Software Provisioning and Configuration Management

LAB FILE SUBMITTED BY:

NAME: Palak Gupta
SAP ID: 500109569
Roll no: R2142221387
BATCH: 1 Devops

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
```

2. Create Terraform Configuration Files:

Create a file named main.tf:

 # main.tf

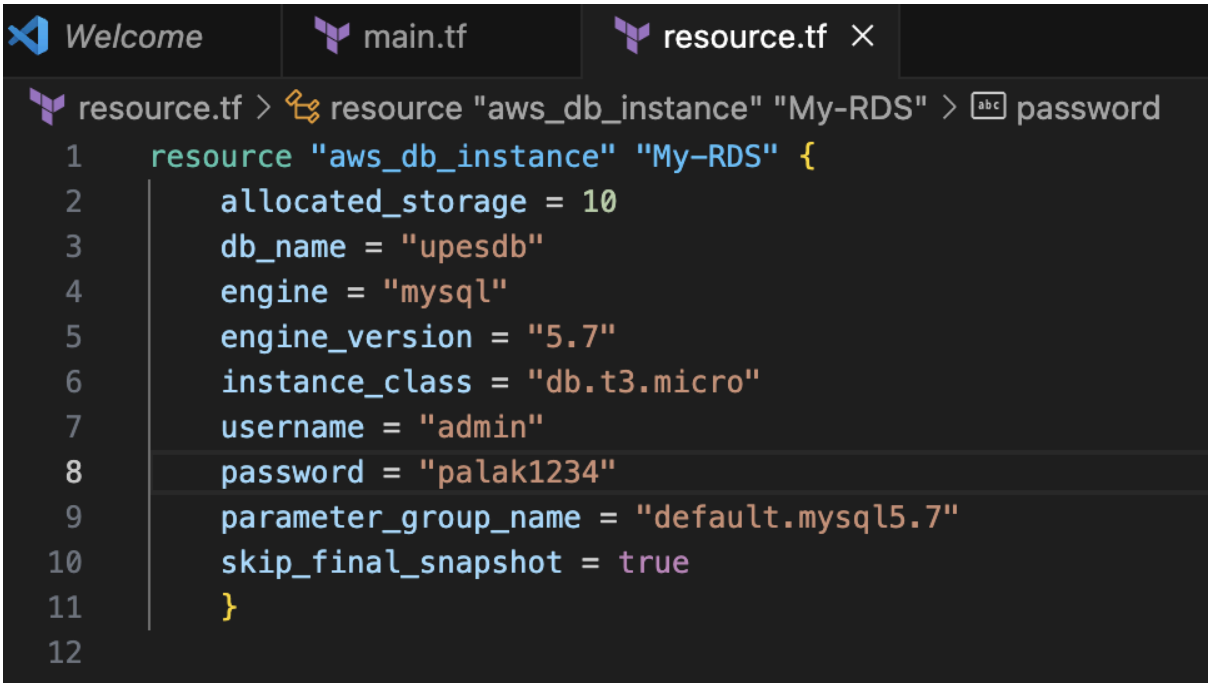
```
provider "aws" {
  region = "us-east-1"
}

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.

main.tf > provider "aws"

