Lab Exercise 10– Creating an AWS RDS Instance in Terraform

Objective:

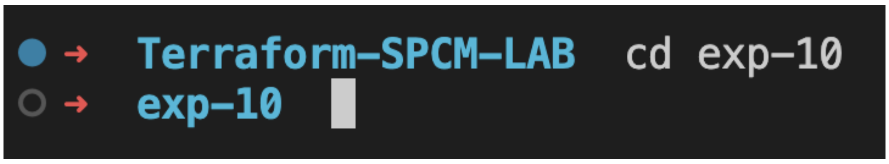
Learn how to use Terraform to create an AWS RDS instance.

Prerequisites:

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

Steps:

1. Create a Terraform Directory:

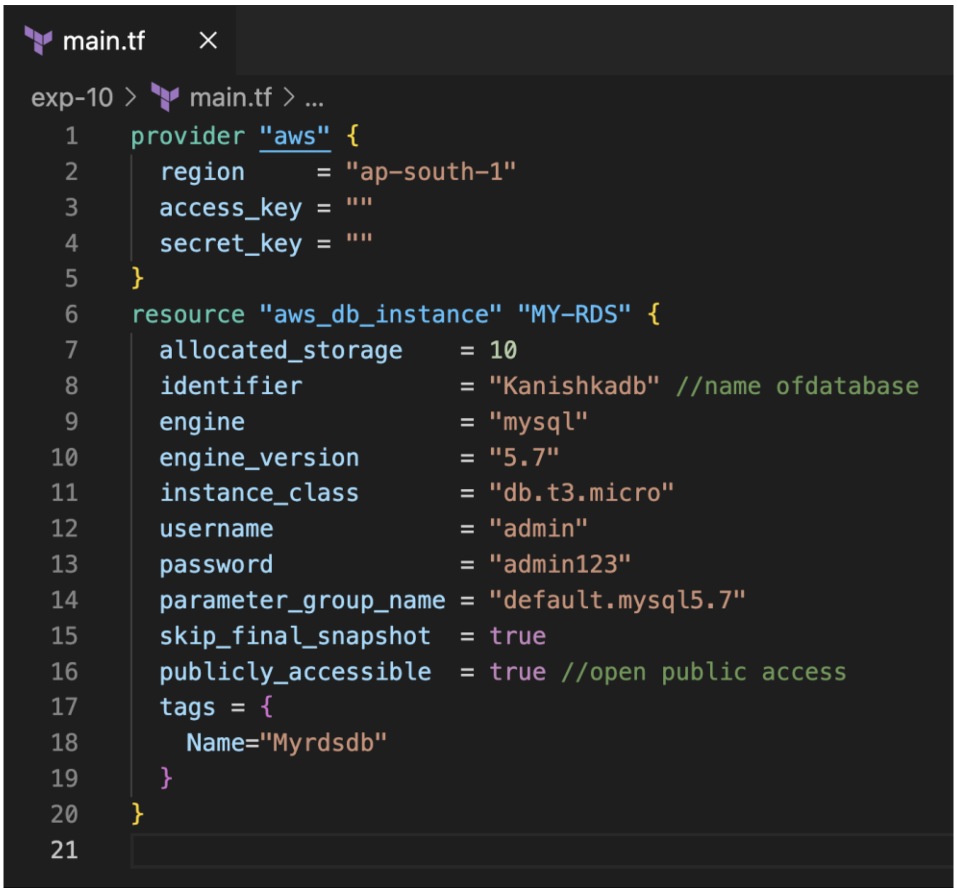


```
● ➜  Terraform-SPCM-LAB  cd exp-10
○ ➜  exp-10
```

2. Create Terraform Configuration Files:

Create a file named main.tf:

# main.tf



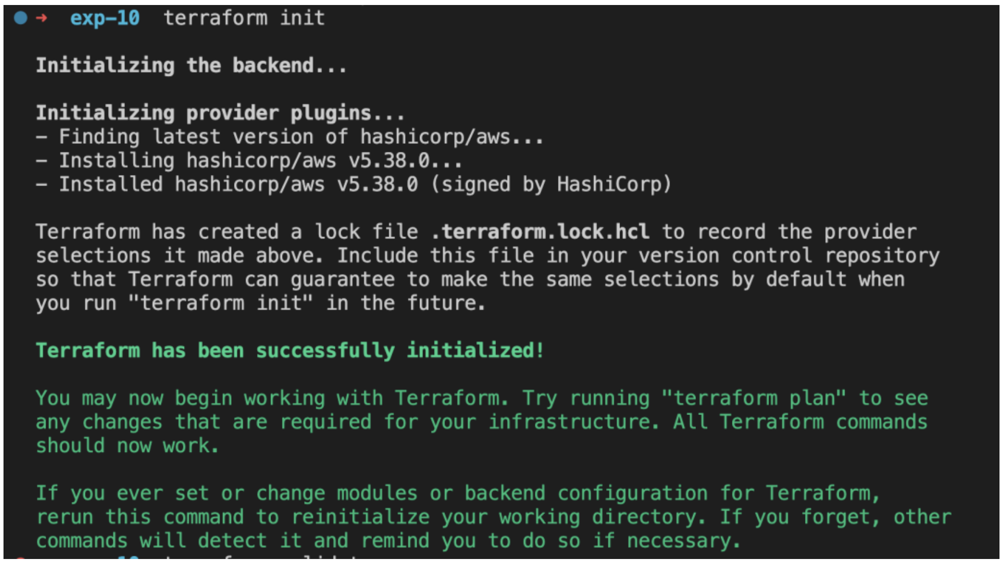
```
provider "aws" {
  region     = "ap-south-1"
  access_key = ""
  secret_key = ""
}
resource "aws_db_instance" "MY-RDS" {
  allocated_storage     = 10
  identifier            = "Kanishkadb" //name ofdatabase
  engine                = "mysql"
  engine_version        = "5.7"
  instance_class        = "db.t3.micro"
  username              = "admin"
  password              = "admin123"
  parameter_group_name = "default.mysql5.7"
  skip_final_snapshot  = true
  publicly_accessible  = true //open public access
  tags = {
    Name="Myrdsdb"
  }
}

```

• Replace "YourPassword123" with a secure password and "your-security-group-id" with your actual security group ID.

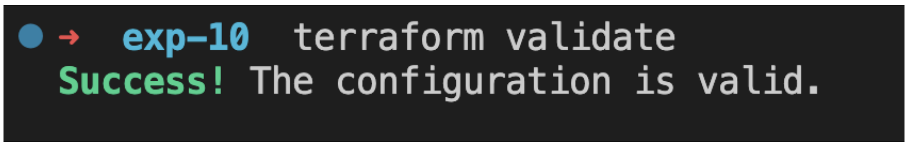
• In this configuration, we define an AWS RDS instance with specific settings, such as engine type, instance class, and security group.

3. Initialize and Apply:

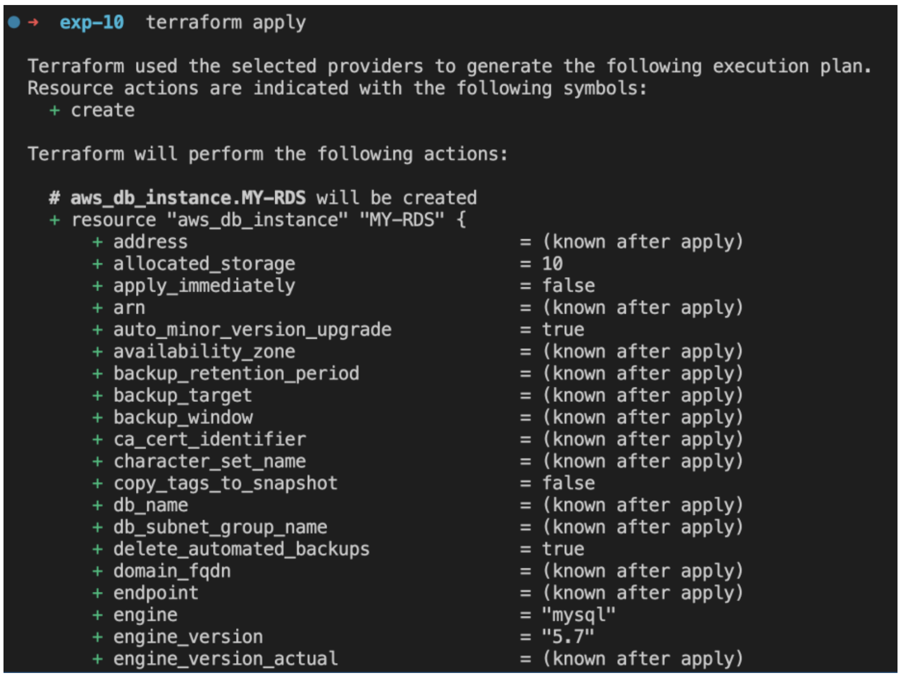
• Run the following Terraform commands to initialize and apply the configuration:



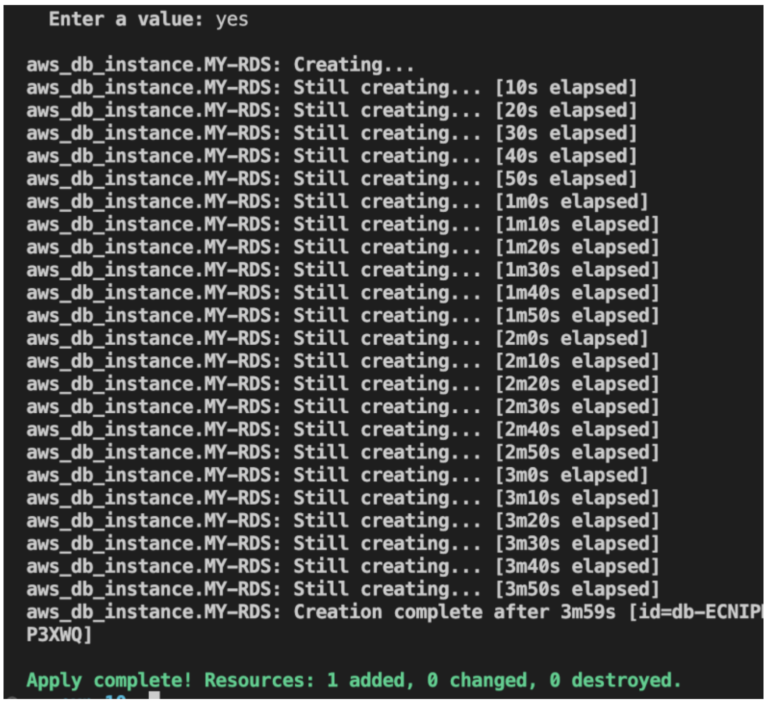
```
● ➜  exp-10  terraform init

 Initializing the backend...

 Initializing provider plugins...
 - Finding latest version of hashicorp/aws...
 - Installing hashicorp/aws v5.38.0...
 - Installed hashicorp/aws v5.38.0 (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.
```



```
● ➜  exp-10  terraform validate
  Success! The configuration is valid.
```



```
● ➜  exp-10  terraform apply

  Terraform used the selected providers to generate the following execution plan.
  Resource actions are indicated with the following symbols:
    + create

  Terraform will perform the following actions:

    # aws_db_instance.MY-RDS will be created
    + resource "aws_db_instance" "MY-RDS" {
        + address                               = (known after apply)
        + allocated_storage                     = 10
        + apply_immediately                     = false
        + arn                                   = (known after apply)
        + auto_minor_version_upgrade            = true
        + availability_zone                     = (known after apply)
        + backup_retention_period               = (known after apply)
        + backup_target                         = (known after apply)
        + backup_window                         = (known after apply)
        + ca_cert_identifier                    = (known after apply)
        + character_set_name                    = (known after apply)
        + copy_tags_to_snapshot                 = false
        + db_name                               = (known after apply)
        + db_subnet_group_name                  = (known after apply)
        + delete_automated_backups              = true
        + domain_fqdn                           = (known after apply)
        + endpoint                              = (known after apply)
        + engine                                = "mysql"
        + engine_version                        = "5.7"
        + engine_version_actual                 = (known after apply)
```



