**Lab Steps**

**Task 1: Sign in to AWS Management Console**

1. Enter your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button
2. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

**Task 2: Setup Visual Studio Code**

1. Open the visual studio code.
2. If you have already installed and using Visual studio code, open a new window.
3. A new window will open a new file and release notes page (only if you have installed or updated Visual Studio Code recently). Close the Release notes tab.
4. Open Terminal by selecting View from the Menu bar and choose Terminal.
5. It may take up to 2 minutes to open the terminal window.
6. Once the terminal is ready, let us navigate to the Desktop.

cd Desktop

1. Create a new folder by running the below command.

mkdir task\_10097\_rds

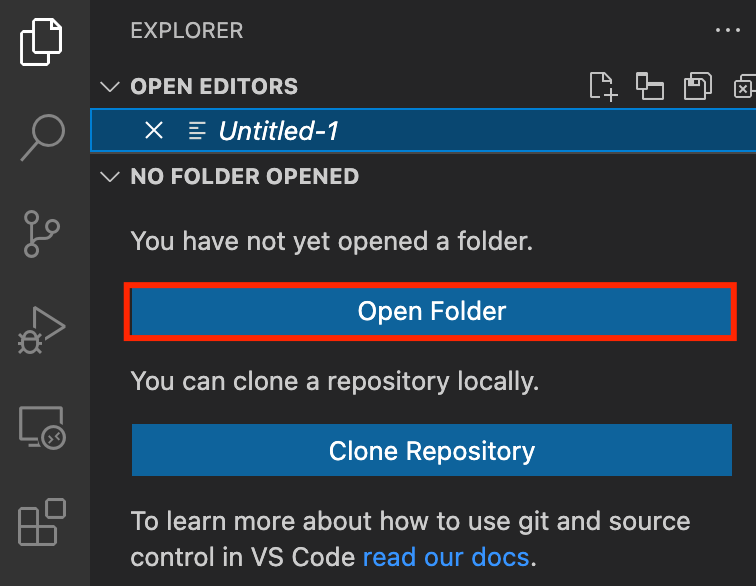
1. Change your present working directory to use the newly created folder by running the below command:

cd task\_10097\_rds

1. Get the location of the present working directory by running the below command:

pwd

1. Note down the location, as you will open the same in the next steps.
2. Now click on the first icon Explorer present on the left sidebar.
3. Click on the button called Open folder and navigate to the location of folder **task\_10097\_rds**.



1. (Optional) Click on Authorize button for allowing Visual Studio Code to use the task\_10097\_rds folder. This will only be asked when you have been using Visual Studio code for a while as you are allowing a new folder to be accessed by VSC.
2. Visual Studio Code is now ready to use.

**Task 3: Create a variable file**

In this task, you will create variable files where you will declare all the global variables with a short description and a default value.

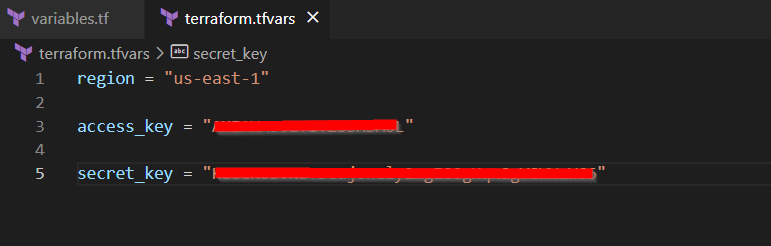
1. To create a variable file, expand the folder **task\_10097\_rds** and click on the **New** **File** icon to add the file.
2. Name the file as **variables.tf** and press **Enter** to save it.
3. **Note:** Don't change the location of the new file, keep it default, i.e. inside the **task\_10097\_rds** folder**.**
4. Paste the below contents in **variables.tf** file.

|  |
| --- |
| variable "access\_key" {  description = "Access key to AWS console"  }  variable "secret\_key" {  description = "Secret key to AWS console"  }  variable "region" {  description = "AWS region"  } |

