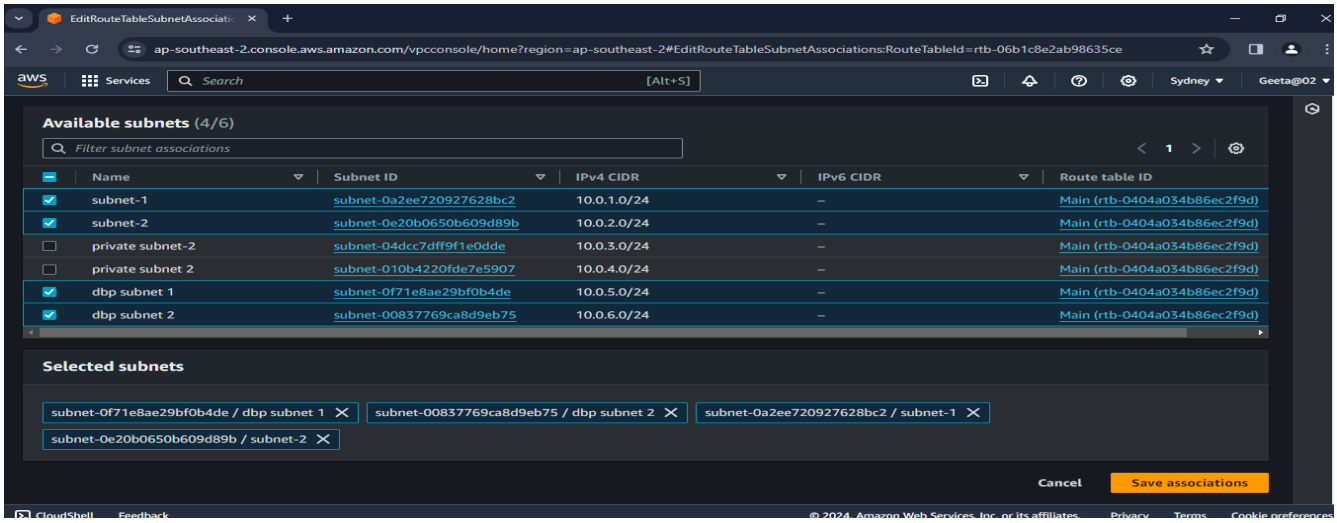
for 2 private subnets of application tire -one route table

for 2 db private subnets one route table.



Then we need to make subnet associations:

Select Subnet associations and then Edit Subnet associations inside your new route table. For each table, you can associate the different subnets. Click on Save association at the end of each configuration

