PROJECT – 1

NAME : P.Mohan Sai Nikil

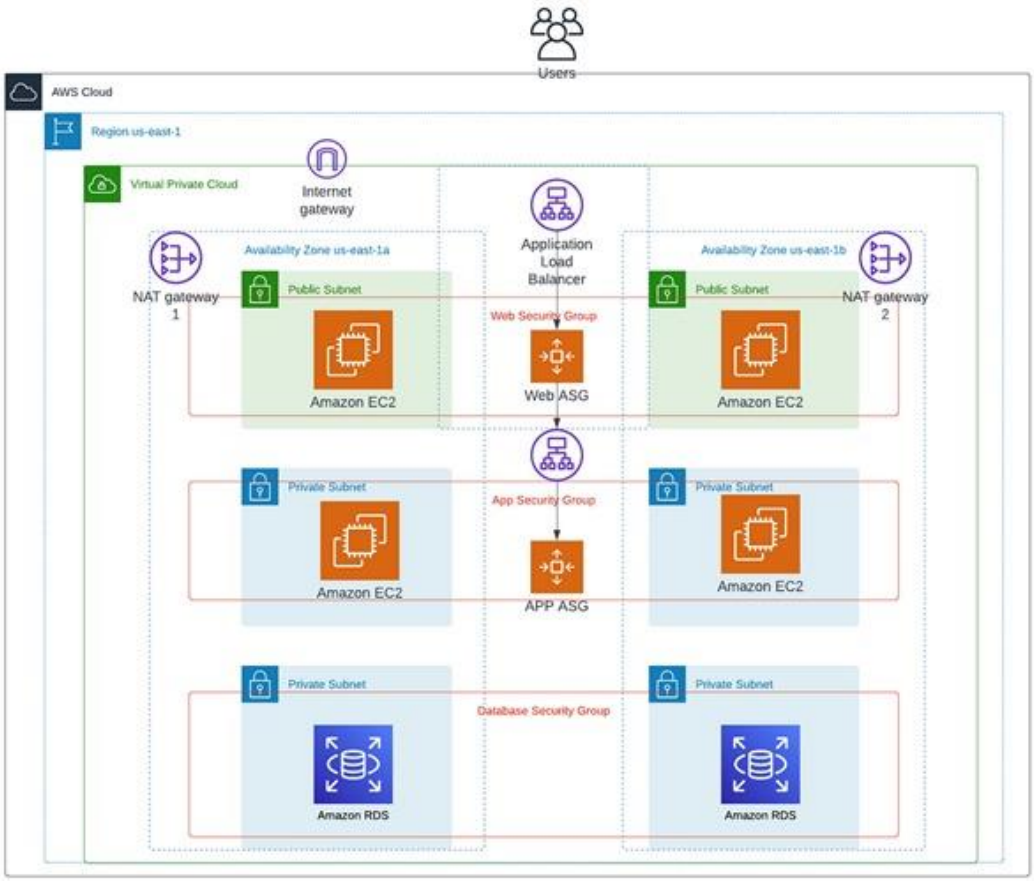
Email : mohanchowdary1404@gmail.com

Ph No : 7013020529

COURSE: AWS DEVOPS

MODULE: AWS 3-Tier Architecture.

TRAINER: Mr. MADHUKAR REDDY



What is a 3-Tier Architecture?

A three-tier architecture comprises three layers, namely the presentation tier, the application tier, and the data tier. The presentation tier serves as the front-end, hosting the user interface, such as the website that users or clients interact with. The application tier, Commonly referred to as the back-end, processes the data. Finally, the data tier is responsible for data storage and management.

Benefits of a 3 Tier Architecture

1. Scalability: Each tier can scale independently, allowing organizations to optimize their resources and minimize costs.

2. Reliability: Each tier can be replicated across multiple servers, improving application availability and reliability.

3. Performance: By dividing the application into separate layers, 3-tier architecture reduces network traffic and enhances application performance.

4. Security: Each tier can have its own security group, allowing different oragnizations to implement customized security measures for each layer.

Tier 1: Creating a VPC and Subnets

Using the architecture diagram as a reference, we will need to satrt by creating a new VPC with 2 Public subnets and 4 Priavte Subnets.

Log into the AWS management console and click the create VPC button.

