

# Complete Project Report: AWS Database Migration using Terraform

**Project Goal:** To provision a cloud infrastructure environment using Infrastructure as Code (Terraform) and execute a successful logical database migration from a self-managed EC2 instance (Source) to a managed AWS RDS instance (Target).

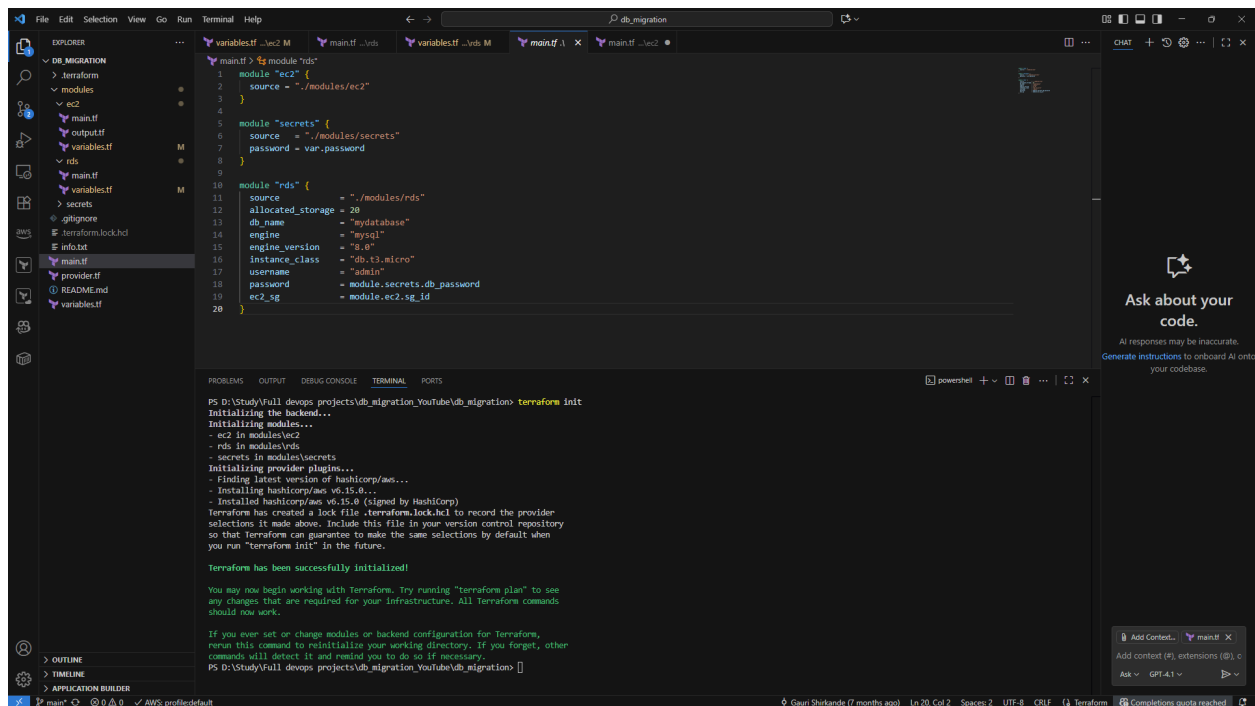
## 1. Infrastructure as Code (IaC) Setup and Deployment

This phase focused on defining and deploying the required AWS resources—an EC2 instance, an RDS MySQL database, and secure credential storage—using Terraform.

### 1.1 Project Initialization and Configuration Review

Step 1: Initialize the Terraform Backend and Modules

The project was initialized using `terraform init` to download the necessary AWS provider and modules, confirming the project's readiness for deployment.



