

School of Computer Science

University of Petroleum and Energy Studies



System Provisioning & Configuration Management

Lab File (6th Sem)

Submitted By:-

Akshat Pandey

500101788

R2142220306

DevOps B1

Submitted To:-

Dr Hitesh Kumar Sharma

EXPERIMENT 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
```

```
C:\Users\aksha\Documents\SPCM_LAB>mkdir terraform-rds
```

```
C:\Users\aksha\Documents\SPCM_LAB>cd terraform-rds
```

```
C:\Users\aksha\Documents\SPCM_LAB\terraform-rds>|
```

2. Create Terraform Configuration Files:

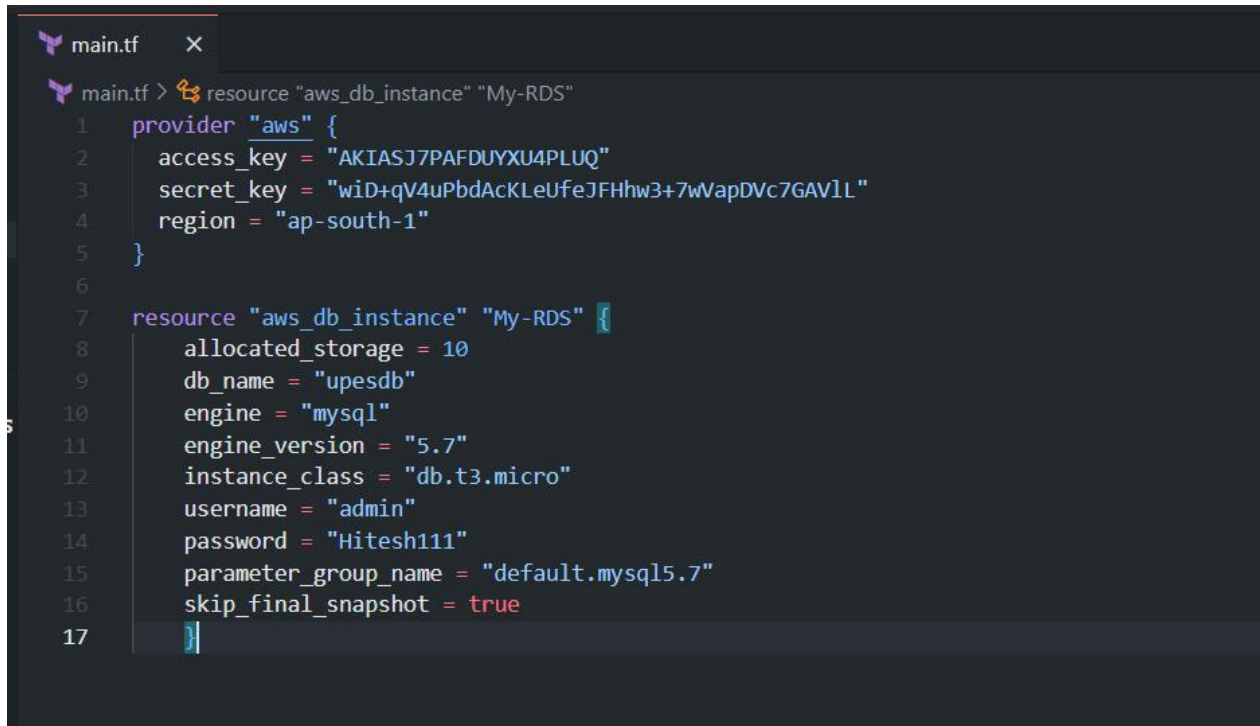
Create a file named main.tf:

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
}
```

