# **AWS DATABASE**

**What is Database:-**

A database is an electronically stored, systematic collection of data. It can contain any type of data, including words, numbers, images, videos, and files. You can use software called a database management system (DBMS) to store, retrieve, and edit data.

**What is Relational Database:-**

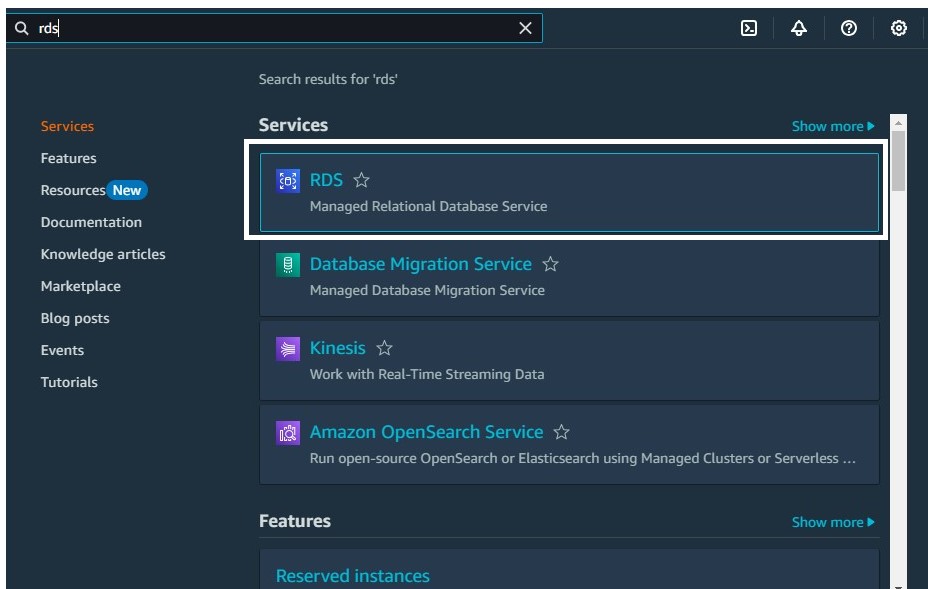
* A relational database (RDB) is a type of database that organizes data into tables, rows, and columns to establish relationships between data points
* The relational model is a standard way of representing and querying data

**What is AWS Database:-**

Aws Database is a collection of databases that offer a high-performance, secure, and reliable foundation for data-driven applications and generative AI solutions

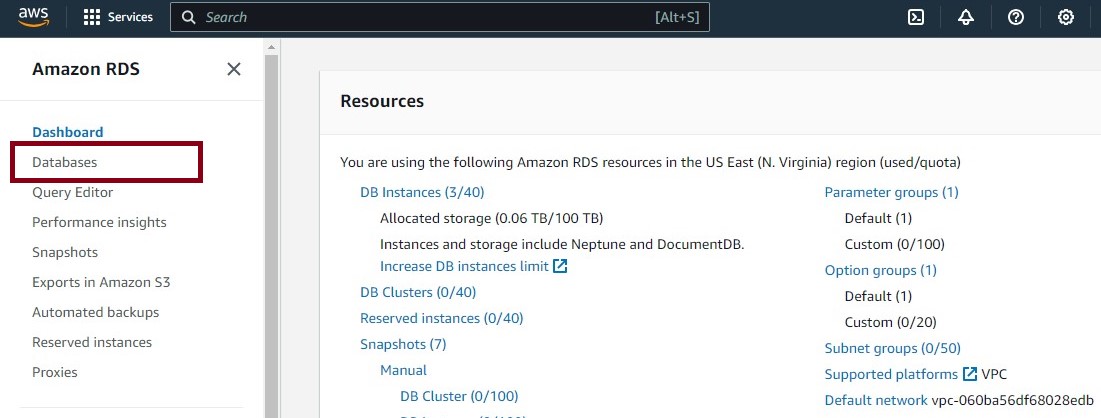
**Steps to create RDS:-**

# **Step 1:- Create RDS**

Login to the Aws Console and search for Database in that we can find RDS, here we will create the RDS Database

