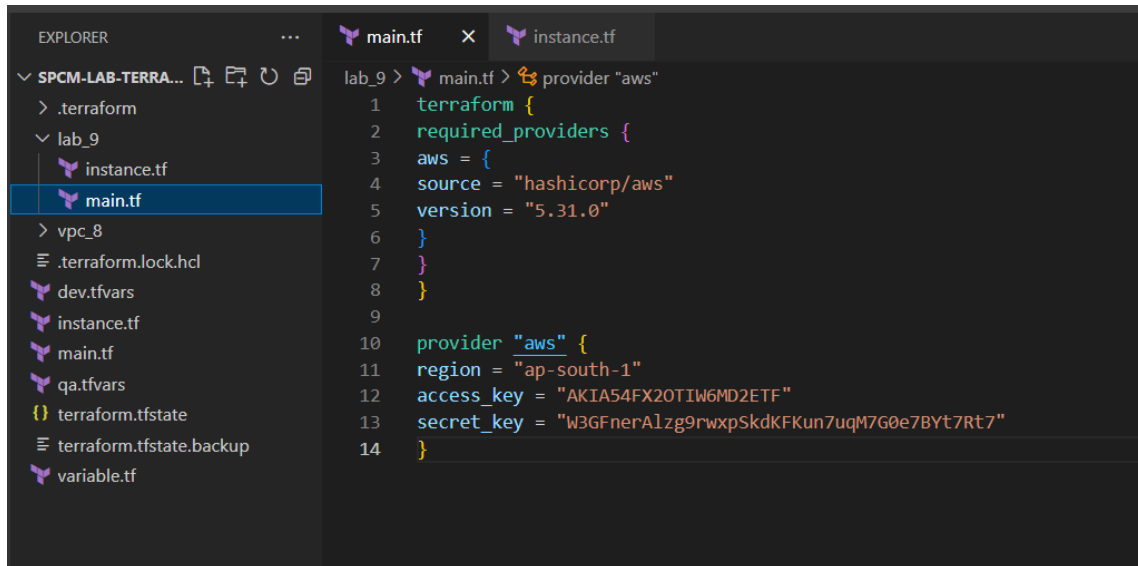


# LAB EXERCISE 9

## Aim: Creating Multiple EC2 Instances with for each in Terraform

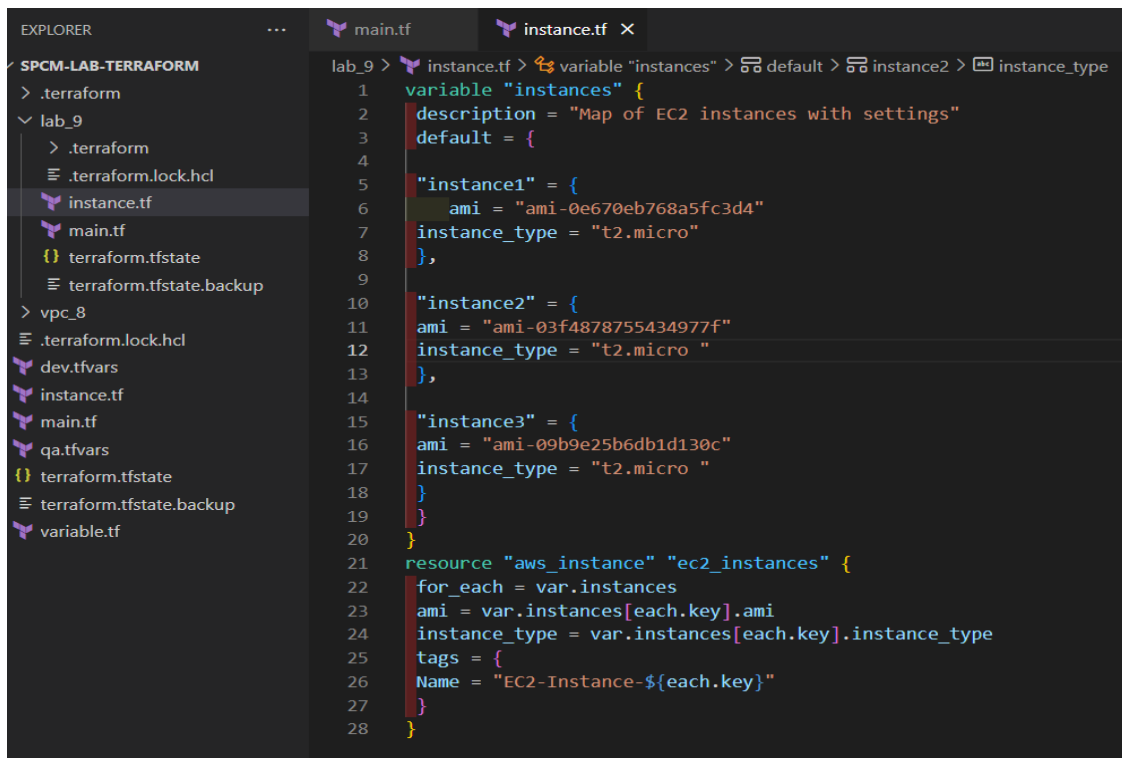
Step 1: Create a main.tf file



The screenshot shows the VS Code interface with the Explorer on the left and the main.tf file open in the editor. The file content is as follows:

```
lab_9 > main.tf > provider "aws"
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 = "ap-south-1"
12   access_key = "AKIA54FX20TIW6MD2ETF"
13   secret_key = "W3GFnerAlzg9rwxpSkdkFKun7uqM7G0e7BYt7Rt7"
14 }
```

Step 2: Create a instance.tf file



The screenshot shows the VS Code interface with the Explorer on the left and the instance.tf file open in the editor. The file content is as follows:

```
lab_9 > instance.tf > variable "instances" > default > instance2 > instance_type
1 variable "instances" {
2   description = "Map of EC2 instances with settings"
3   default = {
4
5     "instance1" = {
6       ami = "ami-0e670eb768a5fc3d4"
7       instance_type = "t2.micro"
8     },
9
10    "instance2" = {
11      ami = "ami-03f4878755434977f"
12      instance_type = "t2.micro "
13    },
14
15    "instance3" = {
16      ami = "ami-09b9e25b6db1d130c"
17      instance_type = "t2.micro "
18    }
19  }
20 }
21 resource "aws_instance" "ec2_instances" {
22   for_each = var.instances
23   ami = var.instances[each.key].ami
24   instance_type = var.instances[each.key].instance_type
25   tags = {
26     Name = "EC2-Instance-${each.key}"
27   }
28 }
```

Step 3: Now run terraform init command to Initialize.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_9>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
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.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_9>
```

Step 4: Now run the terraform validate command to check if any error is present or not.

```
F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_9>terraform validate
Success! The configuration is valid.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_9>
```

Step 5: Now run terraform plan command.

