**Task on RDS**

**Introduction to AWS RDS**

**Amazon Relational Database Service (Amazon RDS)** is a managed database service that simplifies the setup, operation, and scaling of relational databases in the AWS Cloud. It provides cost-efficient, resizable capacity while automating administrative tasks such as backups, patching, and failure recovery.

**Key Features**

* **Managed Service** – AWS handles database maintenance, backups, and scaling.
* **Multi-AZ Deployments** – Ensures high availability with automatic failover.
* **Read Replicas** – Improves performance by distributing read traffic.
* **Security** – Supports encryption, IAM authentication, and VPC isolation.
* **Automated Backups** – Enables point-in-time recovery.

**Supported Database Engines**

* **Amazon Aurora**
* **MySQL**
* **PostgreSQL**
* **MariaDB**
* **Oracle**
* **Microsoft SQL Server**

**Use Cases**

* **Web Applications** – Scalable backend for dynamic websites.
* **Enterprise Databases** – Secure and reliable storage for business applications.
* **Analytics & Reporting** – Supports complex queries and data processing.

1) Create mariadb db on ec2

Go to AWS console

Select EC2 and connect EC2

Why should we run db on EC2

Access to the DB instance OS

Vendor Demands

DB or DB version that aws doesn’t provide.

Install Db on ec2 steps:

Begin Configuration:

=================== s

udo su - yum -y install mariadb-server wget

systemctl enable mariadb

systemctl start mariadb

yum -y update

Set Environmental Variables

===========================

DBName=ec2db

DBPassword=admin123456

DBRootPassword=admin123456

DBUser=ec2dbuser

Database Setup on EC2 Instance:

===============================

echo "CREATE DATABASE ${DBName};" >> /tmp/db.setup

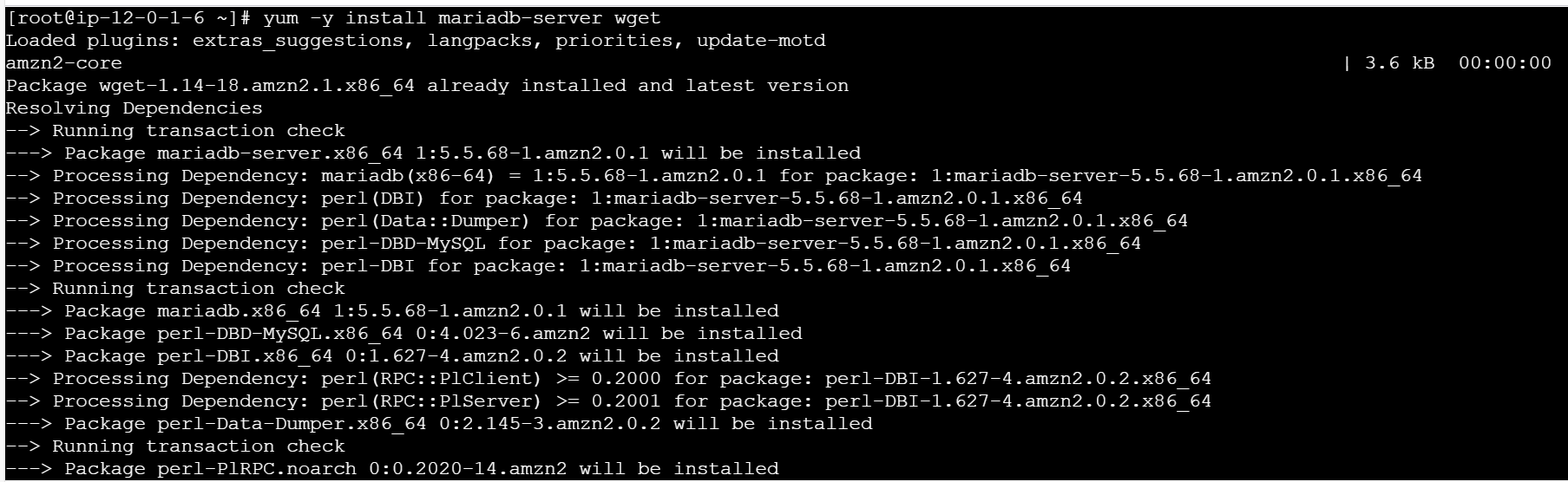
echo "CREATE USER '${DBUser}' IDENTIFIED BY '${DBPassword}';" >> /tmp/db.setup

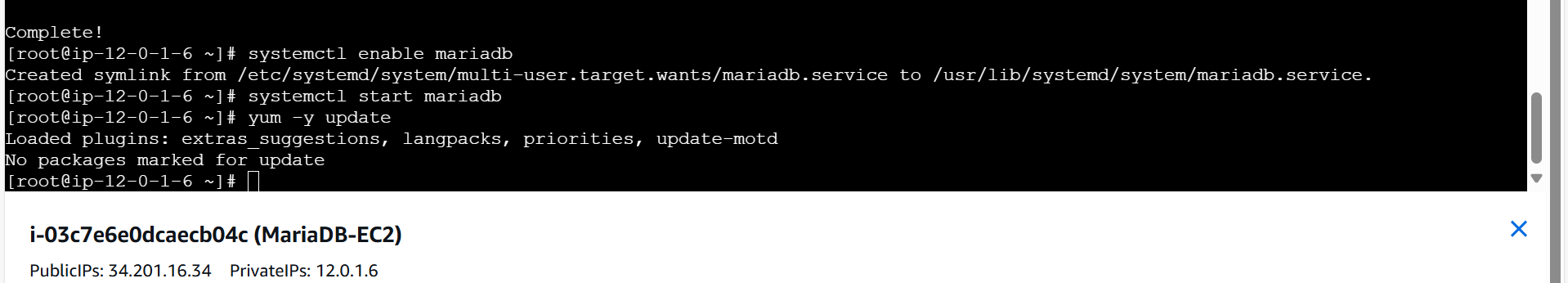
echo "GRANT ALL PRIVILEGES ON TO '${DBUser}'@'%';" >> /tmp/db.setup

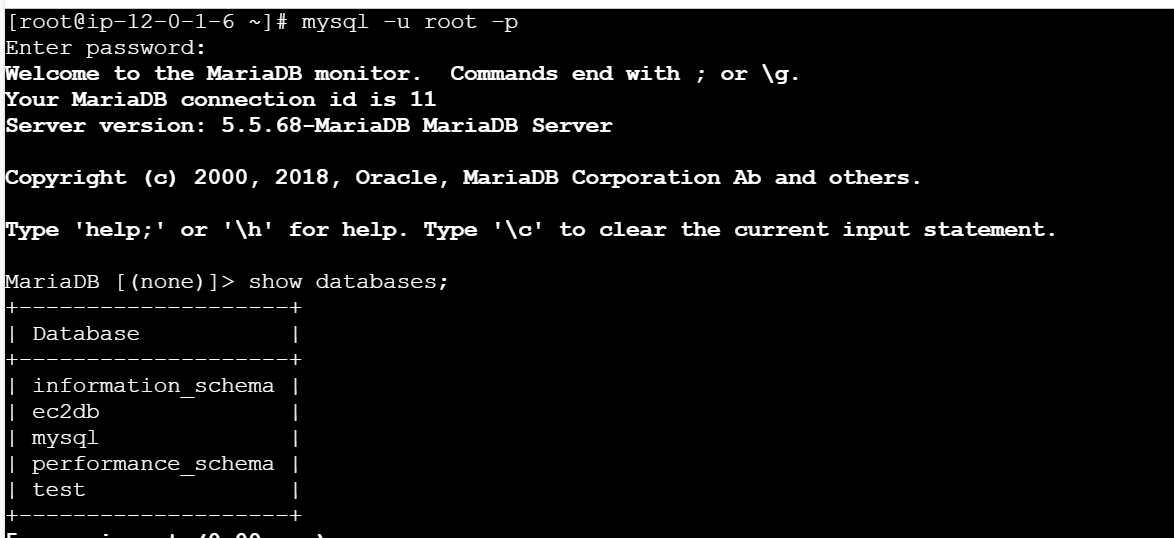
echo "FLUSH PRIVILEGES;" >> /tmp/db.setup

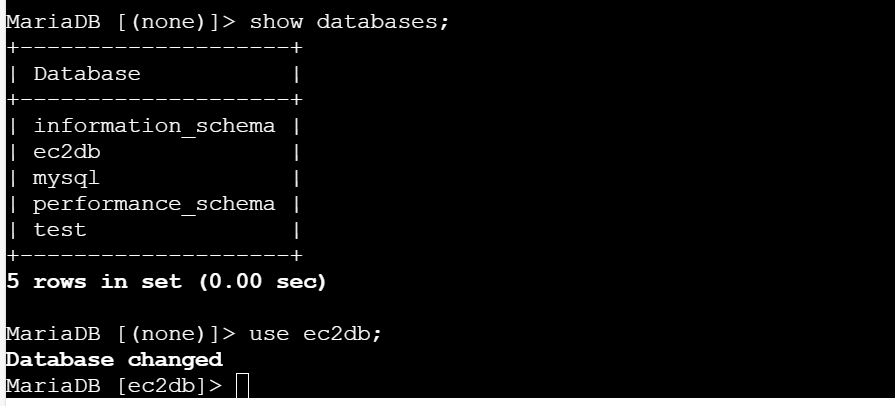
mysqladmin -u root password "${DBRootPassword}" mysql -u root --password="${DBRootPassword}" < /tmp/db.setup

rm /tmp/db.setup









2) Insert some dummy data

To insert some dummy data need to create Table

Create table1 (id INT,name VARCHAR(45));