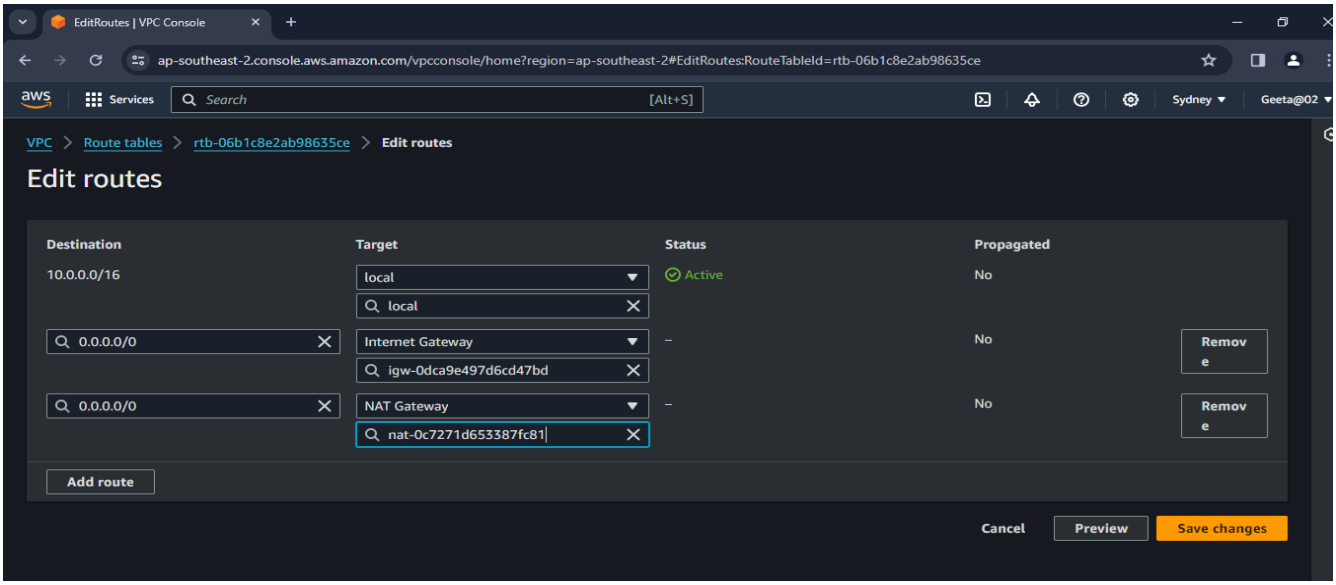


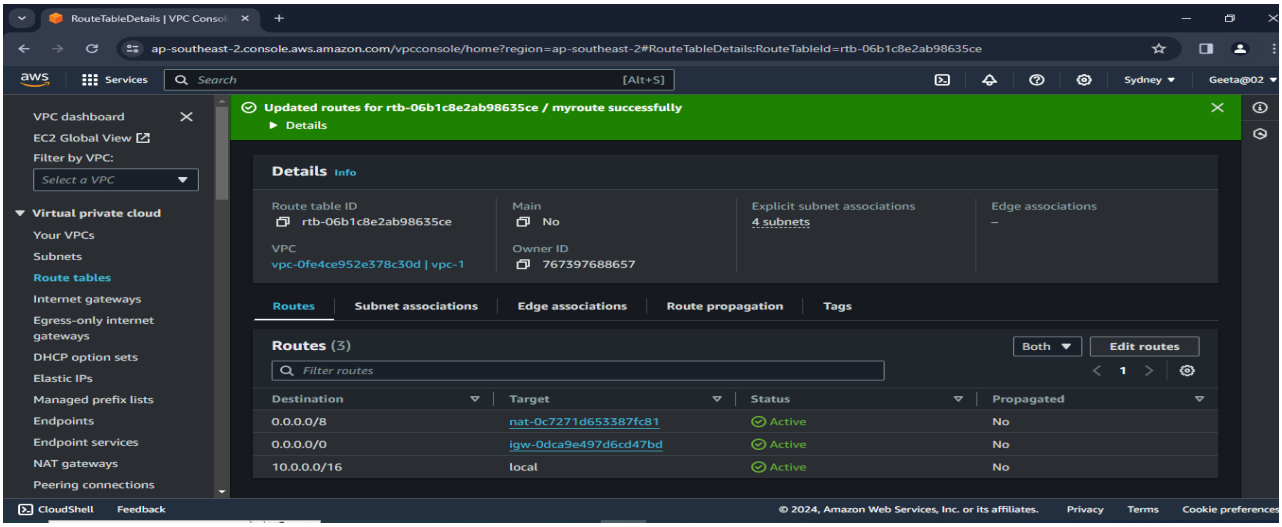
Next we need to do edit routes.

Select the routes and click on edit routes for the

route created for public subnets attach internet gateway.

Routes creates for private subnets attach internet gateway and nat gateway and save the changes.





STEP-6: We should create launch template.

You can build a launch template that has the configuration details to launch an instance. You can use launch templates to store launch parameters so that you do not have to specify them every time you launch an instance. For example, a launch template can contain the instance type, AMI ID, and network settings that you usually utilize to launch instances.

We need to create 2 launch templates – one for the Presentation (web tier) and the other for the Application Tier.

PROCESS:

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

web-lt

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Web Tier Launch Template

Max 255 chars

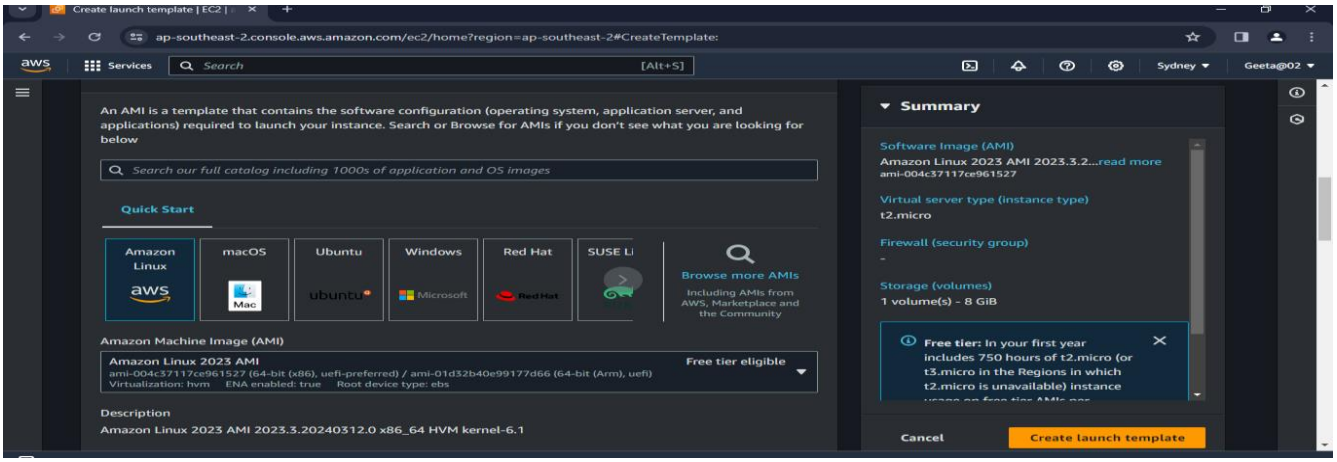
Auto Scaling guidance Info

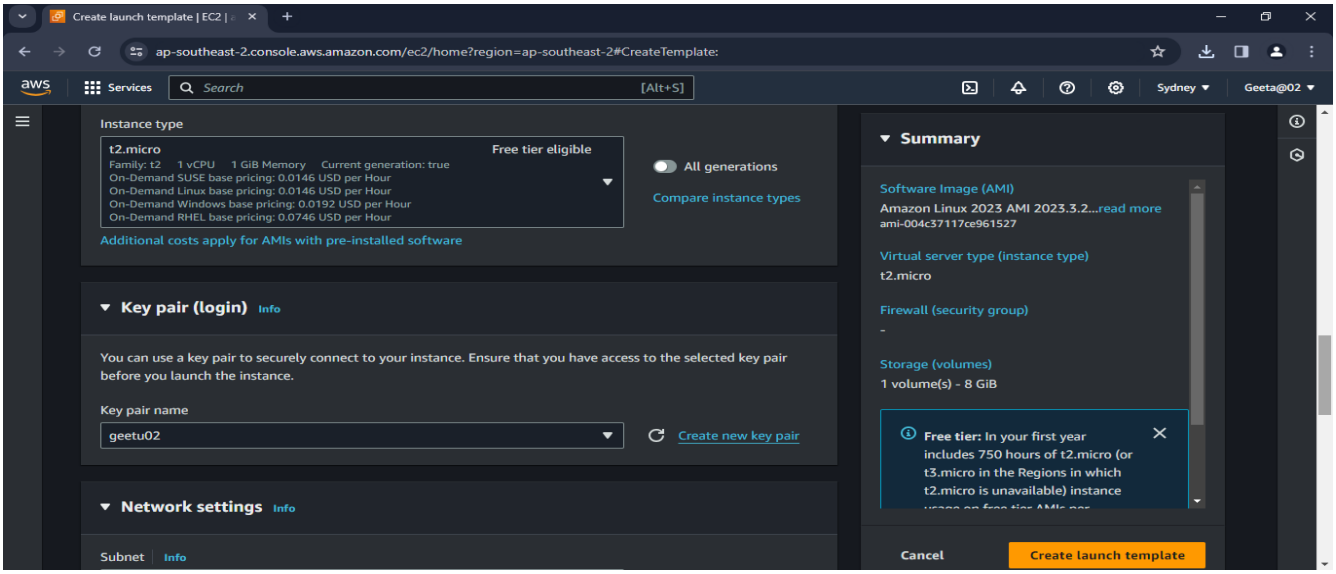
Select this if you intend to use this template with EC2 Auto Scaling

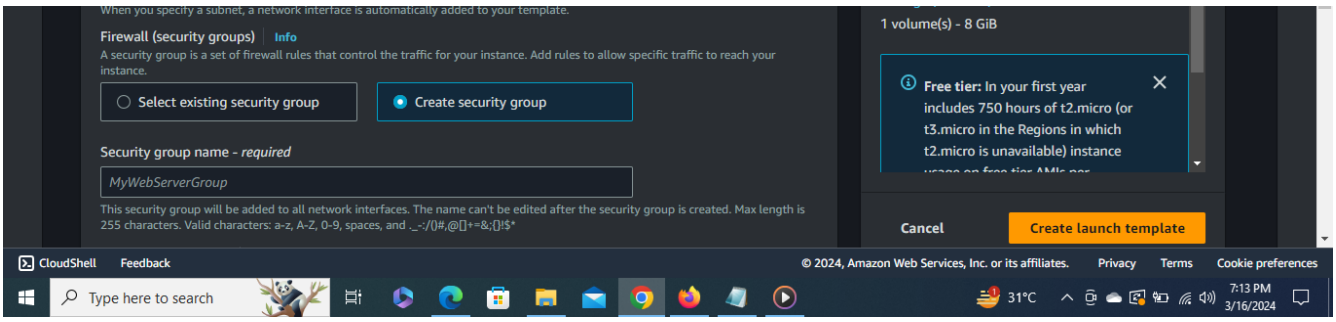
☑ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