```
F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_9>terraform plan
```

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_instance.ec2_instances["instance1"] will be created
+ resource "aws_instance" "ec2_instances" {
  + ami                        = "ami-0e670eb768a5fc3d4"
  + arn                      = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone         = (known after apply)
  + cpu_core_count            = (known after apply)
  + cpu_threads_per_core      = (known after apply)
  + disable_api_stop          = (known after apply)
  + disable_api_termination   = (known after apply)
  + ebs_optimized              = (known after apply)
  + get_password_data         = false
  + host_id                   = (known after apply)
  + host_resource_group_arn    = (known after apply)
  + iam_instance_profile       = (known after apply)
  + id                        = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle         = (known after apply)
  + instance_state             = (known after apply)
  + instance_type              = "t2.micro"
  + ipv6_address_count         = (known after apply)
  + ipv6_addresses             = (known after apply)
  + key_name                   = (known after apply)
  + monitoring                  = (known after apply)
  + outpost_arn                = (known after apply)
  + password_data              = (known after apply)
  + placement_group            = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns                = (known after apply)
  + private_ip                 = (known after apply)
  + public_dns                 = (known after apply)
  + public_ip                  = (known after apply)
  + secondary_private_ips      = (known after apply)
  + security_groups             = (known after apply)
  + source_dest_check           = true
  + spot_instance_request_id   = (known after apply)
  + subnet_id                  = (known after apply)
  + tags                        = {
```

```
cmd C:\Windows\System32\cmd.exe
```

```
  + subnet_id                  = (known after apply)
  + tags                        = {
    + "Name" = "EC2-Instance-instance1"
  }
  + tags_all                    = {
    + "Name" = "EC2-Instance-instance1"
  }
  + tenancy                     = (known after apply)
  + user_data                   = (known after apply)
  + user_data_base64            = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids      = (known after apply)
}

# aws_instance.ec2_instances["instance2"] will be created
+ resource "aws_instance" "ec2_instances" {
  + ami                        = "ami-03f4878755434977f"
  + arn                      = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone         = (known after apply)
  + cpu_core_count            = (known after apply)
  + cpu_threads_per_core      = (known after apply)
  + disable_api_stop          = (known after apply)
  + disable_api_termination   = (known after apply)
  + ebs_optimized              = (known after apply)
  + get_password_data         = false
  + host_id                   = (known after apply)
  + host_resource_group_arn    = (known after apply)
  + iam_instance_profile       = (known after apply)
  + id                        = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle         = (known after apply)
  + instance_state             = (known after apply)
  + instance_type              = "t2.micro "
  + ipv6_address_count         = (known after apply)
  + ipv6_addresses             = (known after apply)
  + key_name                   = (known after apply)
  + monitoring                  = (known after apply)
  + outpost_arn                = (known after apply)
  + password_data              = (known after apply)
  + placement_group            = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns                = (known after apply)
  + private_ip                 = (known after apply)
  + public_dns                 = (known after apply)
  + public_ip                  = (known after apply)
  + secondary_private_ips      = (known after apply)
  + security_groups             = (known after apply)
  + source_dest_check           = true
```

```

C:\Windows\System32\cmd.exe

+ tags = {
  + "Name" = "EC2-Instance-instance2"
}
+ tags_all = {
  + "Name" = "EC2-Instance-instance2"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

# aws_instance.ec2_instances["instance3"] will be created
+ resource "aws_instance" "ec2_instances" {
  + ami = "ami-09b9e25b6db1d130c"
  + arn = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (known after apply)
  + cpu_core_count = (known after apply)
  + cpu_threads_per_core = (known after apply)
  + disable_api_stop = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized = (known after apply)
  + get_password_data = false
  + host_id = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile = (known after apply)
  + id = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle = (known after apply)
  + instance_state = (known after apply)
  + instance_type = "t2.micro"
  + ipv6_address_count = (known after apply)
  + ipv6_addresses = (known after apply)
  + key_name = (known after apply)
  + monitoring = (known after apply)
  + outpost_arn = (known after apply)
  + password_data = (known after apply)
  + placement_group = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns = (known after apply)
  + private_ip = (known after apply)
  + public_dns = (known after apply)
  + public_ip = (known after apply)
  + secondary_private_ips = (known after apply)
  + security_groups = (known after apply)
  + source_dest_check = true
  + spot_instance_request_id = (known after apply)

```

```

+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

```

Plan: 3 to add, 0 to change, 0 to destroy.