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.95.0"
    }
  }
}

provider "aws" {
  access_key = "AKIAQMEY6IA6YOZ2KTVX"
  secret_key = "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"
  region = "us-east-1"
}
```



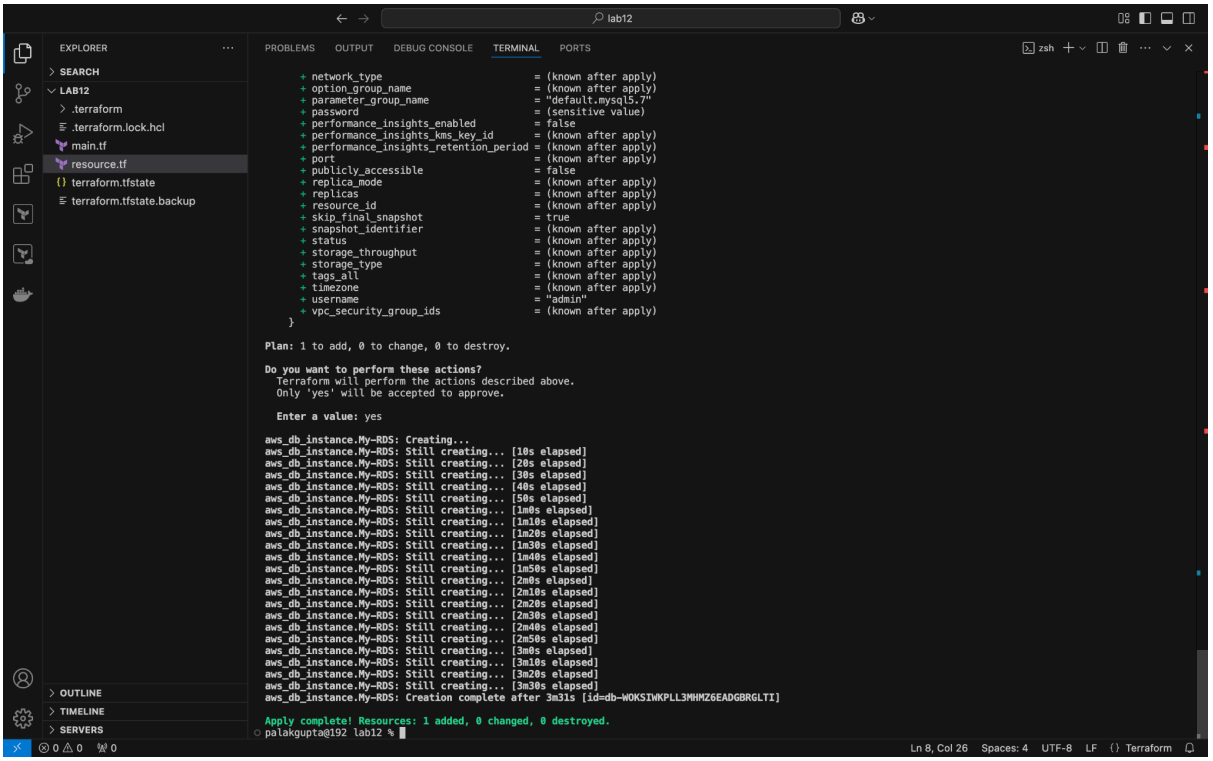
3. Initialize and Apply:

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