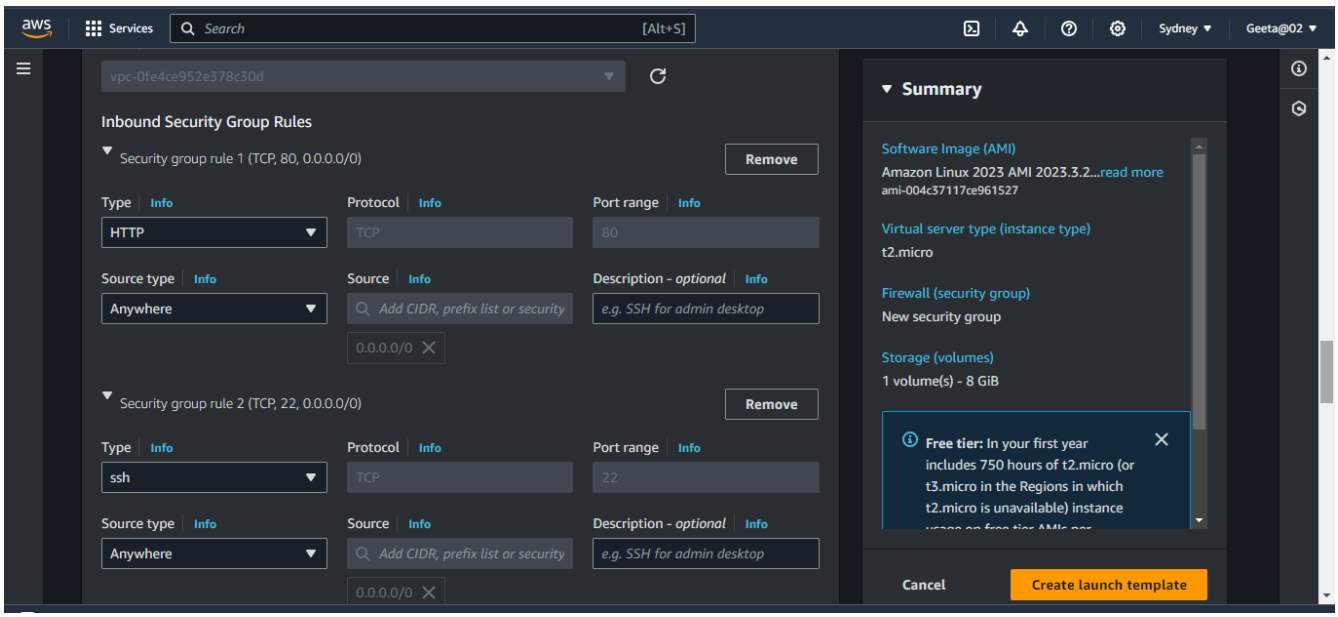
▶ Template tags

▶ Source template

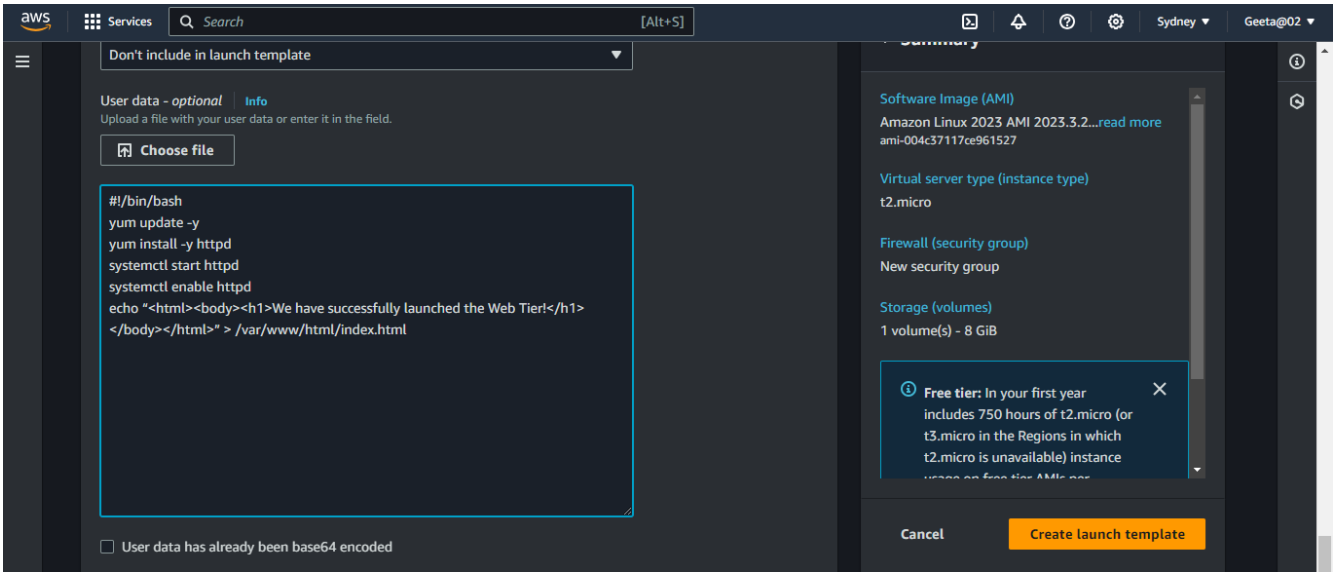






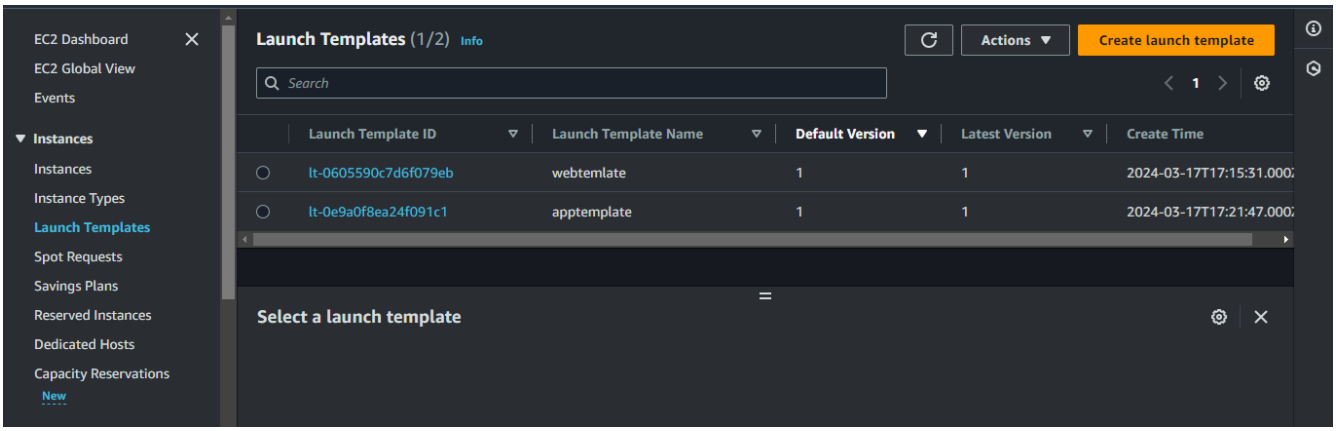


next we need to go advanced settings scroll down and write the code in user data.

