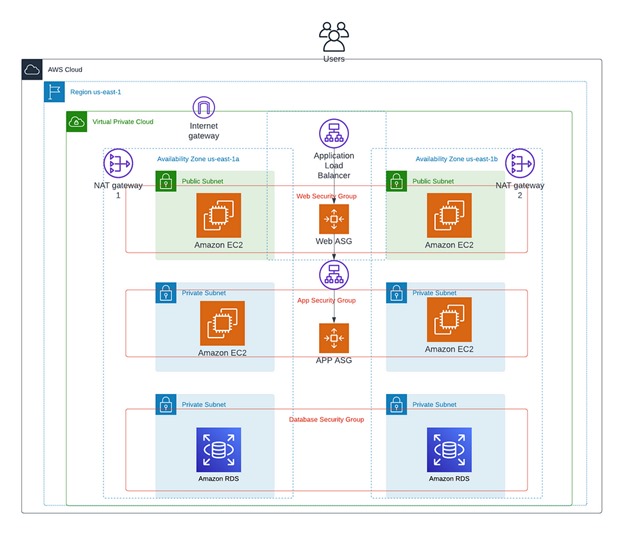
**Building a 3-tier web application architecture with AWS**



**Intro**

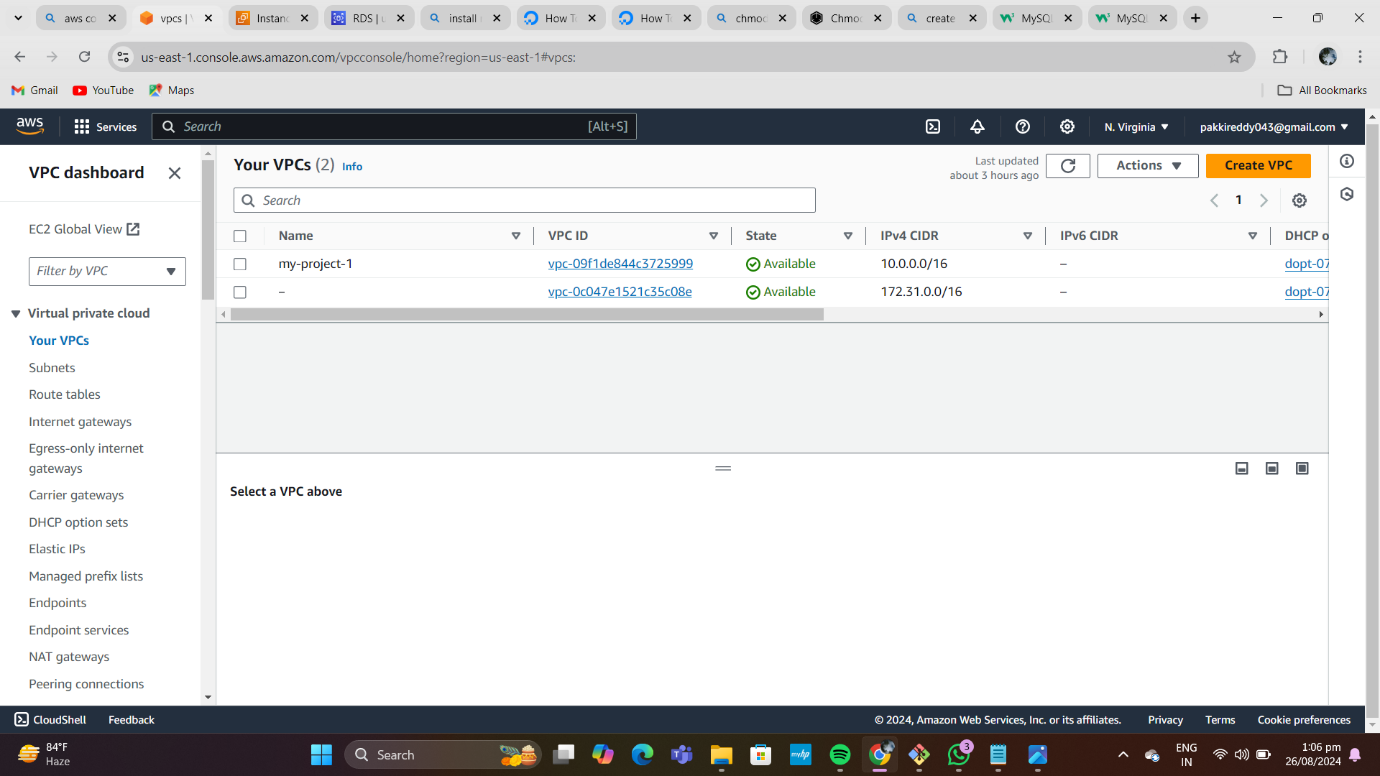
When building a cloud-based application, the underlying architecture and environment are just as critical as the application itself. There are many considerations when deciding on the proper architecture of your app:

1. **Scalability:**How easily and/or frequently does the app need to scale up or down? How much value do you put into not having to constantly micro-manage and monitor resource usage?
2. **Availability:** How readily available is your app? How important is being able to go through long periods of time without failures? If failure does occur in a part of your app, how vulnerable is the rest?
3. **Security:** How secure is your app? How does your app handle security permissions for different parts of your app? If an attack happens in one part of your app, how vulnerable is the rest?

**Required AWS Services:**

1. Virtual Private Cloud ( Subnets – 6, Internet gate way – 1, Route tables – 2, Nat gate way – 1)
2. Elastic cloud Compute (EC2)
3. Load Balancer
4. Auto- Scaling Group
5. Relational Database Service (RDS)

Create VPC :



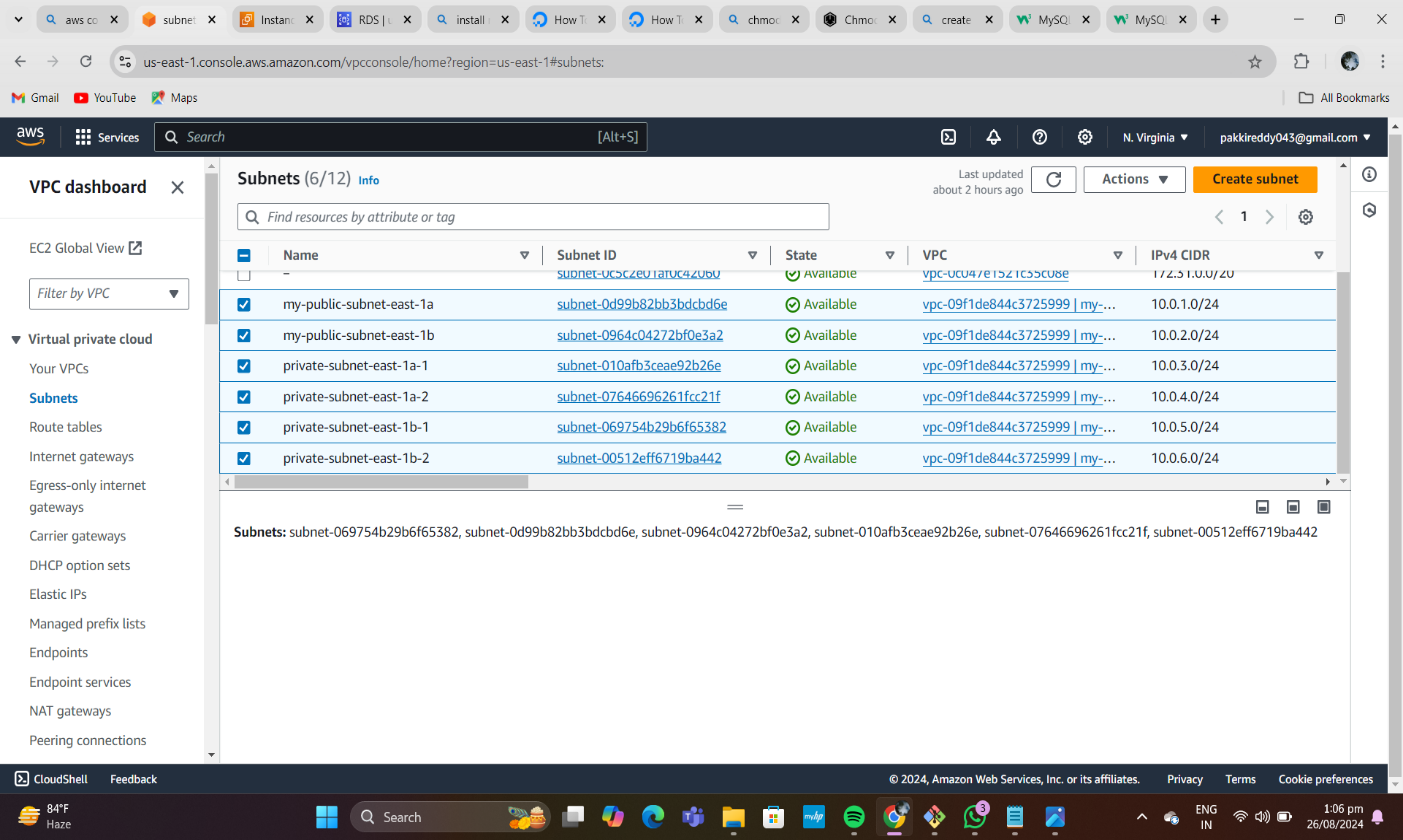
• Go to VPC dashboard click on create VPC.