1. In the above content, you are declaring a variable called, access\_key, secret\_key, and region with a short description of all 3.
2. After pasting the above contents, save the file by pressing **ctrl + S**.
3. Now expand the folder **task\_10097\_rds** and click on the **New File** icon to add the file.
4. Name the file as **terraform.tfvars** and press **Enter** to save it.
5. Paste the below content into the **terraform.tfvars** file.

|  |
| --- |
| region = "us-east-1"  access\_key = "<YOUR AWS CONSOLE ACCESS ID>"  secret\_key = "<YOUR AWS CONSOLE SECRET KEY>" |

1. In the above code, you are defining the dynamic values of variables declared earlier.
2. Replace the values of access\_key and secret\_key by copying from the lab page.
3. After replacing the values of access\_key and secret\_key, save the file by pressing Ctrl + S.



**Task 4: Create a Security group for RDS Instance in main.tf file**

 In this task, you will create a **main.tf** file where you will add details of the provider and resources.

1. To create a **main.tf** file, expand the folder **task\_10097\_rds** and click on the **New** **File** icon to add the file.
2. Name the file as **main.tf** and press **Enter** to save it.
3. Paste the below content into the **main.tf** file.

|  |
| --- |
| provider "aws" {    region     = var.region    access\_key = var.access\_key    secret\_key = var.secret\_key  } |

1. In the above code, you are defining the provider as aws.
2. Next, we want to tell Terraform to create a default VPC, two subnets and a security group for RDS Database Instance
3. To create a security group Paste the below content into the **main.tf** file after the provider

|  |
| --- |
| ####################### Default VPC and Subnets ######################  data "aws\_vpc" "default" {      default = true  }  data "aws\_subnet" "subnet1" {      vpc\_id = data.aws\_vpc.default.id      availability\_zone = "us-east-1a"  }  data "aws\_subnet" "subnet2" {      vpc\_id = data.aws\_vpc.default.id      availability\_zone = "us-east-1b"  }    # Creating a security group  resource "aws\_security\_group" "rds\_sg" {    name        = "rds\_sg"    description = "Security group for RDS instance"    vpc\_id      = data.aws\_vpc.default.id      ingress {      description = "MYSQL"      from\_port   = 3306      to\_port     = 3306      protocol    = "tcp"      cidr\_blocks = ["0.0.0.0/0"]    }      egress {      from\_port   = 0      to\_port     = 0      protocol    = "-1"      cidr\_blocks = ["0.0.0.0/0"]    }  } |

**Task 5: Create a RDS Database Instance in main.tf file**

In this task we are going to create a RDS Database Instance

1. To create a Database Subnet group and RDS Database Instance add another block of code just below the security group code into the **main.tf** file

|  |
| --- |
| ## Creating DB Subnet Group  resource "aws\_db\_subnet\_group" "mydb\_subnet\_group" {    name       = "mydb-subnet-group"    subnet\_ids = [                  data.aws\_subnet.subnet1.id,                  data.aws\_subnet.subnet2.id                  ]    tags = {      Name = "MyDBSubnetGroup"    }  }  # Creating RDS Database Instance  resource "aws\_db\_instance" "myinstance" {    engine               = "mysql"    identifier           = "mydatabaseinstance"    allocated\_storage    = 20    engine\_version       = "8.0"    instance\_class       = "db.t3.micro"    username             = "mydatabaseuser"    password             = "mydatabasepassword"    parameter\_group\_name = "default.mysql8.0"    vpc\_security\_group\_ids = [aws\_security\_group.rds\_sg.id]    db\_subnet\_group\_name  = aws\_db\_subnet\_group.mydb\_subnet\_group.name    skip\_final\_snapshot  = true    publicly\_accessible  = true  } |

1. Save the file by pressing **Ctrl + S**.

**Task 6: Create an Output file**

In this task, you will create an **output.tf** file where you will add details of the provider and resources.