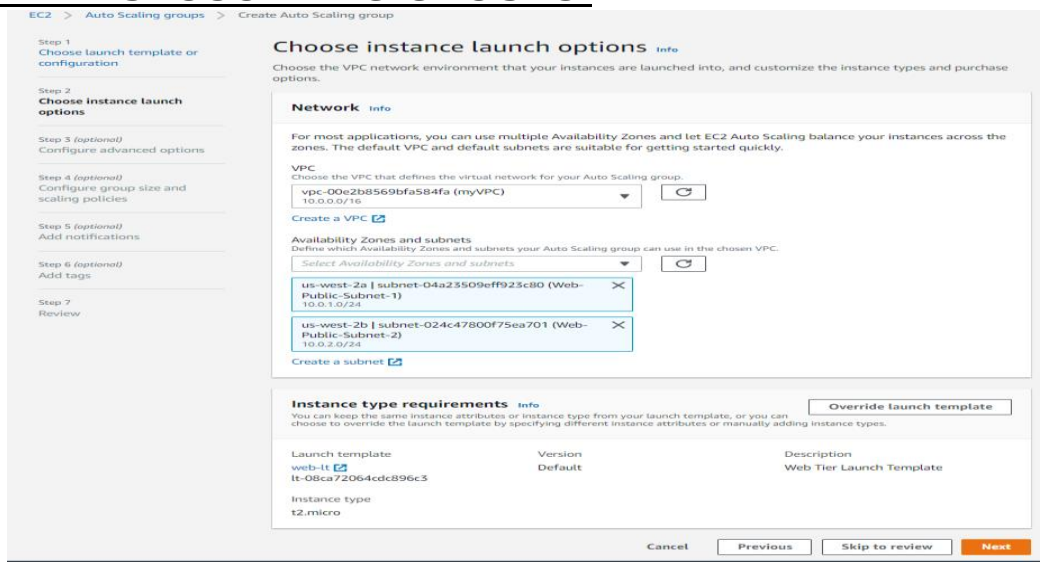


And now we are ready and click on launch template.

Now we need to create another template in the same way just keep the name as app template.

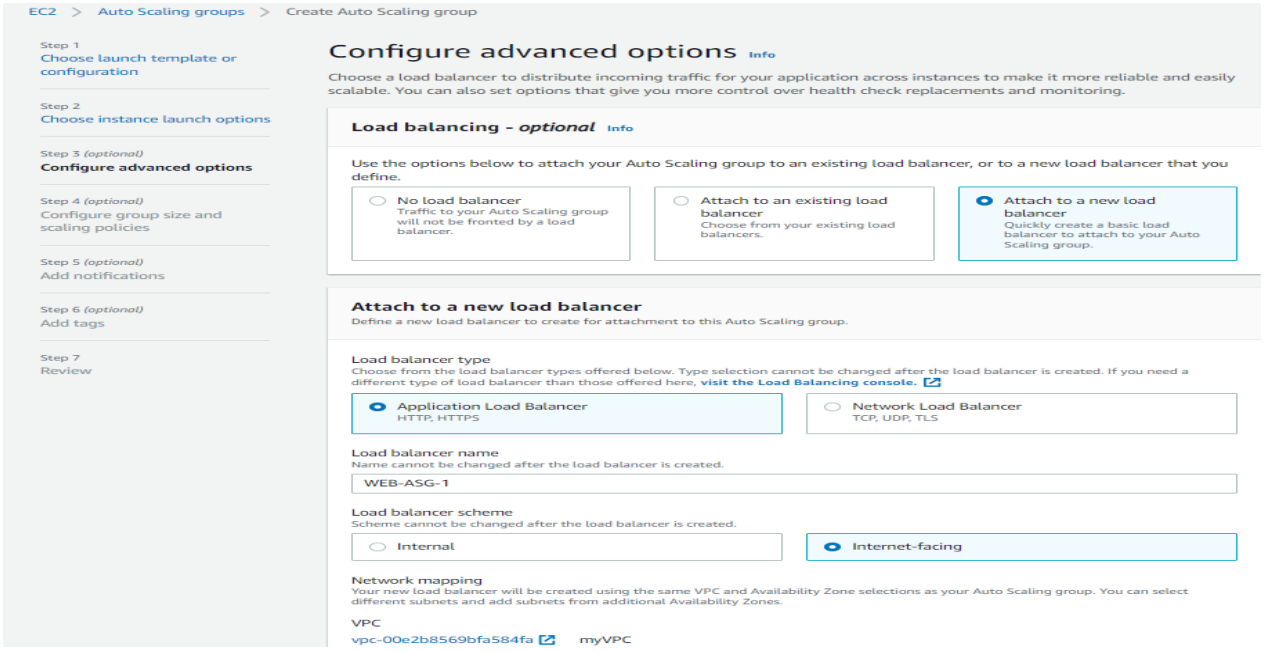


STEP-7:CREATE AUTOSCALING GROUPS:



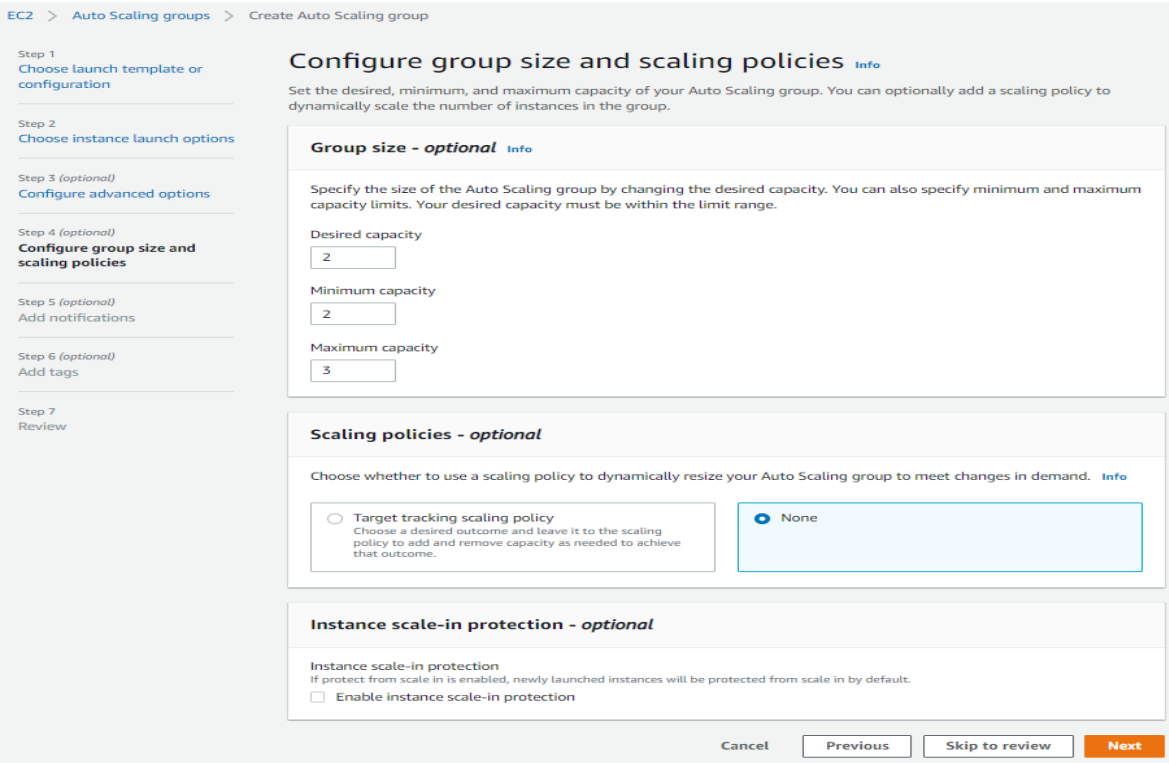
We need to create two autoscaling groups.

Then we need to attach the application load balancer in the next step.



In next step we need to select the subnets create the create a target group.

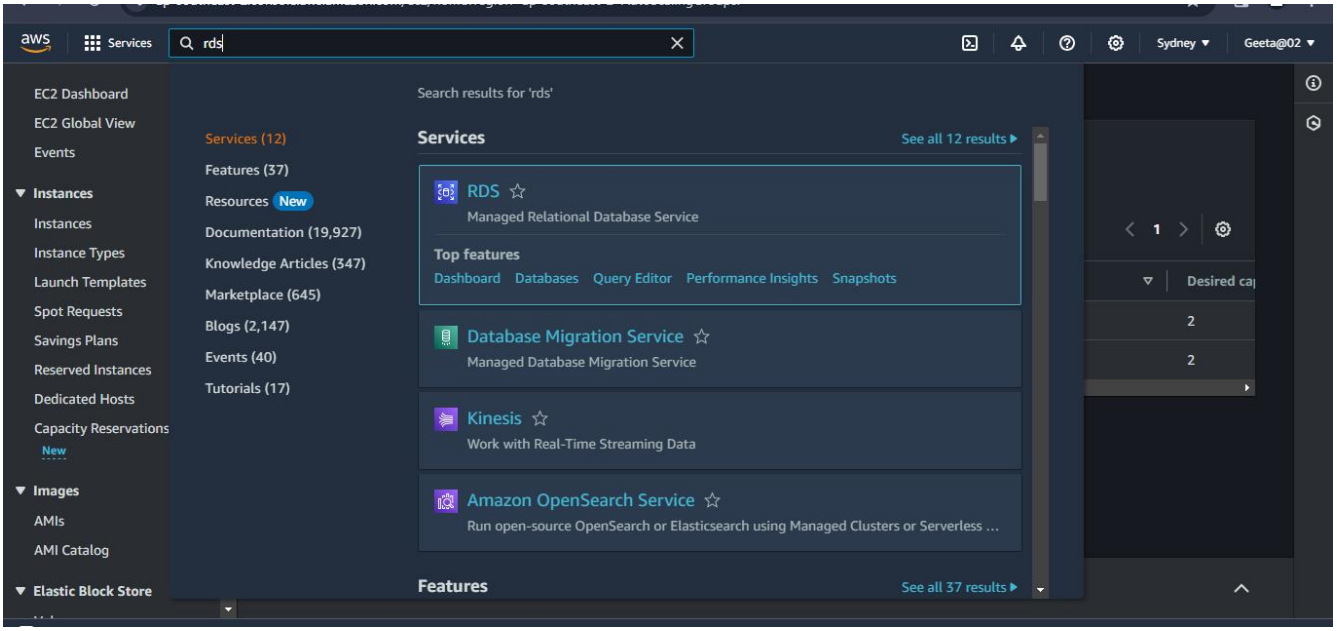
Now we have to configure the group size and scaling policies