# **Step 2:- Create Database**

Select the Database option to create the RDS Database



# **Step 3:- Create Database**

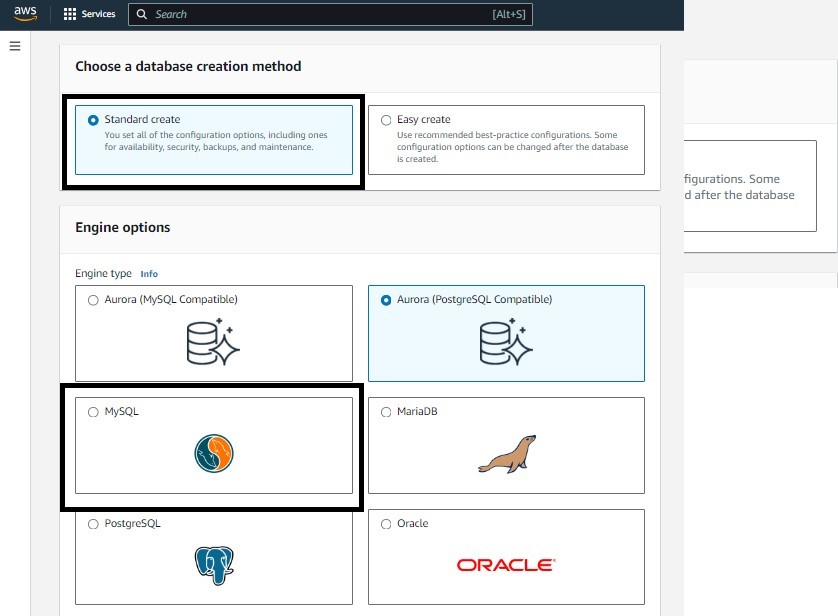
Two options available to create database

Standard create

Easy create

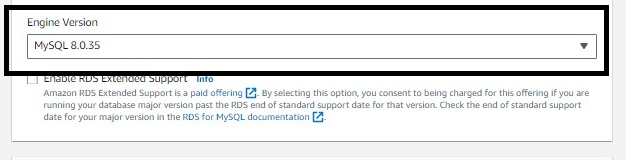
# **Step 4:- Engine Options**

Multiple Engine options available in the AWS Console and choose the database based on your requirement and currently I have chosen **MySQL** for our demo



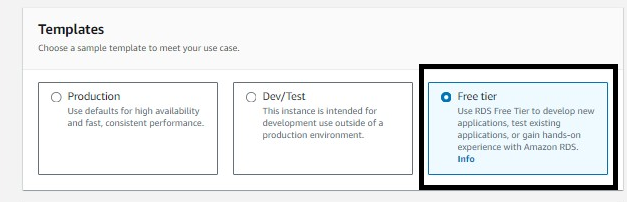
# **Step 5:- Engine Version**

If we want still we can select older version from the version list of items

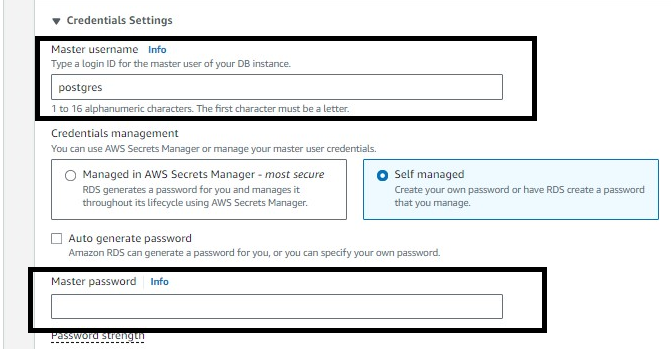
****

# **Step 6:- Template**

Different types of Templates available and select based on your requirement and for the demo I have selected Production, so I can show more options available