The screenshot displays the Visual Studio Code interface with the Terraform project files open. The Explorer pane on the left shows the project structure, including `main.tf`, `variables.tf`, `outputs.tf`, `providers.tf`, and `terraform.lock.hcl`. The main editor shows the content of `main.tf`, which defines three modules: `ec2`, `secrets`, and `rds`. The `ec2` module is configured with a source of `../modules/ec2`. The `secrets` module is configured with a source of `../modules/secrets` and a password variable. The `rds` module is configured with a source of `../modules/rds` and various instance parameters including `allocated_storage`, `db_name`, `engine`, `engine_version`, `instance_class`, `username`, `password`, and `ec2_sg`.

The terminal window at the bottom shows the output of the `terraform init` command, which successfully initializes the Terraform backend and modules. The output includes the following text:

```
PS D:\Study\Full devops projects\db_migration\Youtube\db_migration> terraform init
Initializing the backend...
Initializing modules...
- ec2 in modules\ec2
- rds in modules\rds
- secrets in modules\secrets
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.15.0...
- Installed hashicorp/aws v6.15.0 (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.
PS D:\Study\Full devops projects\db_migration\Youtube\db_migration>
```

Step 2: Review the Core Terraform Configuration

The core configuration, defined in `main.tf`, was reviewed. It shows modular calls for the EC2 instance, RDS database, and Secrets Manager, ensuring a clean and reusable structure.

The screenshot shows the VS Code interface with a Terraform configuration file open. The terminal displays the output of the `terraform init` command, which initializes the working directory by installing the required providers and modules. The output includes the following text:

```

Initializing modules...
- ec2 in modules/ec2
- rds in modules/rds
- secrets in modules/secrets
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.15.0...
- Installed hashicorp/aws v6.15.0 (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.
PS D:\Study\Full devops projects\db_migration\Youtube\db_migration> terraform validate
Success! The configuration is valid.
  
```

## 1.2 Deployment Planning and Execution

### Step 3: Execute terraform plan (EC2 review)

The terraform plan command was executed. The output was reviewed to confirm the planned creation of the EC2 instance and its associated resources.

The screenshot shows the VS Code interface with the same Terraform configuration file open. The terminal displays the output of the `terraform plan` command, which generates an execution plan for the infrastructure. The output includes the following text:

```

PS D:\Study\Full devops projects\db_migration\Youtube\db_migration> terraform plan
var.password
Enter a value: MySecurePass123!

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:

# module.ec2.aws_instance.web will be created
+ resource "aws_instance" "web" {
  + ami              = "ami-0f28064a708f0f0d3"
  + ami              = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (known after apply)
  + disable_api_stop    = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized       = (known after apply)
  + enable_primary_ipv6 = (known after apply)
  + force_destroy       = false
  + 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)
  
```

### Step 4: Review the terraform plan (Secrets Manager review)

The plan output was further reviewed to confirm the planned creation of the AWS Secrets Manager resource, which securely stores the database credentials.

```

1 module "ec2" {
2   source = "../modules/ec2"
3 }
4
5 module "secrets" {
6   source = "../modules/secrets"
7   password = var.password
8 }
9
10 module "rds" {
11   source = "../modules/rds"
12   allocated_storage = 20
13   db_name = "mydatabase"
14   engine = "mysql"
15   engine_version = "8.0"
16   instance_class = "db.t3.micro"
17   username = "admin"
18   password = module.secrets.db_password
19   ec2_sg = module.ec2_sg_id
20 }

```

```

+ policy = (known after apply)
+ recovery_window_in_days = 30
+ region = "us-east-1"
+ tags_all = (known after apply)
+ replica (known after apply)

# module.secrets.aws_secretsmanager_secret_version.secret_version will be created
+ resource "aws_secretsmanager_secret_version" "secret_version" {
+   name = (known after apply)
+   has_secret_string_wo = (known after apply)
+   id = (known after apply)
+   region = "us-east-1"
+   secret_id = (known after apply)
+   secret_string = (sensitive value)
+   secret_string_wo = (write-only attribute)
+   version_id = (known after apply)
+   version_stages = (known after apply)
}

Plan: 11 to add, 0 to change, 0 to destroy.

