Lab Exercise 12– Creating an AWS RDS Instance in Terraform

Objective:

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

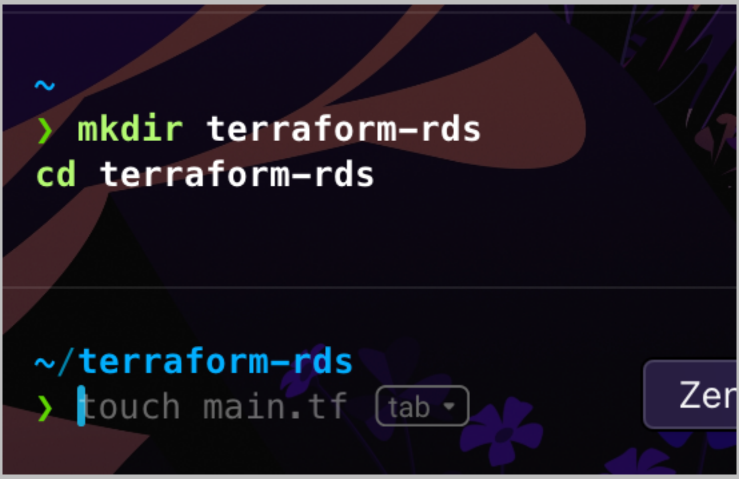
Prerequisites:

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

Steps:

1. Create a Terraform Directory:

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



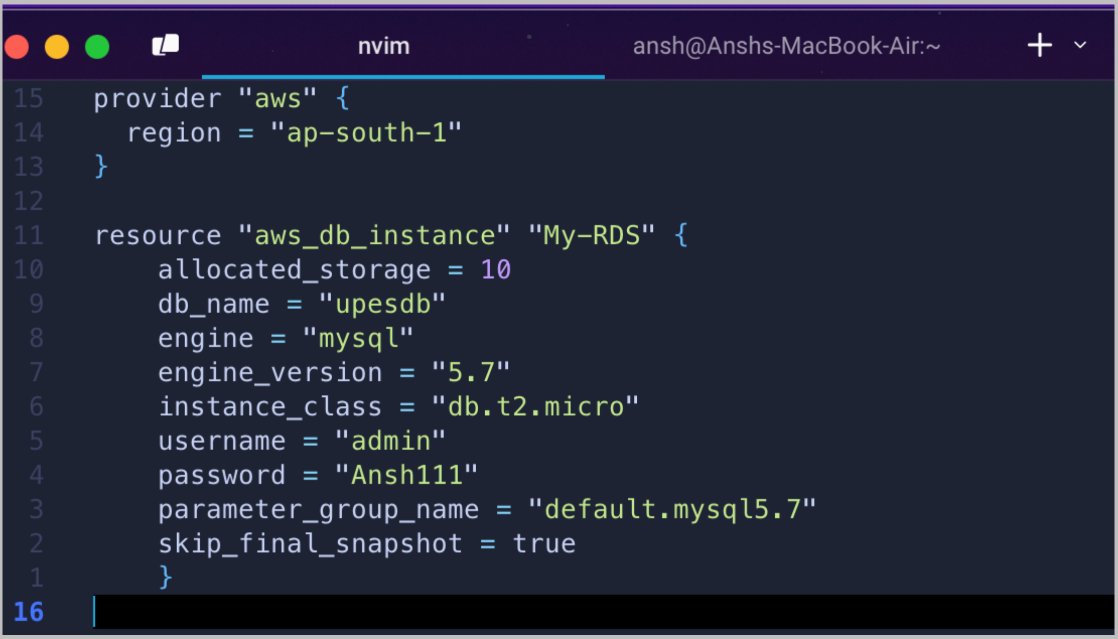
```
~
❯ mkdir terraform-rds
cd terraform-rds

~/terraform-rds
❯ touch main.tf
```