A screenshot of a terminal window with a dark background. The window title is 'main.tf'. The prompt is 'main.tf >'. The user has entered the command 'resource "aws_db_instance" "My-RDS"'. The terminal shows the following configuration for the 'aws_db_instance' resource:

```
1 provider "aws" {
2   access_key = "AKIASJ7PAFDUYXU4PLUQ"
3   secret_key = "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"
4   region = "ap-south-1"
5 }
6
7 resource "aws_db_instance" "My-RDS" {
8   allocated_storage = 10
9   db_name = "upesdb"
10  engine = "mysql"
11  engine_version = "5.7"
12  instance_class = "db.t3.micro"
13  username = "admin"
14  password = "Hitesh111"
15  parameter_group_name = "default.mysql5.7"
16  skip_final_snapshot = true
17 }
```

- 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

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

```
C:\Users\aksha\Documents\SPCM_LAB\terraform-rds>terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.96.0
```

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.

```
C:\Users\aksha\Documents\SPCM_LAB\terraform-rds>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
  + database_insights_mode  = (known after apply)
  + db_name                 = "upesdb"
  + db_subnet_group_name    = (known after apply)
  + dedicated_log_volume   = false
  + delete_automated_backups = true
  + domain_fqdn             = (known after apply)
  + endpoint               = (known after apply)
  + engine                  = "mysql"
  + engine_lifecycle_support = (known after apply)
  + engine_version          = "5.7"
  + engine_version_actual    = (known after apply)
  + hosted_zone_id          = (known after apply)
  + id                     = (known after apply)
  + identifier              = (known after apply)
  + identifier_prefix       = (known after apply)
  + instance_class          = "db.t3.micro"
  + iops                    = (known after apply)
  + kms_key_id              = (known after apply)
  + latest_restorable_time  = (known after apply)
  + license_model            = (known after apply)
  + listener_endpoint       = (known after apply)
  + maintenance_window      = (known after apply)
  + master_user_secret      = (known after apply)

  + status                = (known after apply)
  + storage_throughput     = (known after apply)
  + storage_type           = (known after apply)
  + tags_all               = (known after apply)
  + timezone               = (known after apply)
  + username               = "admin"
  + vpc_security_group_ids = (known after apply)
}
```

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

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

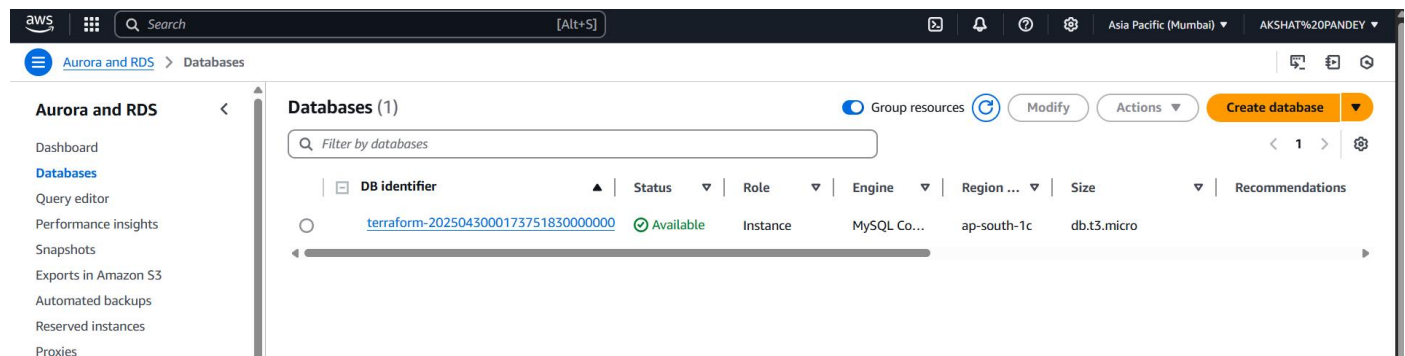
Enter a value: yes

```
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: Creation complete after 3m34s [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI]
```