1. To create an **output.tf** file, expand the folder **task\_10097\_rds** and click on the **New** **File** icon to add the file.
2. Name the file as **output.tf** and press **Enter** to save it.
3. Paste the below content into the **output.tf** file.

|  |
| --- |
| output "security\_group\_id" {  value = aws\_security\_group.rds\_sg.id  }  output "db\_instance\_endpoint" {  value = aws\_db\_instance.myinstance.endpoint  } |

1. In the above code, we will extract the security group id and RDS Database Instance endpoint to confirm that they are created.

**Task 7: Confirm the installation of Terraform by checking the version**

1. In the Visual Studio Code, open Terminal by selecting **View** from the Menu bar and choose **Terminal**.
2. If you are not in the newly created folder change your present working directory by running the below command.

cd task\_10097\_rds

1. To confirm the installation of Terraform, run the below command to check the version:

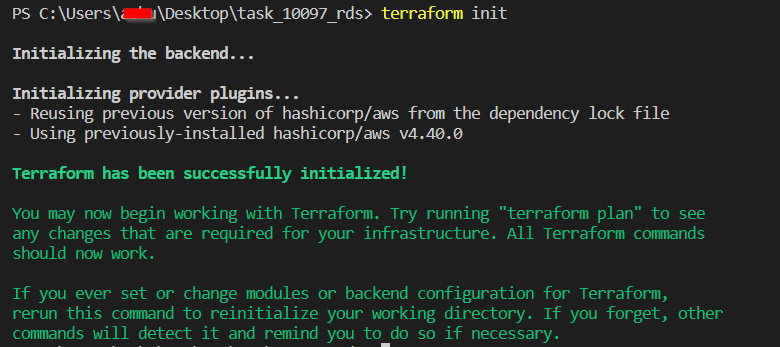
terraform version

1. If you are getting output as command not found: terraform, this means that terraform is not installed on your system, To install terraform follow the official guide link provided in the Prerequisite section above.

**Task 8: Apply terraform configurations**

1. Initialize Terraform by running the below command,

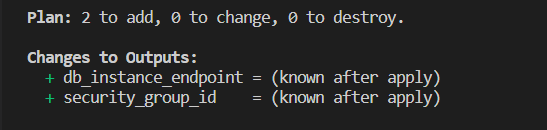
terraform init

****

**Note:** terraform init will check for all the plugin dependencies and download them if required, this will be used for creating a deployment plan

1. To generate the action plans run the below command,

terraform plan

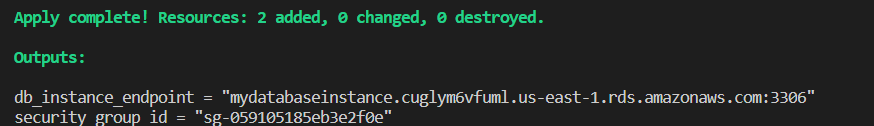
****

1. To create all the resources declared in main.tf configuration file, run the below command,

terraform apply

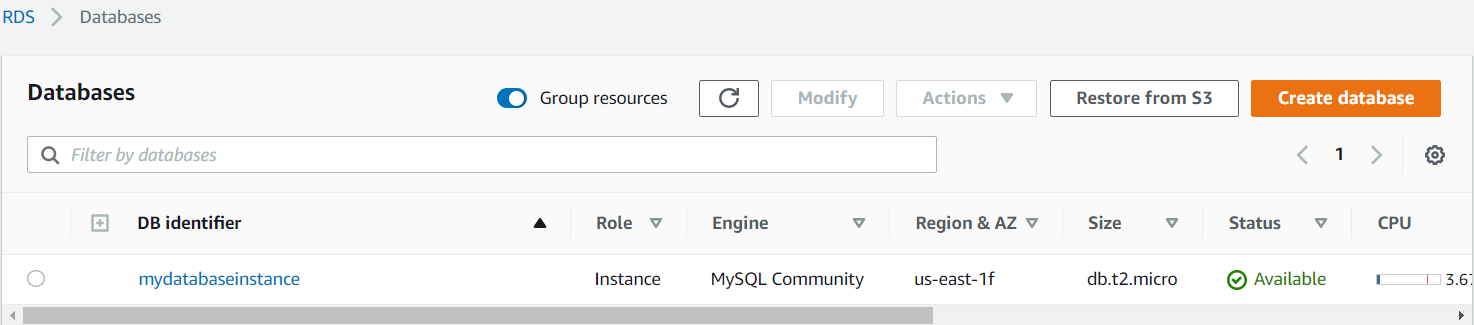
1. Approve the creation of all the resources by entering **yes**.
2. Id’s of all the resources created by terraform will be visible there.

