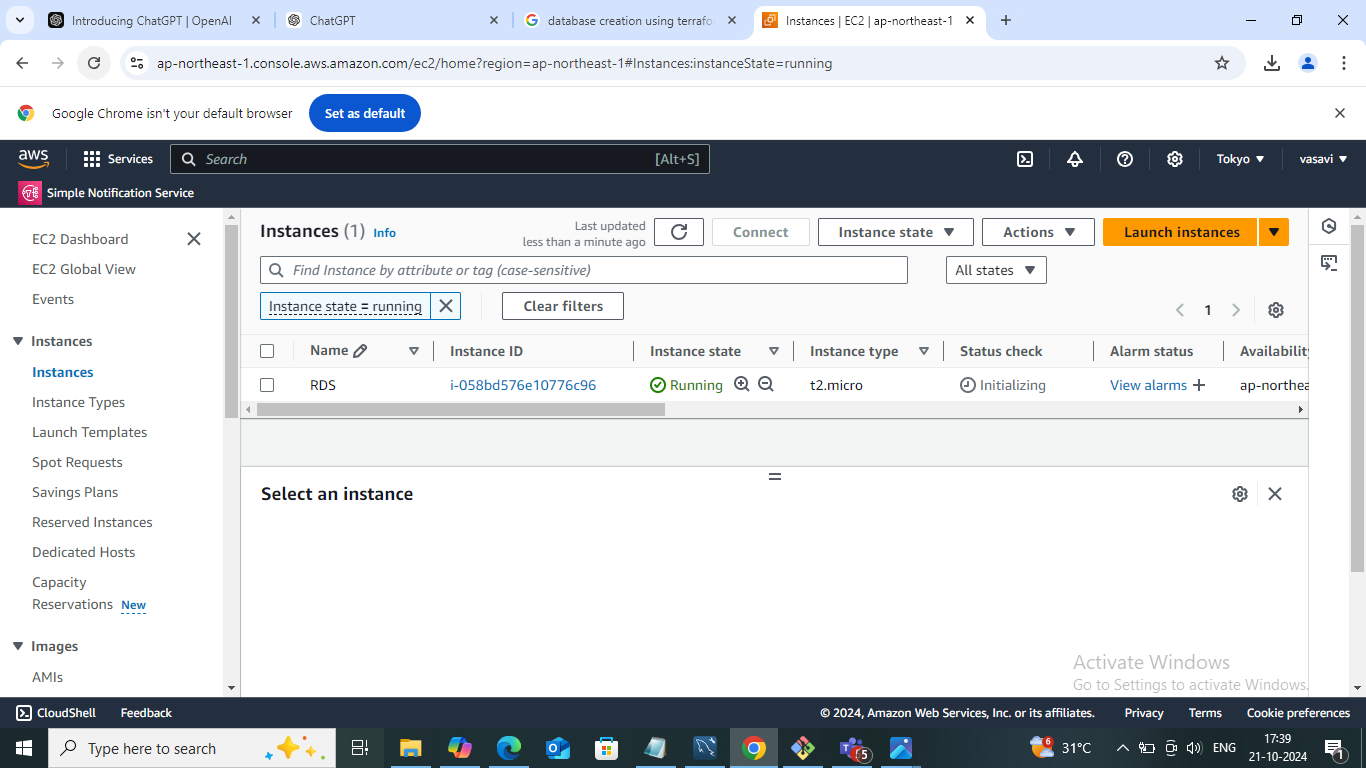
**CREATING RDS USING TERRAFORM**

RDS(RELATIONAL DATABASE) : RDS stands for Amazon Relational Database Service. In simple terms, it's a cloud service that helps you set up, manage, and scale relational databases like MySQL, PostgreSQL, and others. Instead of having to worry about the technical details of running a database, RDS takes care of things like backups, updates, and security, allowing you to focus on using the data instead of managing it.

STEPS TO CREATE RDS USING TERRAFORM :

STEP1🡪Creating an ec2 instance :



Go to Dashboard, click on “Launch Instance,” choose an AMI, and select the instance type, choose an existing keypair or create a new keypair, edit the network settings, then click on launch instance. The instance gets created.

