Lab Exercise 12– Creating an AWS RDS Instance in Terraform

Objective:

Learn how to use Terraform to create an AWS RDS instance.

Prerequisites:

- Terraform installed on your machine.
- AWS CLI configured with the necessary credentials.

Steps:

1. Create a Terraform Directory:

```
mkdir terraform-rds
cd terraform-rds
```

```
D:\College\Sem-6\System Provisioning\Lab>mkdir terraform-rds

D:\College\Sem-6\System Provisioning\Lab>cd terraform-rds

D:\College\Sem-6\System Provisioning\Lab\terraform-rds>
```

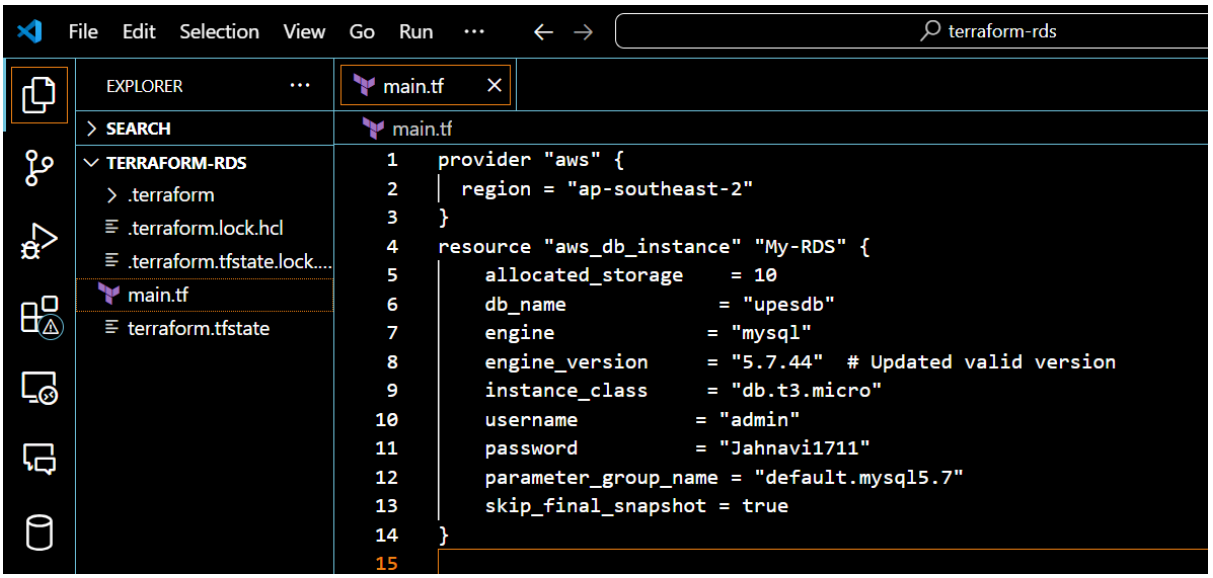
2. Create Terraform Configuration Files:

Create a file named main.tf:

 # main.tf

```
provider "aws" {
  region = "us-east-1"
}
```

```
resource "aws_db_instance" "My-RDS" {
  allocated_storage = 10
  db_name = "upesdb"
  engine = "mysql"
  engine_version = "5.7"
  instance_class = "db.t2.micro"
  username = "admin"
  password = "Hitesh111"
  parameter_group_name = "default.mysql5.7"
  skip_final_snapshot = true
  }
```



- Replace "YourPassword123" with a secure password and "your-security-group-id" with your actual security group ID.
- In this configuration, we define an AWS RDS instance with specific settings, such as engine type, instance class, and security group.

3. Initialize and Apply:

- Run the following Terraform commands to initialize and apply the configuration:

```
terraform init
```