after this a autoscaling group is created in the same process we need to create another auto scaling group with internet configuration for application tire.

STEP-8: CREATE A DATABASE.

Go to the RDS Service on your AWS Management Console.



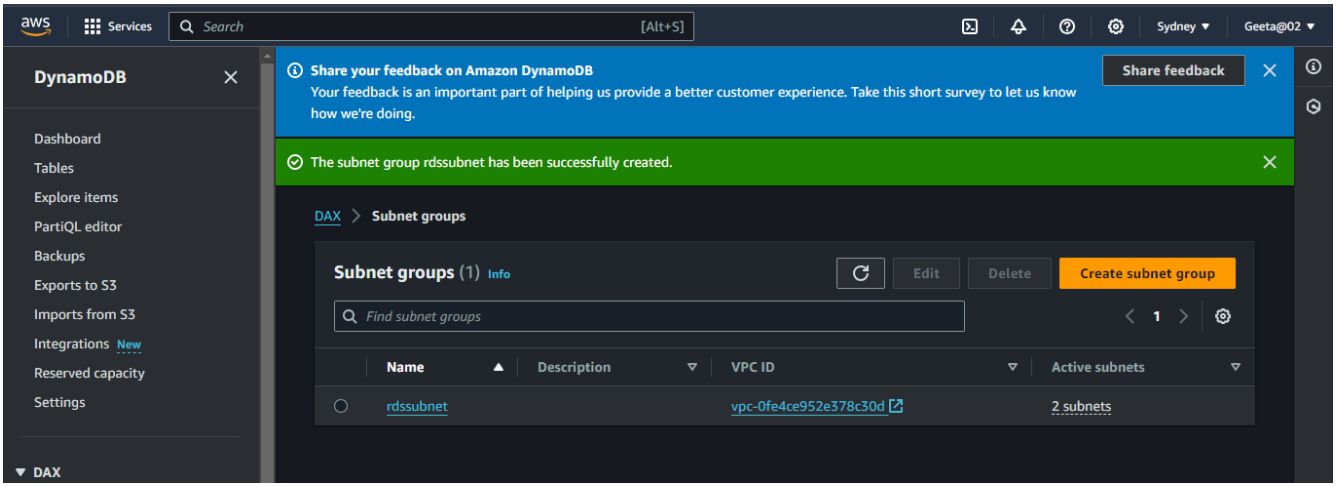
Click on create database. We will use the Standard create

Creation Method: Standard

- Database Engine: MySQL
- Edition: MySQL Community 8. x
- Template: Free Tier
- DB Instance Identifier: Give a friendly name for your
- Master username: Feel free to change or use “admin”
- Password: auto-generate
- DB Instance Size: We are sticking with the Free Tier so DB.t2. micro
- Storage Type: General Purpose
- Storage: 20GB
- Storage Autoscaling: Enabled
- Availability & Durability: For the tutorial and using the Free Tier, this will be greyed out. For Production use cases, ensure you have a Multi - AZ set up as well as a large DB Instance Size, etc.
- VPC: Choose the VPC we made for this series and also select the security groups create if not there create.