STEP2🡪Connect the ec2 instance using gitbash/mobaxterm/putty as of your choice

STEP3🡪Install Terraform by using following commands:

1. sudo yum install -y yum-utils shadow-utils
2. sudo yum-config-manager --add-repo <https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo>
3. sudo yum -y install terraform

STEP4🡪Creating .tf file:

Here I have created vasu.tf file and placed a script to create the database. Also replace the region with which the ec2 instance has been created then change the password as your wish. Save & exit

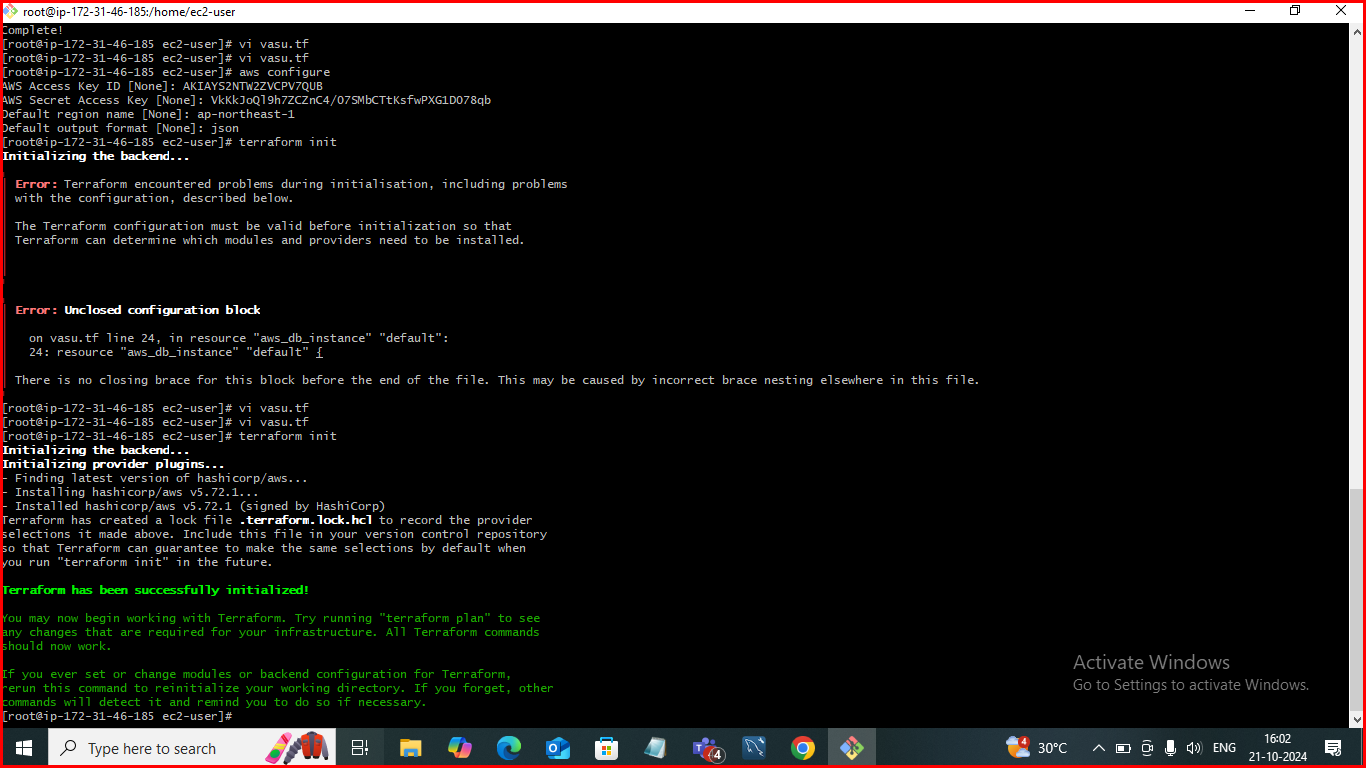
provider "aws" {  
   region = "ap-northeast-1"  
}  
resource "aws\_security\_group" "demo-rds-sg" {  
   name = "secgroup123"  
   description = "awssecuritygroup"  
#vpc\_id = aws\_vpc.demo-vpc.id  
      ingress {  
       from\_port = 3306  
       to\_port = 3306  
       protocol = "tcp"  
       cidr\_blocks = ["0.0.0.0/0"]  
      }  
      egress {  
         from\_port =0  
         to\_port = 65535  
         protocol ="tcp"  
         cidr\_blocks = ["0.0.0.0/0"]  
      }  
   tags = {  
       name ="secgroup"  
   }  
}  
resource "aws\_db\_instance" "default" {  
allocated\_storage    = 30  
db\_name              = "mydb"  
engine               = "mysql"  
engine\_version       = "8.0.35"  
instance\_class       = "db.t3.micro"  
username             = "admin"  
password             = "sreevani"  
publicly\_accessible = "true"  
parameter\_group\_name = "default.mysql8.0"  
skip\_final\_snapshot  = true  
#vpc\_security\_group\_ids =  [aws\_securiy\_group.demo-rds-sg.id]

STEP5 🡪 AWS configuration:

Create security credentials and provide access key and secrete key.

