**19.DevOps-B24-Terraform-TFLog-AccecssingSecrets-RDS-Deploy**

--- in this session, we will discuss about **terraform backend locking with s3**, **Enabling TF\_LOG**, **terraform resources**, **terraform locals**.

**Enabling TF\_LOG**

--- **note** - most of the times, we do not enable logging, you want to trouble shoot some issue then we will enable logging in terraform. You can fallow below procedure to enable logging in terraform.

--- **Reference** - <https://www.terraform.io/internals/debugging>

--- Terraform has detailed logs which can be enabled by setting the TF\_LOG environment variable to any value. This will cause detailed logs to appear on stderr.

--- You can set **TF\_LOG** to one of the log levels (in order of decreasing verbosity) **TRACE**, **DEBUG**, **INFO**, **WARN** or **ERROR** to change the verbosity of the logs.

**Enable tf log in powershell**

--- open your windows powershell and set below environment variables.

--- $env:TF\_LOG="TRACE"

--- $env:TF\_LOG\_PATH="./logs/terraform.log"

**Enable tf log in bash**

--- **note** – if your using git bash on windows machine then use this method, ignore 1st method.

--- **export TF\_LOG="TRACE"** – it will give you more information when you executing terraform commands, TRACE will give you the most of the information.

--- **export TF\_LOG\_PATH="terraform.txt"**-

--- **export TF\_LOG=" WARN "** – it will give you warn information when you executing terraform commands.

--- **export TF\_LOG=" ERROR "** – it will give you **ERROR** information when you executing terraform commands.

--- **export TF\_LOG="INFO"** – it will give you **INFO** information when you executing terraform commands.

--- **export TF\_LOG="DEBUG"** – it will give you **DEBUG** information when you executing terraform commands.

**Create terraform manifest for RDS**

**Provisioning rds using 1st way**

--- rds.tf

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

  allocated\_storage    = 10

  engine               = "mysql"

  engine\_version       = "5.7"

  instance\_class       = "db.t3.micro"

  name                 = "mydb"

  username             = var.username

  password             = var.password

  parameter\_group\_name = "default.mysql5.7"

  skip\_final\_snapshot  = true

}

--- rds-variable.tf

variable "username" {

  sensitive = true

}

variable "password" {

  sensitive = true

}

--- terraform.autotfvars

aws\_region = "us-east-1"

vpc\_cidr = "10.1.0.0/16"

public\_subnet1\_cidr = "10.1.1.0/24"

public\_subnet2\_cidr = "10.1.2.0/24"

public\_subnet3\_cidr = "10.1.3.0/24"

private\_subnet\_cidr = "10.1.20.0/24"

vpc\_name = "terraform-aws-dev"

IGW\_name = "terraform-aws-igw-dev"

public\_subnet1\_name = "Terraform\_Public\_Subnet1-dev"

public\_subnet2\_name = "Terraform\_Public\_Subnet2-dev"

public\_subnet3\_name = "Terraform\_Public\_Subnet3-dev"

private\_subnet\_name = "Terraform\_Private\_Subnet-dev"

Main\_Routing\_Table = "Terraform\_Main\_table-dev"

username="admin"

password="admin123"

key\_name = "terraform-key"

environment = "dev"

**Provisioning rds using 2nd way**

--- Reference - <https://stackoverflow.com/questions/65603923/terraform-rds-database-credentials>

--- rds.tf

data "aws\_secretsmanager\_secret" "password" {

  name = "test-db-password"

  depends\_on = [random\_password.master, aws\_secretsmanager\_secret.password, aws\_secretsmanager\_secret\_version.password]

}

data "aws\_secretsmanager\_secret\_version" "password" {

  secret\_id = data.aws\_secretsmanager\_secret.password.id

  depends\_on = [random\_password.master, aws\_secretsmanager\_secret.password, aws\_secretsmanager\_secret\_version.password]

}

output "secret\_id\_pass" {

  value = data.aws\_secretsmanager\_secret\_version.password.secret\_string

  sensitive = true

}

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

  identifier            = "testdb"

  allocated\_storage    = 20

  storage\_type         = "gp2"

  engine               = "mysql"

  engine\_version       = "5.7"

  instance\_class       = "db.t2.medium"

  name                 = "mydb"

  username             = "dbadmin"

  password             = data.aws\_secretsmanager\_secret\_version.password.secret\_string

  publicly\_accessible = true

  db\_subnet\_group\_name = aws\_db\_subnet\_group.default.id

  depends\_on = [random\_password.master, aws\_secretsmanager\_secret.password, aws\_secretsmanager\_secret\_version.password]

}

--- secret-manager.tf

resource "random\_password" "master"{

  length           = 16

  special          = true

  override\_special = "\_!%^"

}

resource "aws\_secretsmanager\_secret" "password" {

  name = "test-db-password"

}

resource "aws\_secretsmanager\_secret\_version" "password" {

  secret\_id = aws\_secretsmanager\_secret.password.id

  secret\_string = random\_password.master.result

}

# this block will create random password in aws secret manager And then in a separate TF config for your database, you can use the secret from AWS Secrets Manager.