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

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

```
C:\Users\aksha\Documents\SPCM_LAB\terraform-rds>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
  + database_insights_mode  = (known after apply)
  + db_name                 = "upesdb"
  + db_subnet_group_name   = (known after apply)
  + dedicated_log_volume   = false
  + delete_automated_backups = true
  + domain_fqdn             = (known after apply)
  + endpoint               = (known after apply)
  + engine                  = "mysql"
  + engine_lifecycle_support = (known after apply)
  + engine_version          = "5.7"
  + engine_version_actual   = (known after apply)
  + hosted_zone_id         = (known after apply)
  + id                     = (known after apply)
  + identifier              = (known after apply)
  + identifier_prefix       = (known after apply)
  + instance_class          = "db.t3.micro"
  + iops                    = (known after apply)
  + kms_key_id              = (known after apply)
  + latest_restorable_time  = (known after apply)
  + license_model            = (known after apply)
  + listener_endpoint       = (known after apply)
  + maintenance_window      = (known after apply)
  + master_user_secret       = (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: Creation complete after 3m34s [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI]
```

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

6. Clean Up:

After testing, you can clean up the RDS instance:

terraform destroy

```
C:\Users\aksha\Documents\SPCM_LAB\terraform-rds>terraform destroy
aws_db_instance.My-RDS: Refreshing state... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI]
```

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                               = "terraform-20250430001737518300000001.cjaaomym8dm7.ap-south-1.rds.amazonaws.com" -> null
  - allocated_storage                     = 10 -> null
  - apply_immediately                     = false -> null
  - arn                                   = "arn:aws:rds:ap-south-1:158878148841:db:terraform-20250430001737518300000001" -> null
  - auto_minor_version_upgrade            = true -> null
  - availability_zone                     = "ap-south-1c" -> null
  - backup_retention_period                = 0 -> null
  - backup_target                          = "region" -> null
  - backup_window                          = "23:53-00:23" -> null
  - ca_cert_identifier                     = "rds-ca-rsa2048-g1" -> null
  - copy_tags_to_snapshot                  = false -> null
  - customer_owned_ip_enabled              = false -> null
  - database_insights_mode                 = "standard" -> null
  - db_name                                = "upesdb" -> null
  - db_subnet_group_name                   = "default" -> null
  - dedicated_log_volume                   = false -> null
  - delete_automated_backups               = true -> null
  - deletion_protection                    = false -> null
  - domain_dns_ips                         = [] -> null
  - enabled_cloudwatch_logs_exports        = [] -> null
  - endpoint                              = "terraform-20250430001737518300000001.cjaaomym8dm7.ap-south-1.rds.amazonaws.com:3306" -> null
  - engine                                 = "mysql" -> null
  - engine_lifecycle_support               = "open-source-rds-extended-support" -> null
  - engine_version                         = "5.7" -> null
  - engine_version_actual                   = "5.7.44-rds.20250213" -> null
  - hosted_zone_id                         = "Z2VFMSZA74J7XZ" -> null
  - iam_database_authentication_enabled    = false -> null
  - id                                     = "db-2IWQSB4FZDNWVA3XLCFYDM4PHI" -> null
  - identifier                             = "terraform-20250430001737518300000001" -> null
  - identifier_prefix                       = "terraform-" -> null
  - instance_class                         = "db.t3.micro" -> null
  - iops                                   = 0 -> null
```

```

- vpc_security_group_ids          = [
  - "sg-0fe7ead32716b4afa",
  ] -> null
  # (15 unchanged attributes hidden)
}

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-2IWQSB4FZDNWVA3XLCFYDM4PHI]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 1m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 1m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 1m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 1m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 1m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 1m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 2m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 2m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 2m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 2m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 2m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 2m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 3m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 3m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 3m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 3m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 3m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 3m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-2IWQSB4FZDNWVA3XLCFYDM4PHI, 4m0s elapsed]
aws_db_instance.My-RDS: Destruction complete after 4m3s

Destroy complete! Resources: 1 destroyed.

C:\Users\aksha\Documents\SPCM_LAB\terraform-rds>|

```

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