AWS Access Key ID [None]: AKIAYS2NTW2ZVCPV7QUB  
AWS Secret Access Key [None]: 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  
Default region name [None]: ap-northeast-1  
Default output format [None]: json

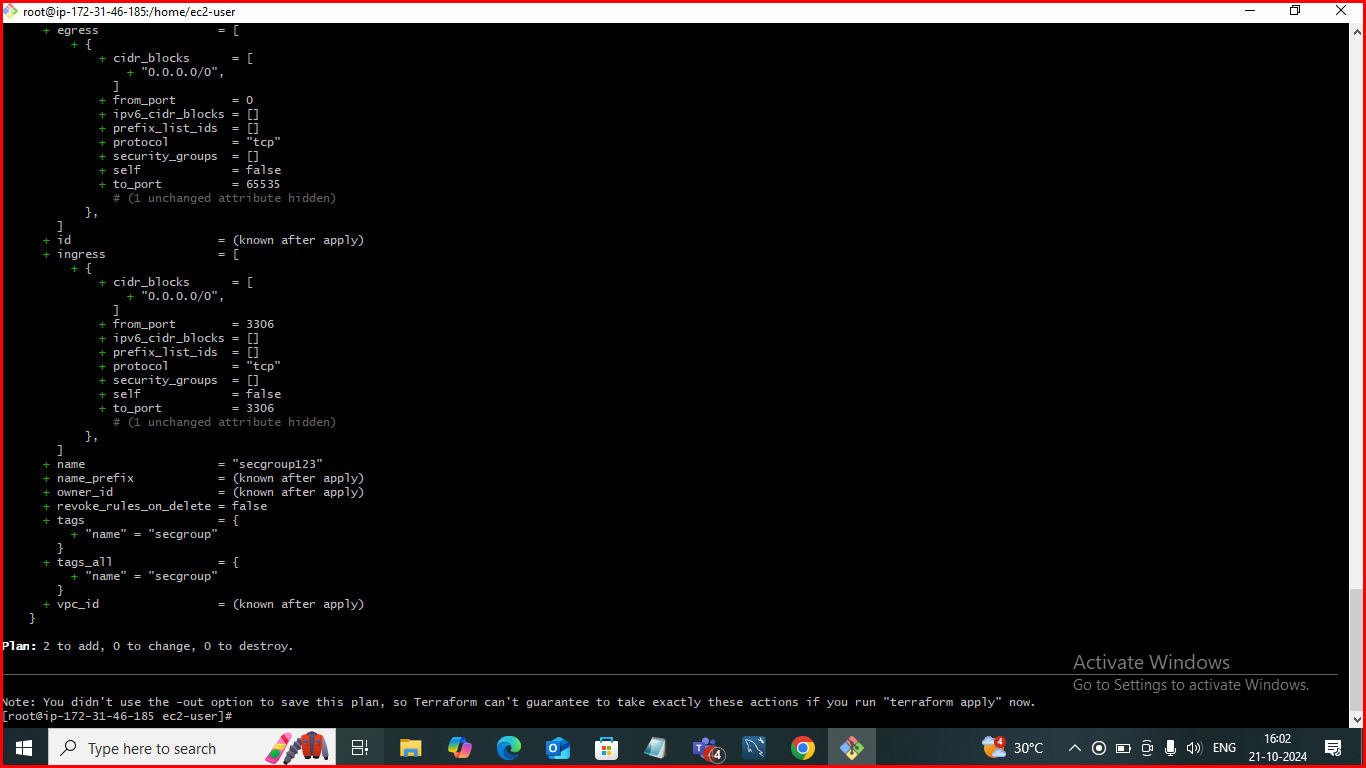
Provide the necessary details from your credentials also the region and default output.

