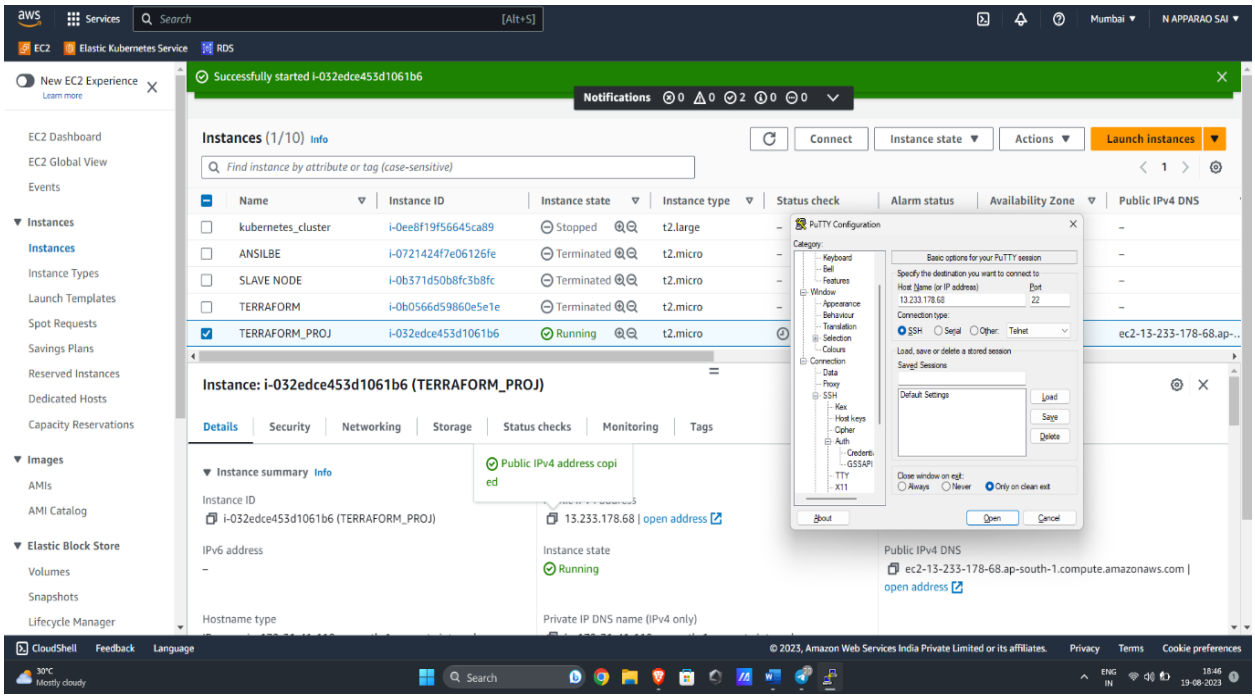
PROJECT 1: Provisioning AWS RDS with Terraform

STEPS TO CREATE AWS RDS :

1. Create an EC2 instance in AWS console, I create an instance with name TERRAFORM_PROJ
2. Login into Putty terminal with public -IP and keypair of that instance



3. After login to centos root user, install Terraform and its supporting packages with linux script

   sudo yum update -y
   sudo yum install -y yum-utils
   sudo yum-config-manager --add-repo
   https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo
   sudo yum -y install terraform

➢ After execution above commands, we installed Terraform and its supporting packages, from the above link we downloaded HASHICORP LANGUAGE packages