We are going to create a VPC with multiple public and private subnets, availability zones, and more, so let's choose "VPC and more".

Name your VPC. I am using the auto-assigned IPV4-CIDR block of "10.0.0.0/16." Choose these settings:

no IPV6

default Tenancy

2 Availability Zones

2 Public subnets

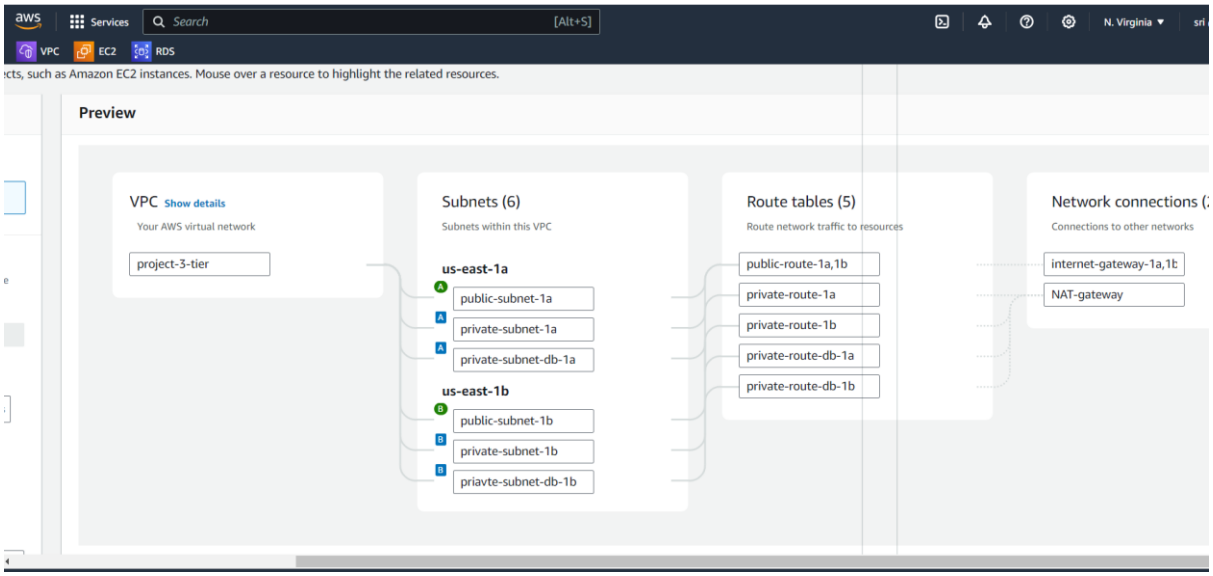
4 Private subnets

Next, for Nat gateway choose "in 1 AZ," none for VPC endpoints, and leave the Enable DNS hostnames and Enable DNS resolutions boxes checked.

Before we create the VPC expand the customize AZ's and customize subnets CIDR block tabs.

Click the "Create VPC" button.

The diagram below highlights the route that your new VPC will take.



Tier 2: Creating a Web Server Tier

Next we will create our first tier that represents our front end user interface (web interface). We will create an auto scaling group of EC2 instances that will host a custom webpage for us. Start by heading to

Next we are going to launch an EC2 instance.