#---

--- rds-subnet-group.tf

resource "aws\_db\_subnet\_group" "default" {

  name       = "main"

  subnet\_ids = [aws\_subnet.subnet1-public.id, aws\_subnet.subnet2-public.id, aws\_subnet.subnet3-public.id]

  tags = {

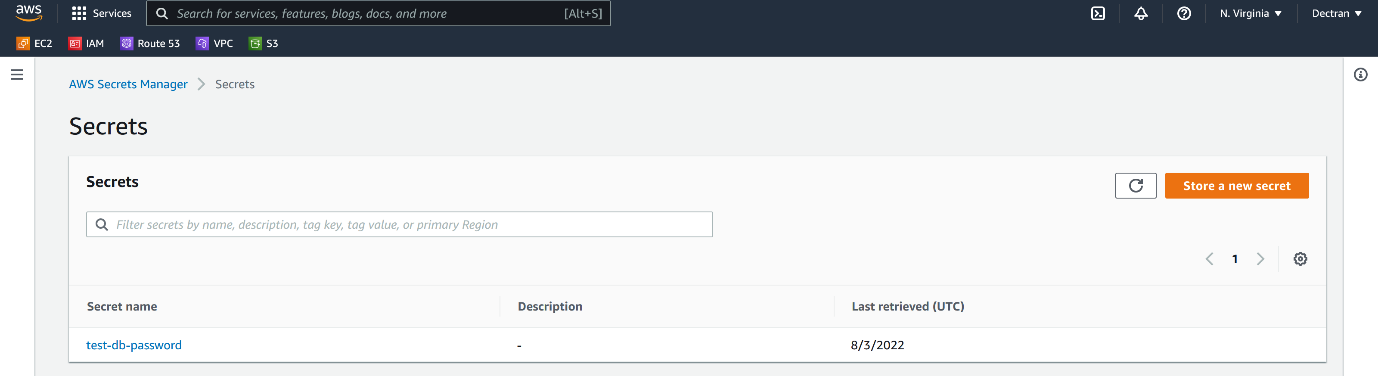
    Name = "My DB subnet group"