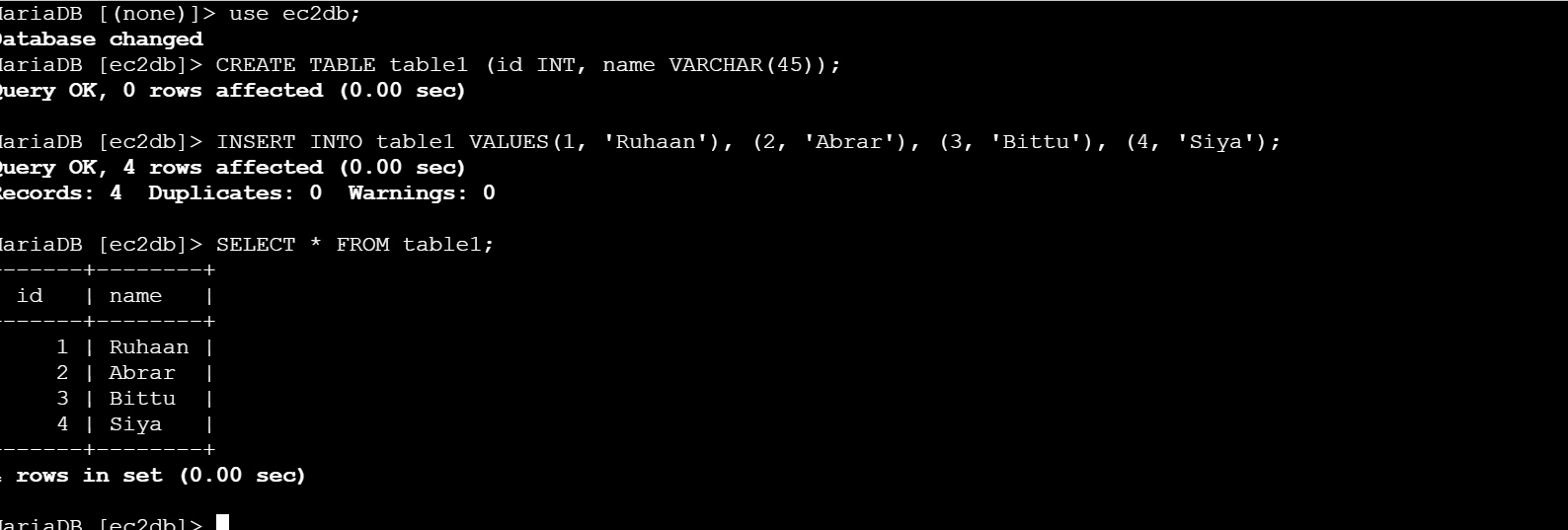
Now insert the data into table1 which we created

INSERT INTO table1 VALUES (1,’Ruhaan’), (2,’Abrar’),(3,’Bittu),(4,’Siya’);

Once we execute this command we can insert the data successfully into table1

Now we can see the output from database

SHOW \* FROM table1;



3) Take the backup of dummy data on ec2

To take the backup of dummy data on ec2 using below command

Mysqldump –u root –p database\_name > file\_name.sql

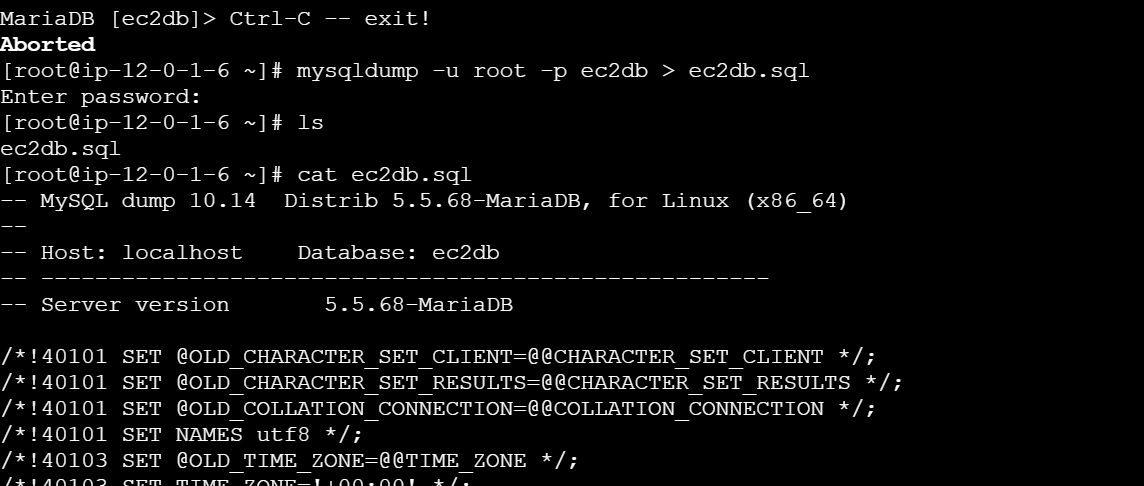
-u –user

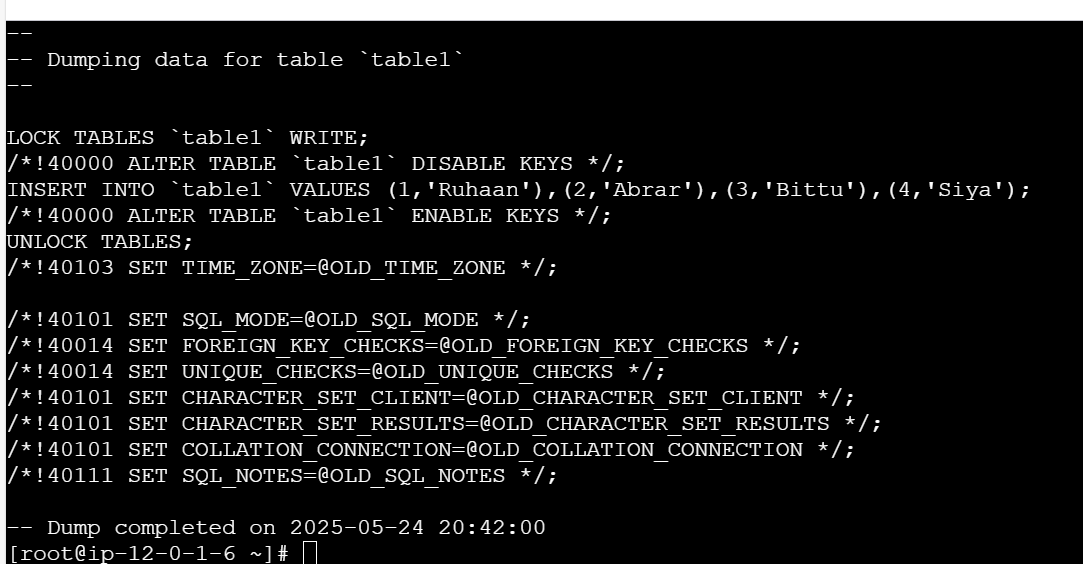
-p—password

Mysqldump –u root –p ec2db > ec2db.sql

Enter the password

Use ls command to list the file created ec2db.sql





5) Migrate database from ec2 to RDS

**Objective:**

Set up a new Amazon RDS instance to host a managed relational database with scalability, security, and automated backups.

**Prerequisites:**

* An AWS account with RDS access.
* IAM permissions for creating and managing RDS instances.
* A well-configured **VPC** and **subnet group** to allow database connectivity.

**Step 1: Open Amazon RDS Console**

1. Sign in to **AWS Management Console**.
2. Navigate to **RDS** → Click **Create Database**.

**Step 2: Choose Database Engine**

1. Under **Database creation method**, select **Standard Create**.
2. Choose your database engine (e.g., **MySQL**, **PostgreSQL**, **MariaDB**, **SQL Server**, or **Aurora**).
3. Select the latest **engine version** for compatibility.

**Step 3: Configure Database Instance**

1. **DB Instance Identifier:** Enter a unique name for the database.
2. **DB Instance Class:** Choose based on workload (e.g., db.t3.micro for free-tier, db.m5.large for production).
3. **Storage Type:**
   1. **General Purpose SSD (gp2)** – balanced performance.
   2. **Provisioned IOPS (io1)** – for high-transaction workloads.
   3. **Magnetic (standard)** – cost-effective option.
4. **Storage Size:** Define storage allocation (e.g., 20GB).

**Step 4: Configure High Availability (Optional)**

1. **Multi-AZ Deployment:** Choose **Yes** for automatic failover.
2. **Read Replica:** Enable if needed for scalability.

**Step 5: Set Up Database Authentication**

1. **Master Username:** Define an admin username.
2. **Master Password:** Set a strong password.

**Step 6: Configure Networking & Security**

1. **VPC:** Choose an existing or create a new VPC.
2. **Subnet Group:** Select an appropriate subnet group.
3. **Public Access:** Enable or disable based on security needs.
4. **Security Groups:**

* Allow inbound traffic for the database (TCP 3306 for MySQL, TCP 5432 for PostgreSQL).
* Restrict access based on application requirements.

**Step 7: Enable Monitoring & Backups**

1. **CloudWatch Enhanced Monitoring:** Enable for performance tracking.
2. **Backup Retention Period:** Set retention days (e.g., 7 days).
3. **Encryption:** Enable encryption for data security.