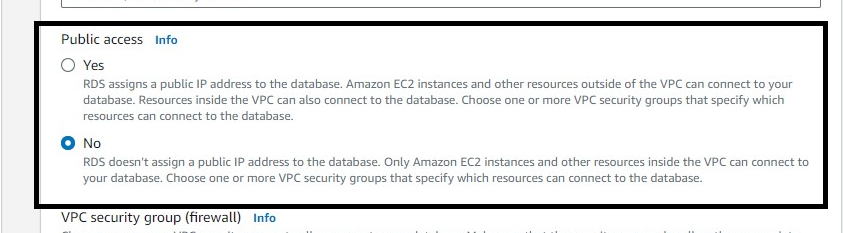


# **Step 7:- credential settings**



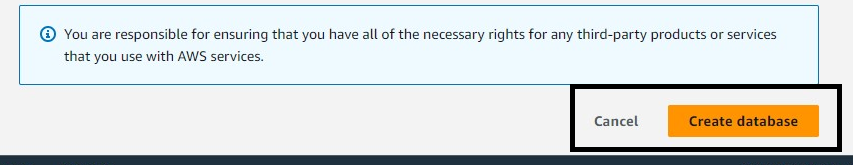
Type a login ID for the master user of your DB instance and enter the master password

# **Step 8:- Public access**



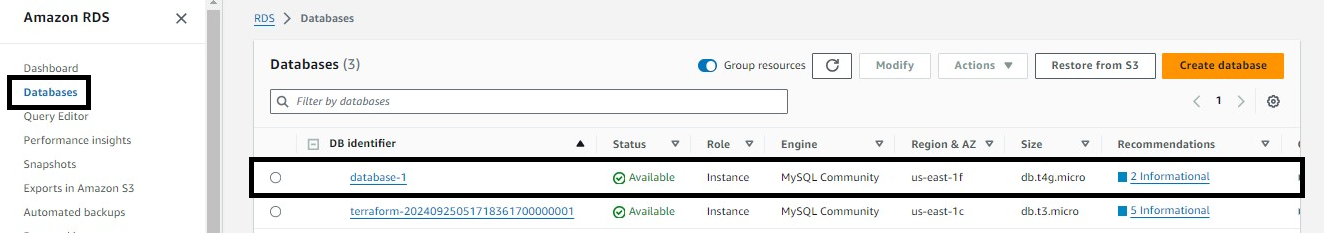
Here we take Public access “yes” if you want EC2 instances and other resources outside of the VPC hosting the database to connect to it.

# **Step9 :- Creation of DB**



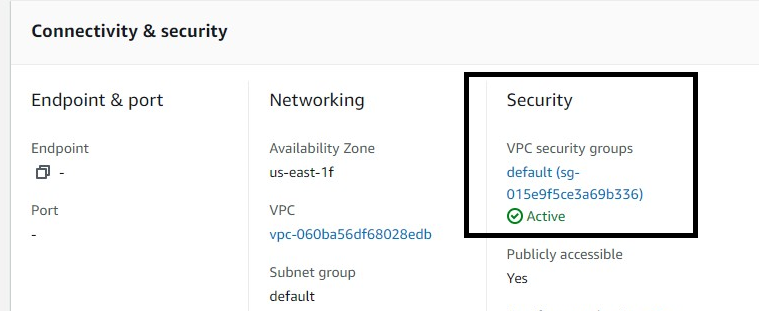
Finally click on the create database to create the DB instance