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
  }
```



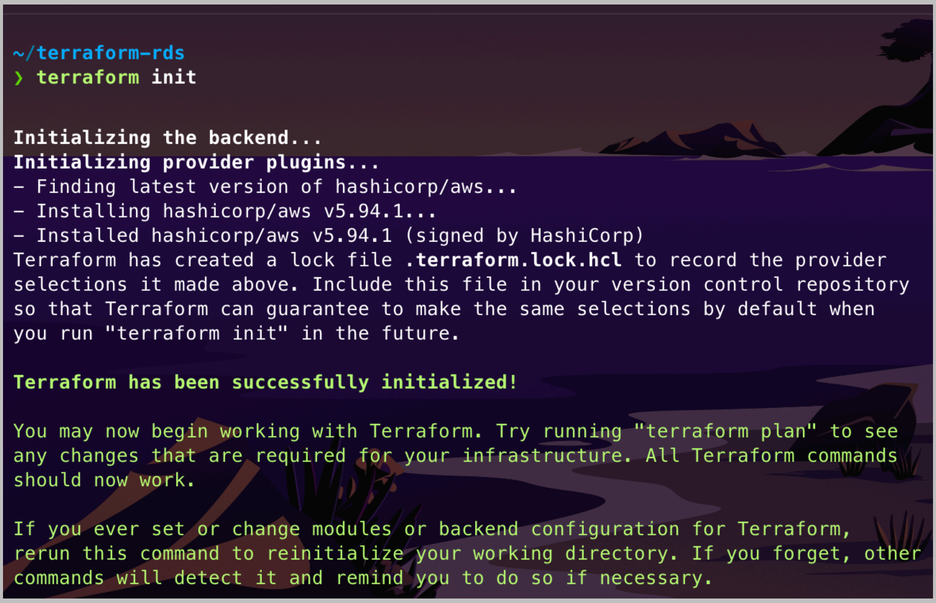
```
provider "aws" {
  region = "ap-south-1"
}

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 = "Ansh111"
    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:

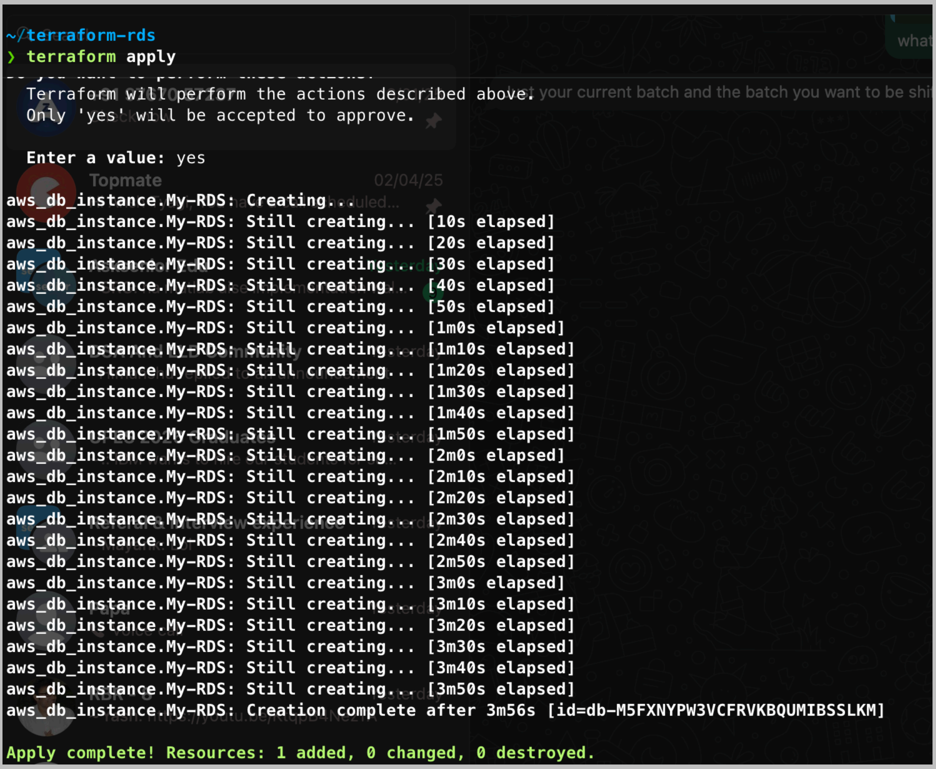
- Run the following Terraform commands to initialize and apply the configuration:

terraform init



```
~/terraform-rds
❯ terraform init

Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.94.1...
- Installed hashicorp/aws v5.94.1 (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.
```