• Click on VPC only and name tag as my-project-1.

• Give IPV4 CIDR (classless inter domain routing) as 10.0.0.0/16.

• Click on VPC, it is created.

CREATE SUBNETS :



• Create 6 subnets (2-public, 4-private).

• Create first subnet.

• Click on subnet, click on create subnet, select our VPC (my-project-1).

• Give name tag as my-public-subnet-east-1a, select availability zone as us-east-1a. Give CIDR as 10.0.1.0/24 and created it.

• Create Second subnet.

• Click on subnet, click on create subnet, select our VPC (my-project-1).

• Give name tag as my-public-subnet-east-1b, select availability zone as us-east-1b. Give CIDR as 10.0.2.0/24 and created it.

Create Third subnet.

• Click on subnet, click on create subnet, select our VPC (my-project-1).

• Give name tag as private-subnet-east-1a-1, select availability zone as us-east-1a. Give CIDR as 10.0.3.0/24 and created it.

Create fourth subnet.

• Click on subnet, click on create subnet, select our VPC (my-project-1).

• Give name tag as my-private-subnet-1a-2, select availability zone as us-east-1a. Give CIDR as 10.0.4.0/24 and created it.

Create FIFTH subnet.

• Click on subnet, click on create subnet, select our VPC (my-project-1).

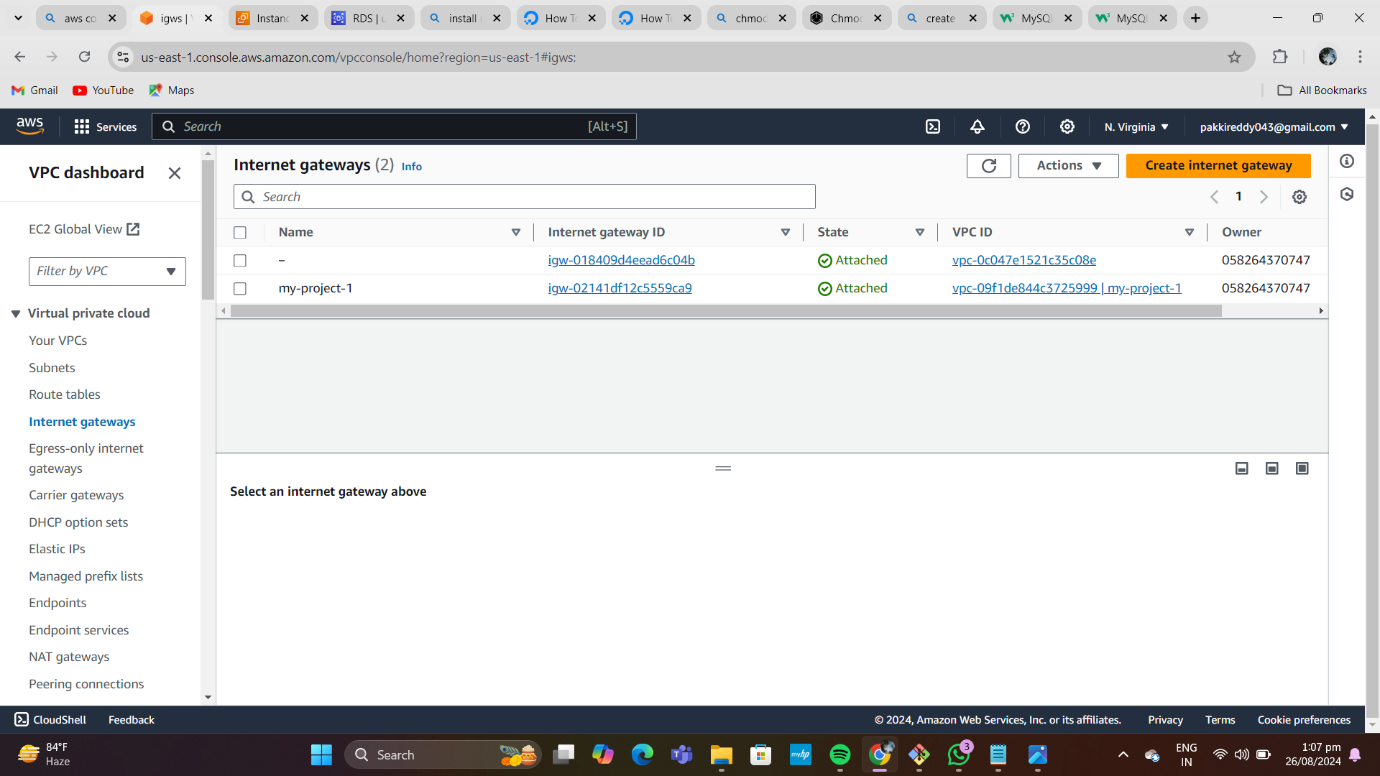
• Give name tag as my-private-subnet-1b-1, select availability zone as us-east-1b. Give CIDR as 10.0.5.0/24 and created it.

Create SIXTH subnet.

• Click on subnet, click on create subnet, select our VPC (my-project-1).

• Give name tag as private-subnet-1b-2, select availability zone as us-east-1b. Give CIDR as 10.0.6.0/24 and created it.

Create internet gateway, name tag as my-project-1



This internet gateway is attached to VPC.

• Go to actions in internet gate way and click on attach to VPC

• Select our VPC (my-project-1). Click on attach internet gateway. Click on attach internet gateway.