```
terraform init
terraform apply
```

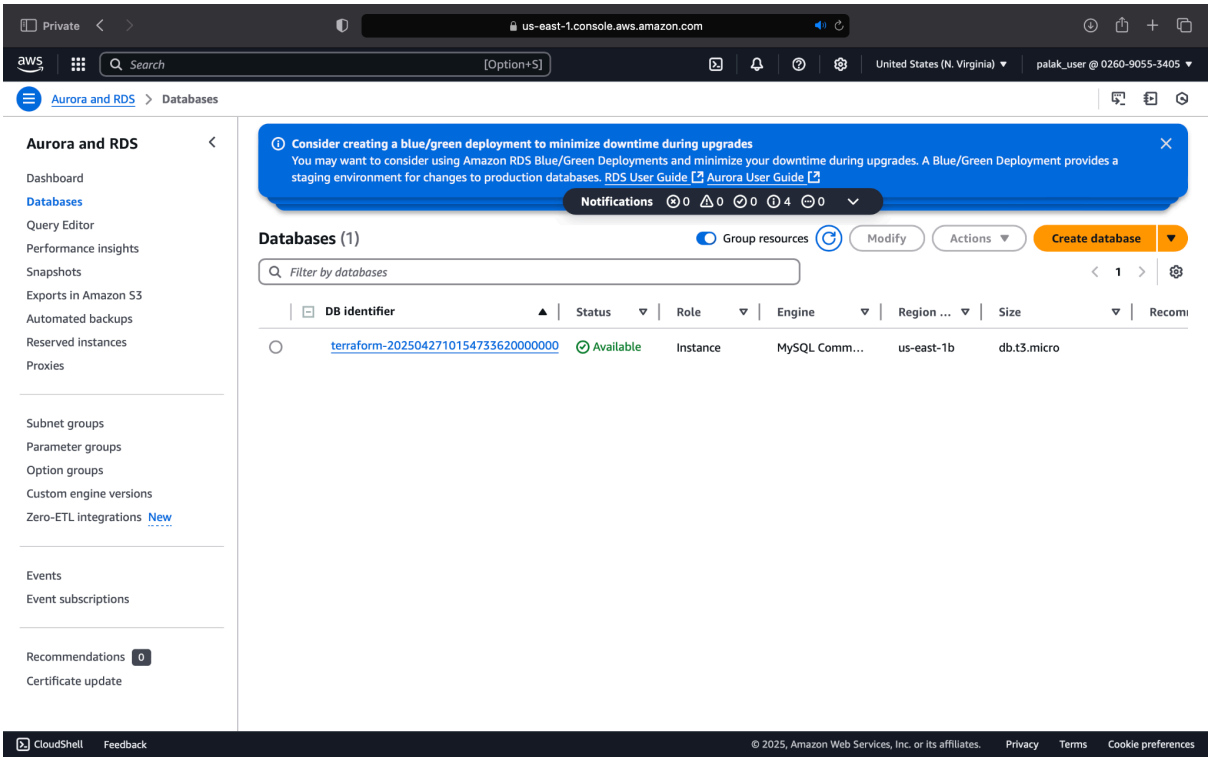
- Terraform will prompt you to confirm the creation of the RDS instance. Type yes and press Enter.

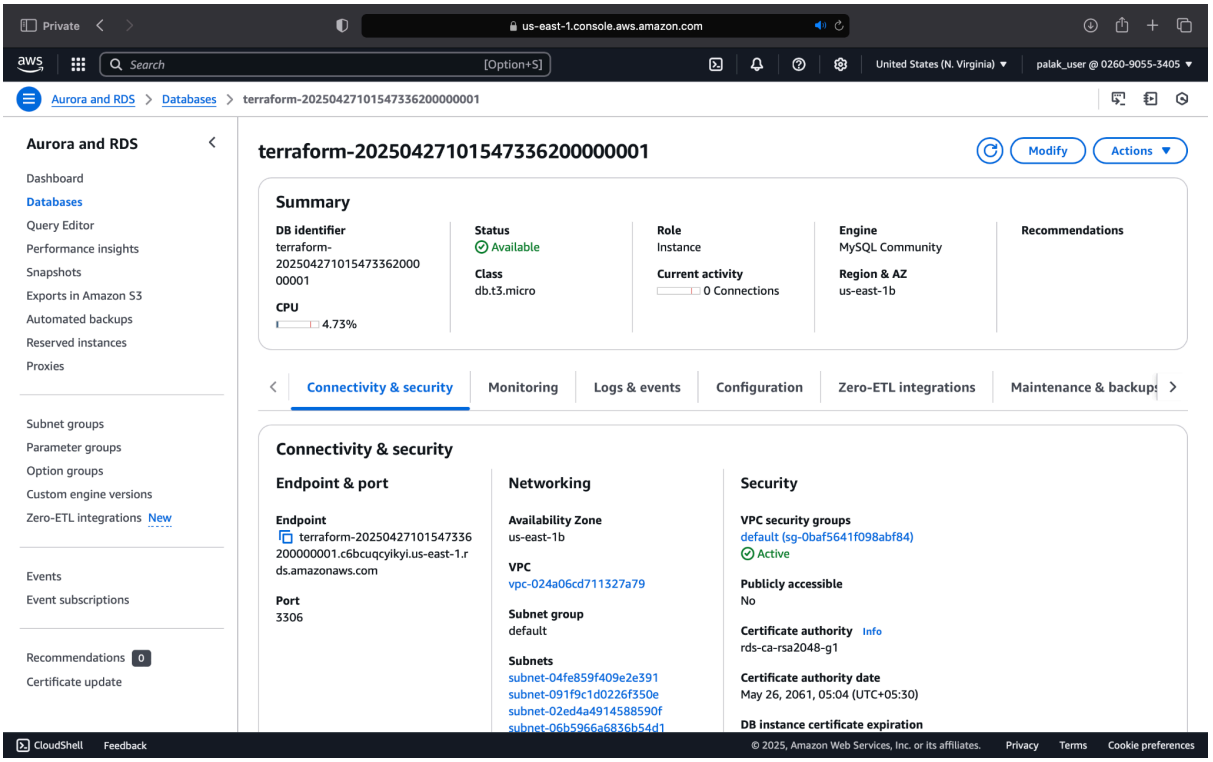
```
palakgupta@192 lab12 % terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.95.0"...
- Installing hashicorp/aws v5.95.0...
- Installed hashicorp/aws v5.95.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl 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.
palakgupta@192 lab12 %
```



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.


Private
us-east-1.console.aws.amazon.com
Search
[Option+S]
United States (N. Virginia)
palak_user @ 0260-9055-3405
Aurora and RDS > Databases
Aurora and RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update
Consider creating a blue/green deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide
Notifications 0 0 0 4 0
Databases (1)
Group resources
Modify
Actions
Create database
Filter by databases
DB identifier
Status
Role
Engine
Region ...
Size
terraform-20250427101547336200000000
Available
Instance
MySQL Comm...
us-east-1b
db.t3.micro
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Private
us-east-1.console.aws.amazon.com
Search
[Option+S]
United States (N. Virginia)
palak_user @ 0260-9055-3405
Aurora and RDS > Databases > terraform-20250427101547336200000001
Aurora and RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update
terraform-20250427101547336200000001
Modify
Actions
Summary
DB identifier
terraform-20250427101547336200000001
Status
Available
Role
Instance
Engine
MySQL Community
Recommendations
Class
db.t3.micro
Current activity
0 Connections
Region & AZ
us-east-1b
CPU
4.73%
Connectivity & security
Monitoring
Logs & events
Configuration
Zero-ETL integrations
Maintenance & backups
Connectivity & security
Endpoint & port
Endpoint
terraform-20250427101547336200000001.c6bcuqcyikyi.us-east-1.rds.amazonaws.com
Port
3306
Networking
Availability Zone
us-east-1b
VPC
vpc-024a06cd711327a79
Subnet group
default
Subnets
subnet-04fe859f409e2e391
subnet-091f9c1d0226f350e
subnet-02ed4a4914588590f
subnet-06b5966a6836b54d1
Security
VPC security groups
default (sg-0baf5641f098abf84)
Active
Publicly accessible
No
Certificate authority Info
rds-ca-rsa2048-g1
Certificate authority date
May 26, 2061, 05:04 (UTC+05:30)
DB instance certificate expiration
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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

resource.tf > resource "aws_db_instance" "My-RDS"