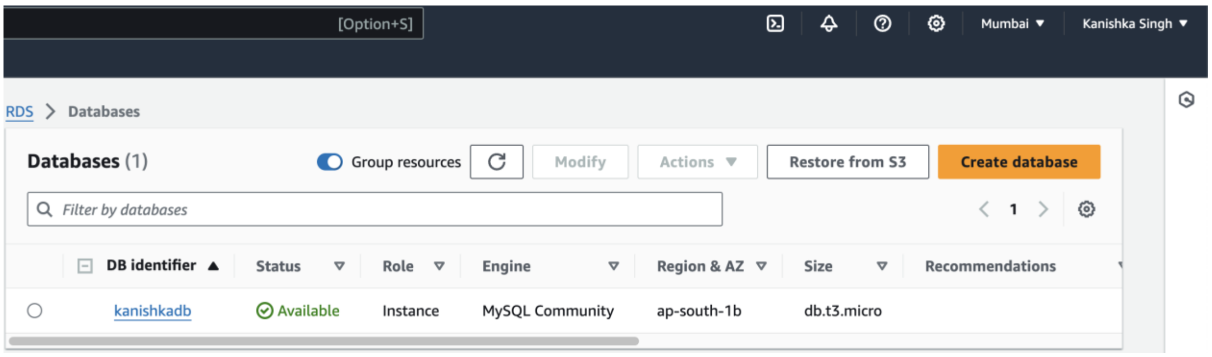
```
  Enter a value: yes

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 3m59s [id=db-ECNIP
P3XWQ]

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:

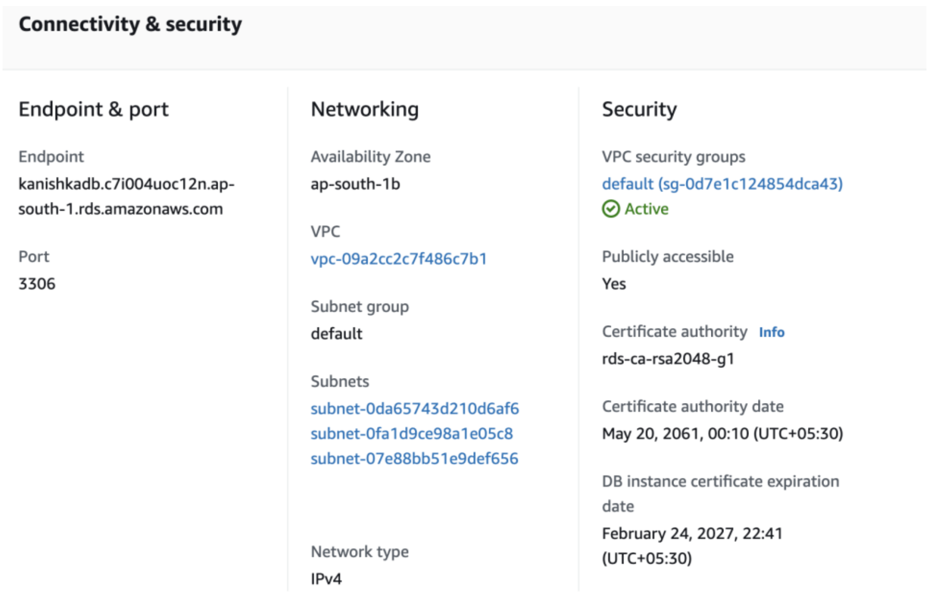
• Log in to the AWS Management Console and navigate to the RDS service.

• Verify that the specified RDS instance with the specified settings has been created.



Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations
kanishkadb	Available	Instance	MySQL Community	ap-south-1b	db.t3.micro	



Connectivity & security

Endpoint & port

Endpoint
kanishkadb.c7i004uoc12n.ap-south-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
ap-south-1b

VPC
vpc-09a2cc2c7f486c7b1

Subnet group
default

Subnets
subnet-0da65743d210d6af6
subnet-0fa1d9ce98a1e05c8
subnet-07e88bb51e9def656

Network type
IPv4

Security

VPC security groups
default (sg-0d7e1c124854dca43)
Active

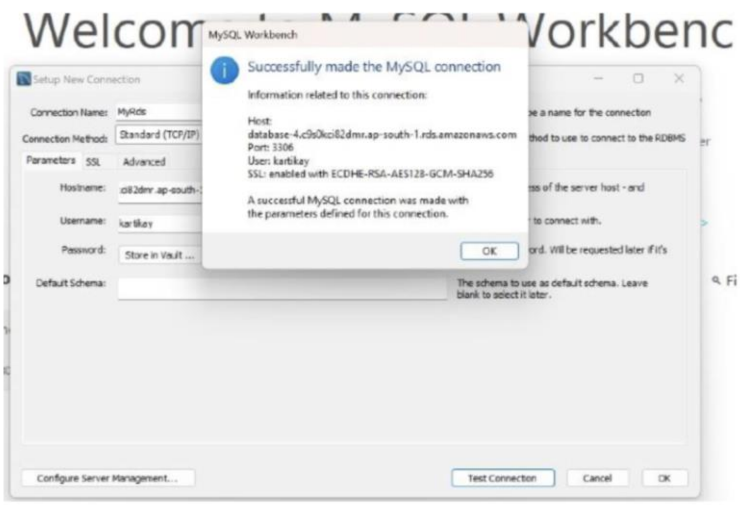
Publicly accessible
Yes

Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 20, 2061, 00:10 (UTC+05:30)

DB instance certificate expiration date
February 24, 2027, 22:41 (UTC+05:30)

5. Connect to MySQL



MySQL Workbench

Successfully made the MySQL connection

Information related to this connection:

Host:
database-4.c9s0kci82dmr.ap-south-1.rds.amazonaws.com
Port: 3306
User: kartikay
SSL: enabled with ECDHE-RSA-AES128-GCM-SHA256