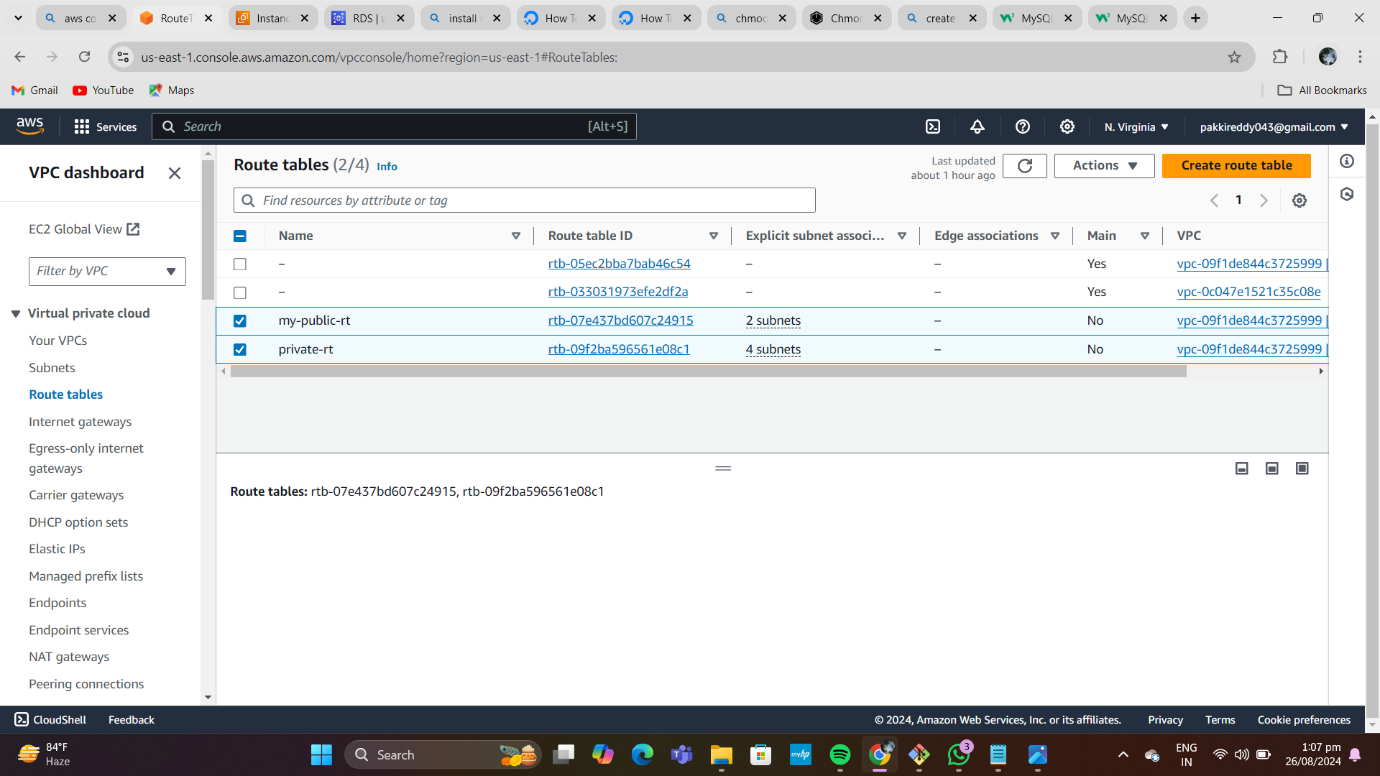
**CREATE ROUTE TABLE** :- give name as my-public-rt

• Select our VPC (my-project-1), create it. Click on route table id, open it.

• Go down click on edit subnet association.

• Select both public subnet and click on save association. Go to actions click on edit routes.

• Click on add routes give all traffic (0.0.0.0/0) and select our internet gateway, save changes.



**Create Private** **route table** name as private-rt.

•Select our VPC, (my-project-1), create it.

• Click on route table id, open it.

• Go down click on edit subnet association.

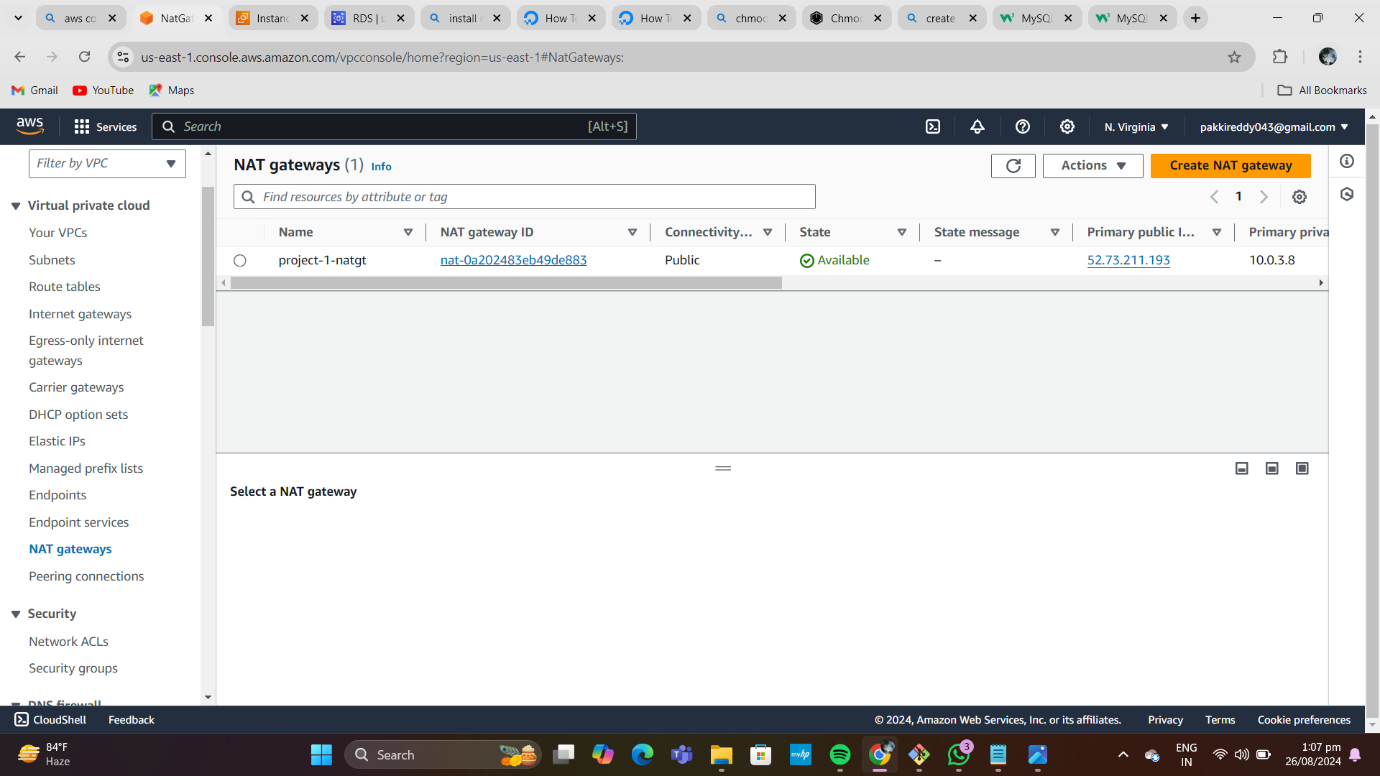
• Select all private subnet and click on save association.

• Create NAT gateway, give name as project-1-natgt.

• Select public subnet(private-subnet-1a).

• Select connectivity type as IPV4.

• Click on allocate Elastic IP



Now go to private route and click on actions.

• Click on edit routes and add route.

• Give all traffic (0.0.0.0/0) and select NAT gateway.