**Step 8: Launch the Database**

1. Review all configurations.
2. Click **Create Database**.
3. Wait for instance initialization (available status).

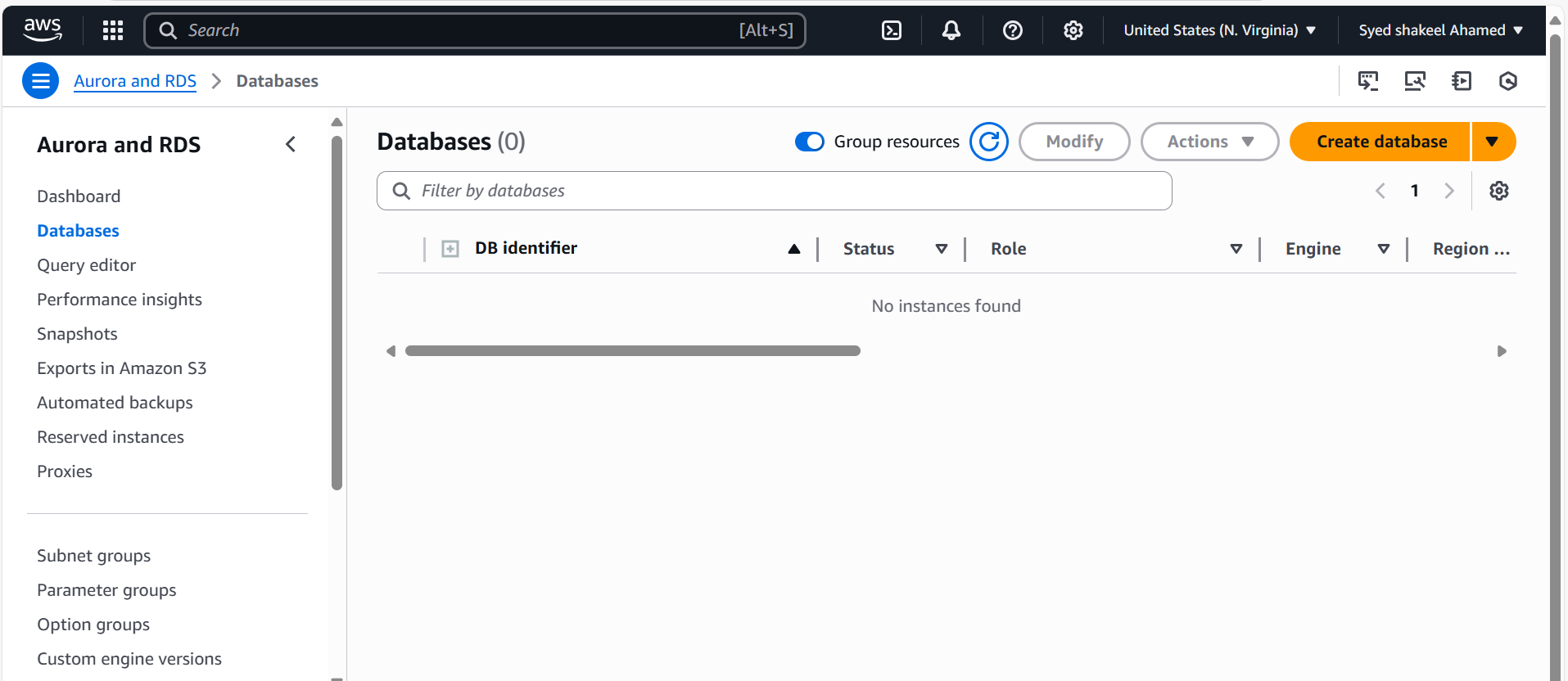
**Step 9: Connect to Your RDS Instance**

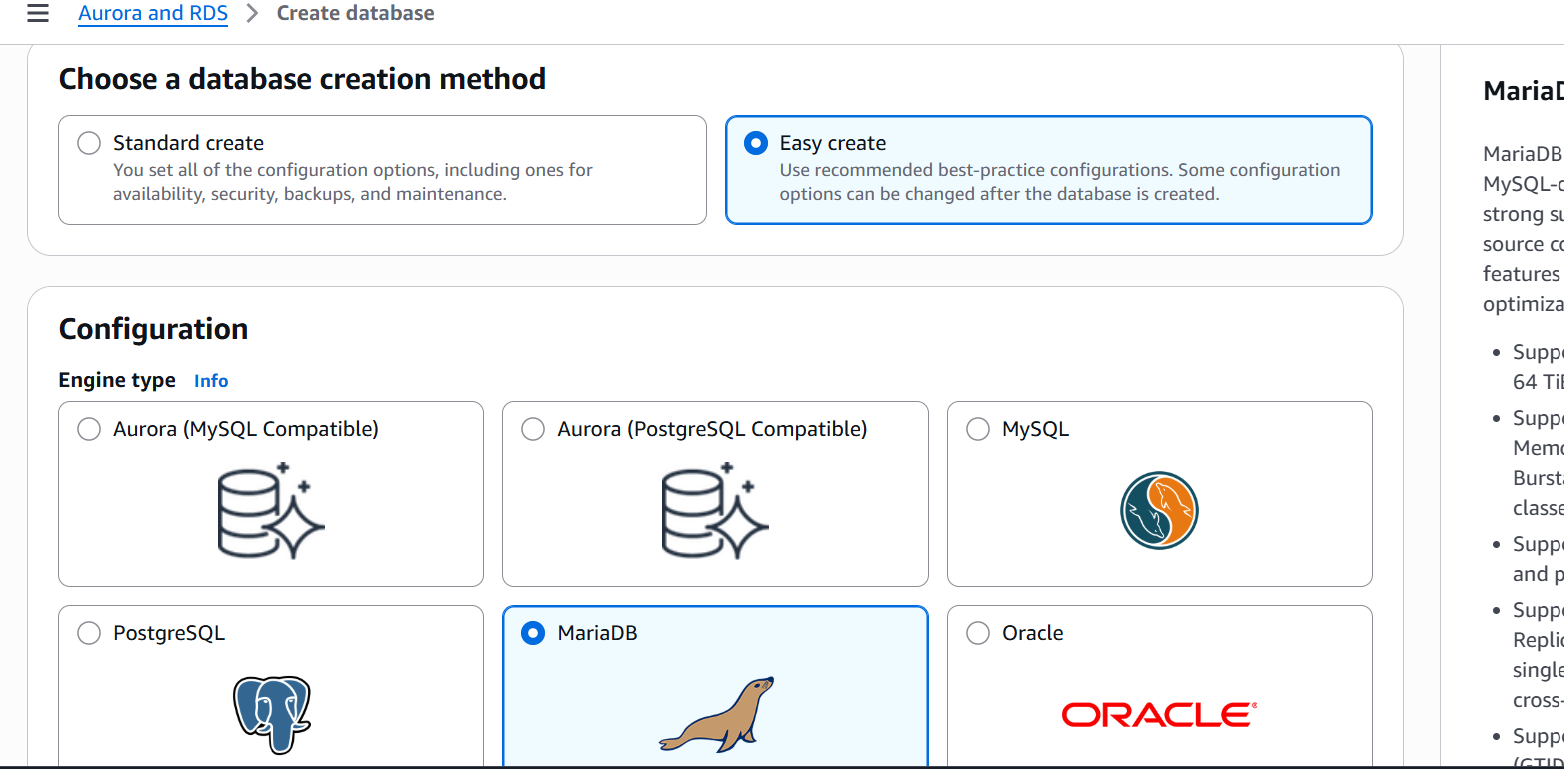
1. Retrieve the **endpoint** from the **RDS Console** → **Connectivity & Security**.
2. Connect via a MySQL/PostgreSQL client:

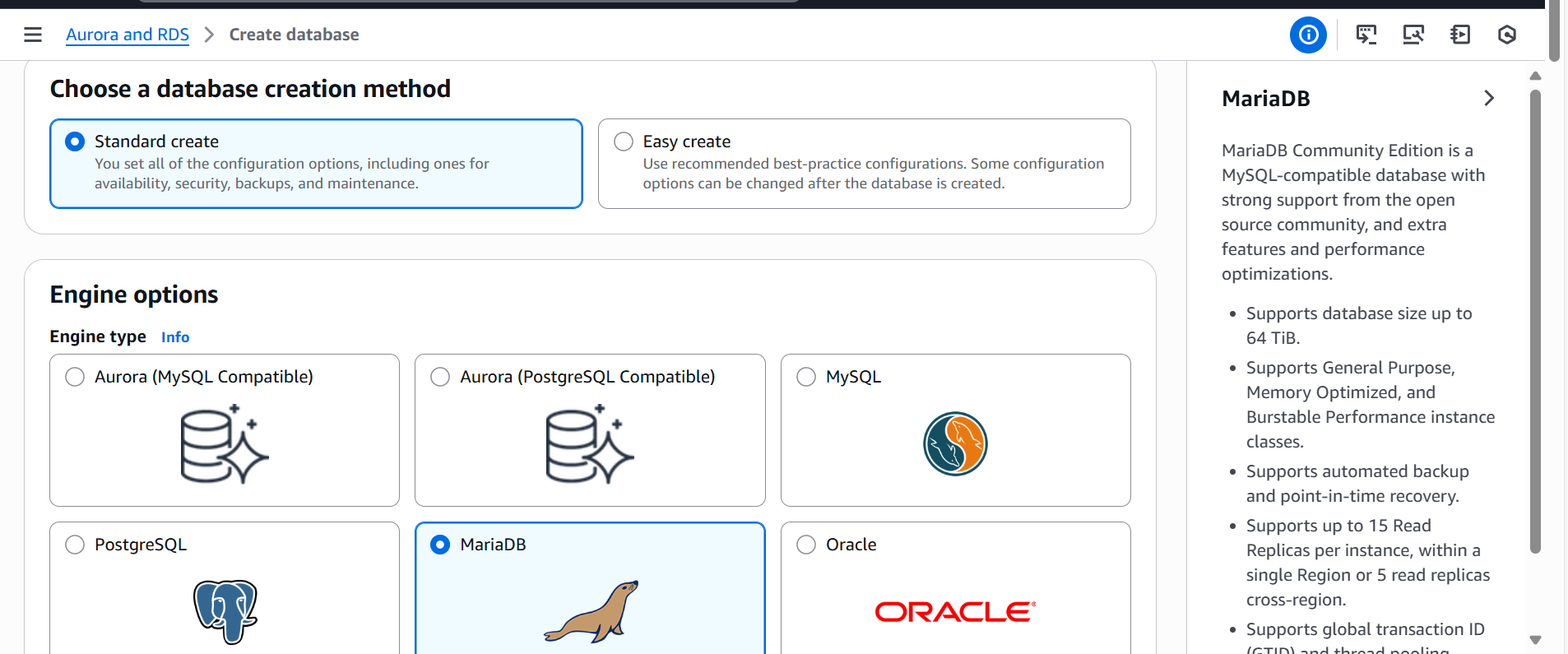
mysql -h -u -p

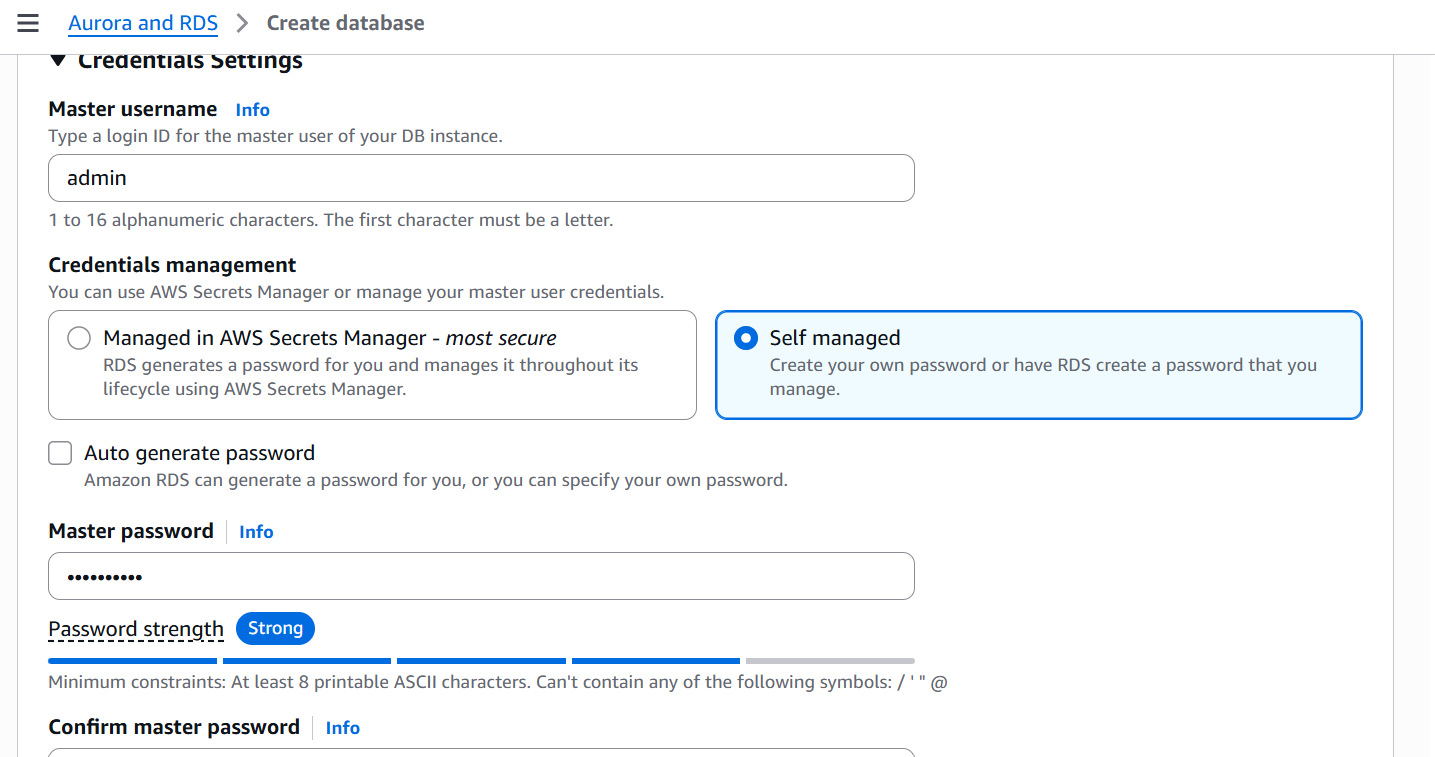
1. Verify connection and test queries:

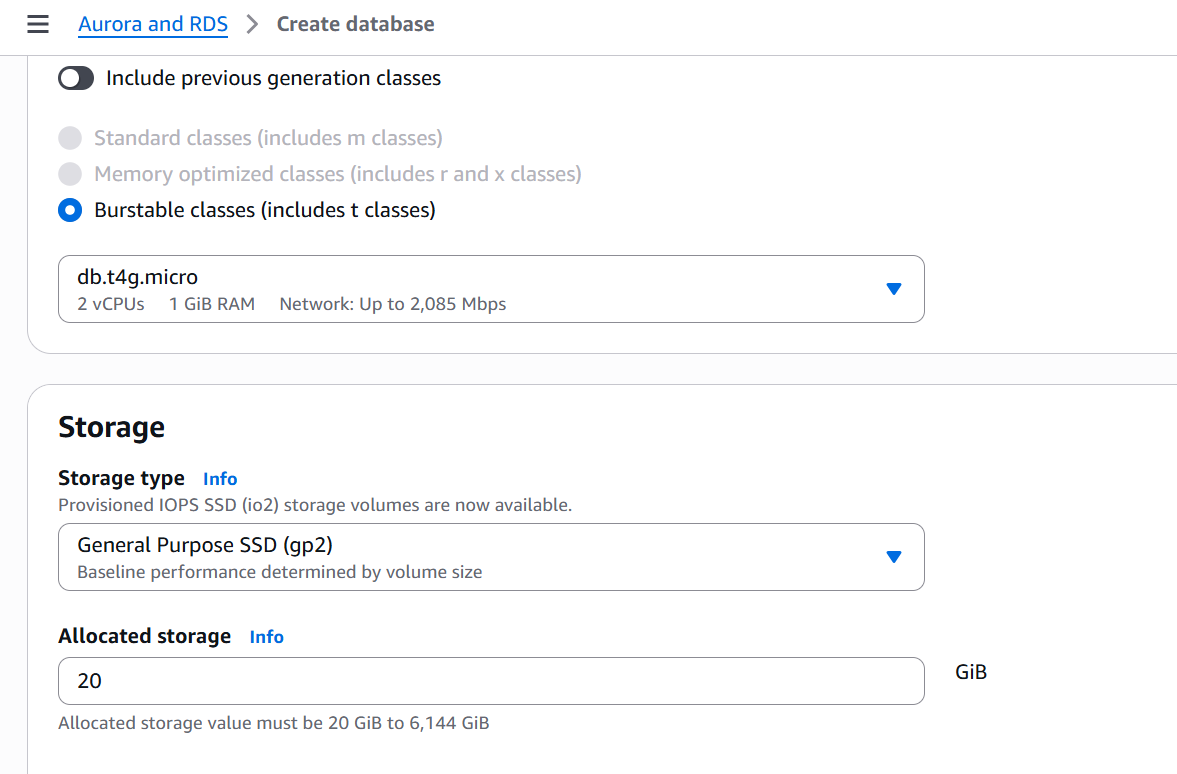
SHOW DATABASES;