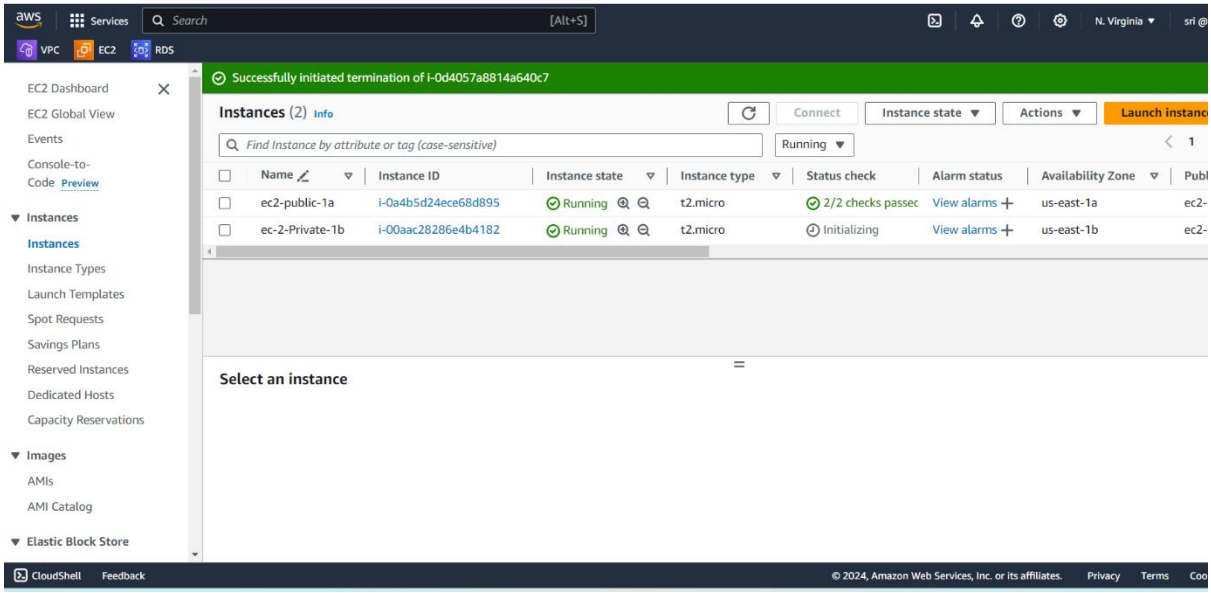
Name your instance and select AMI. Please use Amazon linux2; I used the updated vesrion and was not able to ssh into it

Select the kay pair that you will use, and make sure to select your new VPC and the correct subnet. Auto-assign IP should be enabled.

Create a new security group. For inbound security group rules, add rules for ssh, HTTP, and HTTPS from anywhere. This is not standard or safe, Practice, but for this demonstration it is fine.

Launch your new instance!

Once your instance is up and running, copy the public IP address and paste it into a web server.



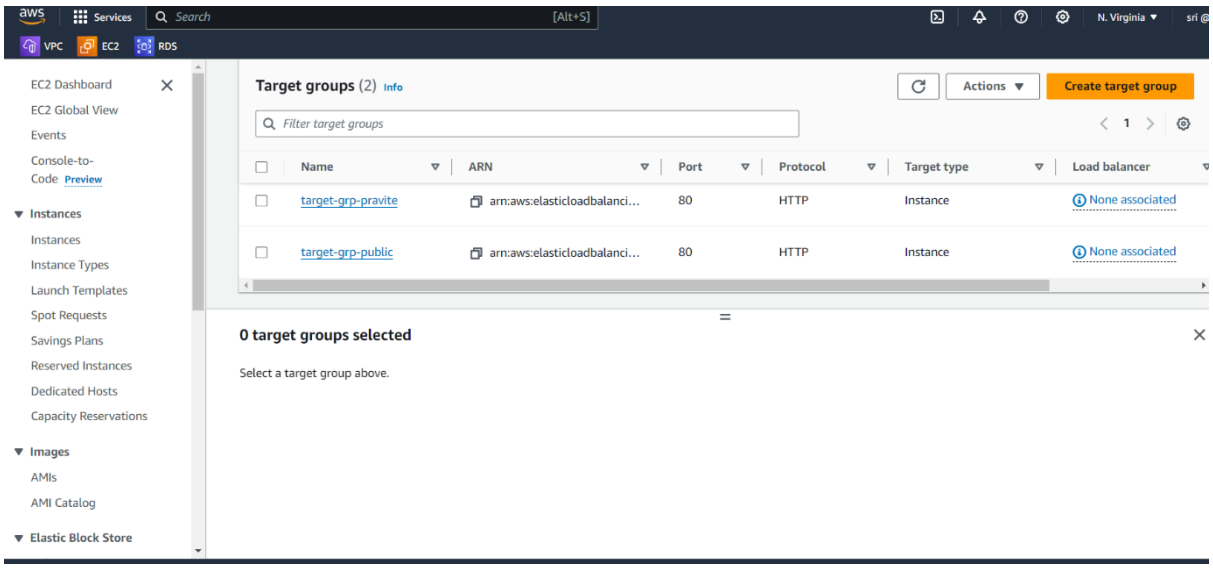
Create load balancer:

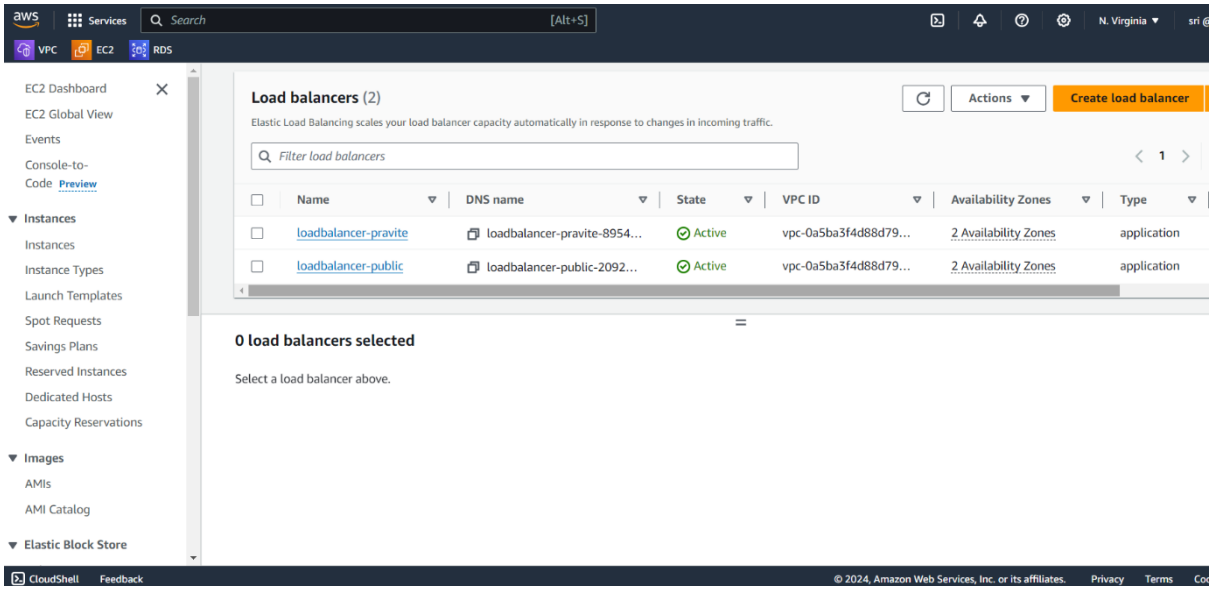
To create a load balancer first you need to create a target group.

Create Target Group:

Now we are given to option to allocate a load balancer for our ASG. A load balancer will distribute the load from incoming traffic across multiple servers. This helps with availability and Performance.

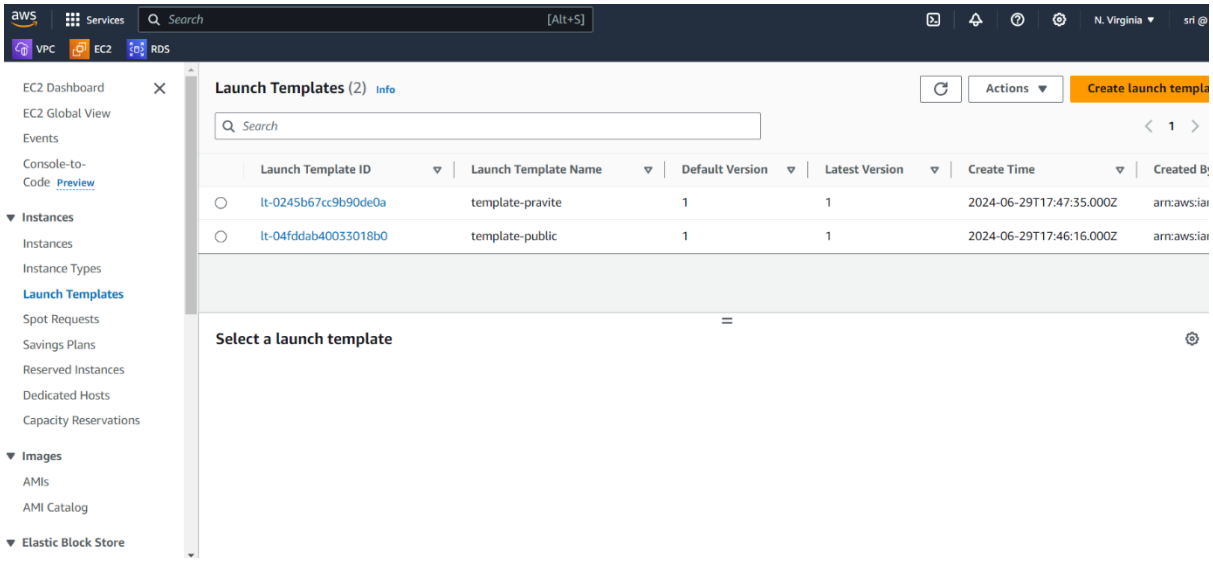
Select “Attach to a new load balancer” and “Application load balancer,” name your load balancer , then select “Internet facing” as this is for our web tier.