```

## Step 5: Execute terraform apply

The terraform apply -auto-approve command was executed to start the resource provisioning in AWS, securely passing the required database password as a variable.

```

PS D:\Study\Full devops projects\db_migration\Youtube\db_migration> terraform apply -auto-approve
var.password
Enter a value: MySecurePass123!

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:

# module.ec2.aws_instance.web will be created
+ resource "aws_instance" "web" {
+   ami = "ami-0c364709f00000000"
+   arm = (known after apply)
+   associate_public_ip_address = (known after apply)
+   availability_zone = (known after apply)
+   disable_api_stop = (known after apply)
+   disable_api_termination = (known after apply)
+   ebs_optimized = (known after apply)
+   enable_primary_ip6 = (known after apply)
+   force_destroy = false
+   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 = "t3.micro"
+   ipv6_address_count = (known after apply)
+   ipv6_addresses = (known after apply)
}

```

## Step 6: Verify Deployment Completion

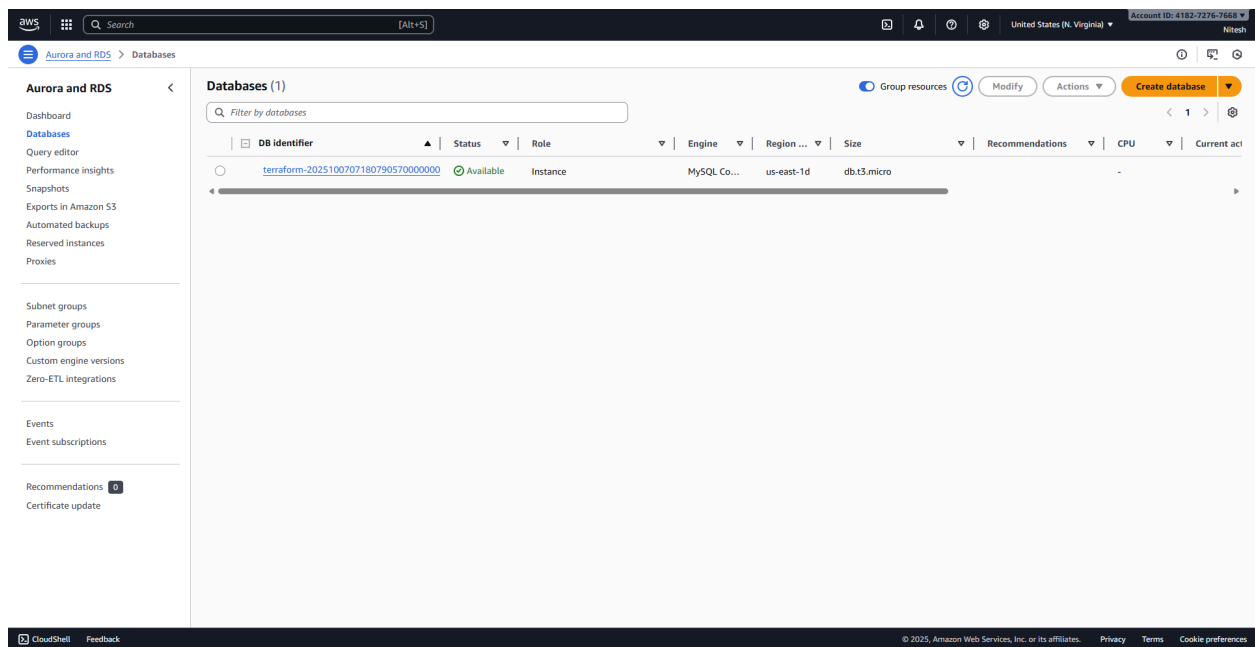
The console confirmed the successful completion of the deployment: "Apply complete! Resources: 11 added, 0 changed, 0 destroyed."



The AWS Management Console was checked to confirm that the EC2 instance (\$db\\_server) was in the Running state.



The AWS Management Console was checked to confirm that the RDS MySQL instance was in the Available state.