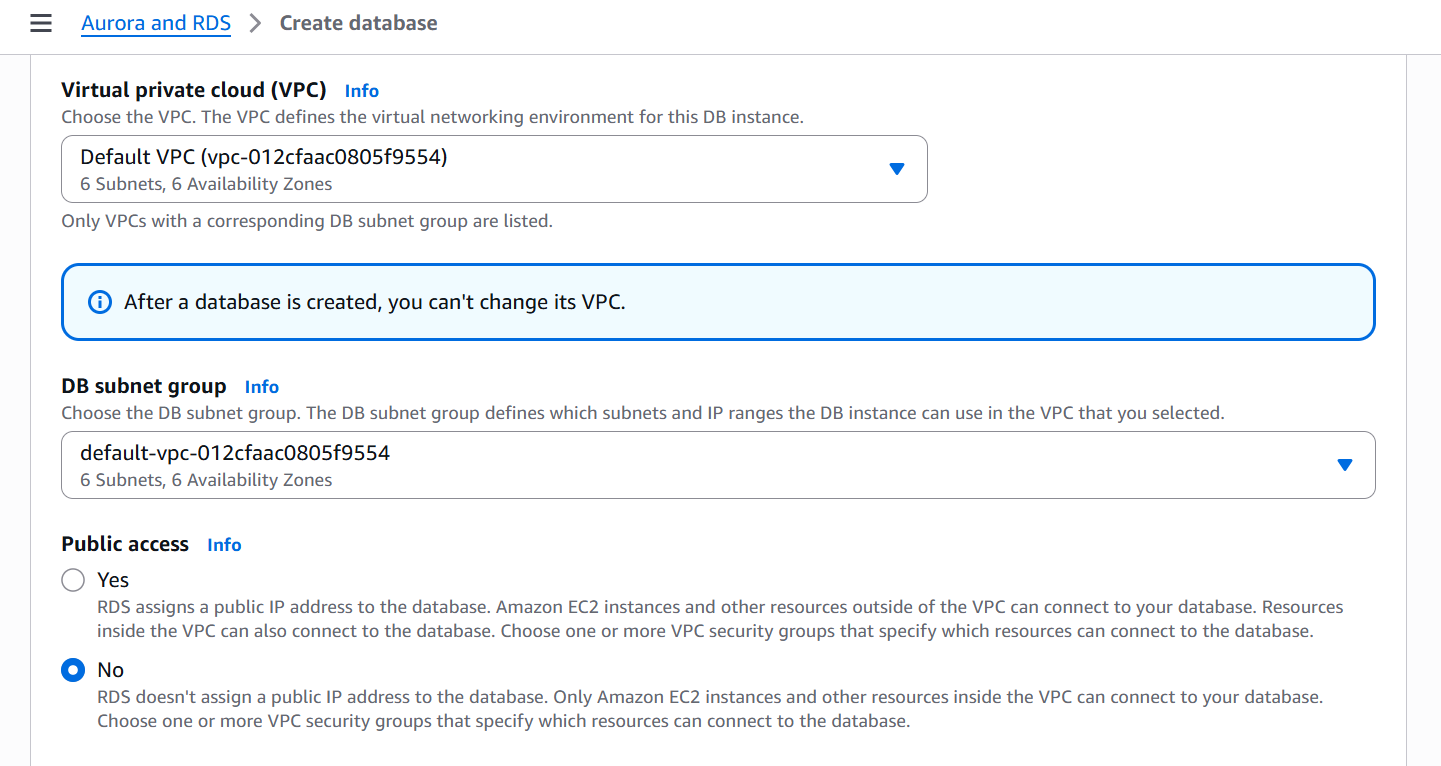


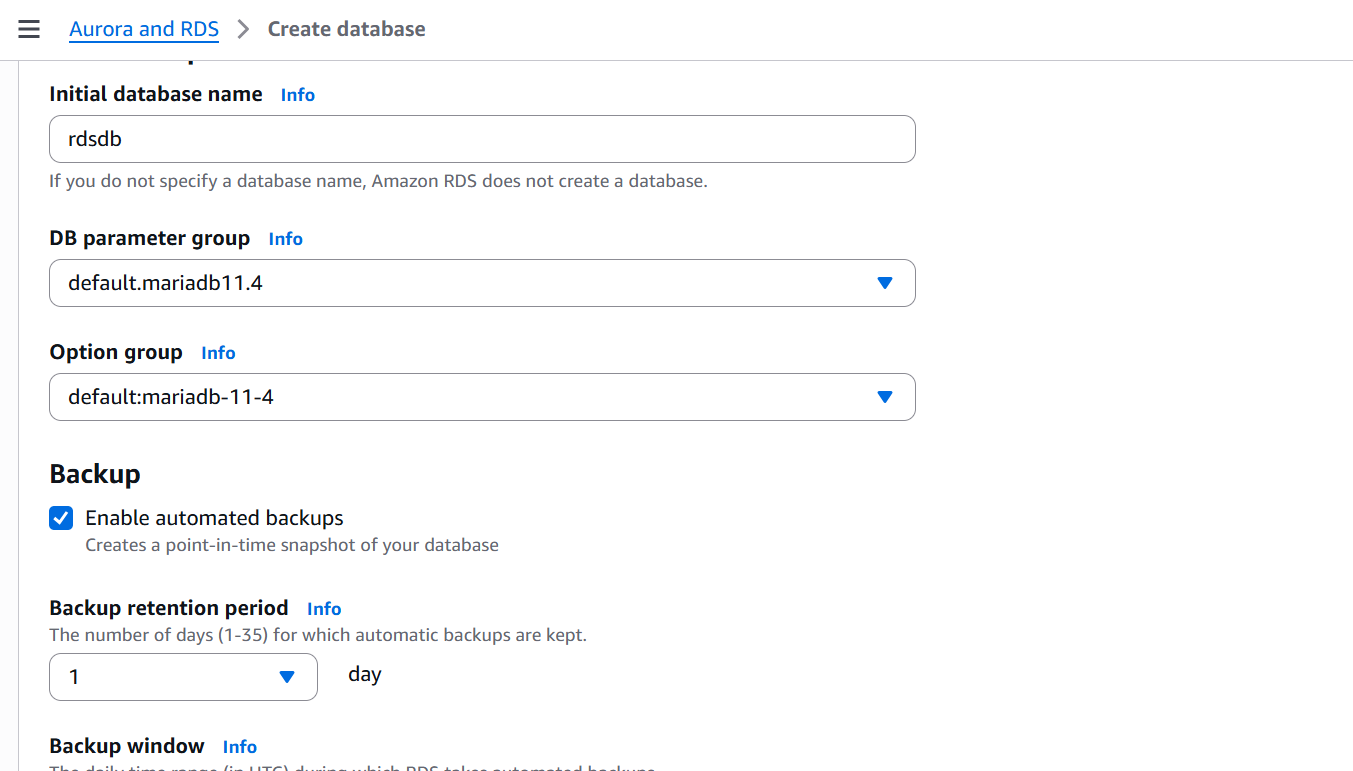


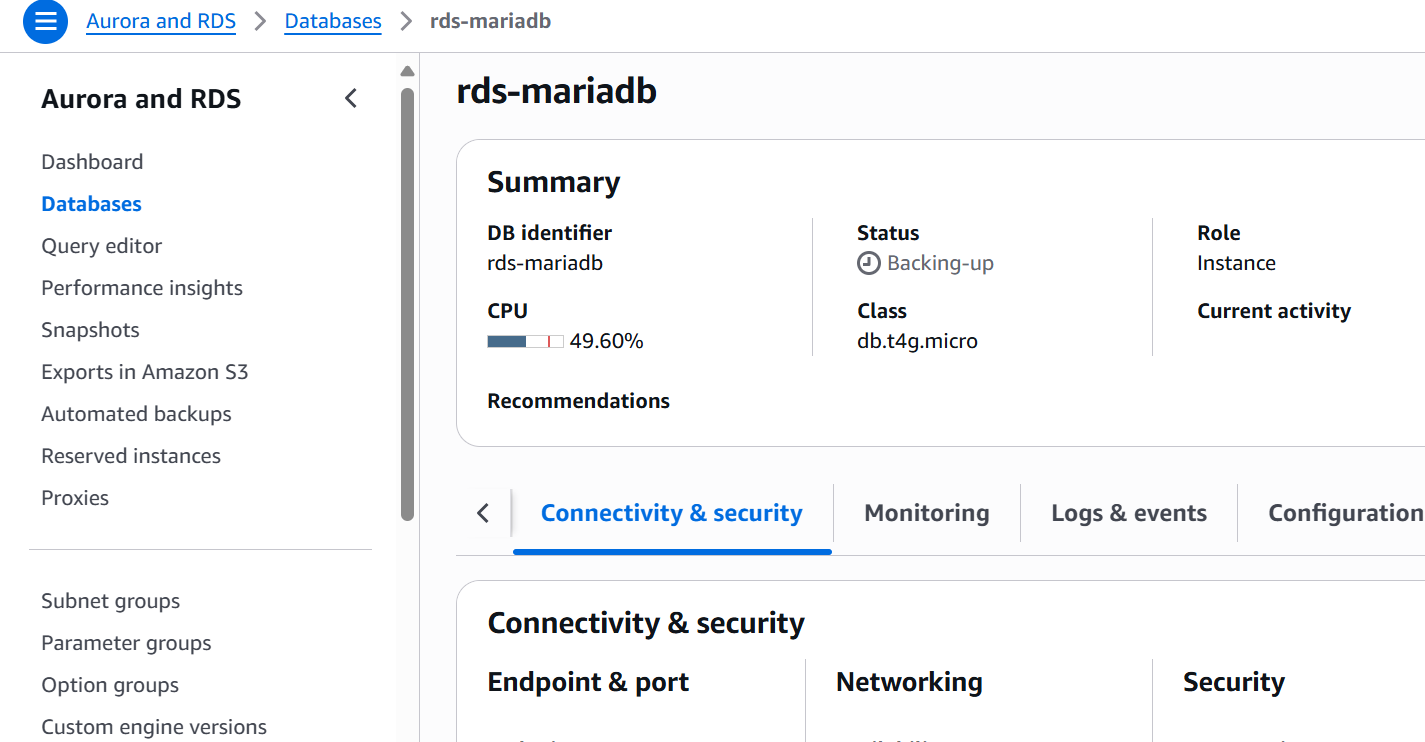


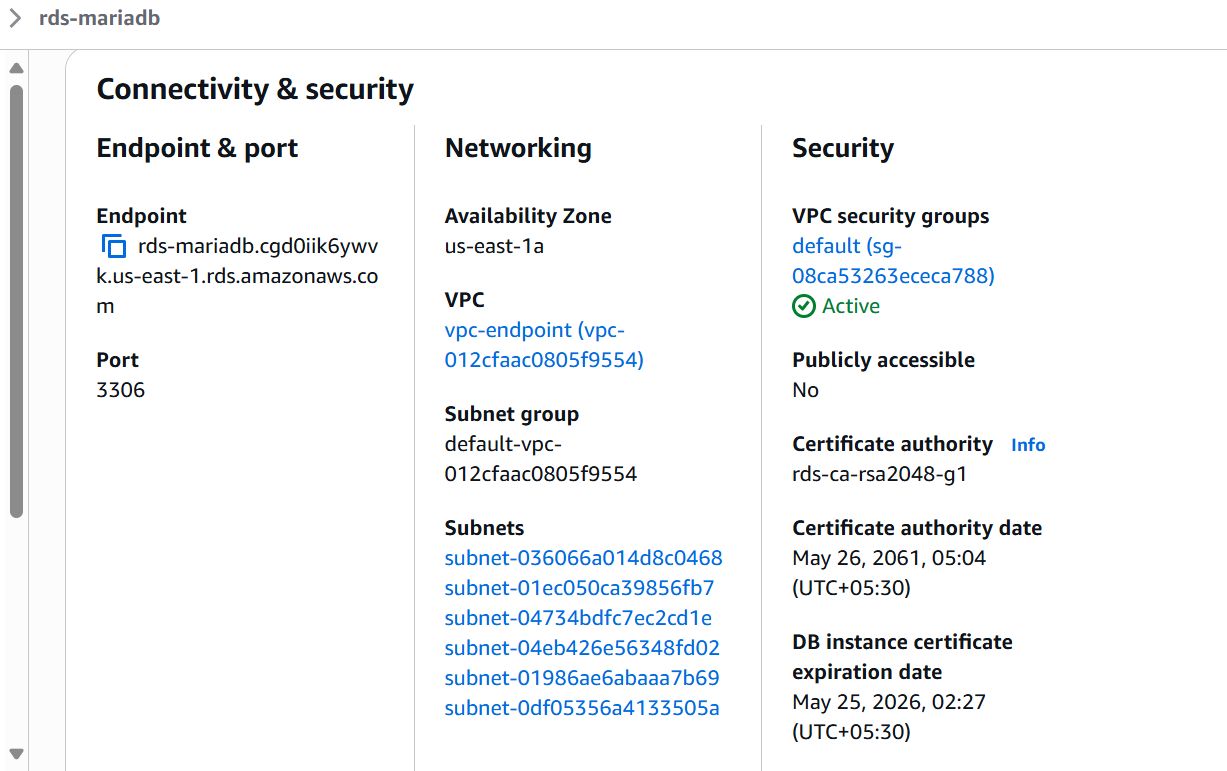












1. Migrate the DB dump that you have taken in step 1 to RDS

mysql -h rds-mariadb.cgd0iik6ywvk.us-east-1.rds.amazonaws.com -P 3306 -u -p database\_name < ec2db.sql