* Note: This process will take around 5-10 minutes.



**Task 9: Check the resources in AWS Console**

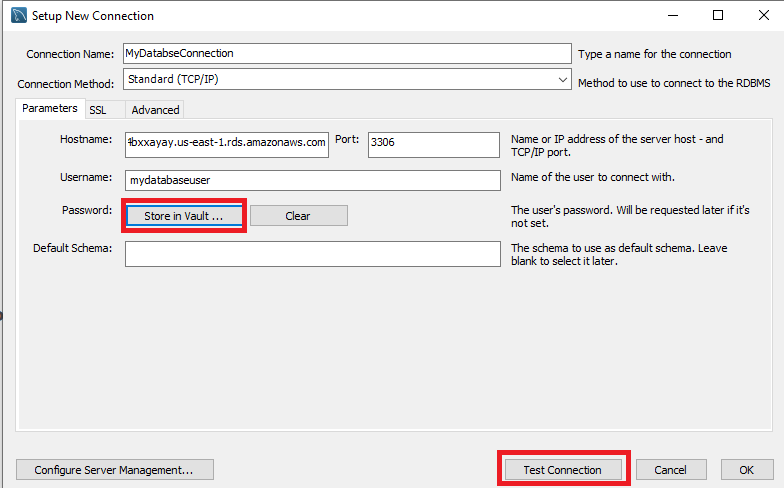
1. Make sure you are in the **US East (N. Virginia) us-east-1** Region.
2. Navigate to **RDS** by clicking on **Services** on the top, then click on **RDS** in the **Database** section.
3. Click on the **Databases** on the left navigation panel. You can see RDS Database Instance created successfully.



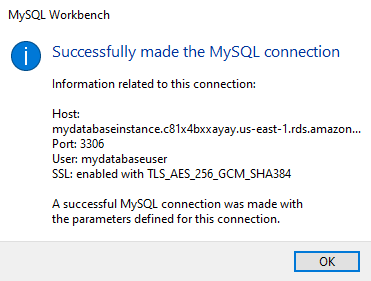
**Task 10: Testing RDS Connection using the MySQL Workbench**

In this task, we will connect to a database on a MySQL DB instance using MySQL monitor commands. One GUI-based application you can use to connect is MySQL workbench, which you have already downloaded and installed based on instructions in the prerequisite section.

1. To connect to a database on a DB instance using MySQL monitor, find the endpoint (DNS name) and port number for your DB Instance.
   * Navigate to **databases** and click on **mydatabaseinstance.**
   * Under the **Connectivity & security** section, copy and note the **endpoint** and **port**.
     + Endpoint: **Copy the endpoint**
     + Port: **3306**
     + You need both the endpoint and the port number to connect to the DB instance.
2. Open MySQL Workbench. Click on the plus icon. **MySQLConnections**
   * Connection Name    : Enter a sample name ***MyDatabseConnection***
   * Host Name         : Enter the endpoint  **mydatabaseinstance.cdegnvsebaim.us-east-1.rds.amazonaws.com**
   * Port            : **3306**
   * Username        : ***mydatabaseuser***
   * Password        : Click on Store in Vault and enter password ***mydatabasepassword***. Click on ok.

****

* Click on **Test Connection** to make sure that you are able to connect to the database properly.

