

Lab Exercise 10– Creating an AWS RDS Instance in Terraform

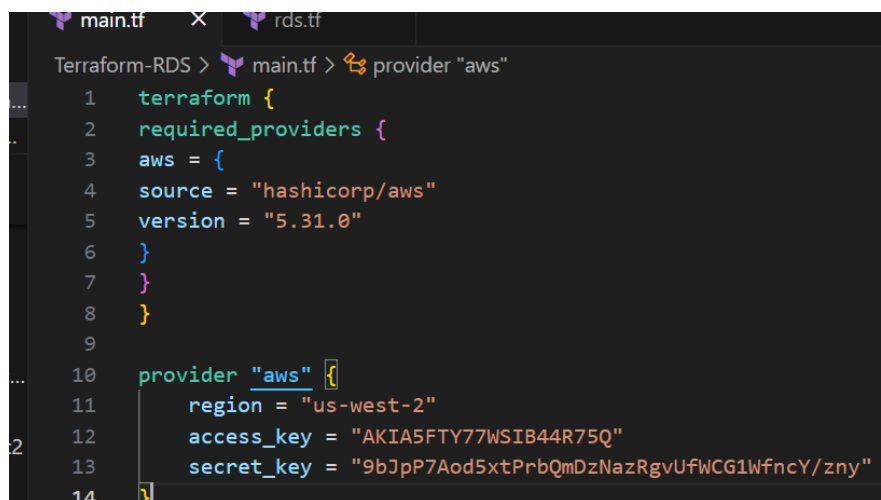
1. Create a Terraform Directory:

```
mkdir Terraform-RDS
cd Terraform-RDS
```

2. Create Terraform Configuration Files:

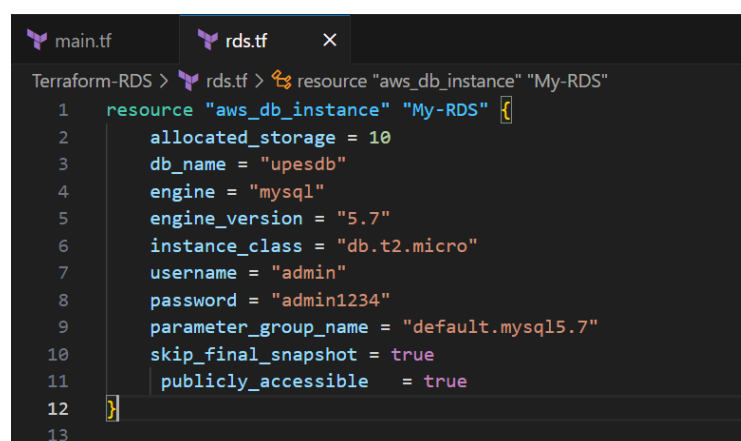
Create a file named main.tf:

main.tf

A screenshot of a code editor with two tabs: 'main.tf' and 'rds.tf'. The 'main.tf' tab is active, showing Terraform configuration for the AWS provider. The code includes a 'terraform' block with 'required_providers' and an 'aws' block with 'provider' configuration. The editor has a dark theme and line numbers on the left.

```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "us-west-2"
12   access_key = "AKIA5FTY77WSIB44R75Q"
13   secret_key = "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"
14 }
```

#rds.tf

A screenshot of a code editor with two tabs: 'main.tf' and 'rds.tf'. The 'rds.tf' tab is active, showing Terraform configuration for an AWS RDS instance. The code includes a 'resource' block for 'aws_db_instance' with various attributes like 'allocated_storage', 'db_name', 'engine', 'instance_class', 'username', 'password', 'parameter_group_name', 'skip_final_snapshot', and 'publicly_accessible'. The editor has a dark theme and line numbers on the left.

```
1 resource "aws_db_instance" "My-RDS" {
2   allocated_storage = 10
3   db_name = "upesdb"
4   engine = "mysql"
5   engine_version = "5.7"
6   instance_class = "db.t2.micro"
7   username = "admin"
8   password = "admin1234"
9   parameter_group_name = "default.mysql5.7"
10  skip_final_snapshot = true
11  publicly_accessible = true
12 }
13
```

3. Initialize and Apply:

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

```
terraform init
terraform apply
```

```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-RDS> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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
you run "terraform init" in the future.

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.
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-RDS> terraform validate
Success! The configuration is valid.
```