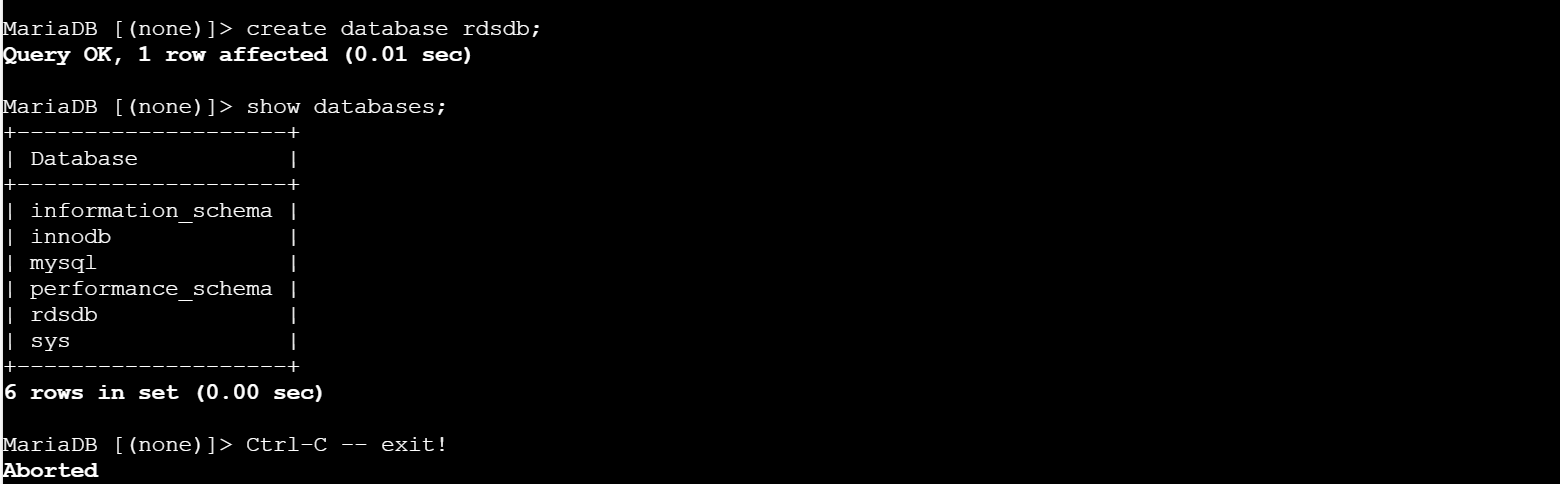
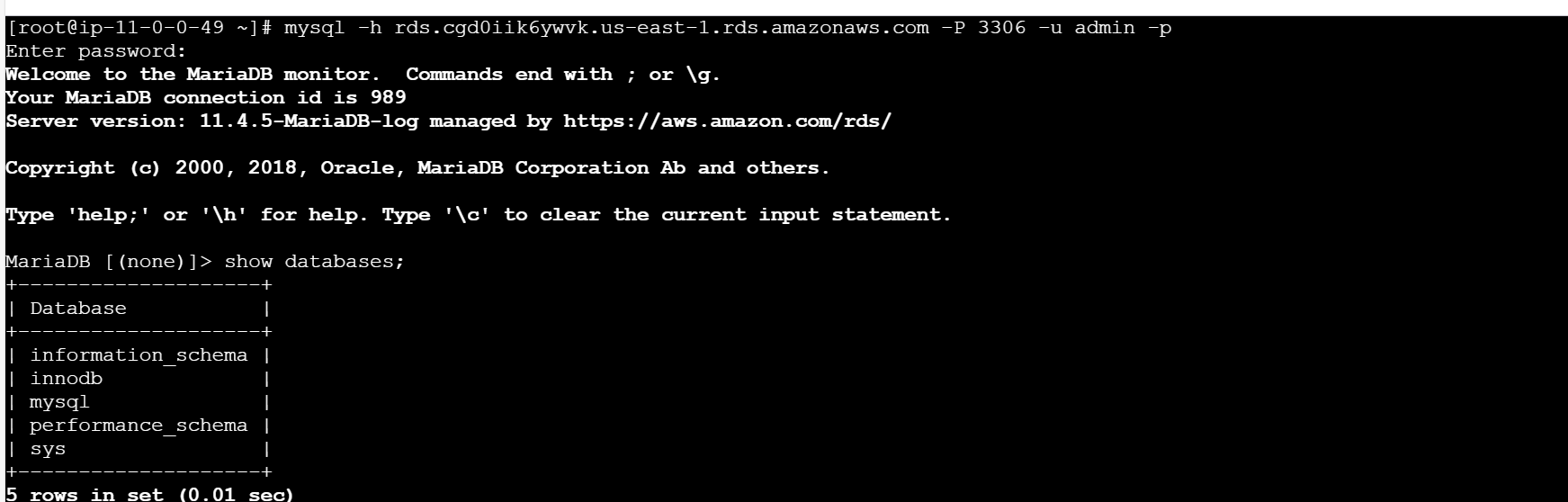
1. Connect to RDS instance

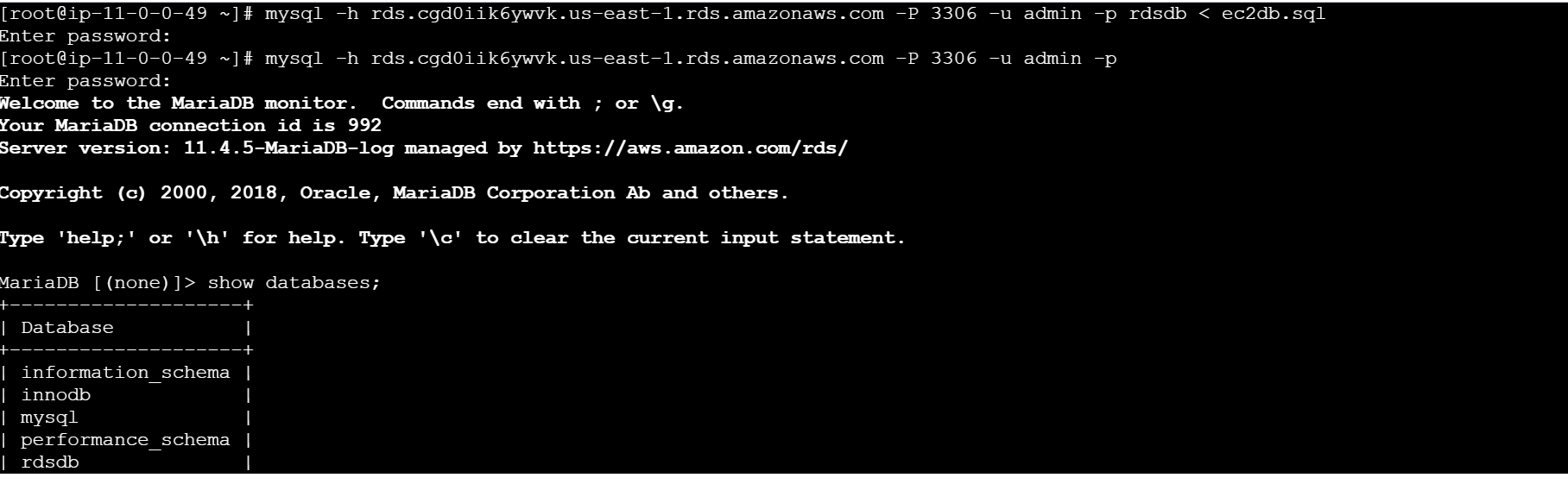
Mysql –h <rds-endpoint> -P 3306 –u admin –p