4. After installation create a directory with command mkdir<directory_name> and go to that directory with command cd < directory_name >.
5. Create a file name variable.tf and provide the variables for your RDS instance in a variables.tf file using HCL script mentioned below. Some variables that are included are the database engine, instance size, storage size, username and password, and backup retention period. Create that file with the command vi variable.tf. it automatically created and entered into that file by entering “I” (insert) the variables script mentioned in the below picture. I had entered my configurations in that script as variables. For access_key and secret_key we have to create a IAM user with suitable permission and generate access_key and secret_key for that IAM user , paste this keys in the variable file script and save it.

```
root@ip-172-31-41-110:~/terra
variable "access_key" {
  description = "AWS access key"
  type        = string
  default     = "AKIATX6ULZXLPSOT7VHR"
}

variable "secret_key" {
  description = "AWS secret key"
  type        = string
  default     = "jTUZkhmCSvqm1BVpducyRQPZOBq75xHPEHnBOSQe"
}
variable "db_engine" {
  description = "Database engine type"
  type        = string
  default     = "mysql"
}

variable "db_instance_class" {
  description = "Database instance class"
  type        = string
  default     = "db.t2.micro"
}

variable "allocated_storage" {
  description = "Allocated storage for the RDS instance (in GB)"
  type        = number
  default     = 20
}

variable "username" {
  description = "Username for the RDS instance"
  type        = string
  default     = "admin"
}

variable "password" {
  description = "Password for the RDS instance"
  type        = string
  default     = "admin123"
}

variable "backup_retention_period" {
  description = "Backup retention period in days"
  type        = number
  default     = 7
}

"variable.tf" 46L, 1018C
```

6. Again create another with command " vi main.tf " and define the Terraform resources for your RDS instance. Use the aws_db_instance resource type to create the RDS instance, and set the resource properties according to your chosen configuration, I had defined my configuration for my database as below script. Save the script in the file.

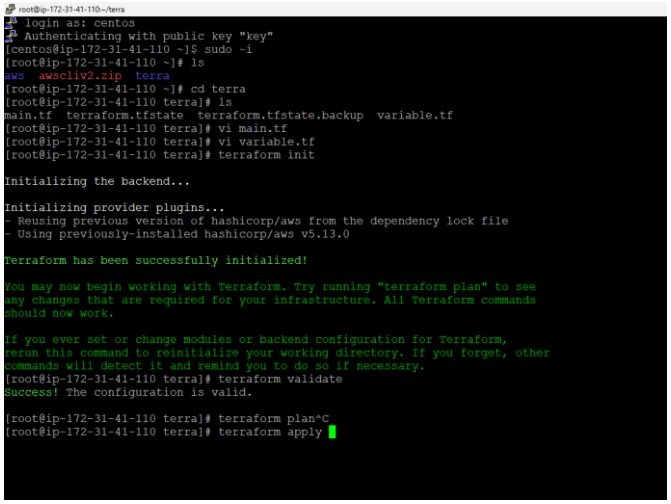
```
root@ip-172-31-41-110:~/terra
provider "aws" {
  region = "ap-south-1"
  access_key = "AKIATX6ULZXLPSOT7VHR"
  secret_key = "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"
}
data "aws_vpc" "existing_vpc" {
  id = "vpc-07639b04723b3cae2"
}
resource "aws_db_subnet_group" "subnet" {
  name       = "subnet"
  subnet_ids = ["subnet-074b79862bb793109","subnet-0ffcc8e0b3e164dc1","subnet-07bd55d8010cbffad"]
}

resource "aws_db_instance" "myinstance" {
  engine               = "mysql"
  engine_version       = "8.0.33"
  allocated_storage    = 20
  storage_type         = "gp2"
  instance_class       = "db.t2.micro"
  identifier           = "mysql-rds-instance"
  username             = "admin"
  password             = "admin123"
  parameter_group_name = "default.mysql8.0"
  vpc_security_group_ids = ["sg-0a754f37be5a64b87"]
  db_subnet_group_name   = aws_db_subnet_group.subnet.name
  skip_final_snapshot    = true
  publicly_accessible    = true
}

output "rds_endpoint" {
  value = aws_db_instance.myinstance.endpoint
}
```

- I had mentioned 3306 port in the security group of that instance which is default port of MYSQL engine