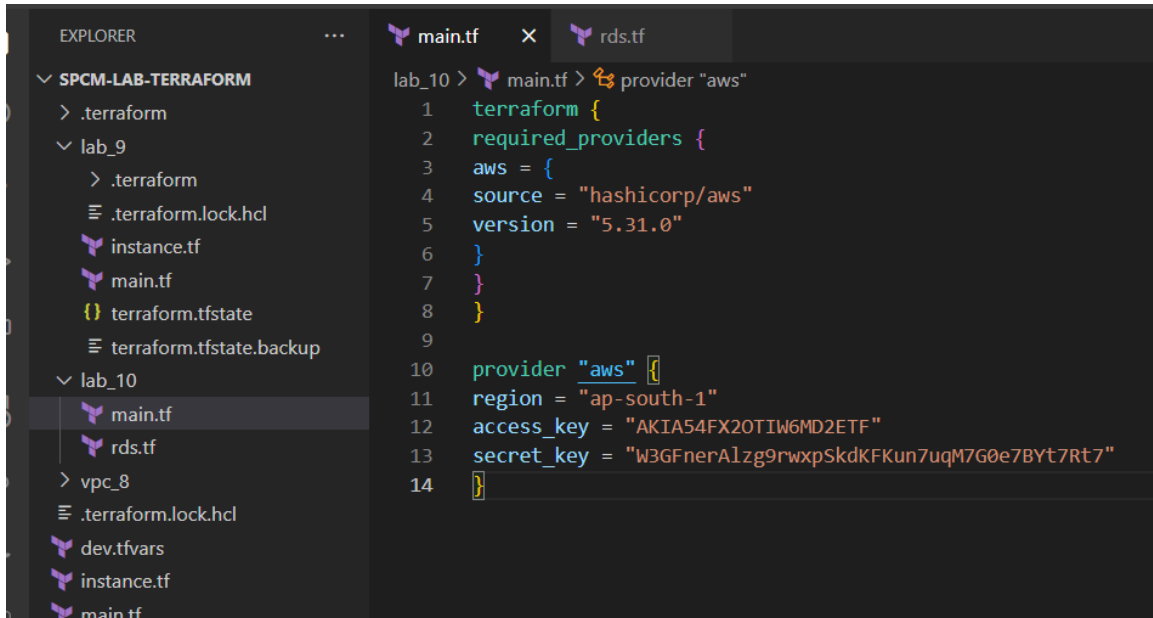
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

F:\sem 6\SPCM LAB\spcm-lab-terraform\lab\_9>

# LAB EXERCISE 10

## Aim: Creating an AWS RDS Instance in Terraform

Step 1: Create a main.tf file

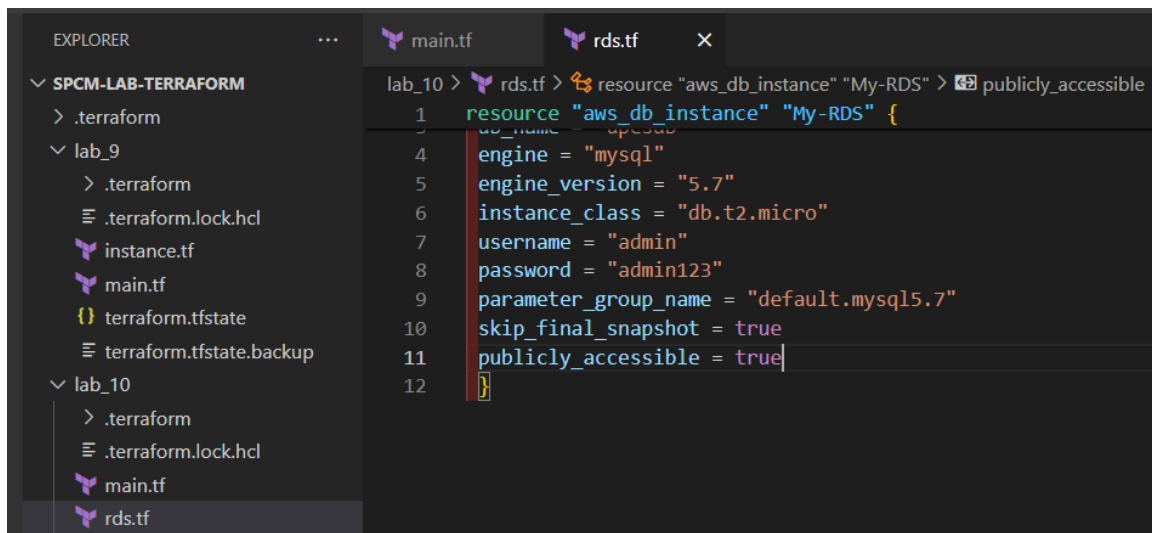


The screenshot shows a code editor with two tabs: 'main.tf' and 'rds.tf'. The 'main.tf' tab is active, displaying the following Terraform configuration:

```
lab_10 > main.tf > provider "aws"
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 = "ap-south-1"
12   access_key = "AKIA54FX20TIW6MD2ETF"
13   secret_key = "W3GFnerAlzg9rwxpSkdkFKun7uqM7G0e7BYt7Rt7"
14 }
```

The Explorer panel on the left shows the project structure for 'SPCM-LAB-TERRAFORM', including directories for 'lab\_9' and 'lab\_10', and files like '.terraform.lock.hcl', 'instance.tf', 'main.tf', 'terraform.tfstate', and 'terraform.tfstate.backup'.