## 2. Source Database Preparation on EC2

The provisioned EC2 instance was prepared as the source database environment.

### 2.1 MySQL Installation and Configuration

Step 9: Install MySQL Server and Client

Commands were executed via SSH to download the MySQL repository and install the server and client components on the EC2 instance.

```

# Amazon Linux 2023
# https://aws.amazon.com/linux/amazon-linux-2023

Last login: Tue Oct 7 07:27:30 2025 from 18.206.107.28
(ec2-user@ip-172-31-94-91 ~)$ # Download the official MySQL 8.0 repo package
sudo wget https://repo.mysql.com/mysql80-community-release-el9-1.noarch.rpm

# Install the repo package
sudo dnf install -y mysql80-community-release-el9-1.noarch.rpm

# Enable the MySQL 8.0 repository (should be enabled by default)
sudo dnf repolist enabled | grep mysql

# Install MySQL server
sudo dnf install -y mysql-community-server

# Start and enable MySQL
sudo systemctl start mysqld
sudo systemctl enable mysqld

# Check service status
sudo systemctl status mysqld

# Get temporary root password (generated during install)
sudo grep 'temporary password' /var/log/mysqld.log

```

i-0ebed951c47e4415e (db\_server)  
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

Step 10: Confirm Service Status and Retrieve Password

The mysqld service was confirmed to be running, and the initial temporary root password was retrieved from the MySQL log file.

```

=====
] Installing      : mysql-community-server-8.0.43-1.el9.x86_64 [
] Installing      : mysql-community-server-8.0.43-1.el9.x86_64 [
] Installing      : mysql-community-server-8.0.43-1.el9.x86_64
6/6
Running scriptlet: mysql-community-server-8.0.43-1.el9.x86_64
6/6
Verifying        : mysql-community-client-8.0.43-1.el9.x86_64
1/6
Verifying        : mysql-community-client-plugins-8.0.43-1.el9.x86_64
2/6
Verifying        : mysql-community-common-8.0.43-1.el9.x86_64
3/6
Verifying        : mysql-community-icu-data-files-8.0.43-1.el9.x86_64
4/6
Verifying        : mysql-community-libs-8.0.43-1.el9.x86_64
5/6
Verifying        : mysql-community-server-8.0.43-1.el9.x86_64
6/6

Installed:
mysql-community-client-8.0.43-1.el9.x86_64  mysql-community-client-plugins-8.0.43-1.el9.x86_64  mysql-community-common-8.0.43-1.el9.x86_64  mysql-community-icu-data-files-8.0.43-1.el9.x86_64
mysql-community-libs-8.0.43-1.el9.x86_64  mysql-community-server-8.0.43-1.el9.x86_64

Complete!
[ec2-user@ip-172-31-94-91 ~]$ sudo systemctl start mysqld
[ec2-user@ip-172-31-94-91 ~]$ sudo systemctl enable mysqld
[ec2-user@ip-172-31-94-91 ~]$ sudo systemctl status mysqld
* mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
   Active: active (running) since Tue 2025-10-07 07:42:50 UTC; 12s ago
     Docs: man:mysqld.8
           https://dev.mysql.com/doc/refman/8.0/using-systemd.html
   Main PID: 27286 (mysqld)
   Status: "Server is operational"
     Tasks: 38 (limit: 1106)
    Memory: 425.2M
       CPU: 5.100s
   CGroup: /system.slice/mysqld.service
           └─27286 /usr/sbin/mysqld

Oct 07 07:42:40 ip-172-31-94-91.ec2.internal systemd[1]: Starting mysqld.service - MySQL Server...
Oct 07 07:42:50 ip-172-31-94-91.ec2.internal systemd[1]: Started mysqld.service - MySQL Server.
[ec2-user@ip-172-31-94-91 ~]$ sudo grep 'temporary password' /var/log/mysqld.log
2025-10-07T07:42:45.441358Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: cd1ep85y1e5
[ec2-user@ip-172-31-94-91 ~]$

i-Oebed951c47e4415e (db_server)
PublicIP: 107.23.50.6 PrivateIP: 172.31.94.91