```
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.t3.small"  //changed to t3.small
7       username = "admin"
8       password = "palak1234"
9       parameter_group_name = "default.mysql5.7"
10      skip_final_snapshot = true
11      }
12
```

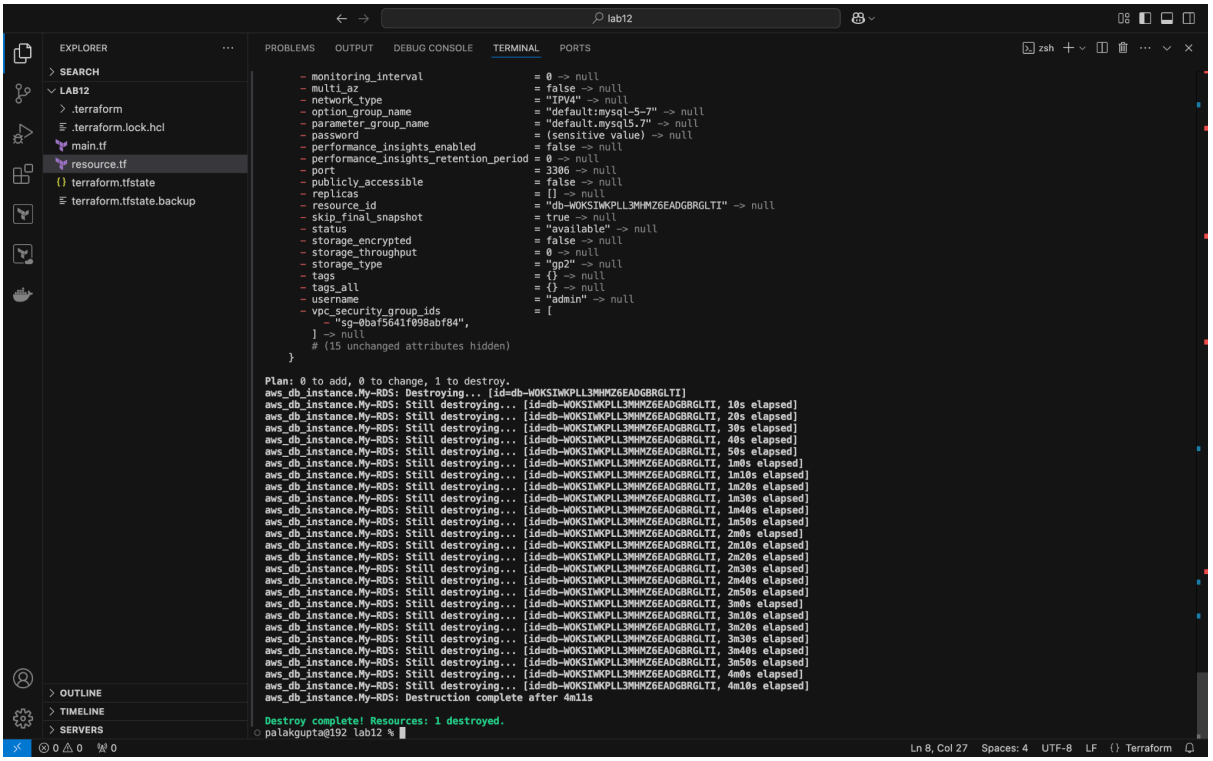
```
palakgupta@192 lab12 % terraform apply -auto-approve
aws_db_instance.My-RDS: Refreshing state... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_db_instance.My-RDS will be updated in-place
  ~ resource "aws_db_instance" "My-RDS" {
        id                                    = "db-WOKSIWKPLL3MHMZ6EADGBRGLTI"
      ~ instance_class                        = "db.t3.micro" -> "db.t3.small"
        tags                                  = {}
        # (70 unchanged attributes hidden)
    }