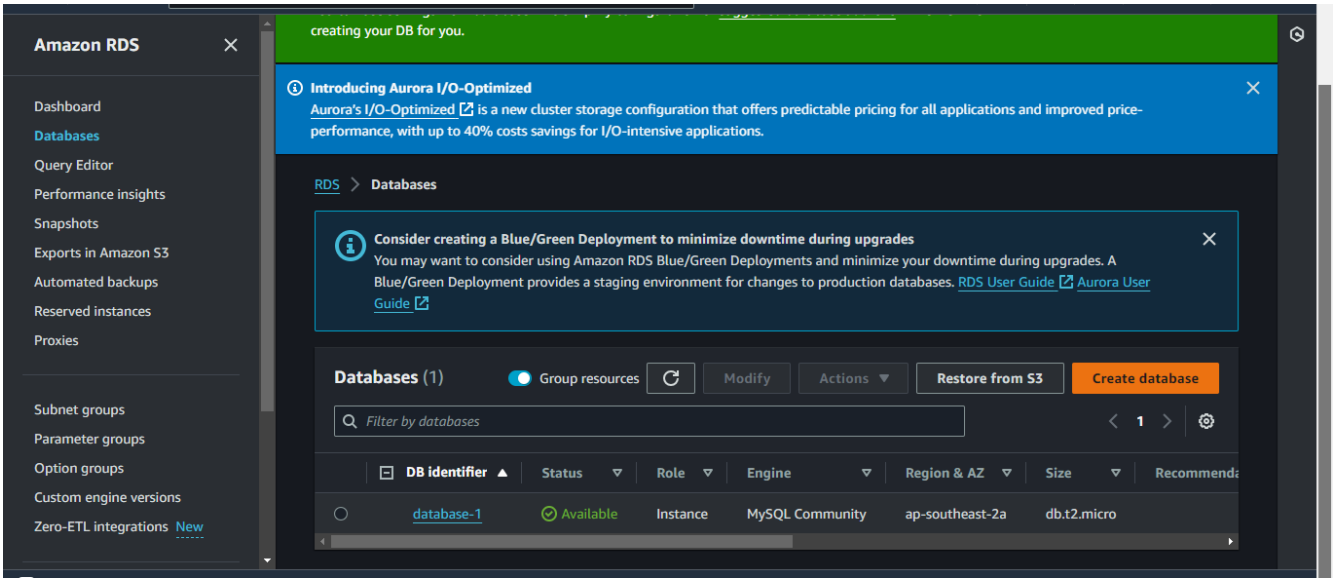
Public Access: No

- VPC subnet Group: Select the Subnet Group we created.



- Availability Zone: Select the zone where you want your DB to reside
- Authentication: A password is fine for our purposes.

Now click on create database it may take a while and our database Is created.



After it comes to available status it is created and ready to use .And the three tire architecture is successfully done .

To check its working :

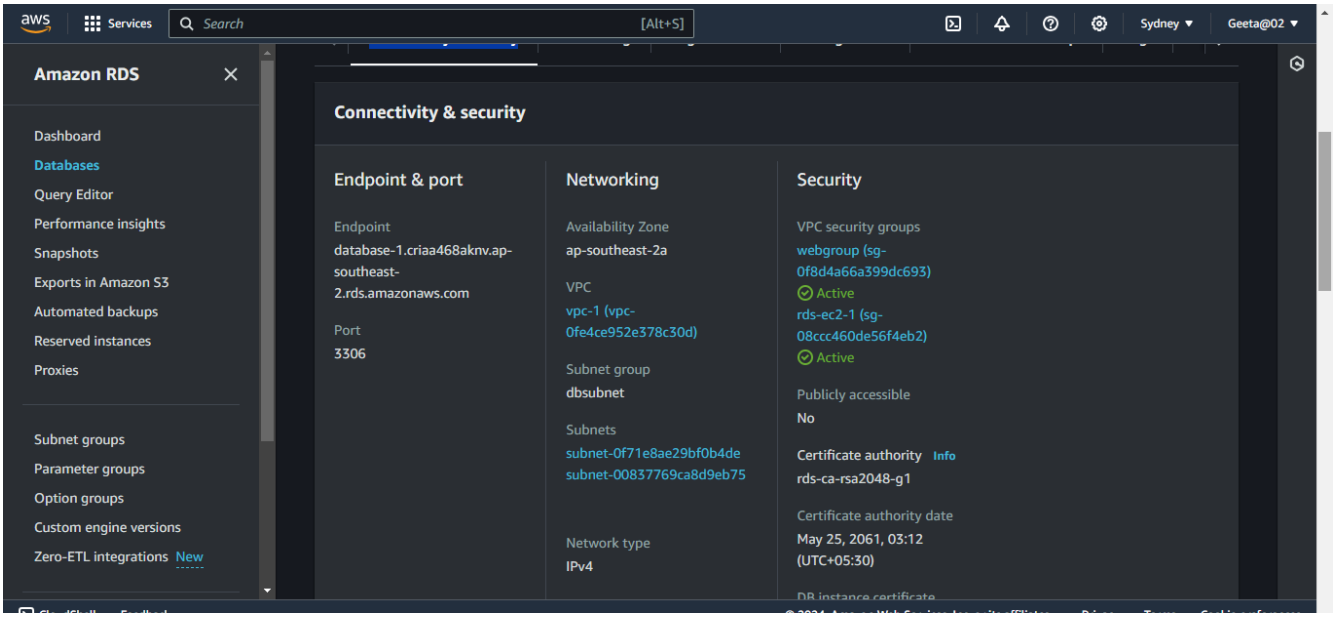
We have instances launched already automatically now choose any one of the instance which are connected in db and connect to the instance to connect to the server.

Now we need to give commands.

Sudo - i

Sudo dnf install mariadb105.

Then copy the dns from the created database and paste it as follows:



mysql -h your_DB_ENDPOINT -P -u admin-p

then it asks for password enter and u get connected to MYSQL.