STEP6🡪Terraform Initialization: 

Terraform has been initialized by using the command

“terraform init”

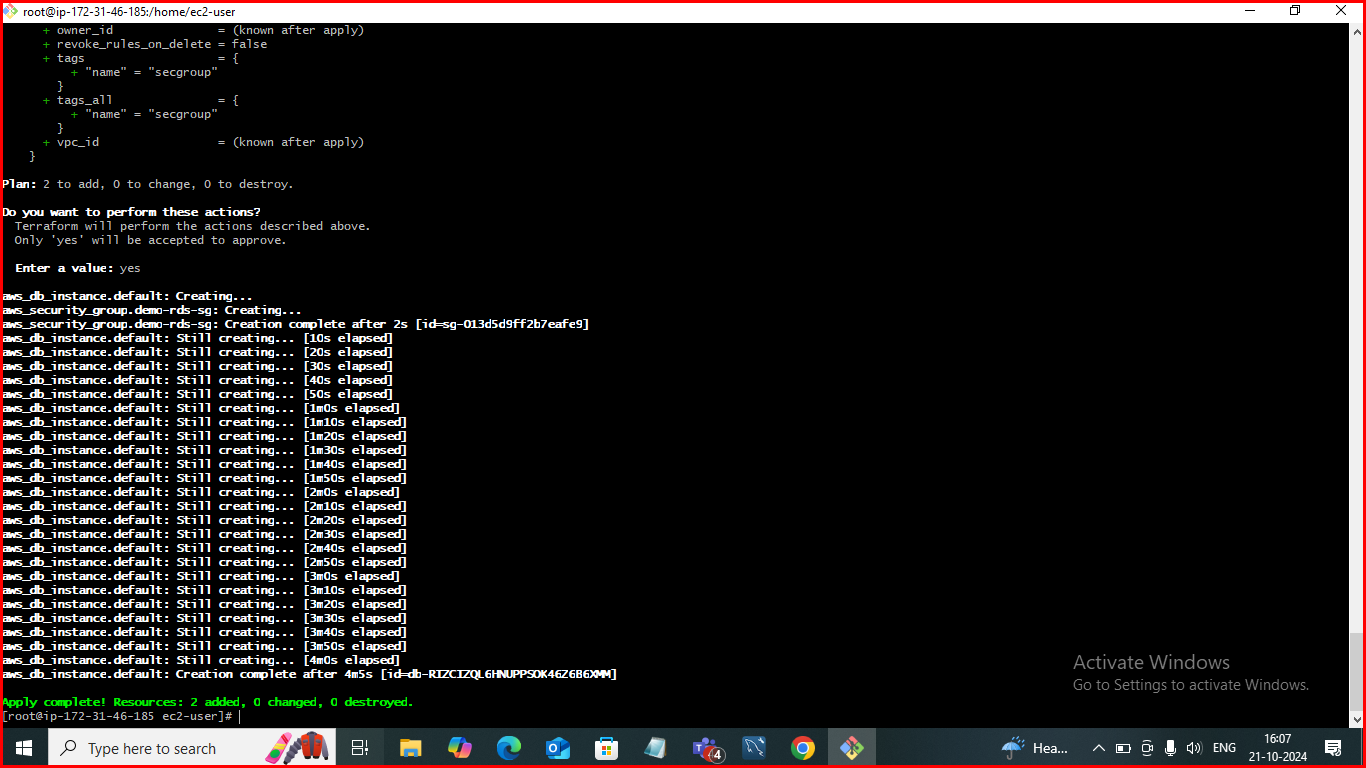
STEP7🡪Terraform planning:



Terraform planning has been done by using the command

“terraform plan”.