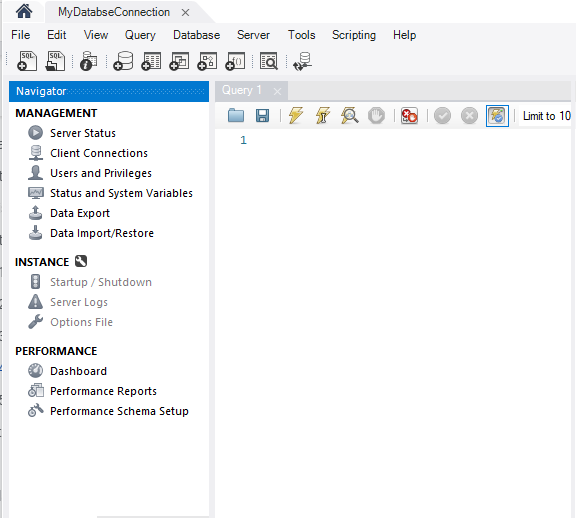


* Click on ok and ok again to save the connection.

1. A database connection will be created in MySQL Workbench



1. Click on it to open the database. Enter the database password if prompted.
2. After successfully connecting and opening the database, you can create tables and perform various queries over the connected database.



1. Navigate to the **Schemas** tab to see databases available to start doing database operations. More details on database operations are available [here](https://dev.mysql.com/doc/).

**Do You Know ?**

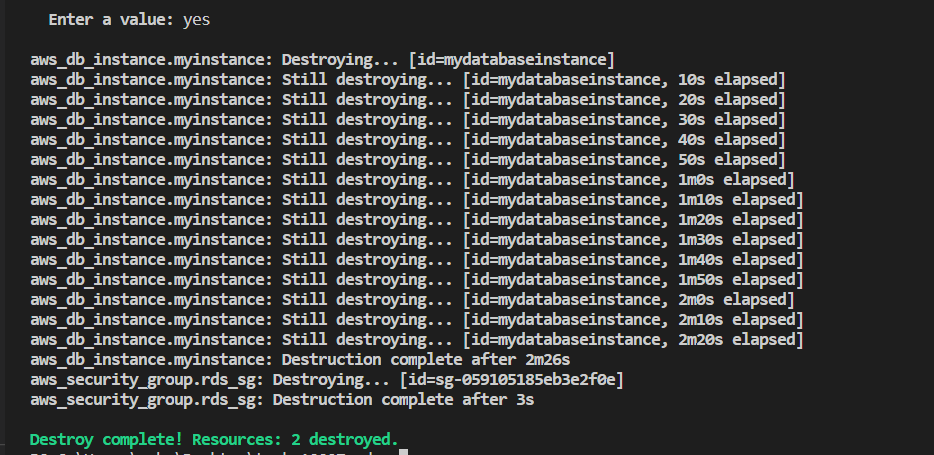
With Terraform, you can pass in a **parameter\_group\_name** to reference a custom parameter group. This parameter group can contain advanced MySQL configuration settings beyond the basic options exposed in the Terraform RDS resource. This enables you to fine-tune various database parameters, such as query caching, buffer sizes, or replication settings, to align the RDS instance more precisely with your application's performance and scalability needs.

**Task 11: Delete AWS Resources**

1. To delete the resources, open Terminal again.
2. Run the below command to delete all the resources.

terraform destroy

1. Enter **yes** to confirm the deletion. You can see the **Destroy complete!** message.
   * Note: This process will take around 5-10 minutes.



**Completion and Conclusion**

1. You have successfully set up the Visual Studio Code editor.
2. You have successfully created variables.tf and terraform.tfvars files.
3. You have successfully created a security group using the terraform.
4. You have successfully created RDS Database Instance using the terraform.
5. You have successfully created output.tf
6. You have successfully executed the terraform configuration commands to create the resources.
7. You have successfully checked all the resources created by opening the Console.
8. You have successfully Tested RDS Connection using MySQL Workbench
9. You have deleted all the resources.