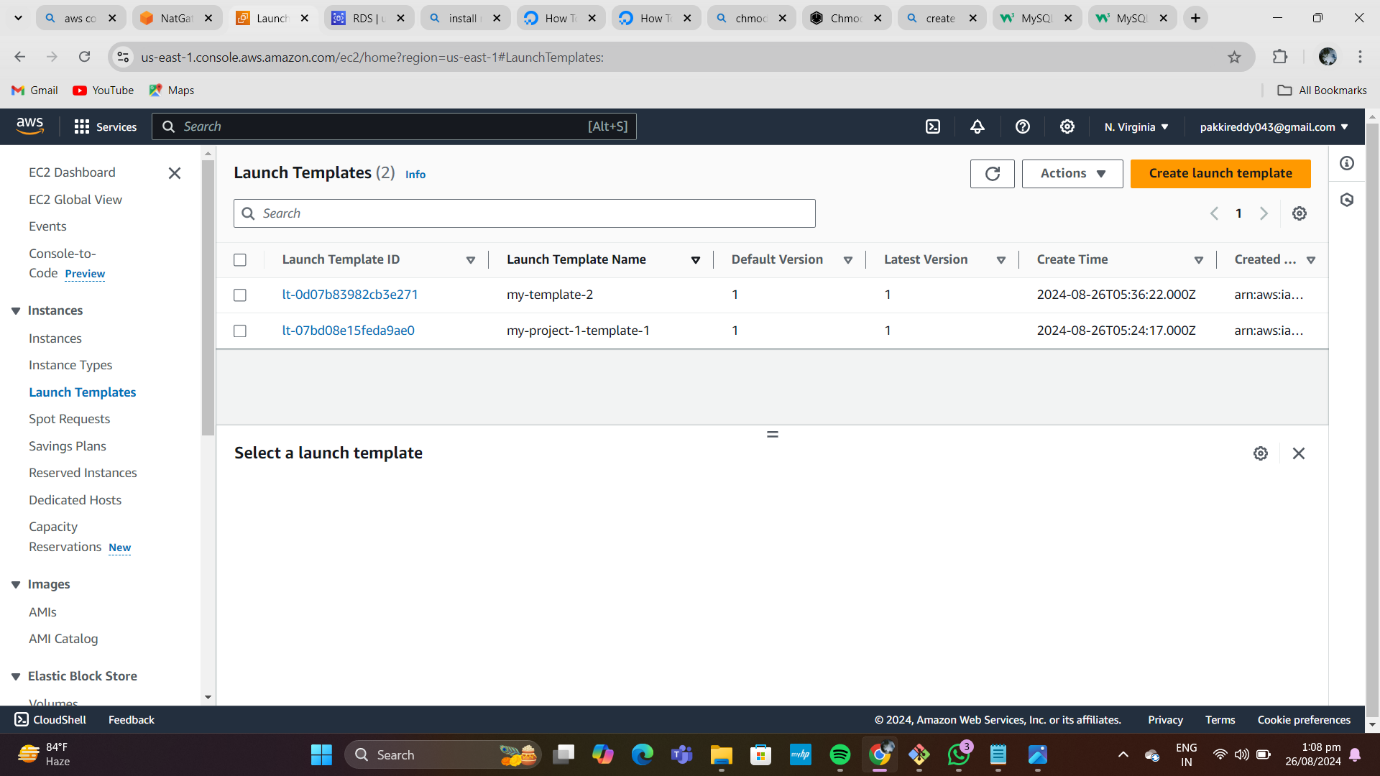
**Create Autoscaling group**.

• For creating autoscaling group we need to create an launch template.

• Click on create a launch template.

• Template name as my-project-1-template-1.

• Create Another Template as my-template-2.



• After creating two templates go to auto scaling groups

• Give name as autoscaling-public-east-1a for 1St one & Give name as private-autoscaling-group-template for 2nd one .

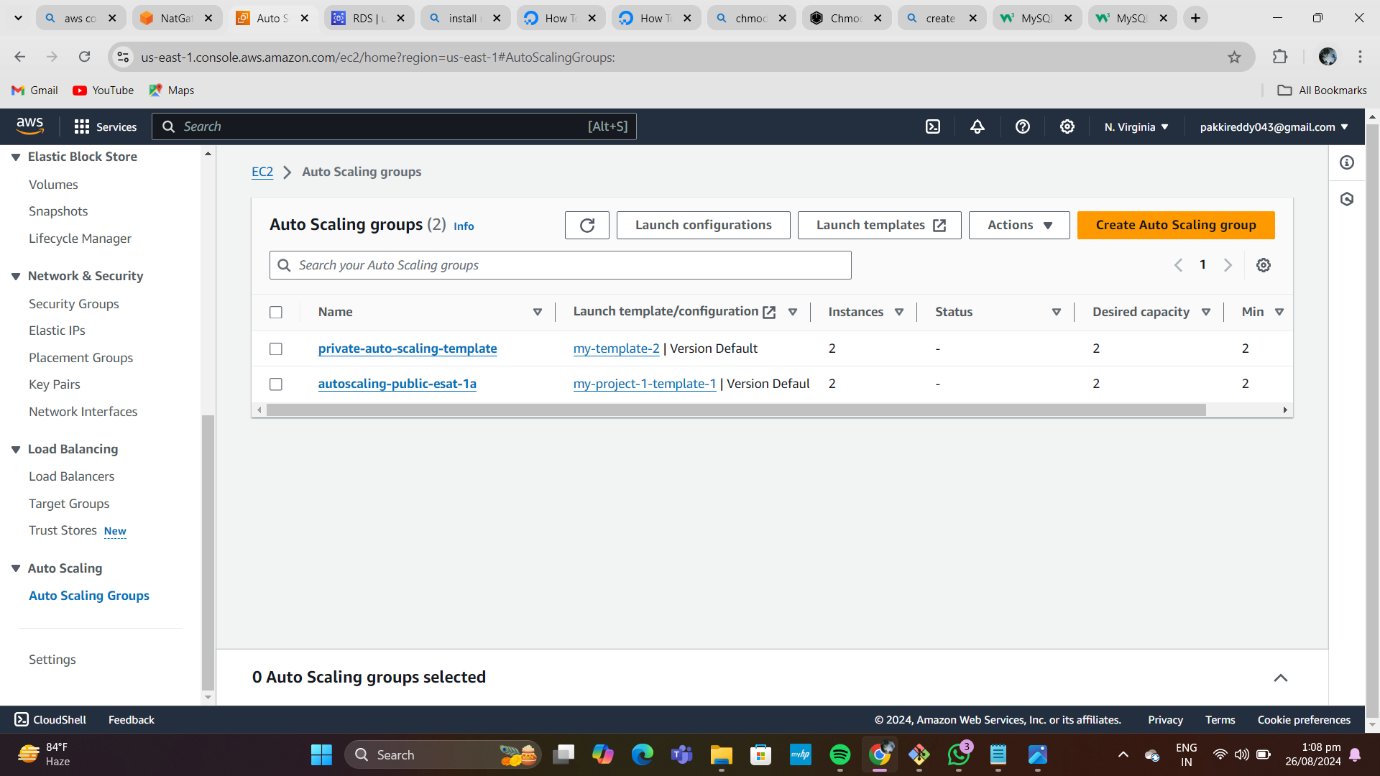
• Choose the created launch template (my-project-1-template-1) and click on next.