# **Step 10:- RDS Instance**

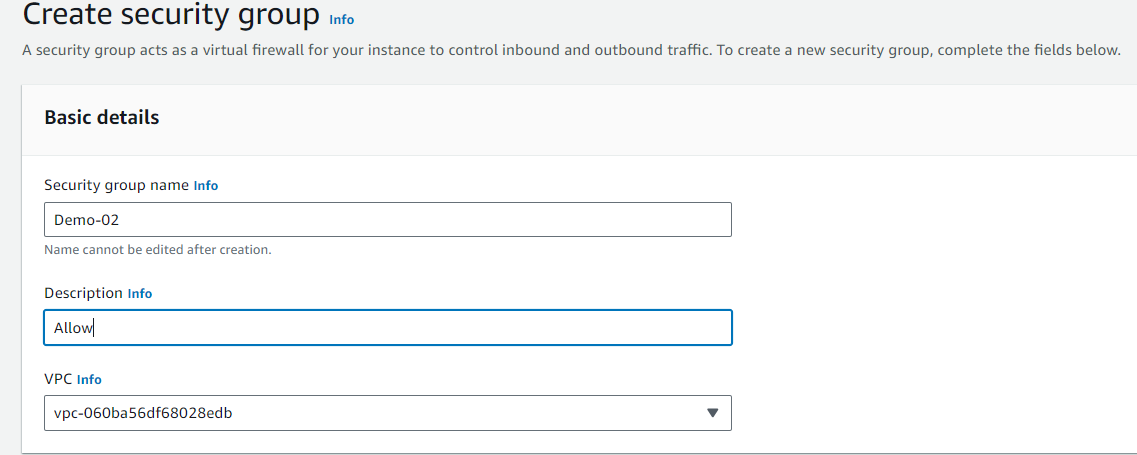


If we see the status is showing as Creating and we need to wait for 5 to 10 minutes for the RDS to be provisioned properly and once the RDS is ready status changes to Available.

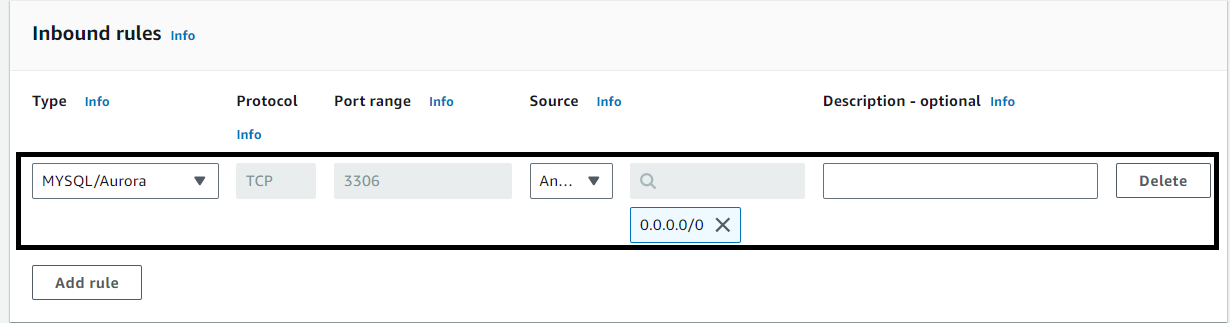
# **Step 11:- Connectivity and Security**



In security we will add security group Details

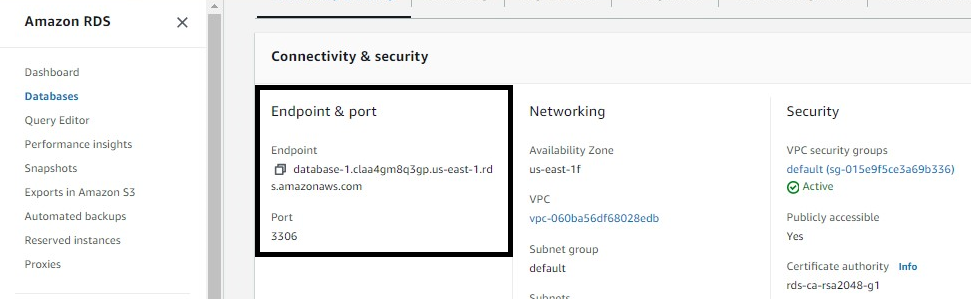


Add Inbound rules



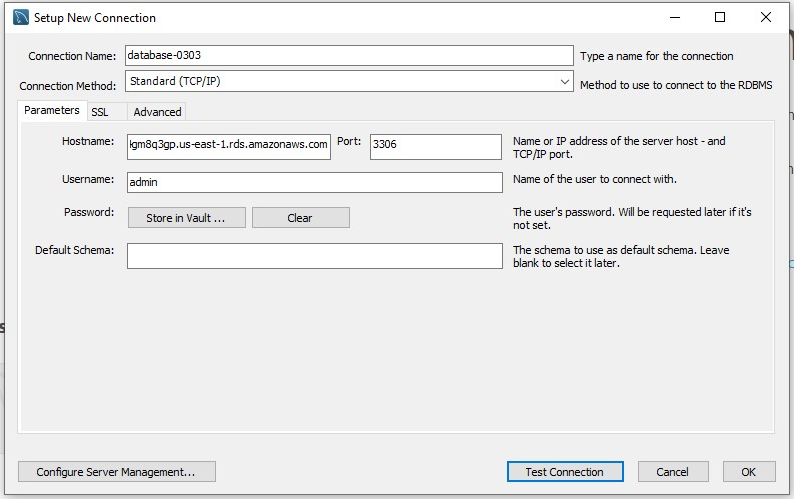
Add type as MySQL and source Anywhere (ipv4)