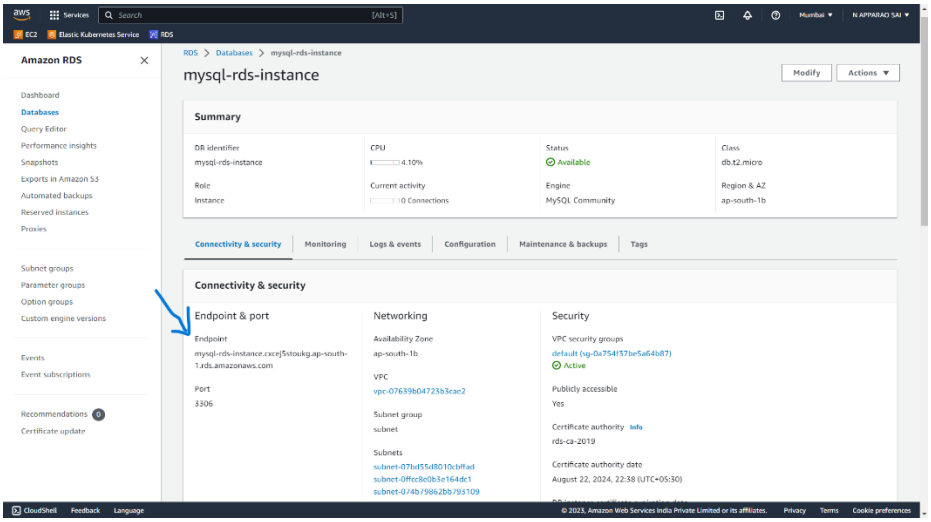
7. To execute the scrips in the files, we have to run the following commands

```
terraform init      # It downloads required plugins for that infrastructure
terraform plan      #  Print output, what it's going to create
terraform apply     #   creates the infrastructure (RDS in this project)
```

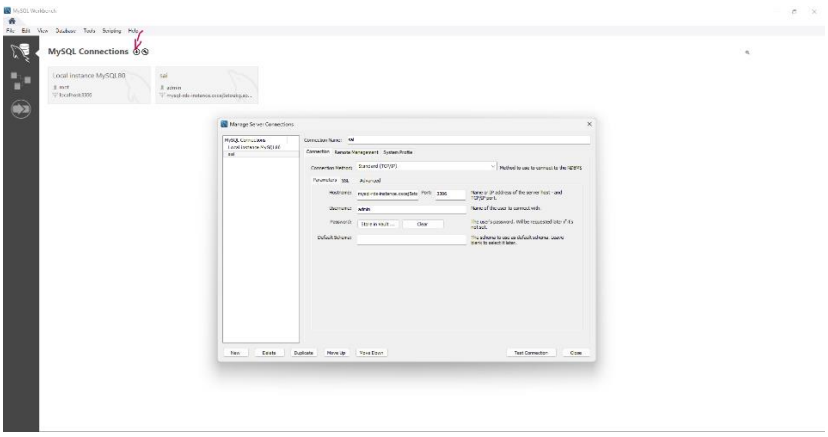
Overall commands that are given in this instance are shown in the above figure.



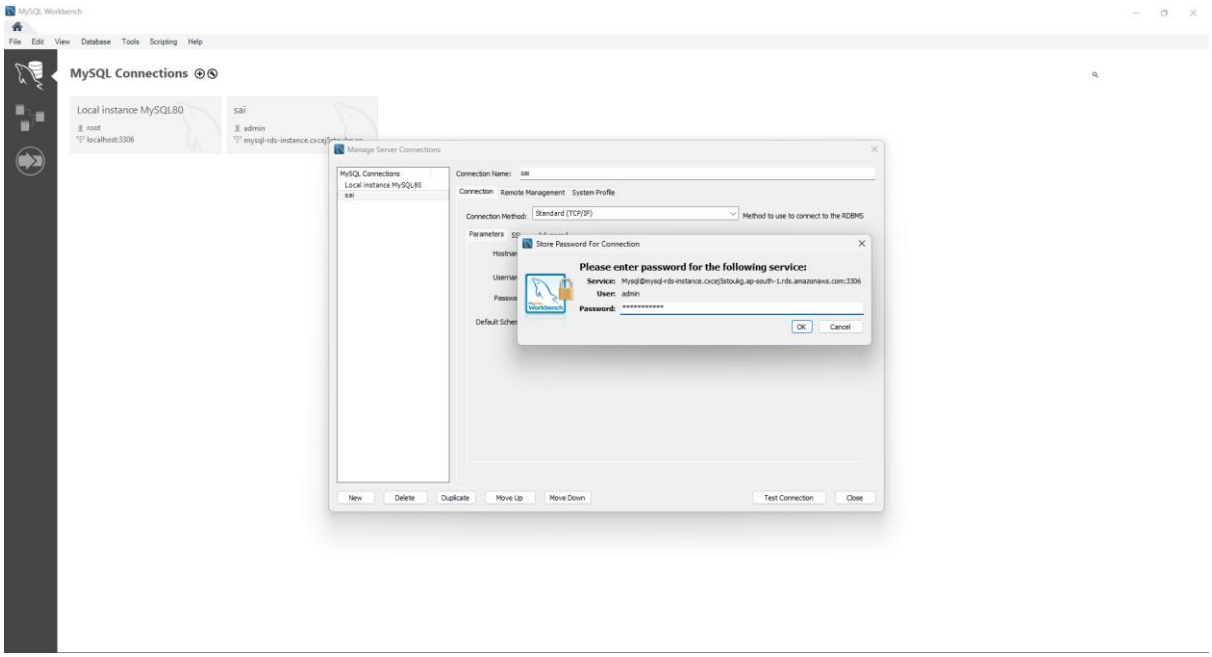
8. After creation of AWS RDS we will get endpoint which is like host name to connect the database to MYSQL workbench , go to the RDS >database>click on the created database there we get Endpoint address , copy that endpoint.