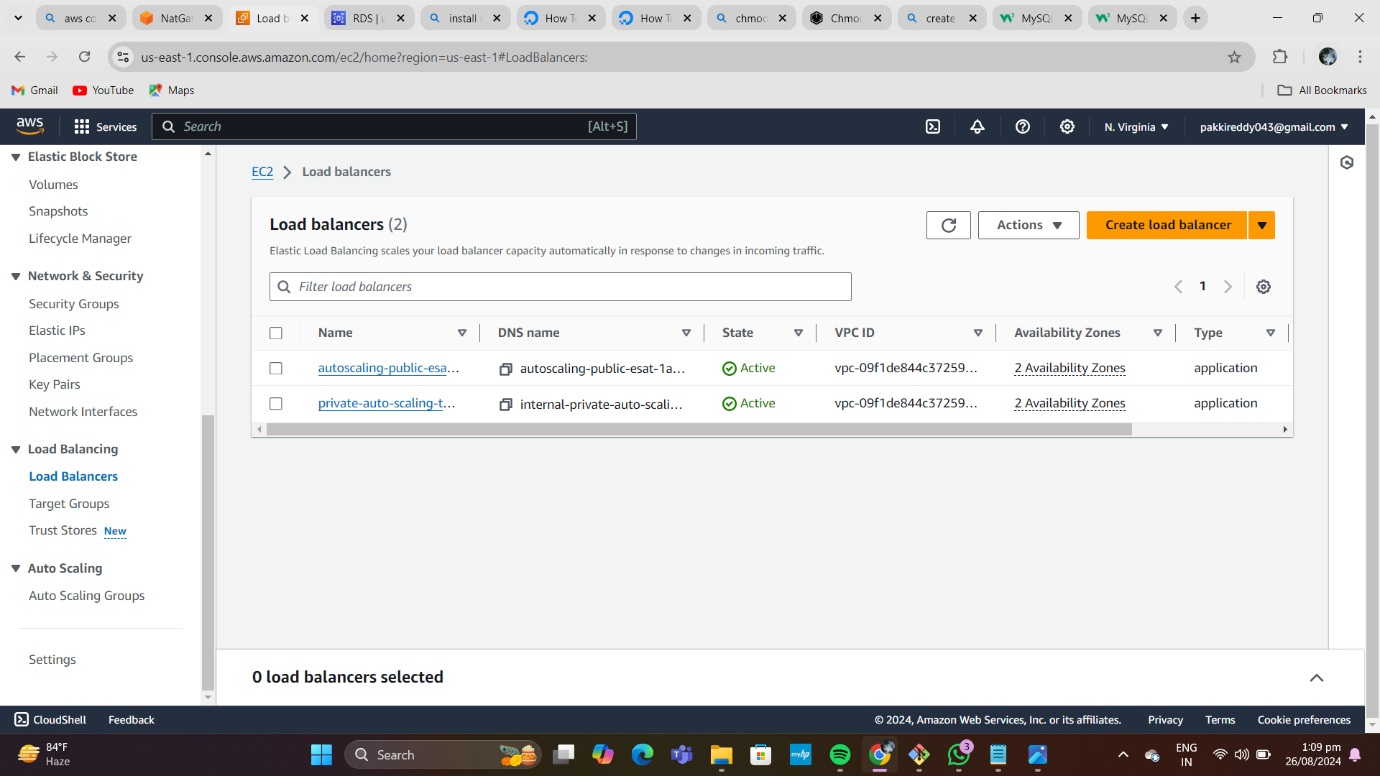
• Select our VPC (my-project-1), and both public subnets for autoscaling-public-east-1a .

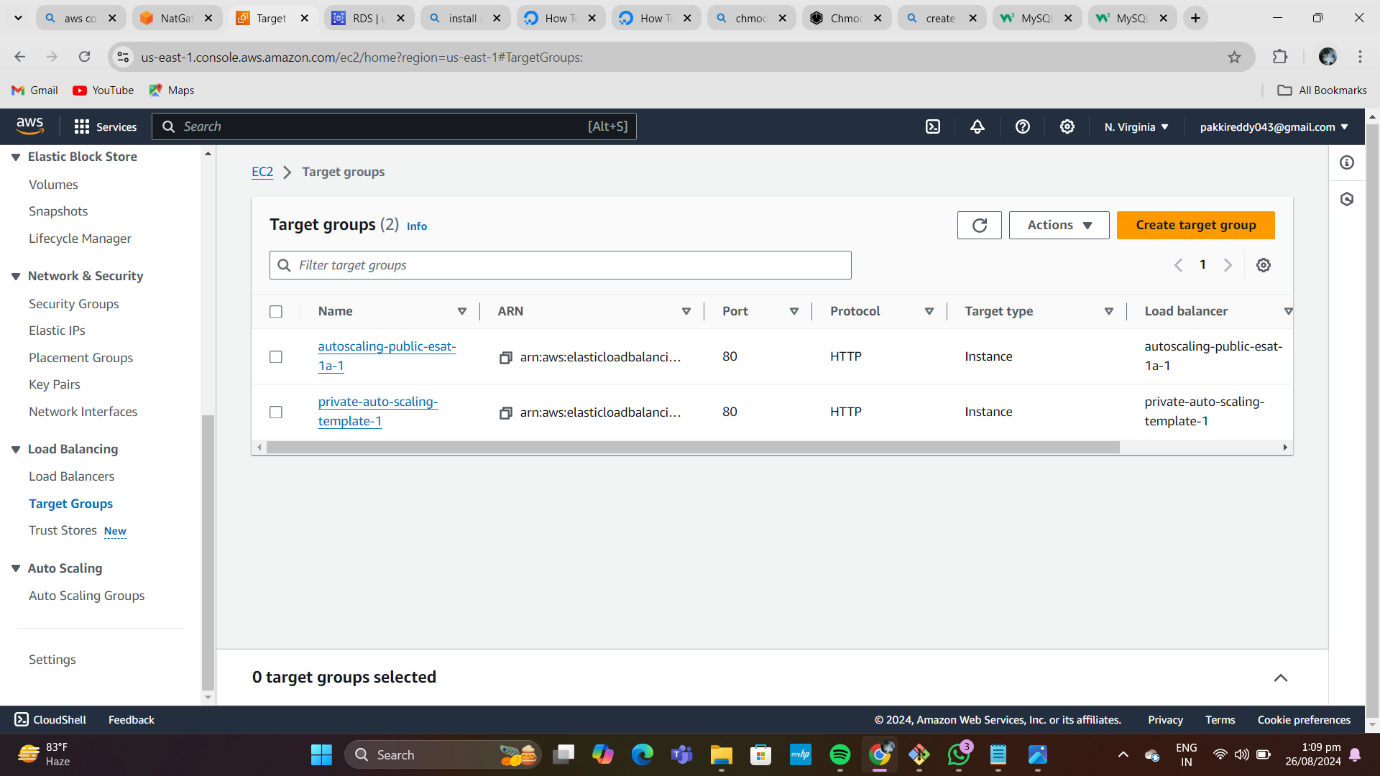
Select our VPC (my-project-1), and both private subnets for autoscaling-group-template .