# **Step 12:- RDS endpoint**

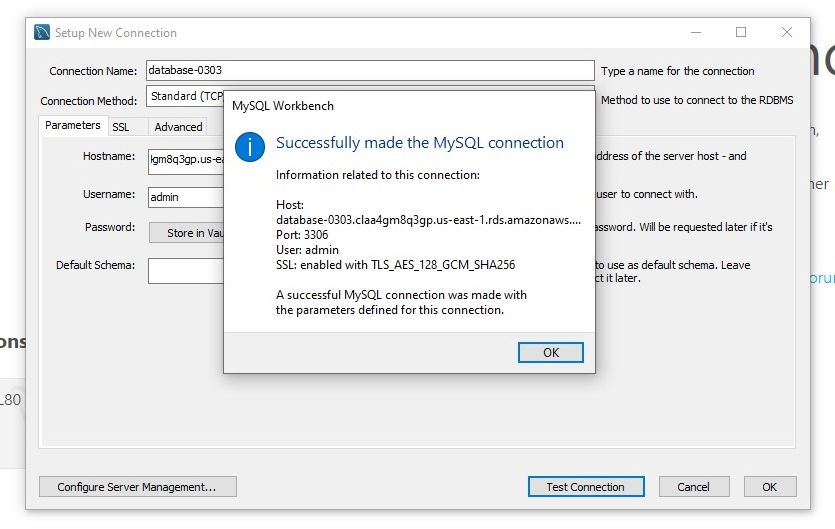


Here we copy the endpoint link and paste it on the MySQL workbench

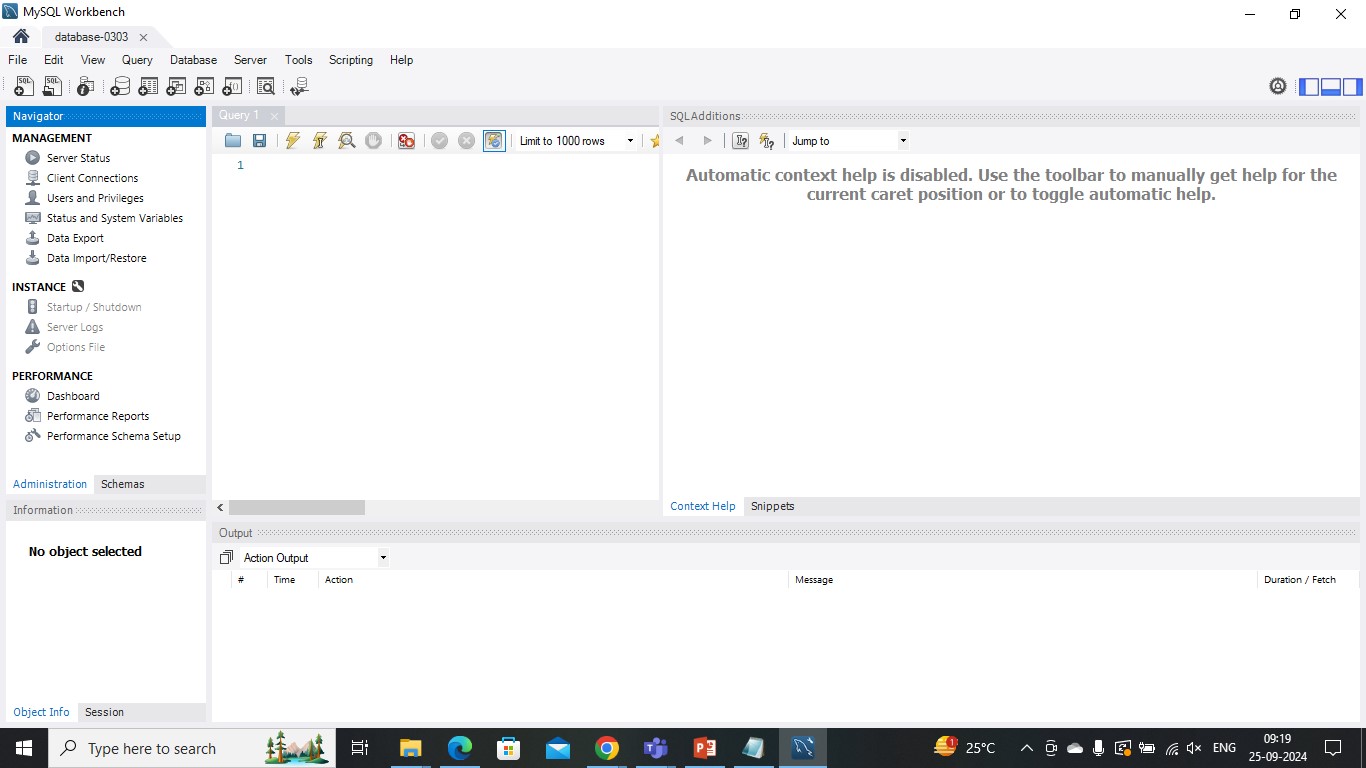
# **Step 13:- MYSQL Workbench**



Add Connection name, Hostname (endpoint in RDS Instance), Username and password in MySQL workbench and click on Test Connection