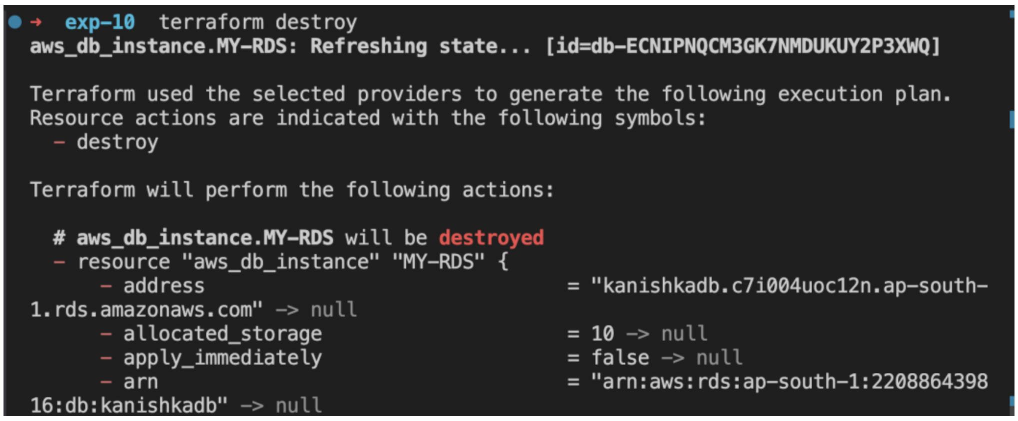
A successful MySQL connection was made with
the parameters defined for this connection.

Welcom
Vorkbenc
Setup New Connection
Connection Name: MyRds
Connection Method: Standard (TCP/IP)
Parameters
SSL
Advanced
Hostname: ci82dmr.ap-south-
Username: kartikay
Password: Store in Vault ...
Default Schema:
The schema to use as default schema. Leave blank to select it later.
OK
Configure Server Management...
Test Connection
Cancel
OK

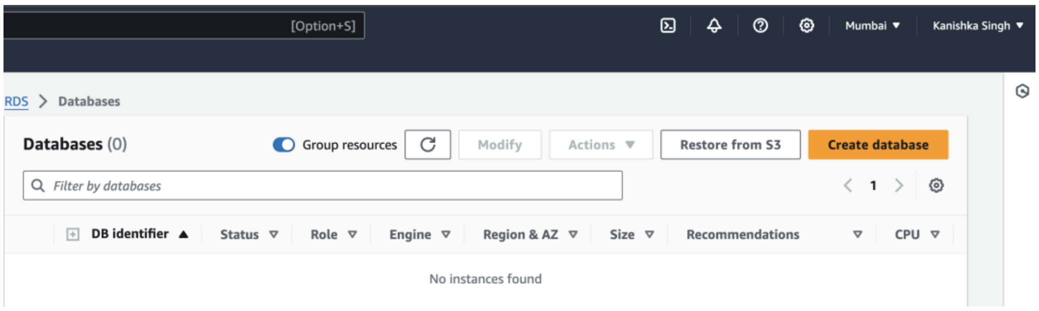
6. Clean Up:

After testing, you can clean up the RDS instance:

terraform destroy



```
● ➜  exp-10  terraform destroy
aws_db_instance.MY-RDS: Refreshing state... [id=db-ECNIPNQCM3GK7NMDUKUY2P3XWQ]

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_db_instance.MY-RDS will be destroyed
  - resource "aws_db_instance" "MY-RDS" {
      - address                              = "kanishkadb.c7i004uoc12n.ap-south-
1.rds.amazonaws.com" -> null
      - allocated_storage                    = 10 -> null
      - apply_immediately                    = false -> null
      - arn                                  = "arn:aws:rds:ap-south-1:2208864398
16:db:kanishkadb" -> null
```



Databases (0)

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU
No instances found							

Confirm the destruction by typing yes.

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.