Create a launch template

We are going to create a launch template. Before we do this we will need to define a launch template; this template will outline what resources are going to be allocated when an auto scaling group launches on-demand instances. Under the EC2 dashboard. Select Launch Templates, and click the “Create launch template” button.



Name your template, and check the box to “provide guidance”.

Use our recently AMI t2.micro instance type and select your key pair

For the firewall, use “select existing security group,” and make sure the security group(SG) that we created for the web tier is selected. Under Advanced network configuration, enable Auto-assign public IP.

We are going to leave the storage options alone for now. Click on the Advanced details tab, Scroll down, and enter the same script as we did earlier for our EC2 instance.

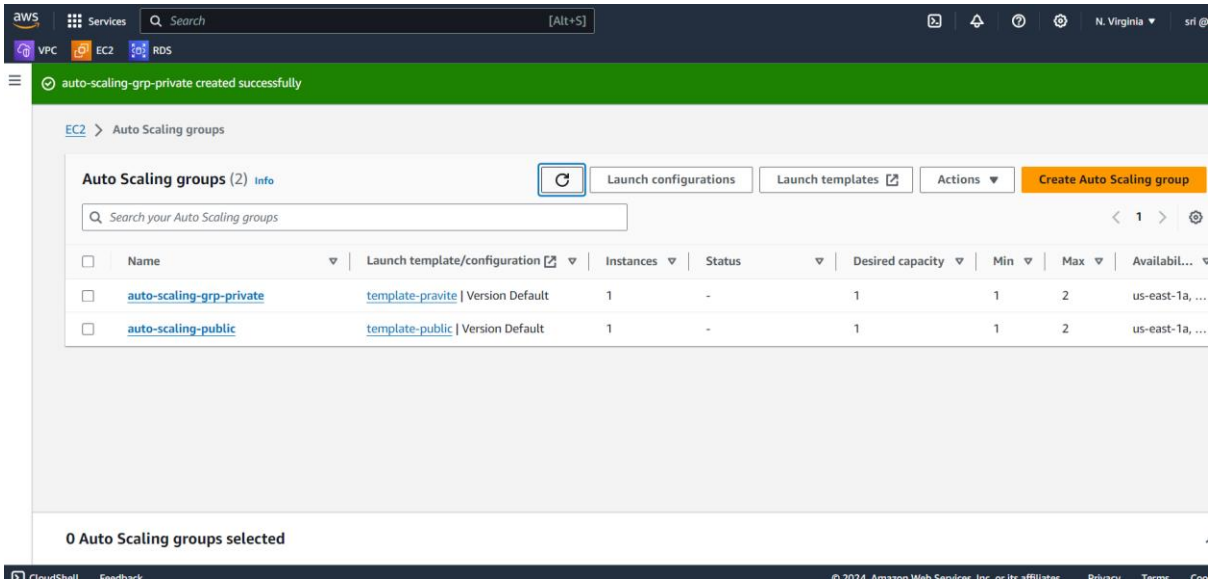
Click the “Create launch template” button.

Navigate to the Autoscaling tab at the bottom of the EC2 dashboard. Click “Create auto scaling group.” The launch template that we just finished creating is the template that our auto scaling will use to launch new EC2 instances when scaling up.

Name your auto scaling group (ASG). choose the launch template that you created, then click the Next button.

Under Networking, make sure to select the VPC that you created earlier, then also under availability zones and subnets select the public Subnets that were created; yours may differ.