Step 2: Create a rds.tf file



The screenshot shows the same code editor with the 'rds.tf' tab now active. The 'main.tf' tab is still open in the background. The 'rds.tf' file contains the following Terraform configuration:

```
lab_10 > rds.tf > resource "aws_db_instance" "My-RDS" > publicly_accessible
1 resource "aws_db_instance" "My-RDS" {
2   db_name = "vpcdb"
3   engine = "mysql"
4   engine_version = "5.7"
5   instance_class = "db.t2.micro"
6   username = "admin"
7   password = "admin123"
8   parameter_group_name = "default.mysql5.7"
9   skip_final_snapshot = true
10  publicly_accessible = true
11 }
12
```

The Explorer panel on the left shows the project structure, with 'lab\_10' expanded to show 'main.tf' and 'rds.tf'.

Step 3: Now run terraform init command to Initialize.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_10>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
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.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_10>

```

Step 4: Now run the terraform validate command to check if any error is present or not.

```

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_10>terraform validate
Success! The configuration is valid.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_10>

```

Step 5: Now run terraform plan command.

```

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_10>terraform plan

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                               = "upesdb"
  + db_subnet_group_name                  = (known after apply)
  + delete_automated_backups              = true
  + endpoint                             = (known after apply)
  + engine                               = "mysql"
  + 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.t2.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)
  + master_user_secret_kms_key_id          = (known after apply)
  + monitoring_interval                    = 0
  + monitoring_role_arn                   = (known after apply)
  + multi_az                              = (known after apply)
  + nchar_character_set_name              = (known after apply)
  + network_type                          = (known after apply)
  + option_group_name                     = (known after apply)
  + parameter_group_name                  = "default.mysql5.7"
  + password                              = (sensitive value)

```

```

+ performance_insights_enabled = false
+ performance_insights_kms_key_id = (known after apply)
+ performance_insights_retention_period = (known after apply)
+ port = (known after apply)
+ publicly_accessible = true
+ replica_mode = (known after apply)
+ replicas = (known after apply)
+ resource_id = (known after apply)
+ skip_final_snapshot = true
+ snapshot_identifier = (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.

Note: You didn't use the -out option to save this plan, so Terraform c
F:\sem 6\SPCM LAB\spcm-lab-terraform\lab_10>_

```

Step 6: Now run the terraform apply command to apply the rds.

```

F:\sem 6\SPCM LAB\spcm-lab-terraform\lab_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 = "upesdb"
+   db_subnet_group_name = (known after apply)
+   delete_automated_backups = true
+   endpoint = (known after apply)
+   engine = "mysql"
+   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.t2.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)
+   master_user_secret_kms_key_id = (known after apply)
+   monitoring_interval = 0
+   monitoring_role_arn = (known after apply)
+   multi_az = (known after apply)
+   nchar_character_set_name = (known after apply)
+   network_type = (known after apply)

```

```

C:\Windows\System32\cmd.exe

+ network_type                = (known after apply)
+ option_group_name            = (known after apply)
+ parameter_group_name         = "default.mysql5.7"
+ password                     = (sensitive value)
+ performance_insights_enabled = false
+ performance_insights_kms_key_id = (known after apply)
+ performance_insights_retention_period = (known after apply)
+ port                         = (known after apply)
+ publicly_accessible           = true
+ replica_mode                  = (known after apply)
+ replicas                      = (known after apply)
+ resource_id                  = (known after apply)
+ skip_final_snapshot           = true
+ snapshot_identifier           = (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

```