  }

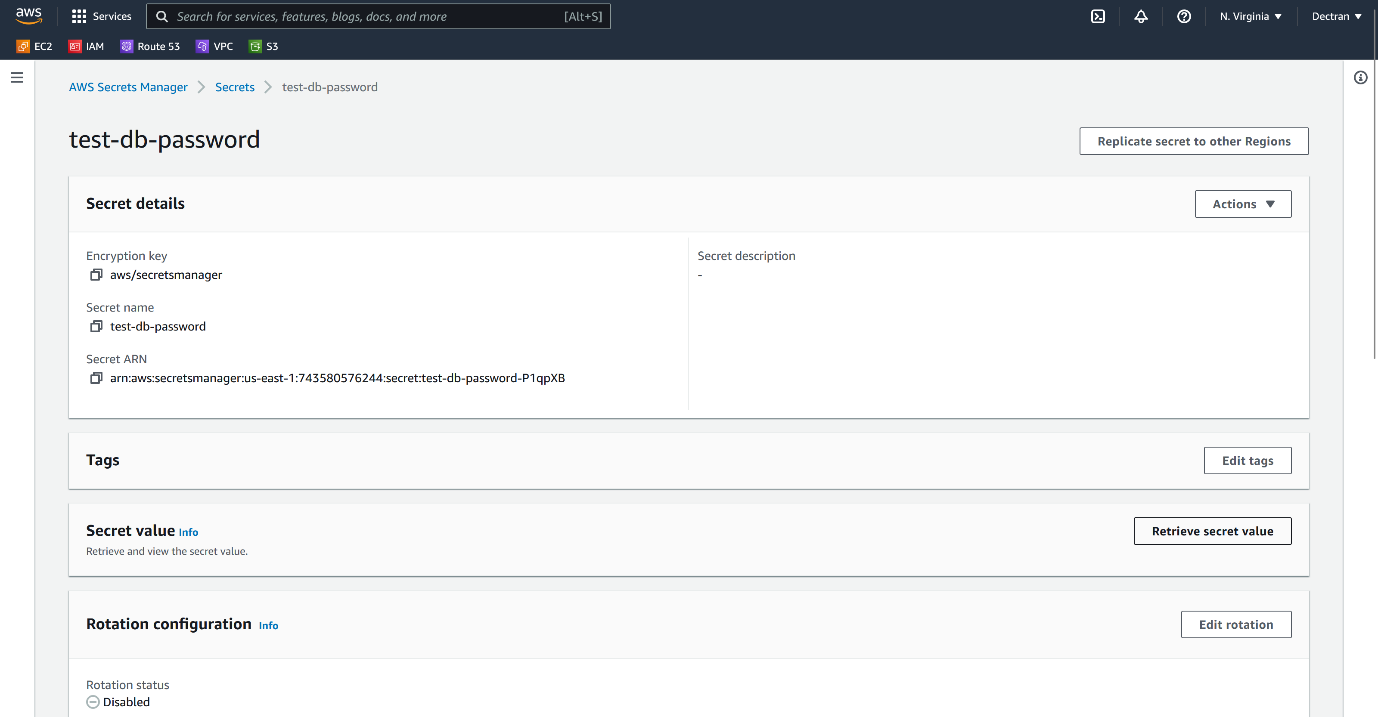
}

**Retrieve password form aws console**

--- go to aws secret manager



--- click on test-db-password



--- click on retrieve secret value.

--- **note** – this is how we retrieve the password form aws console.