Click the Next button.



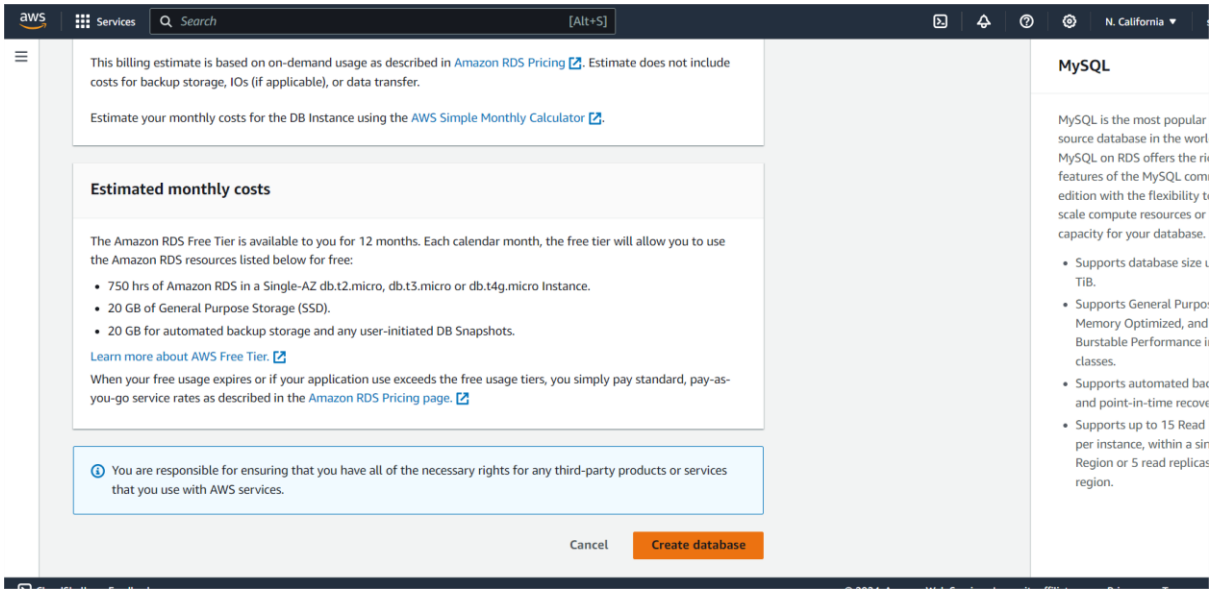
Now we are given to option to allocate a load balancer for our ASG. A load balancer will distribute the load from incoming traffic across multiple servers. This helps with availability and performance.

Select “Attach to a new load balancer” and “Application load balancer,” name your load balancer, then select “Internet facing” as this is for our web tier.