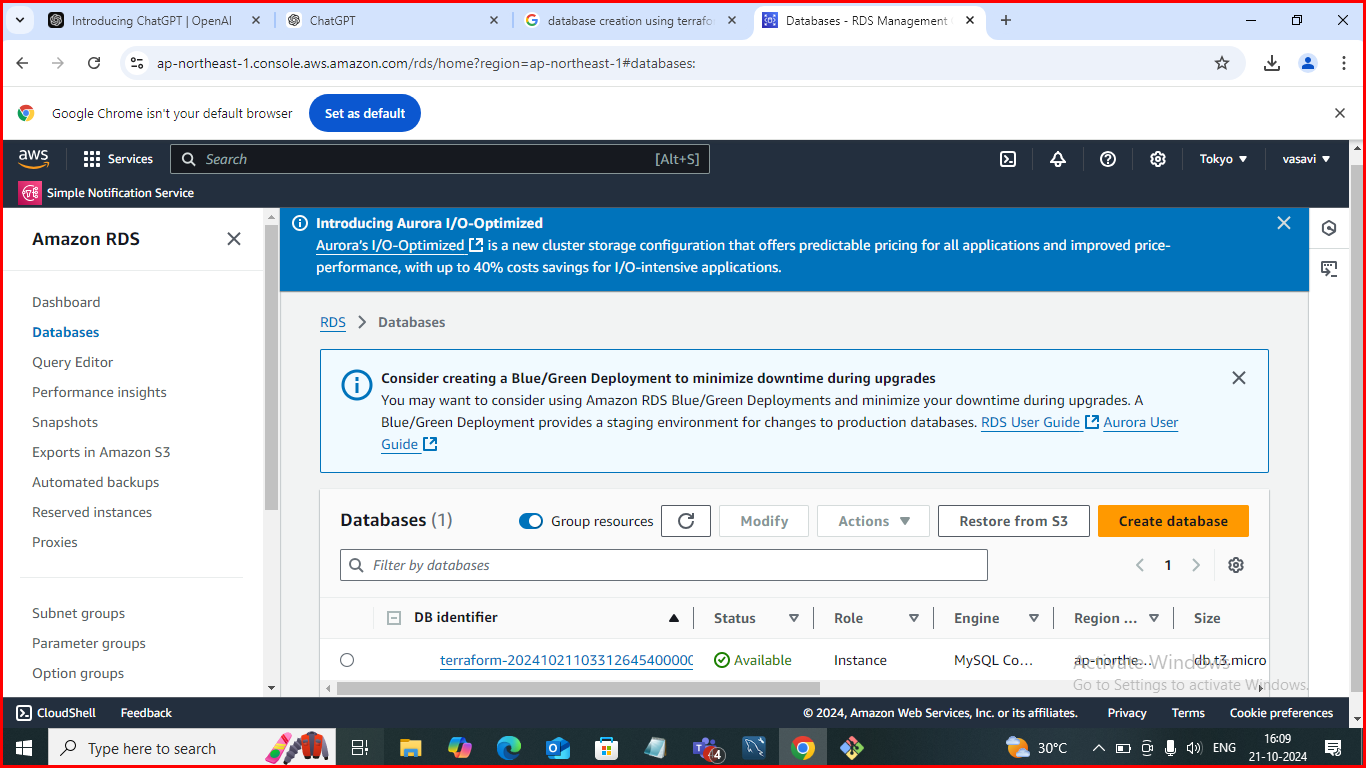
Step8🡪Terraform apply:



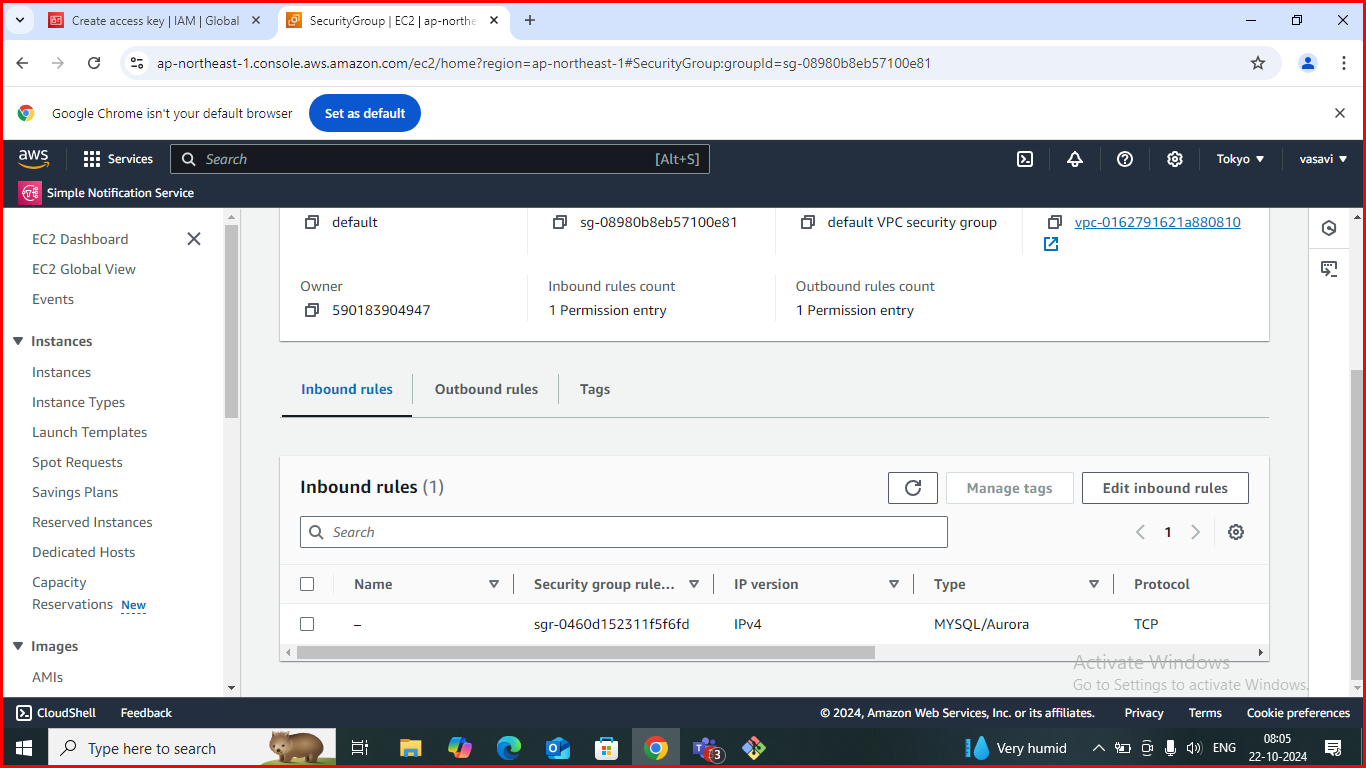
Terraform apply has been done by using the command

“terraform apply”

STEP9: Database gets created:



STEP10: Checking the inbound rules whether security group rule is added or not:

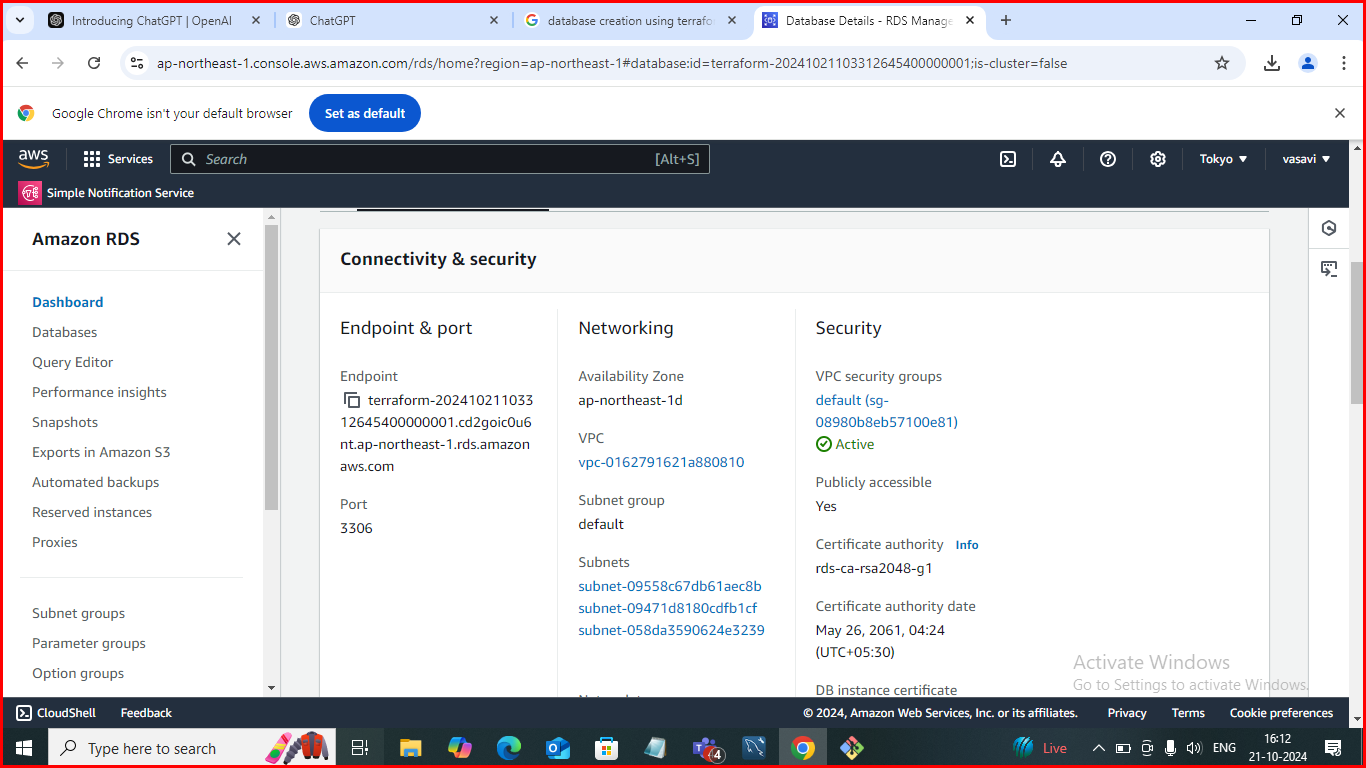


As shown in the figure we can see that the that the security group rule has been added with IPV4 version and type as MYSQL/Aurora.

Only then it gets connected to the database

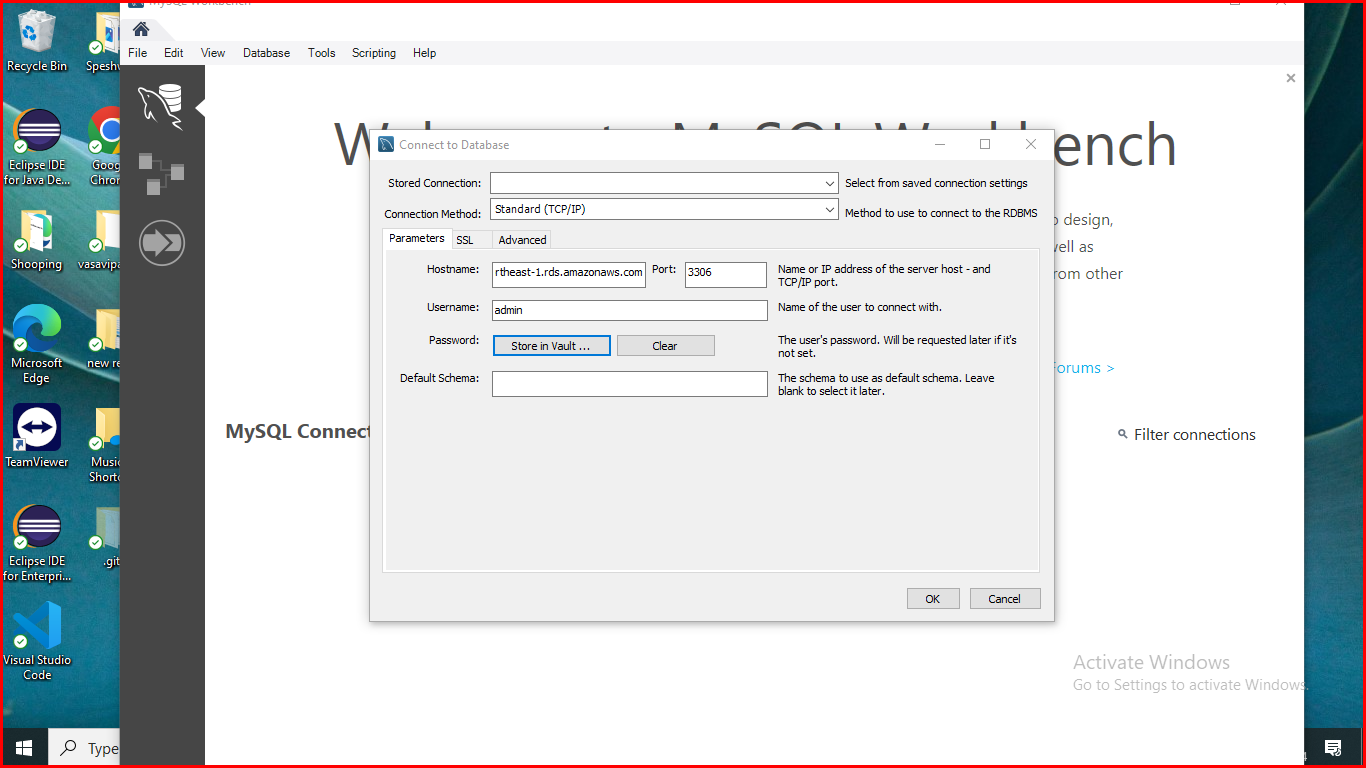
STEP11🡪Connecting to the database:

1. Copy the endpoint link-

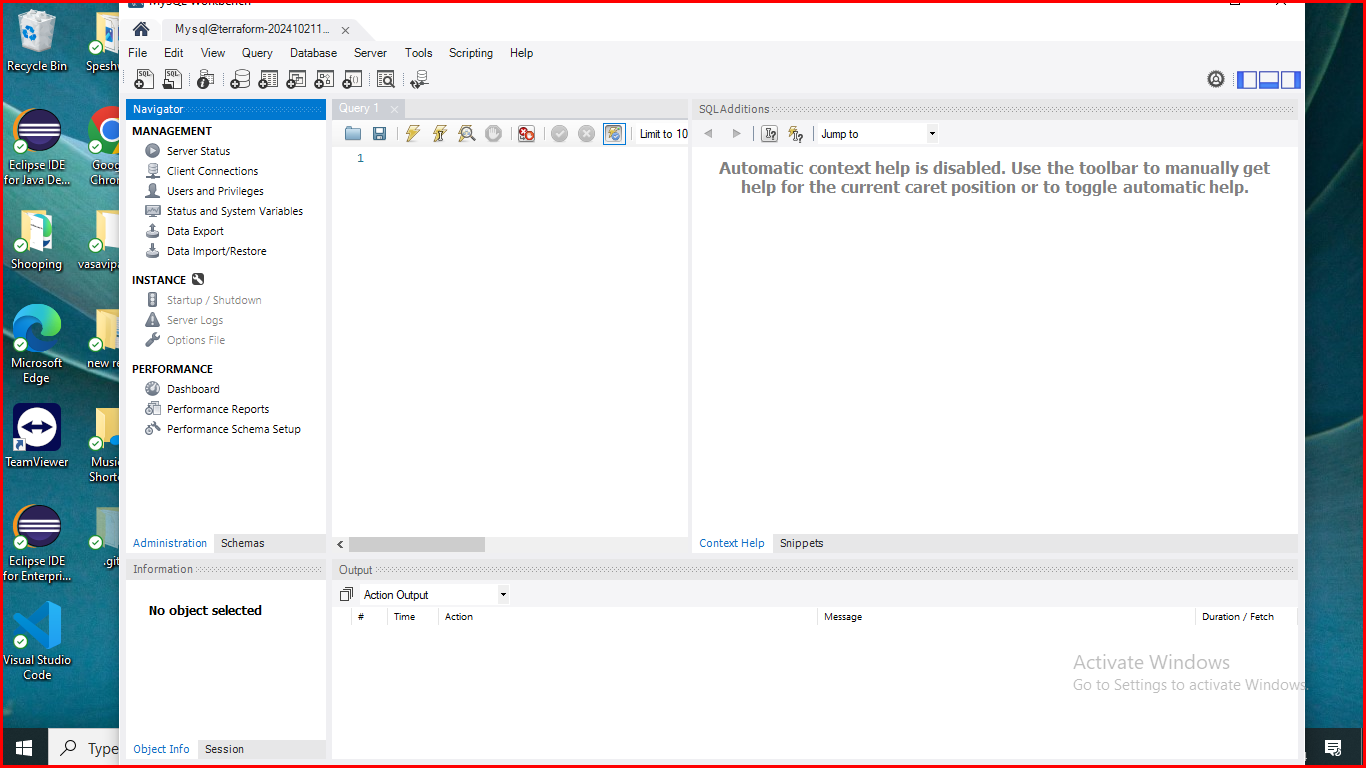


1. navigating to SQL workbench-

* We will open SQL workbench here
* Click on database
* Click on connect to database
* At the place of hostname paste the endpoint &port link that we have copied
* Give the username and password. The username and password should be same as given in the code
* Then click on ok.



STEP12🡪Connecting to the Database:



This page gets displayed only after successful connection of the database.

CONCLUSION: Following these steps we can create RDS with terraform.