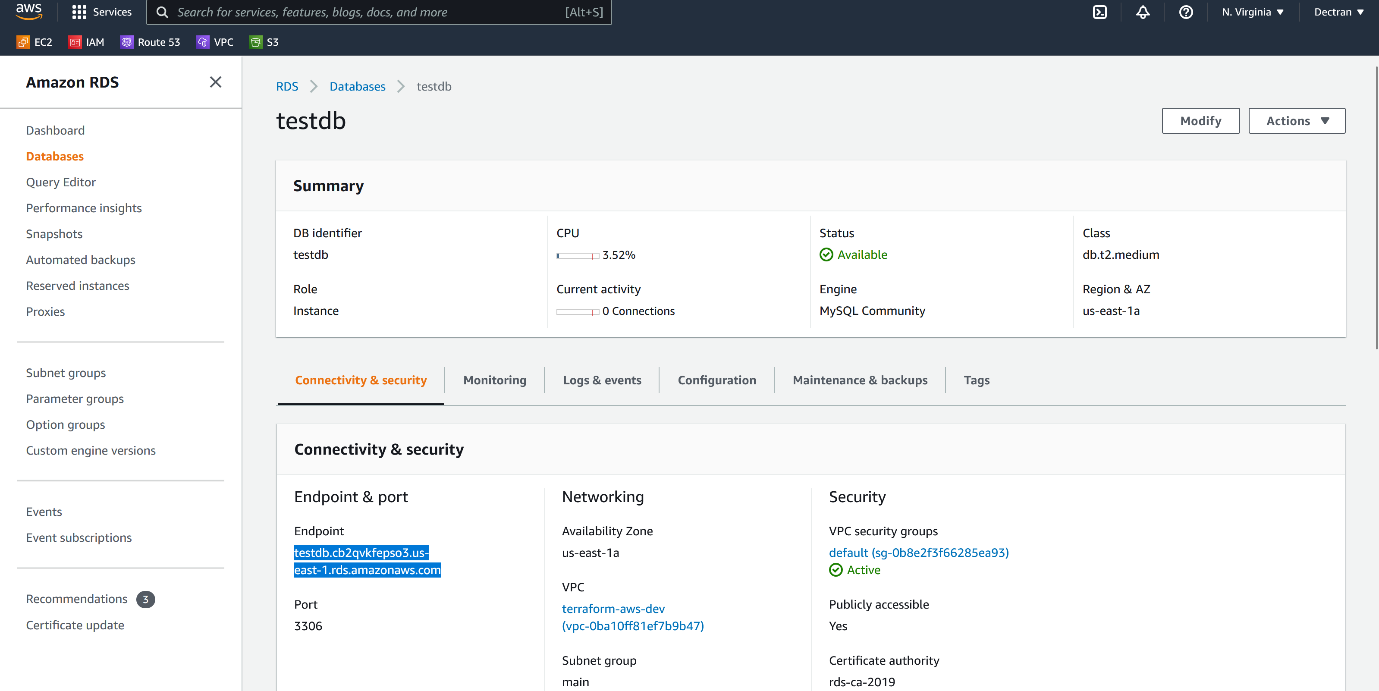
**Connect rds db**

--- Reference - <https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170>

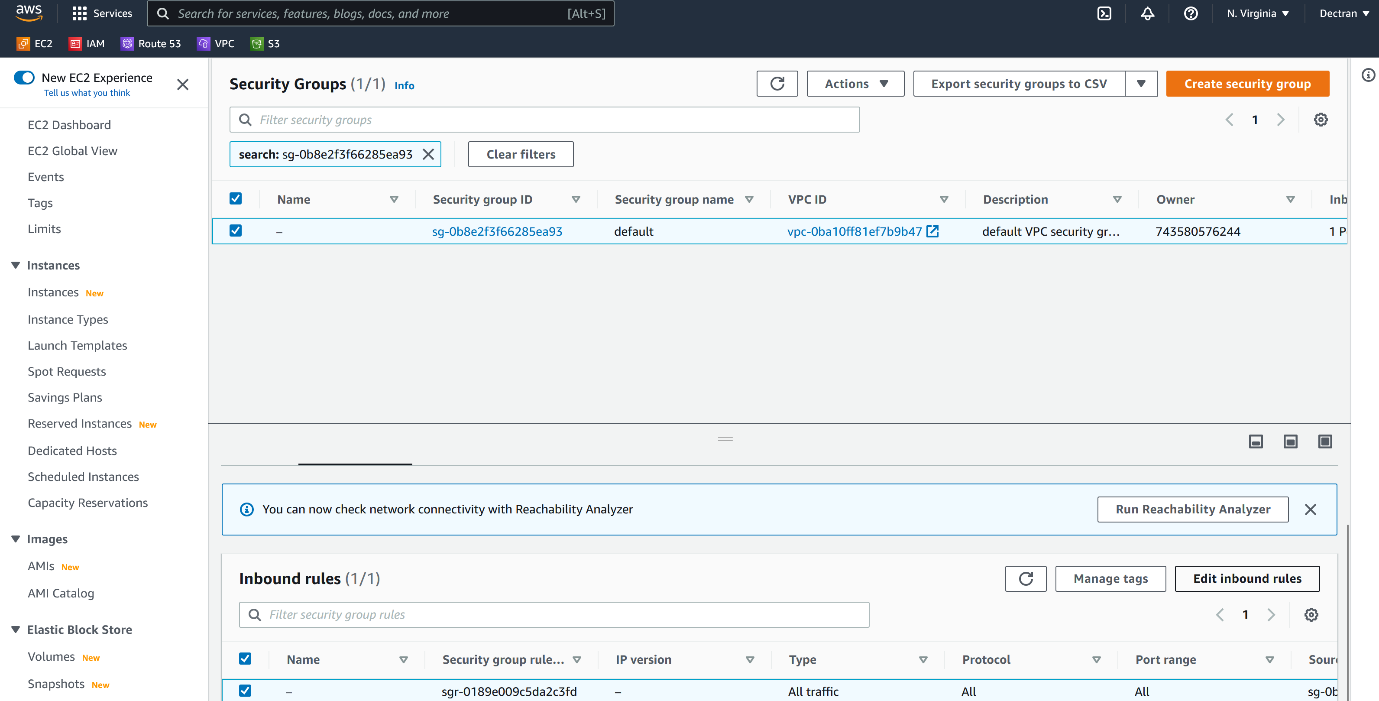
--- mysql workbench reference - <https://dev.mysql.com/downloads/workbench/>

--- download and install the both tools.