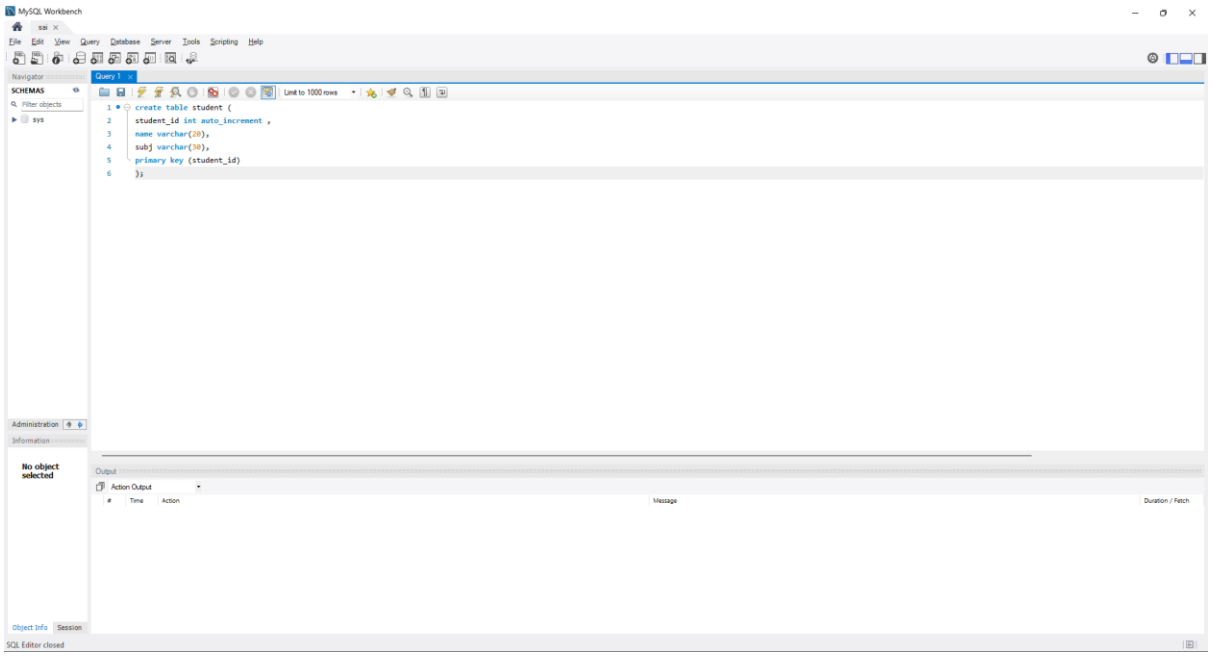
9. Install and open the MYSQL workbench click on ⊕ symbol of mysql connections then add all credentials to connect with our database



➢ Enter the details: Connection name: can be any; Hostname: endpoint of rds , username: one which we gave in “main.tf”file and after clicking on store in vault we can enter password the same which we gave in “main.tf”. Click on test connection and click on connection.



- You will enter into the mysql workbench sever with RDS connection as below, we can excecute sql commands in that workspace.



10. By this we had verified that RDS instance is working correctly by connecting to it using a SQL client, such as MySQL Workbench or pgAdmin.

- By this project we will be able to demonstrate our ability to use Terraform to provision AWS resources, as well as we can understand of how RDS works and how to configure it using Terraform.