```

## 2.2 Source Database Setup

### Step 11: Log into the MySQL Monitor

The temporary password was used to log into the MySQL monitor to begin the configuration process.

```

[ec2-user@ip-172-31-94-91 ~]$ sudo su -
[root@ip-172-31-94-91 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43

Copyright (c) 2000, 2025, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

i-Oebed951c47e4415e (db_server)
PublicIP: 107.23.50.6 PrivateIP: 172.31.94.91

```

### Step 12: Set a Secure Root Password

A secure, permanent password was set for the root user to secure the source database instance.

```
[ec2-user@ip-172-31-94-91 ~]$ sudo su -
[root@ip-172-31-94-91 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER root@localhost IDENTIFIED BY 'Admin@123'
->
Query OK, 0 rows affected (0.01 sec)

mysql>
```

i-0ebed951c47e4415e (db\_server)  
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### Step 13: Create the Source Database

The source database, named ec2db, was created and its existence was verified.

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER root@localhost IDENTIFIED BY 'Admin@123'
->
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW DATABASES
->
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> CREATE DATABASE ec2db;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| ec2db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql>
```

i-0ebed951c47e4415e (db\_server)  
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### Step 14: Populate Test Data

A sample Employee table was created, and two test records were inserted and verified, creating the data payload for the migration.

```
mysql>
+-----+
| Database |
+-----+
| ec2db    |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.01 sec)

mysql> USE ec2db;
Database changed
mysql> CREATE TABLE Employee (FirstName VARCHAR(20), LastName VARCHAR(20), DoB DATE);
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO Employee (FirstName, LastName, DoB) VALUES ("Nitesh", "Punia", "2005-08-15");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (FirstName, LastName, DoB) VALUES ("Tony", "Stark", "2004-07-14");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Employee
+-----+
| FirstName | LastName | DoB      |
+-----+
| Nitesh    | Punia    | 2005-08-15 |
| Tony      | Stark    | 2004-07-14 |
+-----+
2 rows in set (0.00 sec)

mysql>
```

i-Oebed951c47e4415e (db\_server)  
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 3. Database Migration and Verification

This phase involved data export from the source, import to the target, and validation.

### 3.1 Target Connectivity and Readiness Check

Step 15: Connect to the Target RDS Endpoint

A connection was successfully established from the EC2 source instance to the RDS target endpoint, confirming network security group rules were correct.

```
[root@ip-172-31-94-91 ~]# mysql -h terraform-20251007071807905700000004.cizia041acrr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

i-Oebed951c47e4415e (db\_server)  
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 16: Confirm Target Database Existence

The RDS databases were listed, confirming the presence of the Terraform-created target, mydbdatabase.



```
[root@ip-172-31-94-91 ~]# mysql -h terraform-20251007071807905700000004.cisla041acrr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES
+-----+
| Database |
+-----+
| information_schema |
| mydatabase |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql>
```

i-Oebed951c47e4415e (db\_server)

PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

## Step 17: Confirm Target Database is Empty

The target database was inspected (SHOW TABLES) to confirm it was empty prior to the migration process.

```
[root@ip-172-31-94-91 ~]# mysql -h terraform-20251007071807905700000004.cisla041acrr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES
+-----+
| Database |
+-----+
| information_schema |
| mydatabase |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> USE mydatabase
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql>
```

i-Oebed951c47e4415e (db\_server)

PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

## 3.2 Data Export and Import

### Step 18: Execute Data Dump (Export)

The mysqldump utility was used to export the entire ec2db schema and data into a file named db.sql on the EC2 file system.

```
[root@ip-172-31-94-91 ~]# mysqldump -u root -p ec2db > db.sql
Enter password:
[root@ip-172-31-94-91 ~]# ls
db.sql
[root@ip-172-31-94-91 ~]#
```

i-0ebd951c47e4415e (db\_server)  
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### Step 19: Execute Data Load (Import)