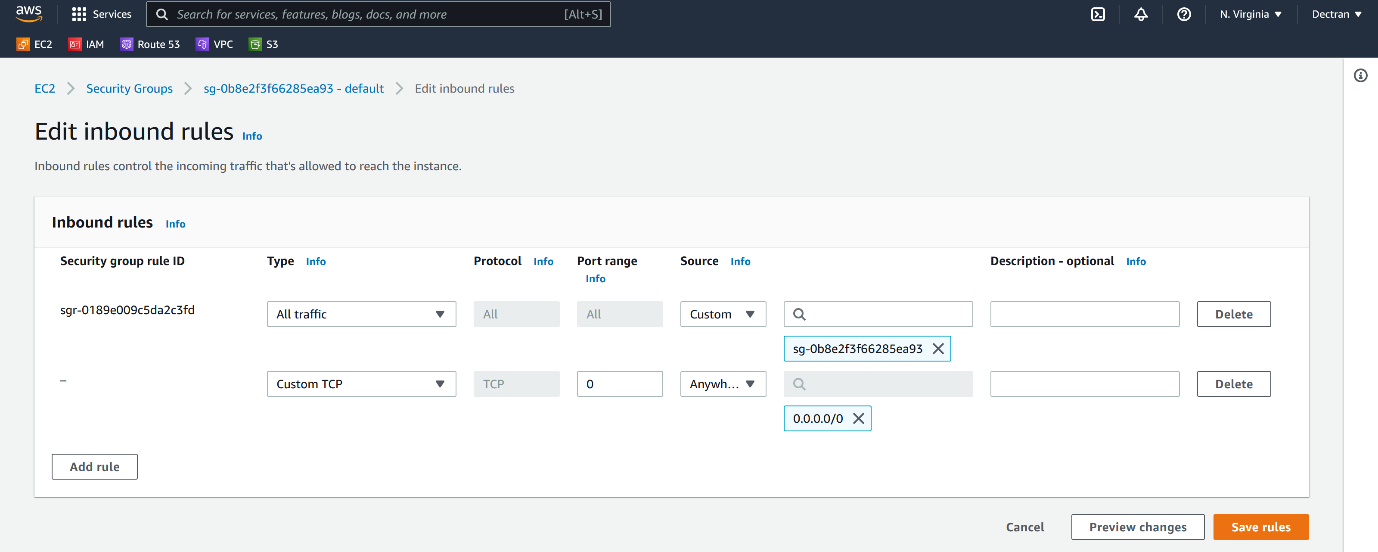
--- go to aws console and rds.



--- copy the rds endpoint and click on security group.

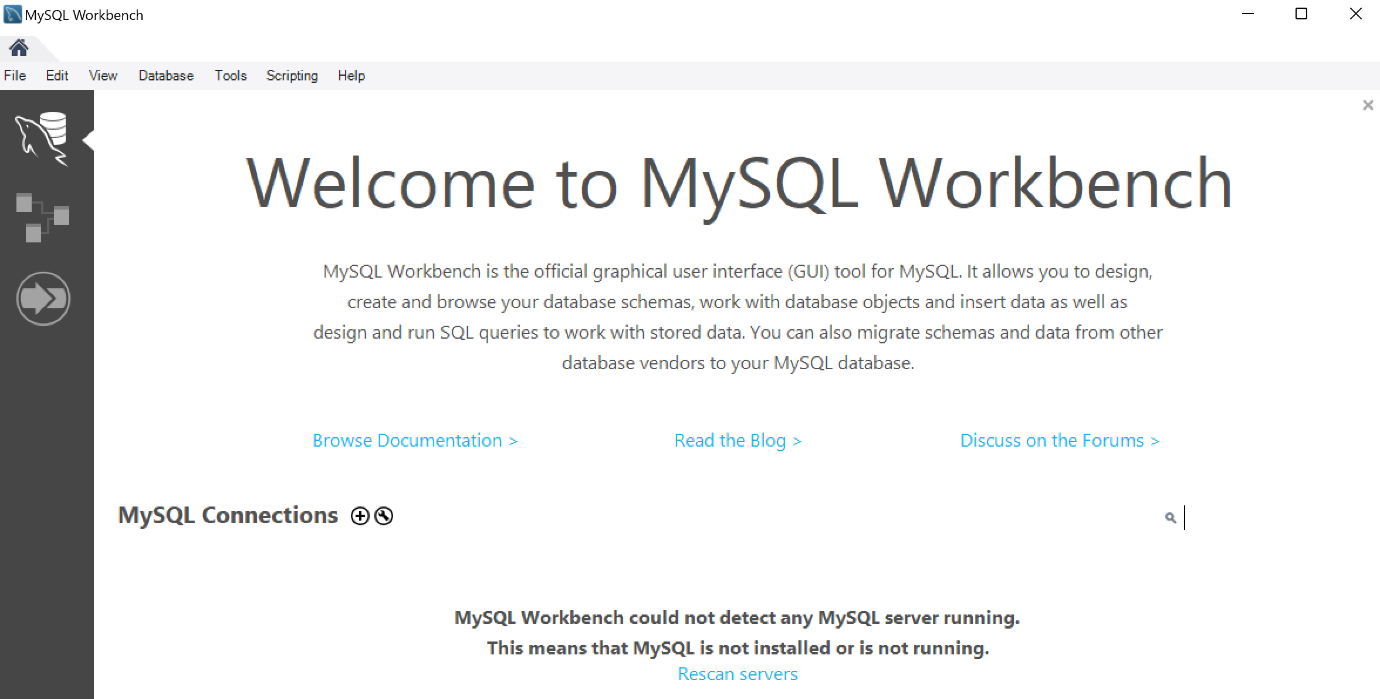


--- click on edit inbound rules.