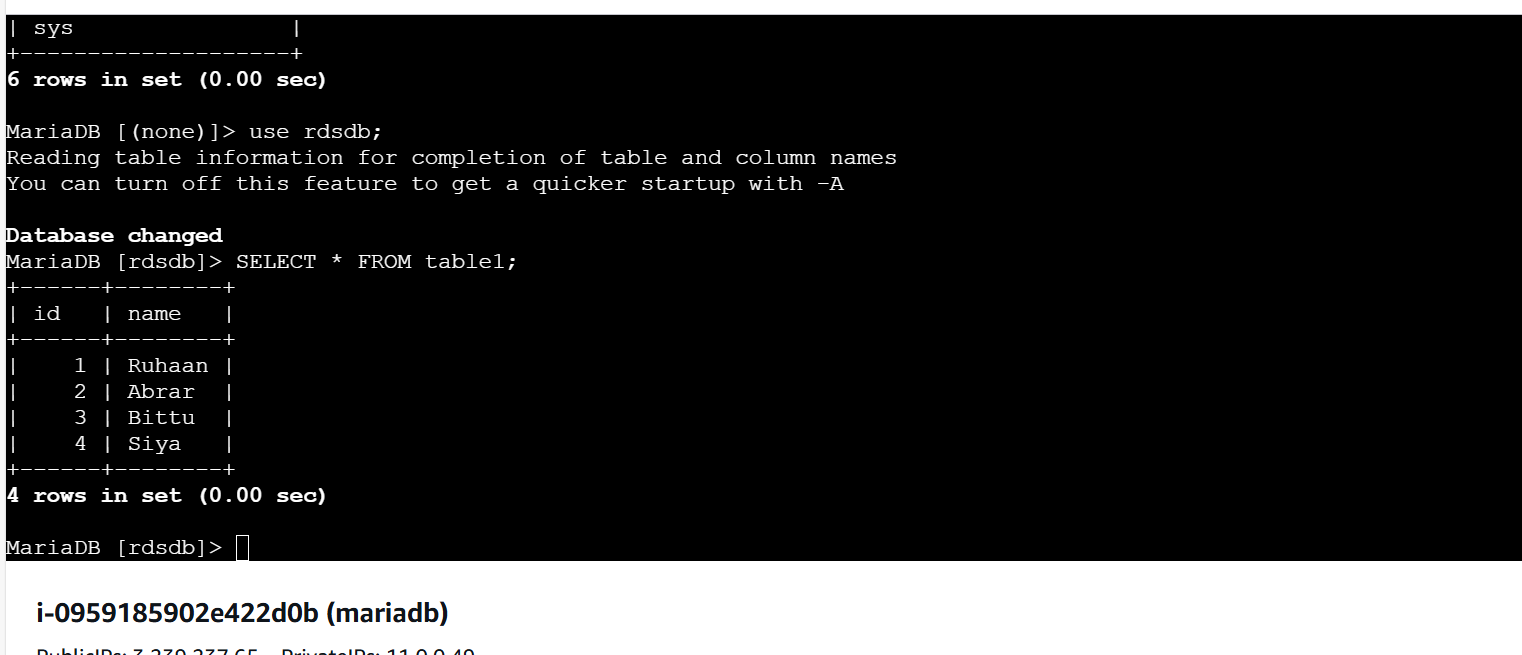
Enter Password

4. Switch to the database and verify the details

This is how we can migrate ,connect and verify the databse from EC2 to RDS







6) Install mysql db on ec2

To install MySQL on an EC2 instance running Amazon Linux 2, first the MySQL repository is added  
using the command:  
wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm  
rpm -ivh mysql57-community-release-el7-11.noarch.rpm

After adding the repository, MySQL is installed using:  
yum install -y mysql-community-server