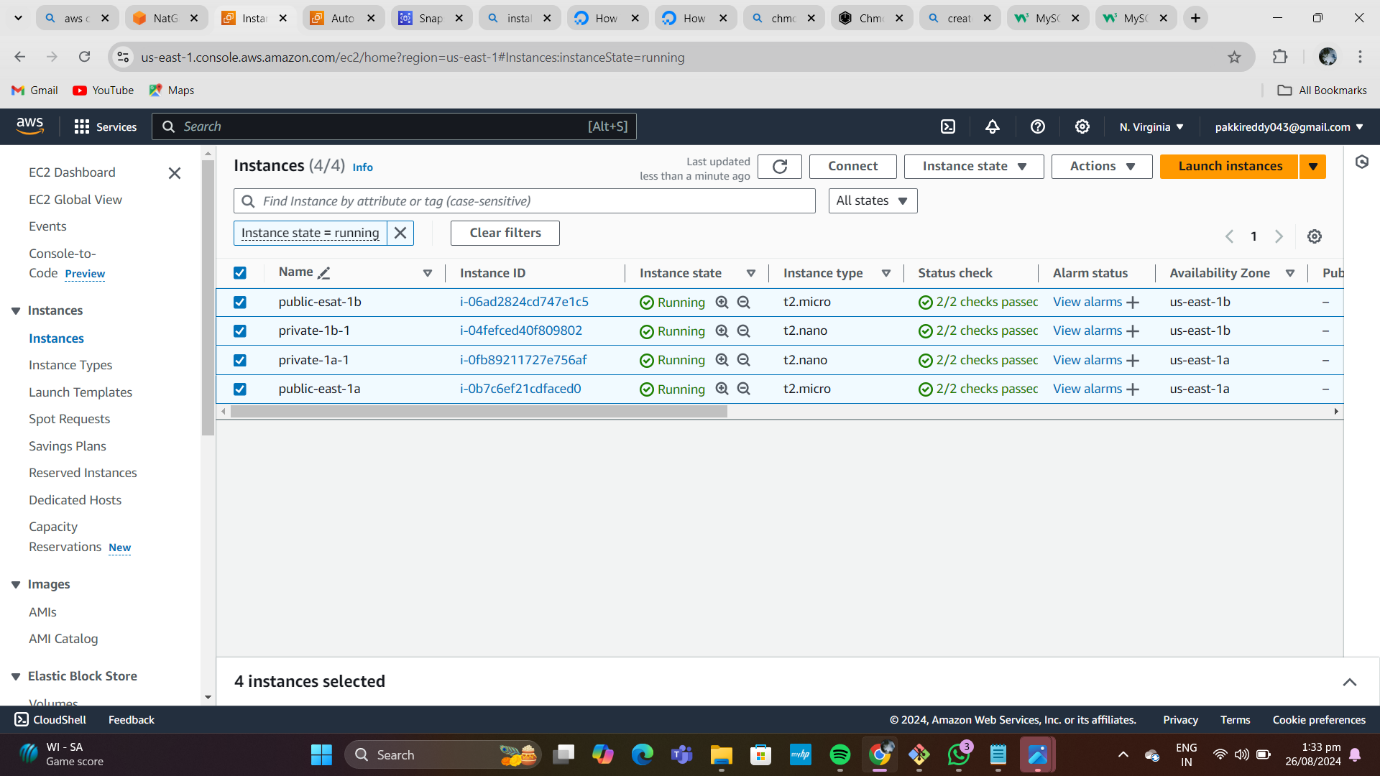
• Create two auto scaling groups.

• at the time of creating auto scaling groups automatically we will create load balancer & target groups & Ec2 instances.









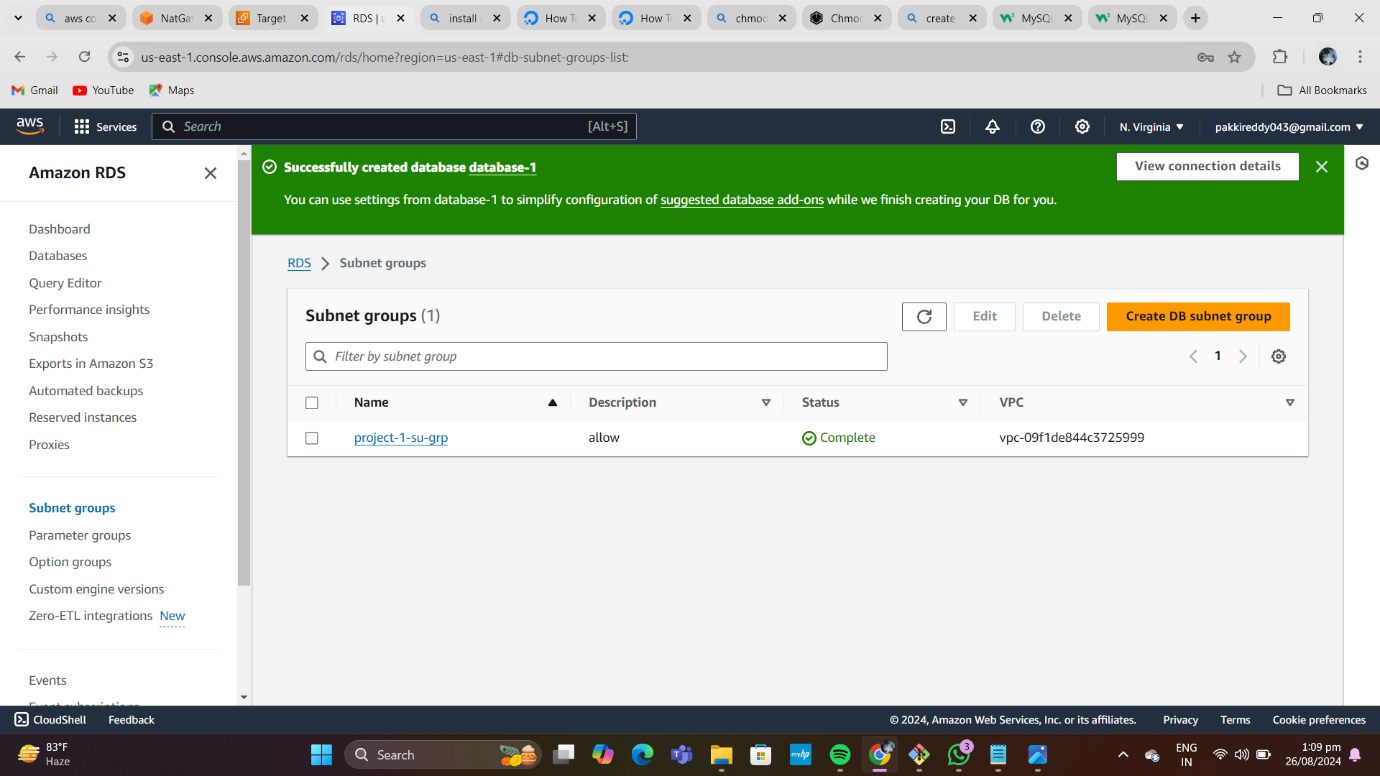
**Create subnet groups**

• Give name as project-1-su-grp and description nothing.

• Select created VPC .

• Give availability zones and select private subnets from each zone.

• Create the DB subnet group**.**

****

**Create Database(RDS)**

• Click on create database, select standard create, select engine type as MySQL.

• Select templates as production and select multi-AZ DB cluster.

• Select on self-managed, give password and confirm the password.

• Select memory optimized class.