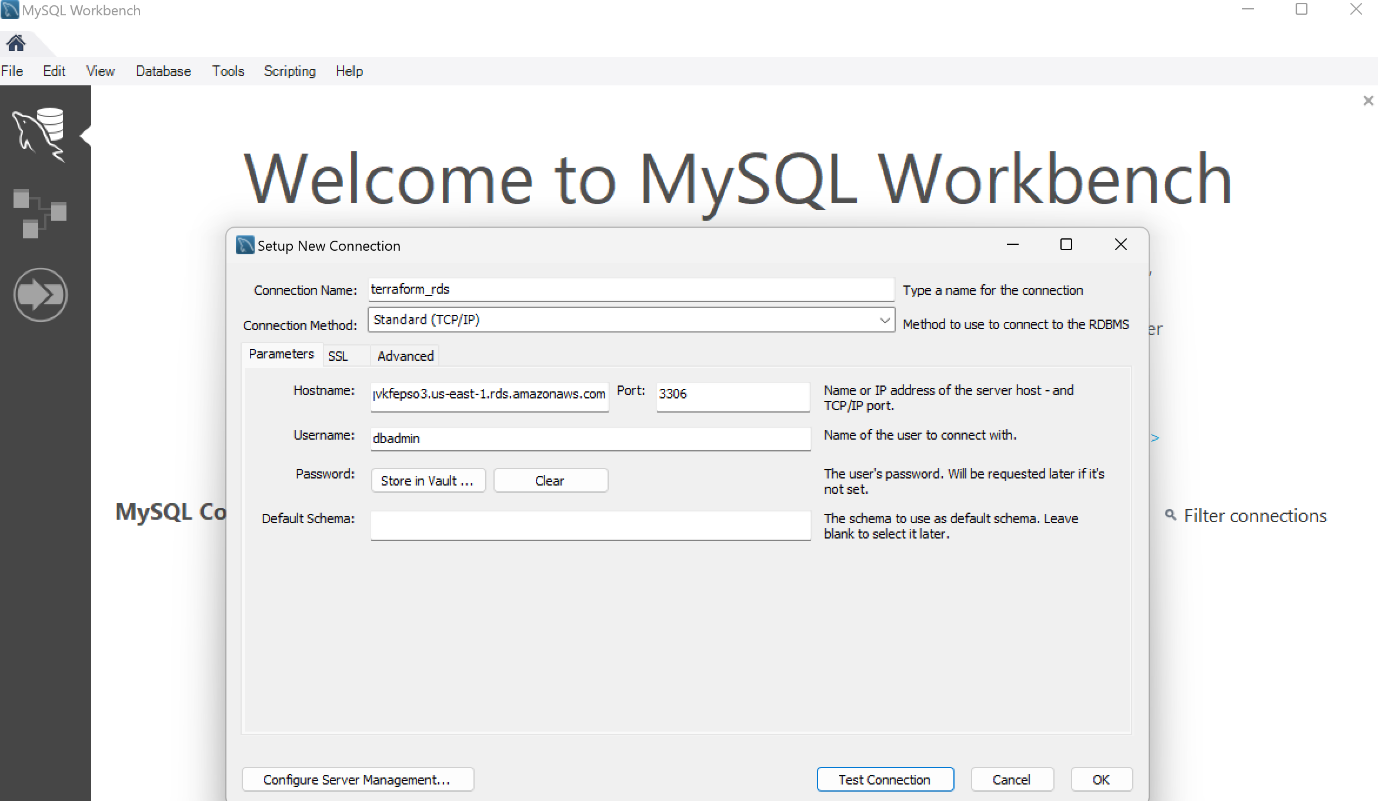


--- click on save rules. Please add mysql port in security group.

--- go to the mysql workbench



--- click on mysql connection



--- give the terraform endpoint in hostname, user name and connection name. now click on test connection.