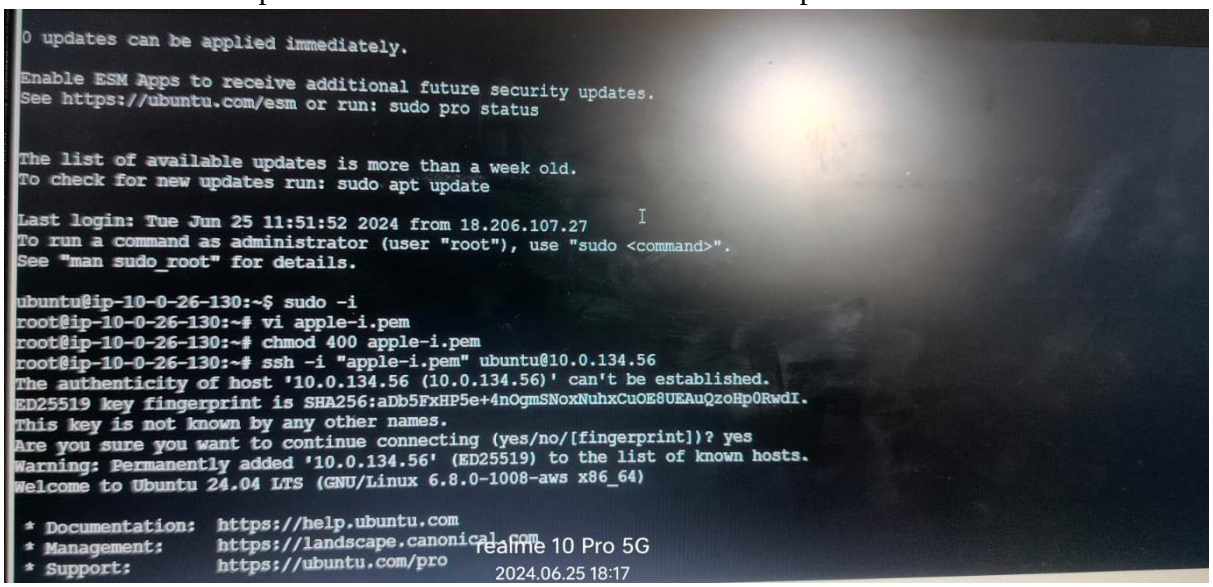
Tier 3: Creating a Database Tier

To create a database first you need you to create a subnet group

Create the database



First connect to the public server and from that connect to the private Server



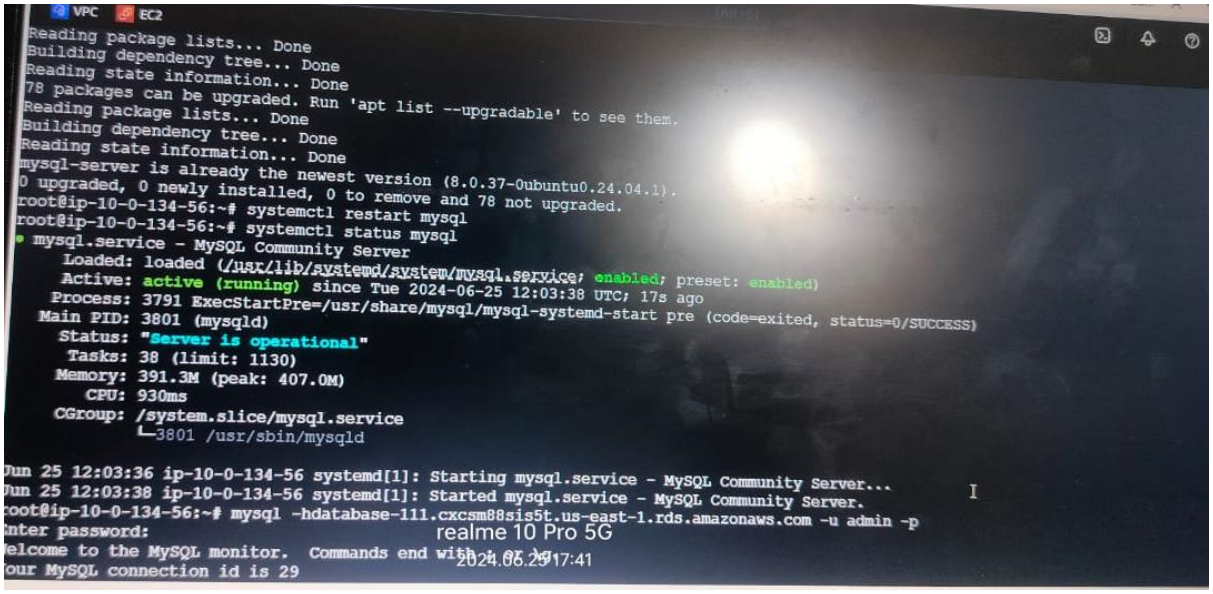
Install the mysql and go to root (sudo -i)