terraform apply

```
- Installed hashicorp/aws v5.88.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

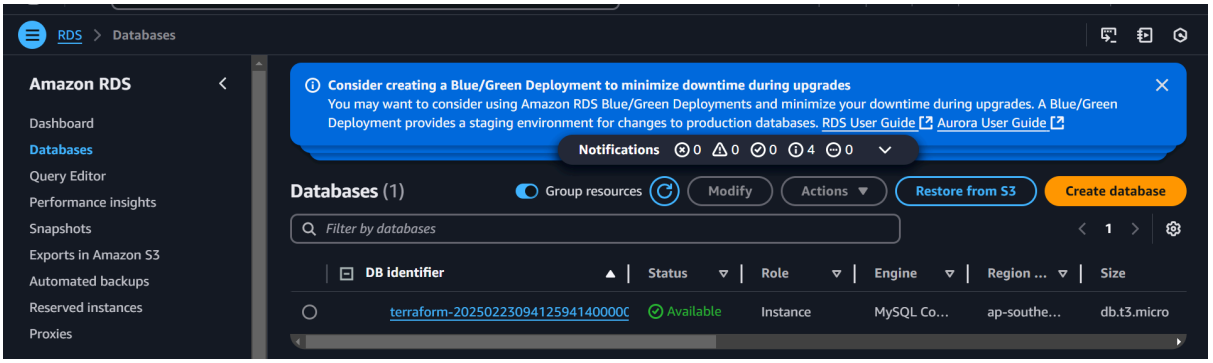
```
aws_db_instance.My-RDS: Still creating... [2m0s elapsed]
aws_db_instance.My-RDS: Still creating... [2m10s elapsed]
aws_db_instance.My-RDS: Still creating... [2m20s elapsed]
aws_db_instance.My-RDS: Still creating... [2m30s elapsed]
aws_db_instance.My-RDS: Still creating... [2m40s elapsed]
aws_db_instance.My-RDS: Still creating... [2m50s elapsed]
aws_db_instance.My-RDS: Still creating... [3m0s elapsed]
aws_db_instance.My-RDS: Still creating... [3m10s elapsed]
aws_db_instance.My-RDS: Still creating... [3m20s elapsed]
aws_db_instance.My-RDS: Still creating... [3m30s elapsed]
aws_db_instance.My-RDS: Still creating... [3m40s elapsed]
aws_db_instance.My-RDS: Still creating... [3m50s elapsed]
aws_db_instance.My-RDS: Creation complete after 3m59s [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

- Terraform will prompt you to confirm the creation of the RDS instance. Type yes and press Enter.

4. Verify RDS Instance in AWS Console:

- Log in to the AWS Management Console and navigate to the RDS service.
- Verify that the specified RDS instance with the specified settings has been created.



5. Update RDS Configuration:

- If you want to modify the RDS instance configuration, update the main.tf file with the desired changes.
- Rerun the terraform apply command to apply the changes:

```
terraform apply
```

6. Clean Up:

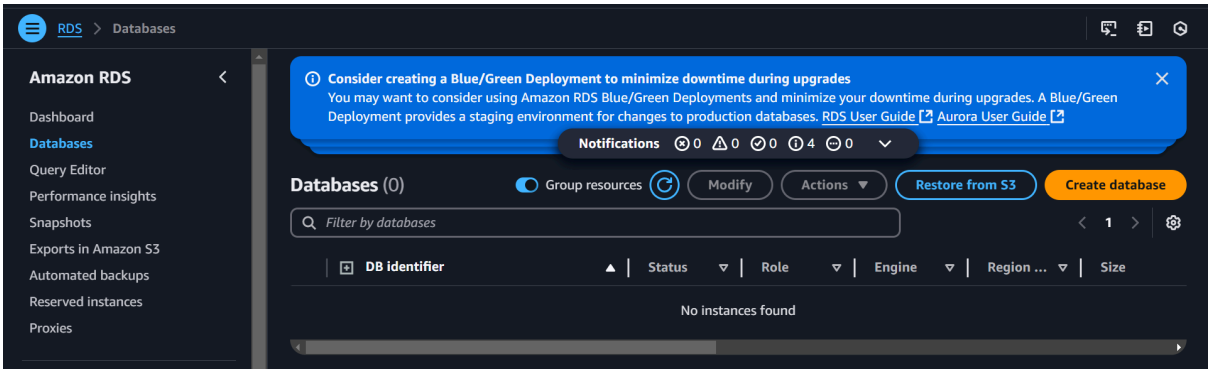
After testing, you can clean up the RDS instance:

```
terraform destroy
```

Confirm the destruction by typing yes.

```
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 2m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 2m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 2m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 2m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 2m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 3m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 3m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 3m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 3m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 3m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 3m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-IFIL7VSV3I22R4AKNAHS4MWPOU, 4m0s elapsed]
aws_db_instance.My-RDS: Destruction complete after 4m7s

Destroy complete! Resources: 1 destroyed.
```



7. Conclusion:

This lab exercise demonstrates how to use Terraform to create an AWS RDS instance. You learned how to define RDS settings, initialize and apply the Terraform configuration, and verify the creation of the RDS instance in the AWS Management Console. Experiment with different RDS settings in the main.tf file to observe how