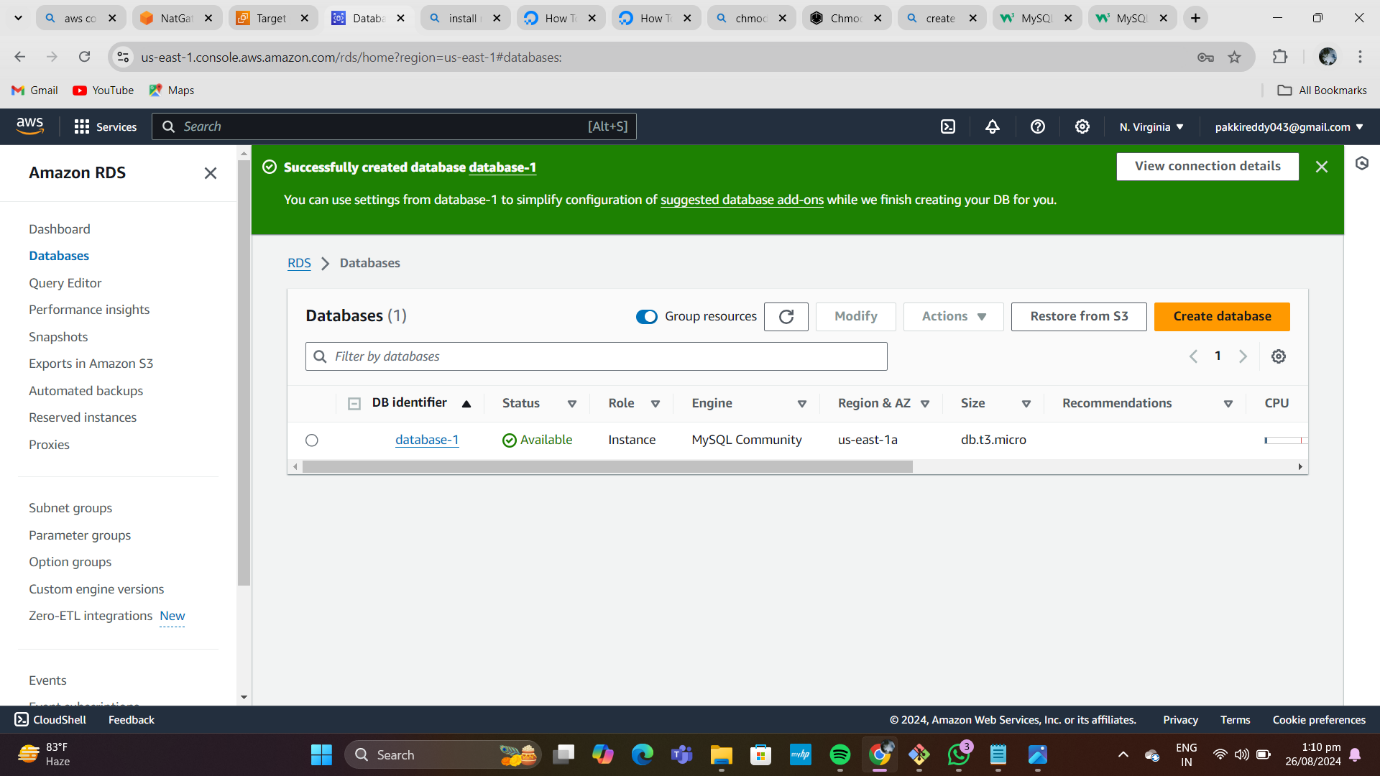
• In connectivity, click on Don’t connect to the EC2 compute resource and select created vpc (my-project-1).

• Select subnet group (extra) and give public access as yes.

• Choose existing security group (PUBLIC).

• Go to VPC dashboard, click on VPC, click on actions, go to edit VPC settings and click on the enable DNS hostnames.

• Create the database.



**Establish connection**

• Go to EC2 instance, click on public-east-1a, through public instance we will connect to private instance through pem file and connect to private instance and after that .

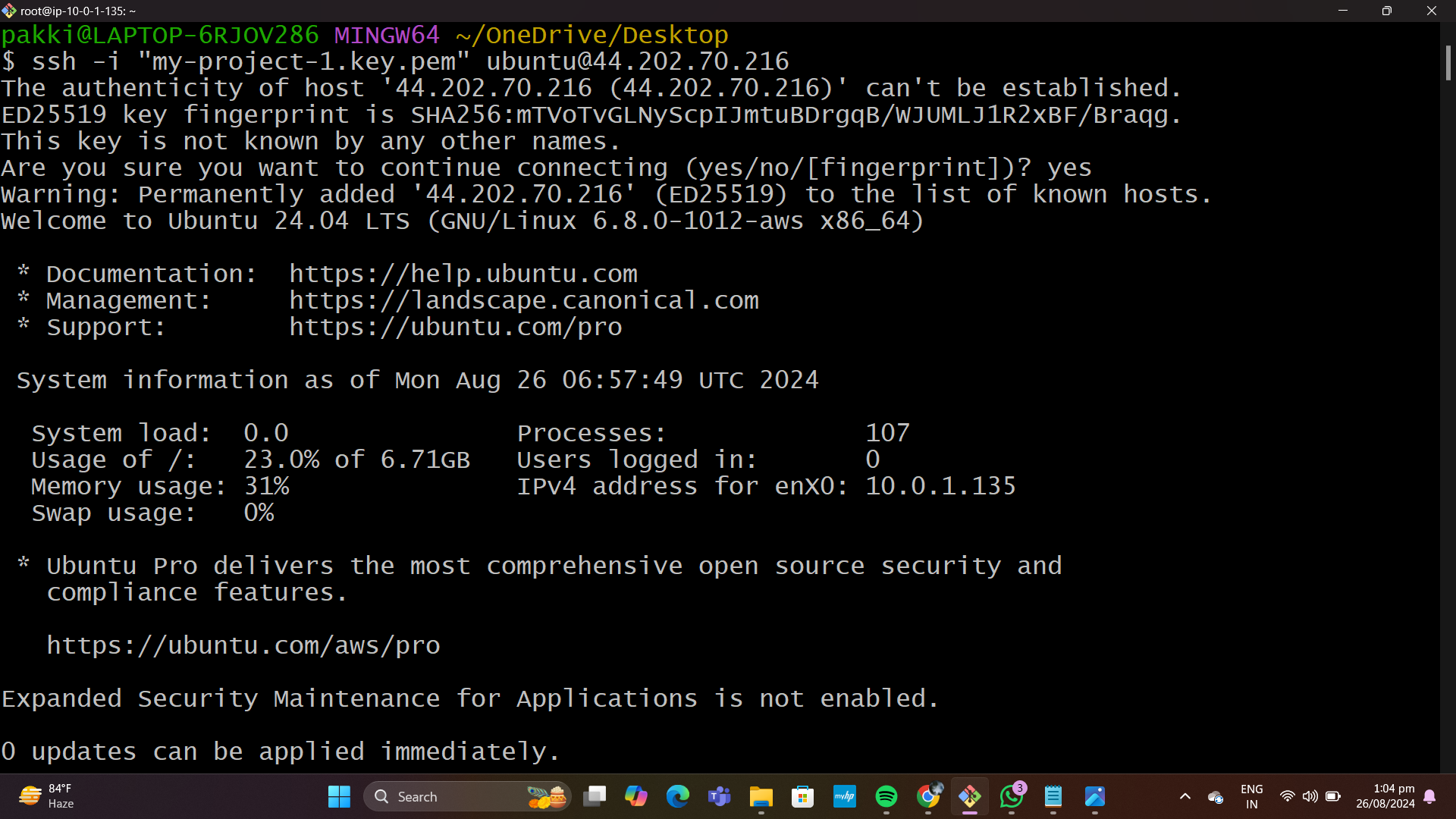
. Give commands as:

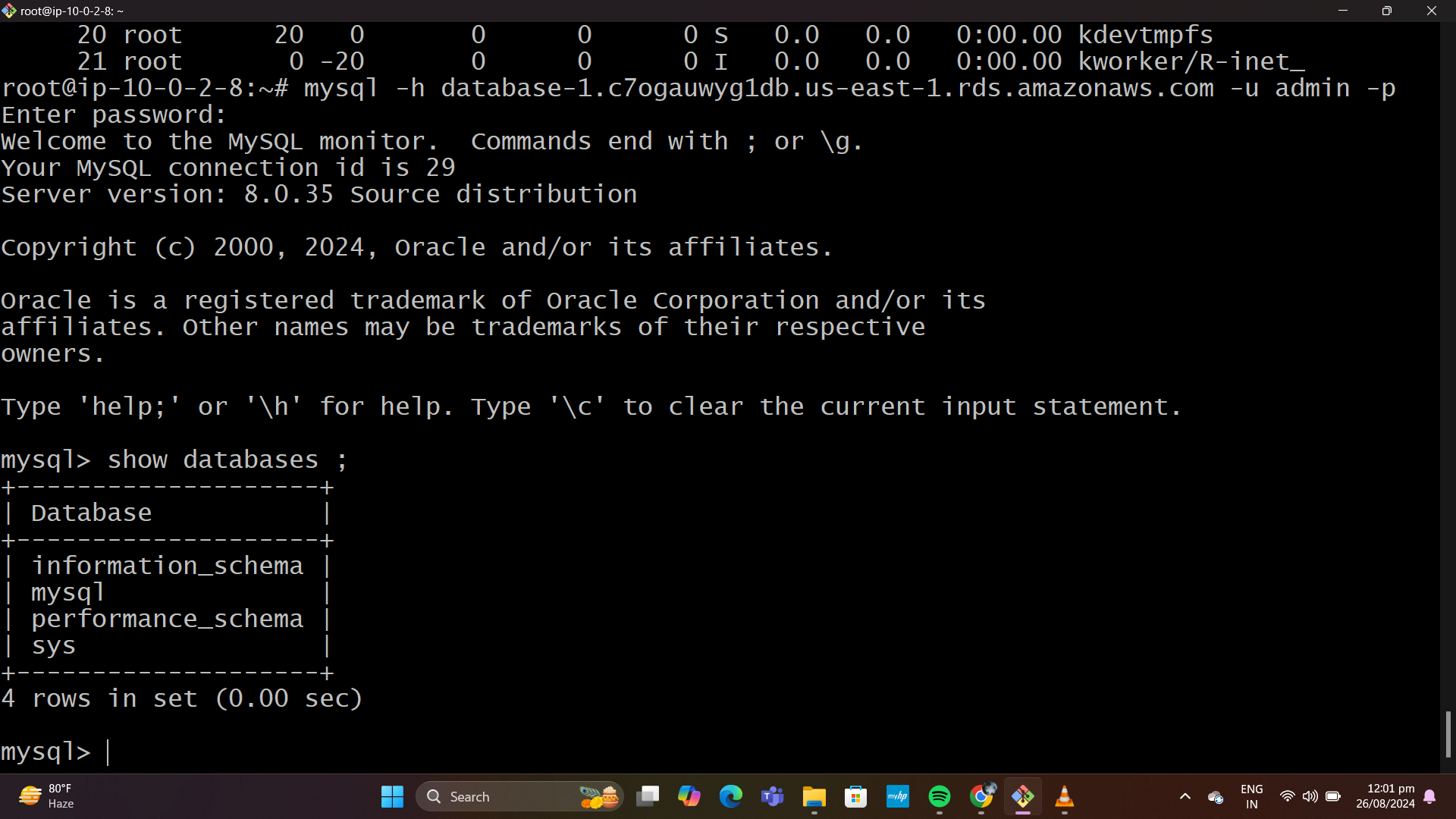
1. sudo -i (convert from normal user to root user).

2. apt update -y

3. sudo apt install mysql-server (to install mysql).

• after installig mysql connect to rds server through the RDS EndPoint and provide user name and passward .





Now we can see that we are connected to MySQL server.

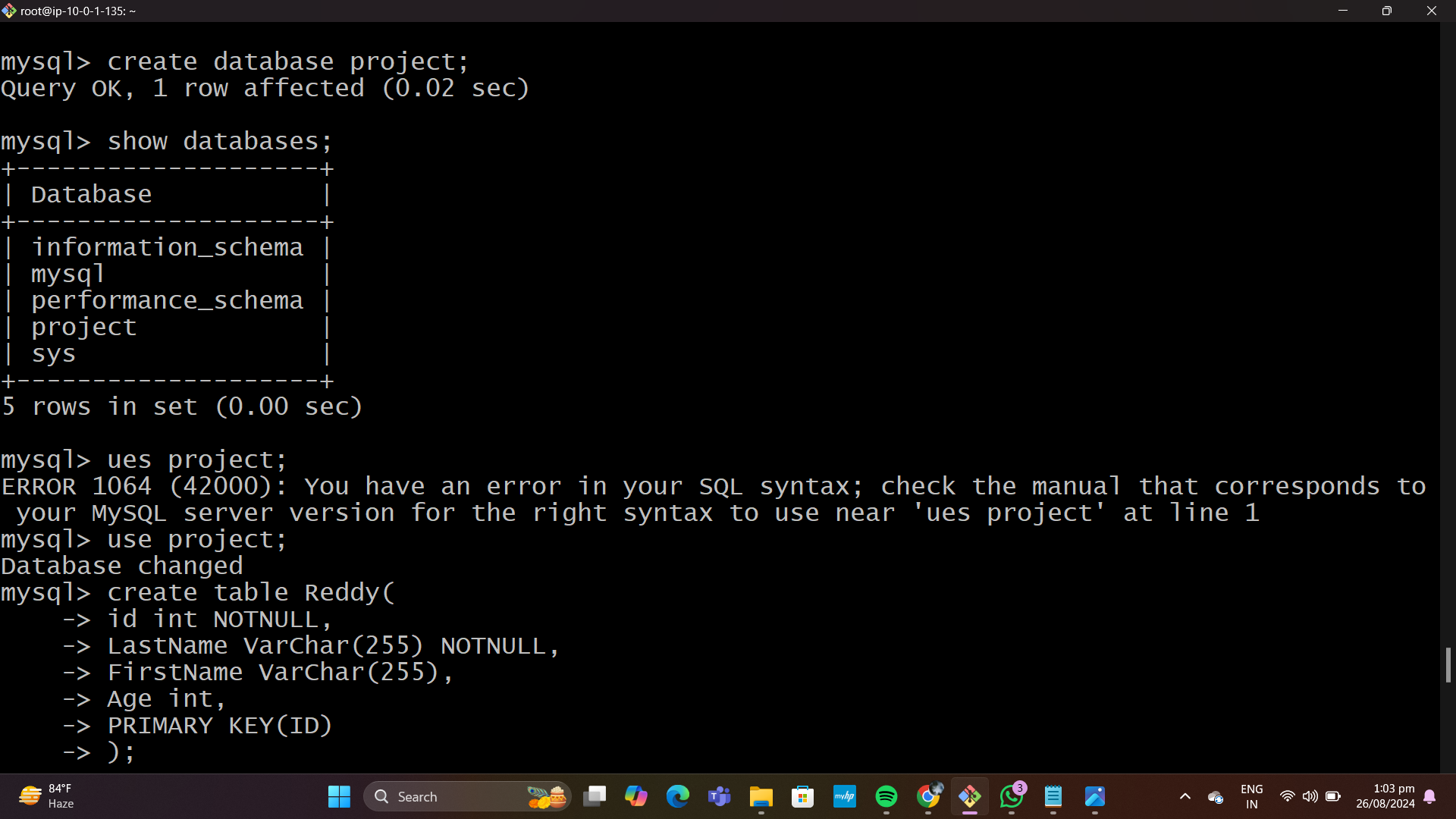
Use commands like:

• show databases; (to show the list of databases are there )

• create database project; (to create a new database)

•use project (database).

• create a table using this command



CREATE TABLE Reddy (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);

**Insert the fiels and the values**

INSERT INTO Reddy (PersonID, LastName, FirstName, Address, City)

Values ('101', 'Pakkireddy', 'palla', 'cherukucherla', 'nandikotkur');

• to the content in the table

• use command select \* from Persons; ( to check the content in table)