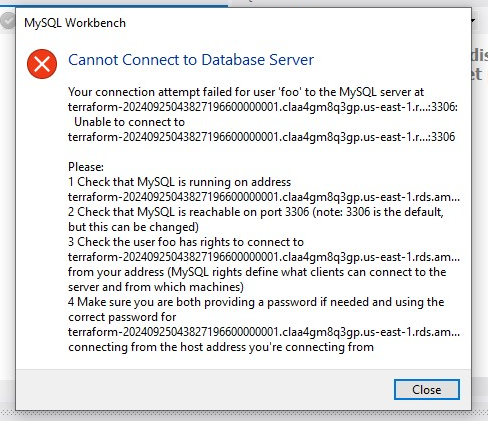


After Testing Connection it should be shown as Successfully made the MySQL Connection



Workbench Console should be open like the above after successful MySQL Connection

# **Step 14:- Error message:-**

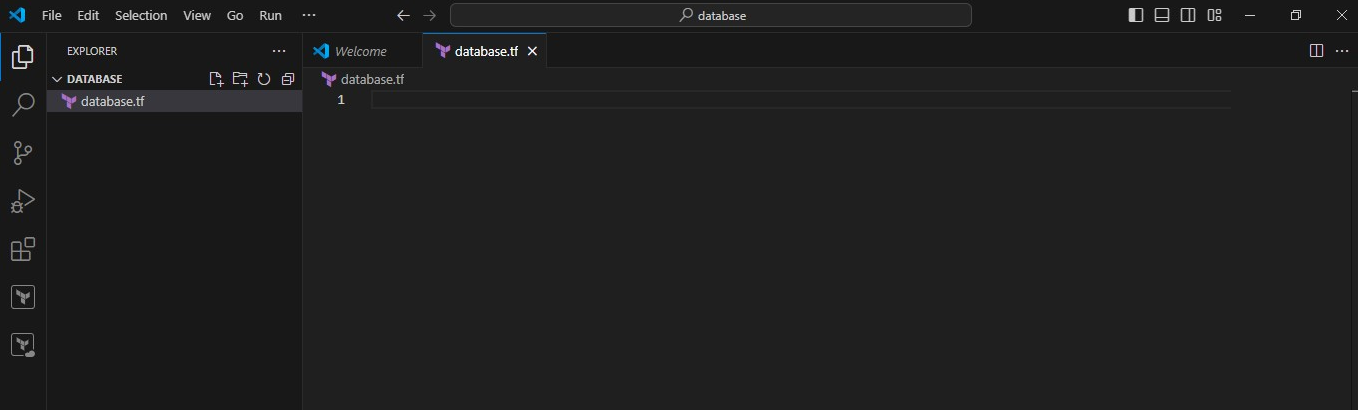


An error message will be shown when didn’t provide the proper port number and hostname