Vi apple -i.pem

Chmod 400 apple -i.pem

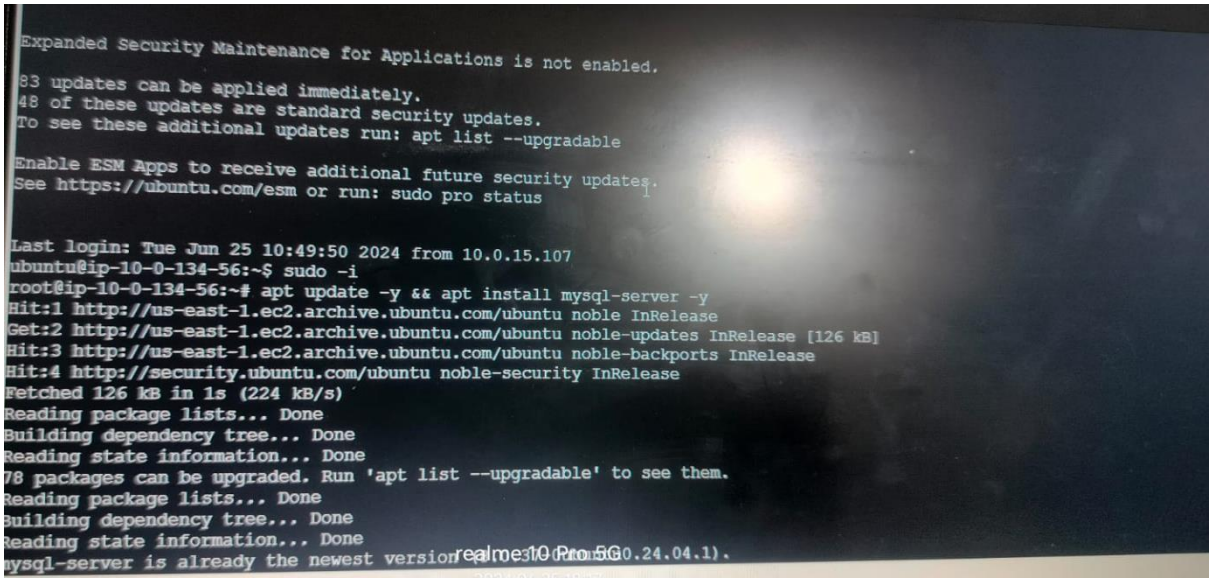
Insert, esc shift:X

Connect to pravite from public and go to root



System restart mysql

System status mysql

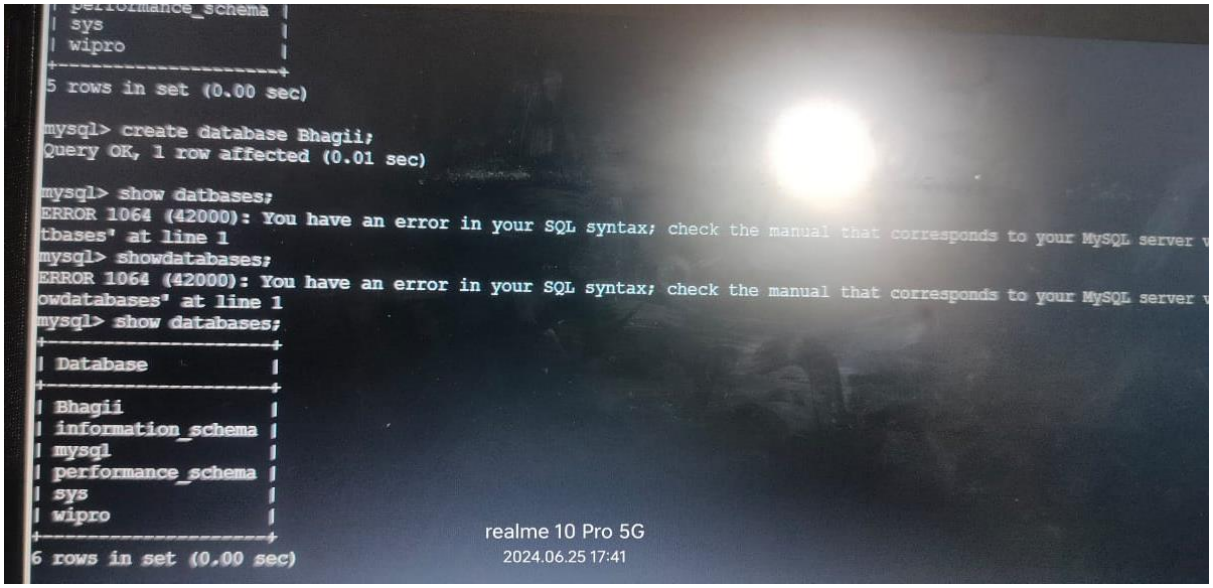


Mysql -h end point of RDS -u admin -p

Enter password: admin 123456

Mysql> show databases;

Mysql> create databases google;



Mysql> show databases;

Mysql> create databases google;

Show databases;