```

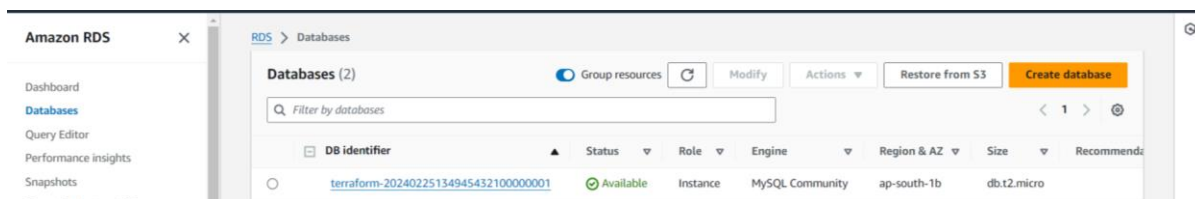
aws_db_instance.My-RDS: Creating...
aws_db_instance.My-RDS: Still creating... [10s elapsed]
aws_db_instance.My-RDS: Still creating... [20s elapsed]
aws_db_instance.My-RDS: Still creating... [30s elapsed]
aws_db_instance.My-RDS: Still creating... [40s elapsed]
aws_db_instance.My-RDS: Still creating... [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... [1m51s elapsed]
aws_db_instance.My-RDS: Still creating... [2m1s elapsed]
aws_db_instance.My-RDS: Still creating... [2m11s elapsed]
aws_db_instance.My-RDS: Still creating... [2m21s elapsed]
aws_db_instance.My-RDS: Still creating... [2m31s elapsed]
aws_db_instance.My-RDS: Still creating... [2m41s elapsed]
aws_db_instance.My-RDS: Still creating... [2m51s elapsed]
aws_db_instance.My-RDS: Still creating... [3m1s elapsed]
aws_db_instance.My-RDS: Still creating... [3m11s elapsed]
aws_db_instance.My-RDS: Still creating... [3m21s elapsed]
aws_db_instance.My-RDS: Still creating... [3m31s elapsed]
aws_db_instance.My-RDS: Still creating... [3m41s elapsed]
aws_db_instance.My-RDS: Still creating... [3m51s elapsed]
aws_db_instance.My-RDS: Still creating... [4m1s elapsed]
aws_db_instance.My-RDS: Still creating... [4m11s elapsed]
aws_db_instance.My-RDS: Still creating... [4m21s elapsed]
aws_db_instance.My-RDS: Creation complete after 4m29s [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI]

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

F:\sem 6\SPCM_LAB\spcm-lab-terraform\lab_10>

```

## Step 7: Verify the RDS Instance in AWS Console.





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
Events
Event subscriptions

### Connectivity & security

#### Endpoint & port

Endpoint  
terraform-20240225134945432100000001.c94q48wo224l.ap-south-1.rds.amazonaws.com

Port  
3306

#### Networking

Availability Zone  
ap-south-1b

VPC  
vpc-063259622707b83c5

Subnet group  
default

Subnets  
subnet-0dc8309ad06d01eb0  
subnet-021abd05255720165  
subnet-06fc0f3f6f22ff3c1

Network type  
IPv4

#### Security

VPC security groups  
default (sg-08d9cef06ed45a975)  
Active

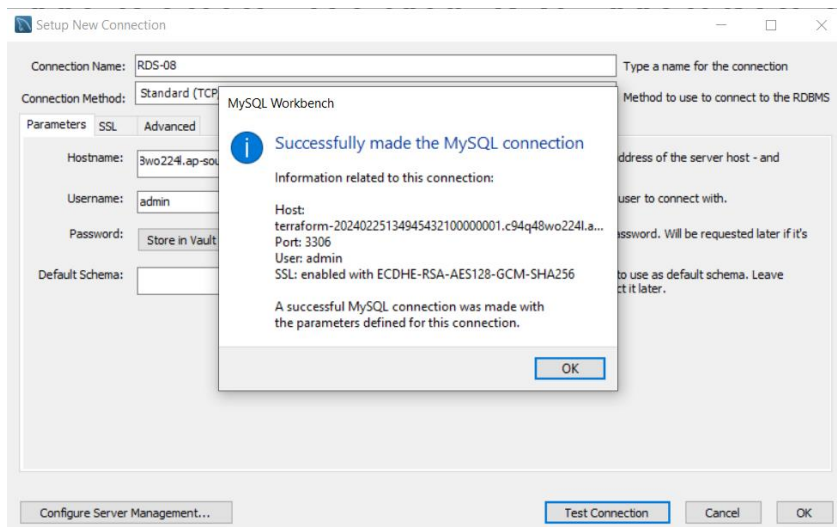
Publicly accessible  
Yes

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, 19:22 (UTC+05:30)

## Step 8: Connect To MYSQL Workbench.



## Step 9: Now Destroying the rds created.