terraform apply

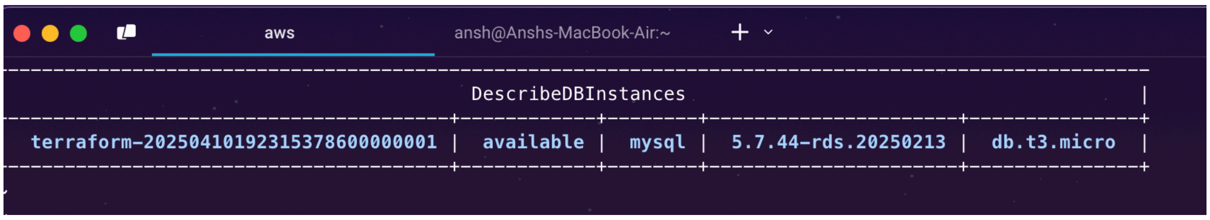


```
~/terraform-rds
❯ terraform apply
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_db_instance.My-RDS: Creating...
aws_db_instance.My-RDS: Still creating... [10s elapsed]
aws_db_instance.My-RDS: Still creating... [20s elapsed]
aws_db_instance.My-RDS: Still creating... [30s elapsed]
aws_db_instance.My-RDS: Still creating... [40s elapsed]
aws_db_instance.My-RDS: Still creating... [50s elapsed]
aws_db_instance.My-RDS: Still creating... [1m0s elapsed]
aws_db_instance.My-RDS: Still creating... [1m10s elapsed]
aws_db_instance.My-RDS: Still creating... [1m20s elapsed]
aws_db_instance.My-RDS: Still creating... [1m30s elapsed]
aws_db_instance.My-RDS: Still creating... [1m40s elapsed]
aws_db_instance.My-RDS: Still creating... [1m50s elapsed]
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 3m56s [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM]

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.

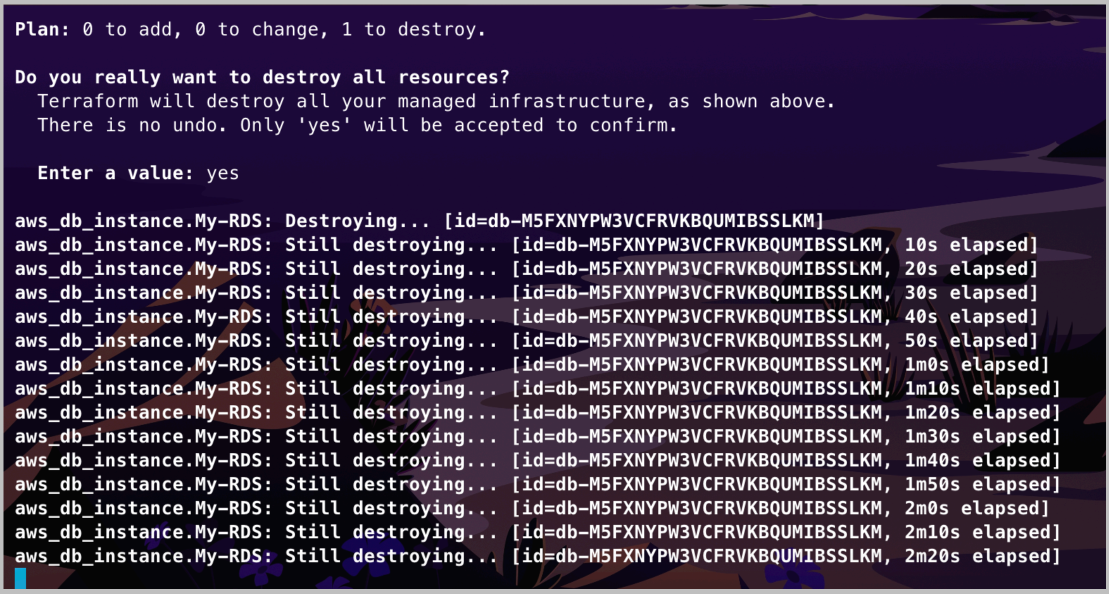


```
---------------------------------------------------------------------------------------------------------
|                                          DescribeDBInstances                                          |
+---------------------------------------+------------+--------+-----------------------+---------------+
|  terraform-20250410192315378600000001 |  available |  mysql |  5.7.44-rds.20250213  |  db.t3.micro  |
+---------------------------------------+------------+--------+-----------------------+---------------+
```

5. Clean Up:

After testing, you can clean up the RDS instance:

```
terraform destroy
```



```
Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_db_instance.My-RDS: Destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 1m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 1m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 1m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 1m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 1m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 1m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 2m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 2m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-M5FXNYPW3VCFRVKBQUMIBSSLKM, 2m20s elapsed]
```

Confirm the destruction by typing yes.

6. 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