Plan: 0 to add, 1 to change, 0 to destroy.
aws_db_instance.My-RDS: Modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 10s elapsed]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 20s elapsed]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 30s elapsed]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 40s elapsed]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 50s elapsed]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 1m0s elapsed]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 1m10s elapsed]
aws_db_instance.My-RDS: Still modifying... [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI, 1m20s elapsed]
aws_db_instance.My-RDS: Modifications complete after 1m24s [id=db-WOKSIWKPLL3MHMZ6EADGBRGLTI]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
palakgupta@192 lab12 %
```

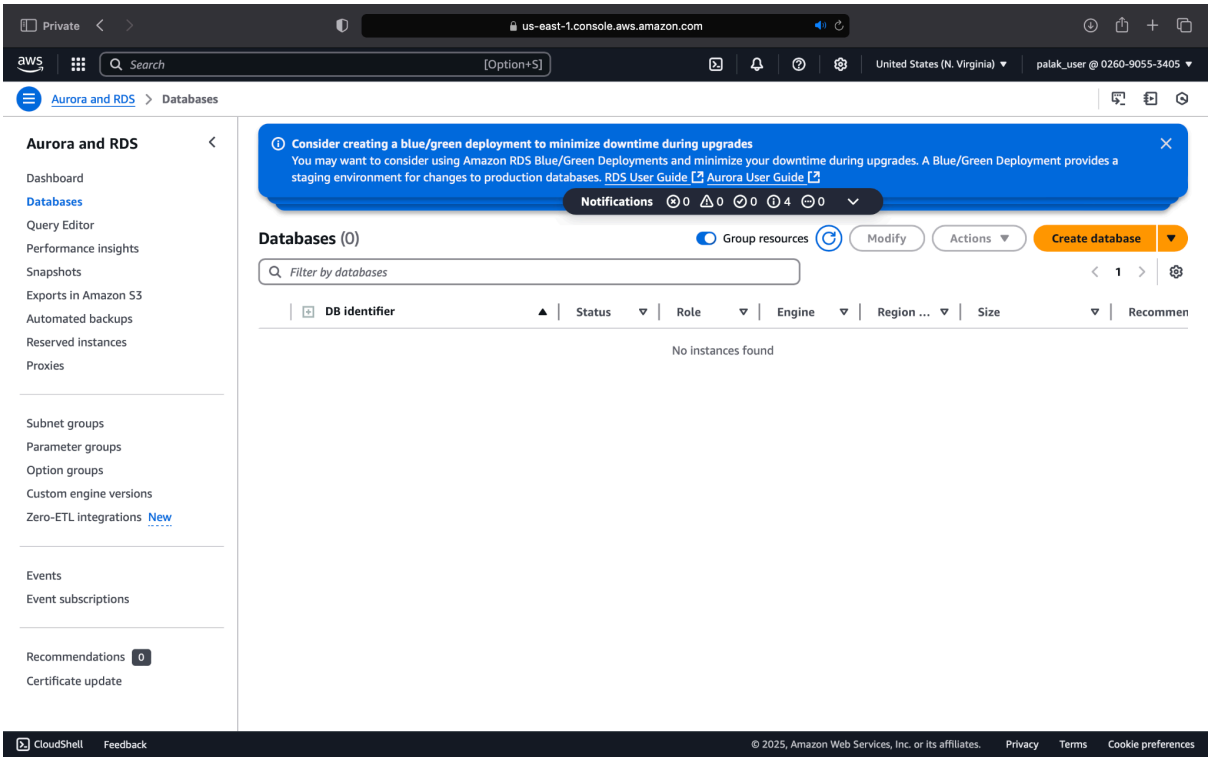
6. Clean Up:

After testing, you can clean up the RDS instance:

```
terraform destroy
```

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