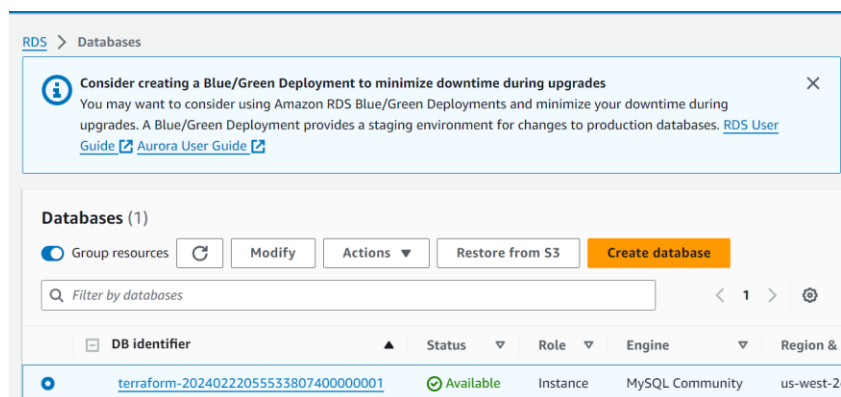
```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-RDS> terraform apply
aws_db_instance.My-RDS: Refreshing state... [id=db-A0J3U3CA73SP0T4E0ZBC5V1IKM]

Terraform used the selected providers to generate the following execution plan. Resource actions
+ create

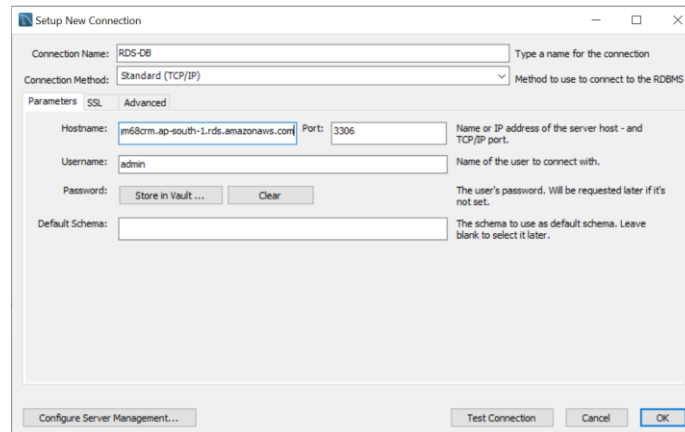
Terraform will perform the following actions:

# 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)
```

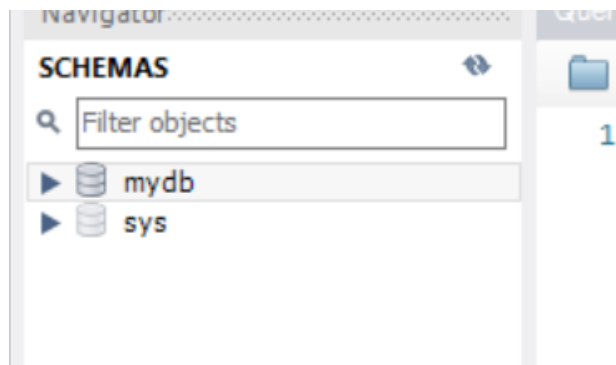
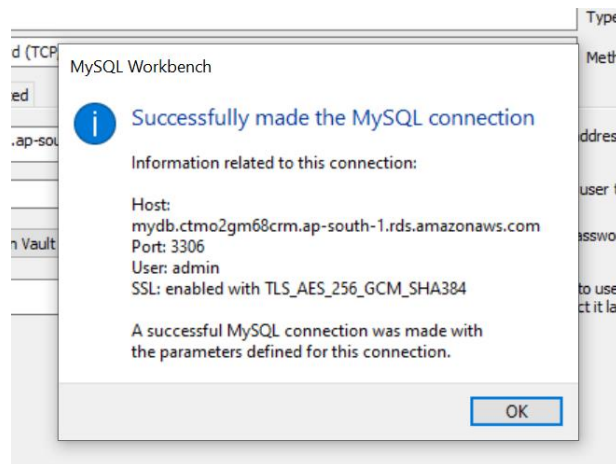
4. Verify RDS Instance in AWS Console:



5. Connect to MySQL Workbench



The 'Setup New Connection' dialog box in MySQL Workbench. It has a title bar with standard window controls. The 'Connection Name' field is set to 'RDS-DB'. The 'Connection Method' is set to 'Standard (TCP/IP)'. The 'Parameters' tab is selected, showing fields for 'Hostname' (mydb.ctmo2gm68crm.ap-south-1.rds.amazonaws.com), 'Port' (3306), 'Username' (admin), and 'Password' (with 'Store in Vault' and 'Clear' buttons). The 'Default Schema' field is empty. At the bottom are buttons for 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'.



6. Clean Up:

```
terraform destroy
```

Confirm the destruction by typing yes.

```
Apply Complete! Resources: 1 added, 0 changed, 0 destroyed.  
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-RDS> terraform destroy  
aws_db_instance.My-RDS: Refreshing state... [id=db-TZBCYTB2HG2ZT2PWKEPK3PCNQ]  
  
Terraform used the selected providers to generate the following execution plan. Resource actions a  
- destroy  
  
Terraform will perform the following actions:  
# aws_db_instance.My-RDS will be destroyed
```