Port no :3306 is the port number for RDS Database

# **Step15:- Creating of Database in Terraform**

Open the Visual Studio and add the new folder name the folder as required (database.tf)



Add the required code to create the instance in EC2

Provider “aws”

Region = “ap-south-1”

resource "aws\_db\_instance" "default" {

  allocated\_storage    = 10

  db\_name              = "mydb"

  engine               = "mysql"

  engine\_version       = "8.0"

  instance\_class       = "db.t3.micro"

  username             = "admin"

  password             = "Password"

  parameter\_group\_name = "default.mysql8.0"

  skip\_final\_snapshot  = true

}

By applying the above code instance was created in the given region

# **Step 16:- security and inbound rules**

To create security group and inbound rules we use the code

provider "aws" {

    region = "us-east-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             = "Password"

  publicly\_accessible = "true"

  parameter\_group\_name = "default.mysql8.0"

  skip\_final\_snapshot  = true

  #vpc\_security\_group\_ids =  [aws\_securiy\_group.demo-rds-sg.id]

}

After applying the code run the code in new terminal by using the commands

# **Step 17:- terraform commands**

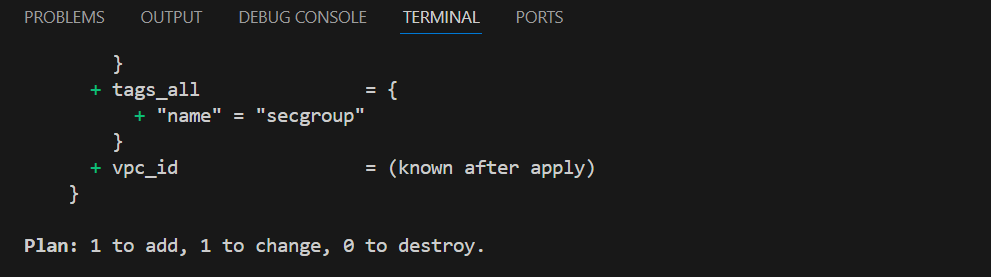
* Aws configure

Here we configure the aws by creating the access key and security key in aws console security credentials

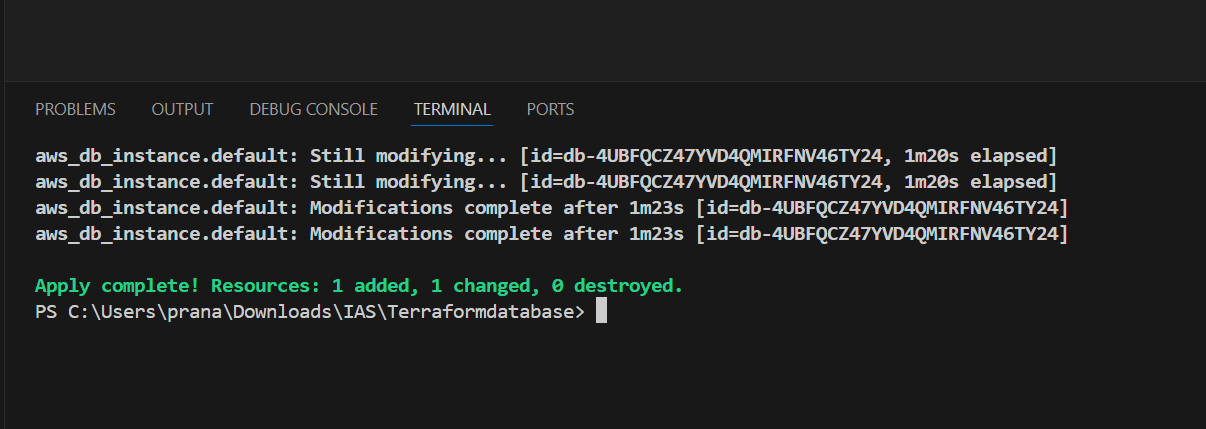
* Access key
* Security key
* Region
* Format

After configuring the above steps we initialize the terraform with commands

* Terraform init
* Terraform validate
* Terraform plan
* Terraform apply



After configuring the terraform, we get the changes like 1 files is added that indicates instance was created in the ec2



* To delete the instances in terraform we use “Terraform destroy”