The MySQL client was used to import the db.sql file into the target RDS database (mydbdatabase), completing the logical migration.

```
[root@ip-172-31-94-91 ~]# mysqldump -u root -p ec2db > db.sql
Enter password:
[root@ip-172-31-94-91 ~]# ls
db.sql
[root@ip-172-31-94-91 ~]# mysql -h terraform-20251007071807905700000004.cizia04isrr.us-east-1.rds.amazonaws.com -u admin -P 3306 -p mydatabase < db.sql
Enter password:
[root@ip-172-31-94-91 ~]#
```

i-0ebd951c47e4415e (db\_server)  
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 3.3 Post-Migration Validation

### Step 20: Verify Table Existence in Target RDS

The target RDS database was checked, confirming the successful creation of the Employee table.

```
[root@ip-172-31-94-91 ~]# mysqldump -u root -p ec2db > db.sql
Enter password:
[root@ip-172-31-94-91 ~]# ls
db.sql
[root@ip-172-31-94-91 ~]# mysql -h terraform-20251007071807905700000004.cizia041acrr.us-east-1.rds.amazonaws.com -u admin -P 3306 -p mydatabase < db.sql
Enter password:
[root@ip-172-31-94-91 ~]# mysql -h terraform-20251007071807905700000004.cizia041acrr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mydatabase |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> USE mydatabase
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_mydatabase |
+-----+
| Employee |
+-----+
1 row in set (0.00 sec)

i-Oebed951c47e4415e (db_server)
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91
```

## Step 21: Verify Data Integrity in Target RDS

A SELECT query was executed on the Employee table in RDS, confirming that all source data rows were successfully migrated and intact.

```
Server version: 8.0.42 Source distribution
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mydatabase |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> USE mydatabase
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_mydatabase |
+-----+
| Employee |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM Employee;
+-----+
| FirstName | LastName | DoB |
+-----+
| Nitesh | Punia | 2005-08-15 |
| Tony | Stark | 2004-07-14 |
+-----+
2 rows in set (0.00 sec)

mysql>

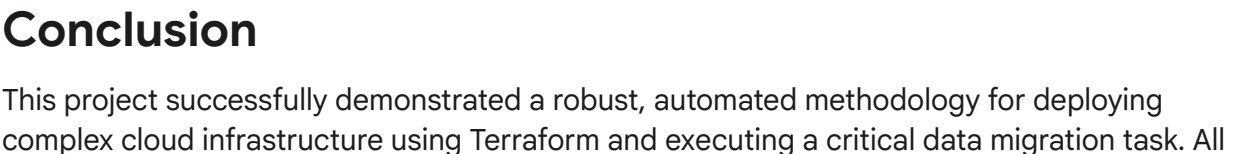
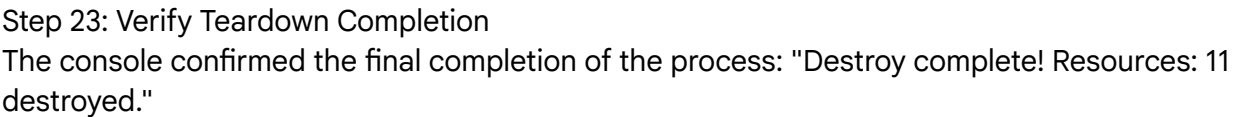
i-Oebed951c47e4415e (db_server)
PublicIPs: 107.23.50.6 PrivateIPs: 172.31.94.91
```


## 4. Infrastructure Teardown

To manage cloud costs, the entire environment was systematically destroyed using Terraform.

### Step 22: Execute terraform destroy

The terraform destroy -auto-approve command was executed to initiate the resource removal process.





phases, from infrastructure provisioning and security configuration to data transfer and environment cleanup, were completed efficiently and verified at each stage.