```

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-20240225134945432100000001.c94q48wo224l.ap-south-1.rds.amazonaws.com" -> null
  allocated_storage    = 10 -> null
  apply_immediately    = false -> null
  arn                  = "arn:aws:rds:ap-south-1:953868252369:db:terraform-20240225134945432100000001" -> null
  auto_minor_version_upgrade = true -> null
  availability_zone     = "ap-south-1b" -> null
  backup_retention_period = 0 -> null
  backup_target         = "region" -> null
  backup_window         = "20:38-21:08" -> null
  ca_cert_identifier    = "rds-ca-rsa2048-g1" -> null
  copy_tags_to_snapshot = false -> null
  customer_owned_ip_enabled = false -> null
  db_name               = "upesdb" -> null
  db_subnet_group_name  = "default" -> null
  delete_automated_backups = true -> null
  deletion_protection    = false -> null
  enabled_cloudwatch_logs_exports = [] -> null
  endpoint              = "terraform-20240225134945432100000001.c94q48wo224l.ap-south-1.rds.amazonaws.com:3306" -> null
  engine                = "mysql" -> null
  engine_version        = "5.7" -> null
  engine_version_actual  = "5.7.44" -> null
  hosted_zone_id        = "ZZVFM5ZA7437X2" -> null
  iam_database_authentication_enabled = false -> null
  id                    = "db-ZBR2QLP2LUV01W2F2RX2ULG3G1" -> null
  identifier             = "terraform-20240225134945432100000001" -> null
  identifier_prefix      = "terraform-" -> null
  instance_class         = "db.t2.micro" -> null
  iops                   = 0 -> null
  license_model          = "general-public-license" -> null
  listener_endpoint      = [] -> null
  maintenance_window     = "tue:12:23-tue:12:53" -> null
  master_user_secret      = [] -> null
  max_allocated_storage  = 0 -> null
  monitoring_interval     = 0 -> null
  multi_az               = false -> null
  network_type           = "IPv4" -> null
  option_group_name       = "default:mysql-5-7" -> null
  parameter_group_name    = "default:mysql5.7" -> null
  password               = (sensitive value) -> null
  performance_insights_enabled = false -> null
}

```

```
C:\Windows\System32\cmd.exe

- performance_insights_retention_period = 0 -> null
- port = 3306 -> null
- publicly_accessible = true -> null
- replicas = [] -> null
- resource_id = "db-ZBR2QLPZLUVOIW2F2RX2ULG3GI" -> null
- skip_final_snapshot = true -> null
- status = "available" -> null
- storage_encrypted = false -> null
- storage_throughput = 0 -> null
- storage_type = "gp2" -> null
- tags = {} -> null
- tags_all = {} -> null
- username = "admin" -> null
- vpc_security_group_ids = [
  - "sg-08d9cef06ed45a975",
] -> null
}

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
```

```
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-ZBR2QLPZLUVOIW2F2RX2ULG3GI]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 1m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 1m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 1m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 1m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 1m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 1m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 2m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 2m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 2m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 2m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 2m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 2m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 3m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 3m10s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 3m20s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 3m30s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 3m40s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 3m50s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 4m0s elapsed]
aws_db_instance.My-RDS: Still destroying... [id=db-ZBR2QLPZLUVOIW2F2RX2ULG3GI, 4m10s elapsed]
aws_db_instance.My-RDS: Destruction complete after 4m14s

Destroy complete! Resources: 1 destroyed.

F:\sem 6\SPCM LAB\spcm-lab-terraform\lab_10>
```