sudo yum update -y

sudo amazon-linux-extras enable mysql8.0

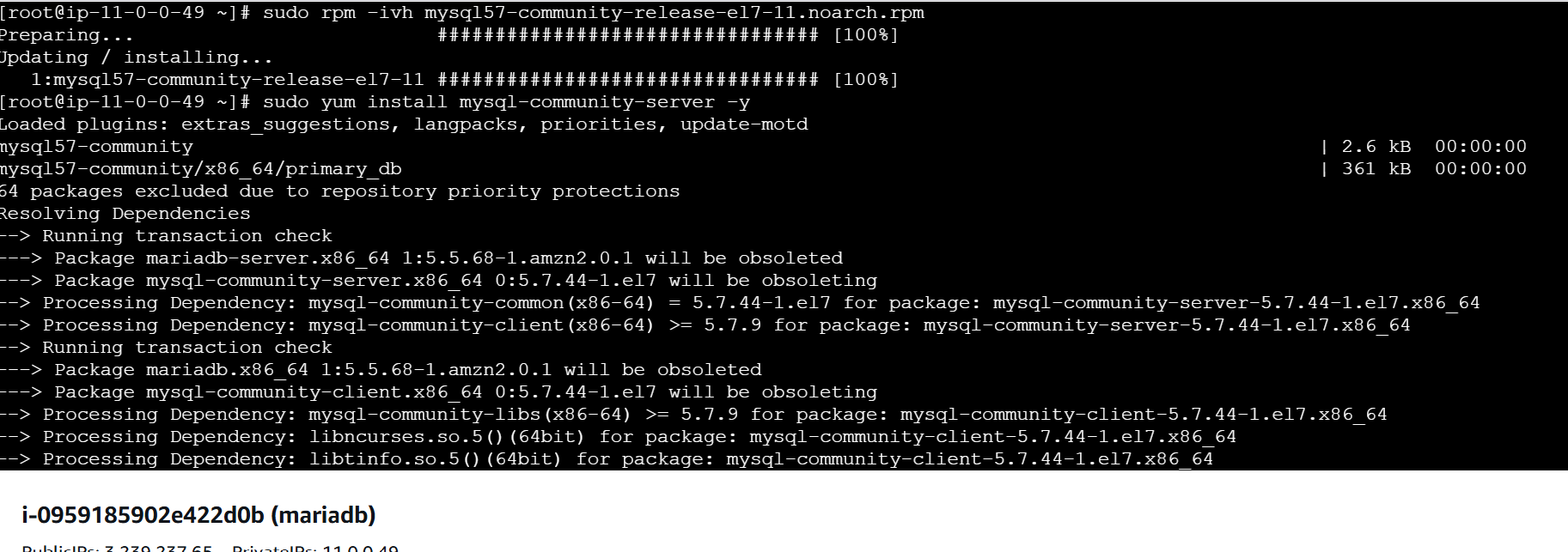
sudo yum install mysql-server -y

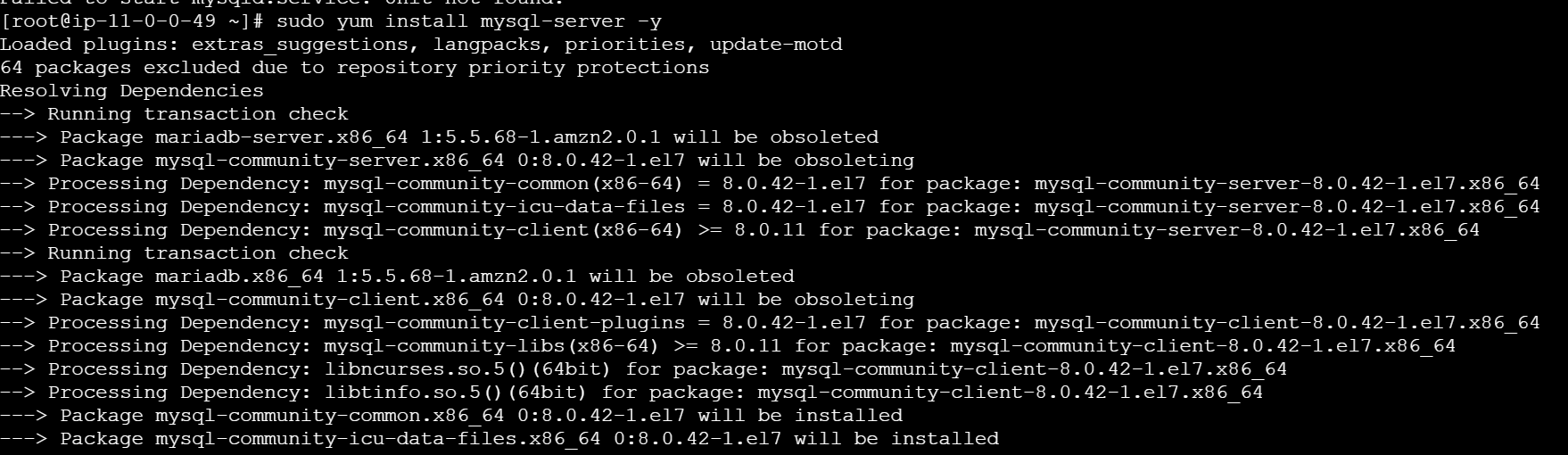
sudo systemctl start mysqld

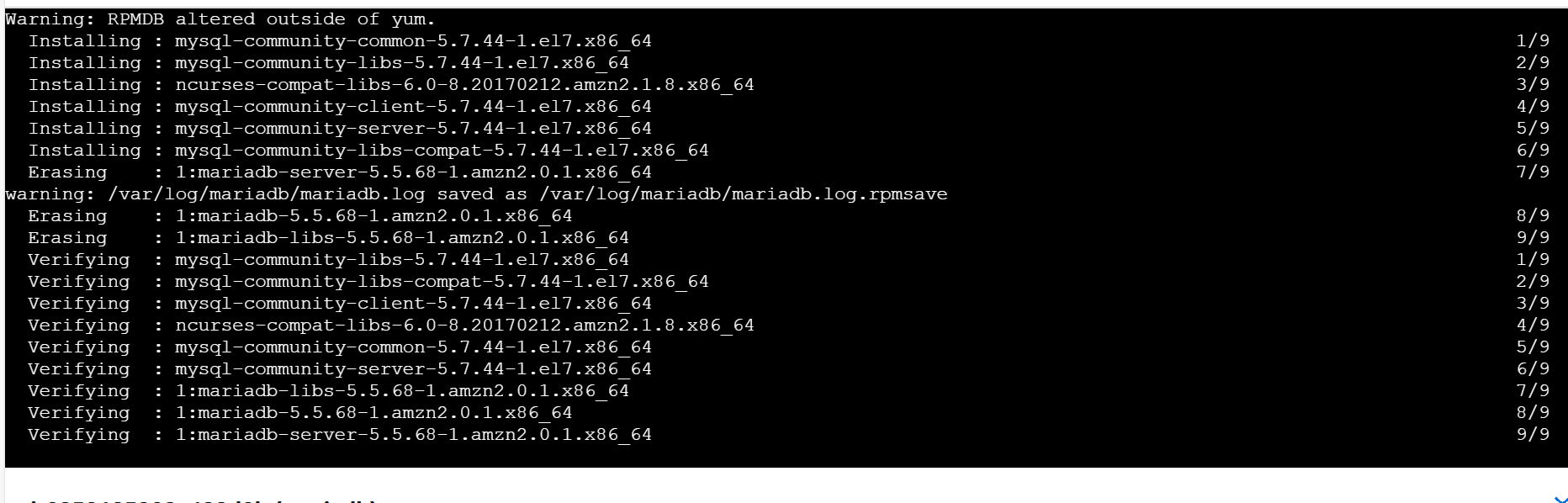
sudo systemctl enable mysqld

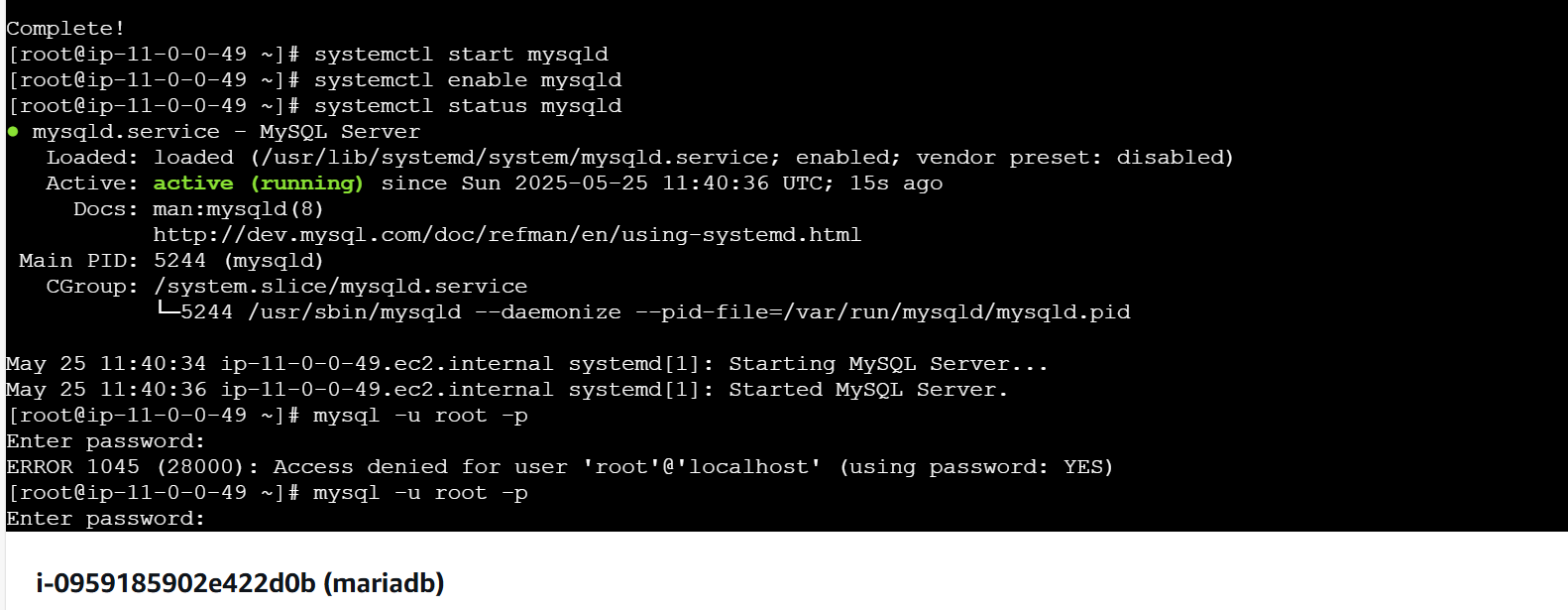
sudo mysql\_secure\_installation

mysql --version



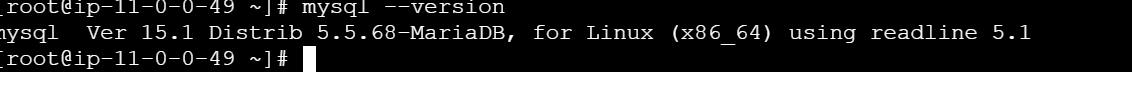












7) Launch mysql RDS image

**Objective:**

Deploy an Amazon RDS instance running MySQL, ensuring scalability, security, and optimized database performance.

**Prerequisites:**

* AWS account with RDS access.
* IAM permissions to create and manage RDS instances.
* Network setup with proper security groups and VPC configurations

**Open Amazon RDS Console**

1. Sign in to the **AWS Management Console**.
2. Navigate to **RDS** service.
3. Click **Create Database**.

**Step 2: Choose Database Engine & Edition**

1. Under **Database creation method**, select **Standard create**.
2. Choose **MySQL** as the database engine.
3. Select the **latest version** for compatibility.
4. Choose the **Edition** (Community Edition or AWS Aurora MySQL).

**Step 3: Configure Instance Settings**

1. **DB Instance Identifier** → Enter a unique name.
2. **DB Instance Class** → Choose based on workload (db.t3.micro for free-tier, db.m5.large for production).
3. **Storage Type**:
   1. General Purpose SSD (gp2) for balanced performance.
   2. Provisioned IOPS (io1) for high-performance workloads.
   3. Magnetic (standard) for low-cost deployments.
4. **Allocate Storage** → Enter required size (e.g., 20GB).

**Step 4: Configure High Availability (Multi-AZ)**

1. Under **Availability & Durability**, select **Multi-AZ deployment** for automatic failover.
2. If required, enable **Replica** for scaling read operations.

**Step 5: Set Up Database Authentication**

1. **Master Username** → Define an admin username.
2. **Master Password** → Set a strong password.
3. **Confirm Password** → Retype for verification.

**Step 6: Configure Network & Security**

1. **VPC** → Select an existing or create a new VPC.
2. **Subnet Group** → Choose an appropriate subnet for availability.
3. **Public Access** → Enable or disable based on security needs.
4. **Security Groups**:

* Allow inbound traffic for MySQL (TCP 3306).
* Restrict access based on application requirements.

**Step 7: Enable Monitoring & Backups**

1. **CloudWatch Enhanced Monitoring** → Enable for performance tracking.
2. **Backup Retention Period** → Set retention days (e.g., 7 days).
3. **Encryption** → Enable encryption for data security.

**Step 8: Create and Launch Database**

1. Review all settings.
2. Click **Create Database**.
3. Wait for instance initialization (available status).

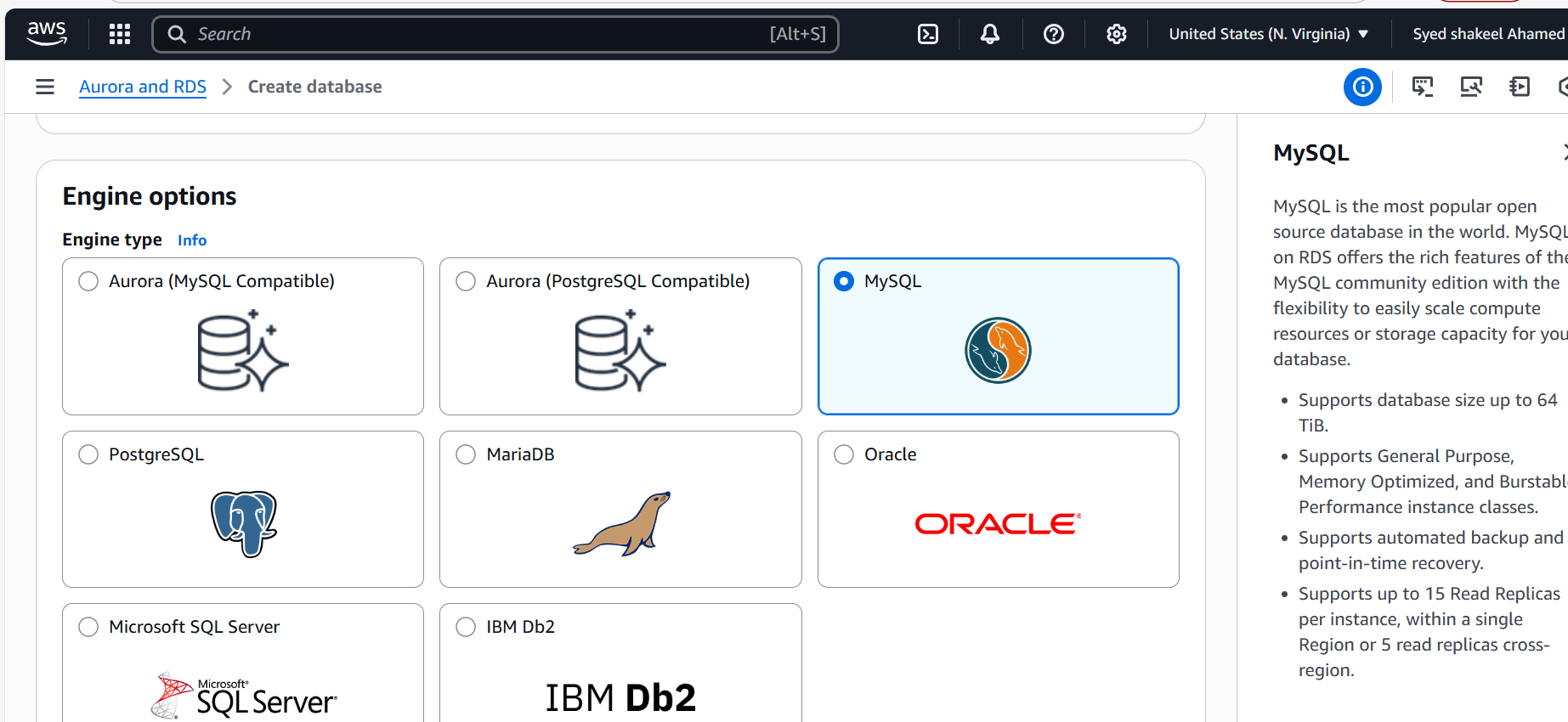
**Step 9: Connect to MySQL RDS**

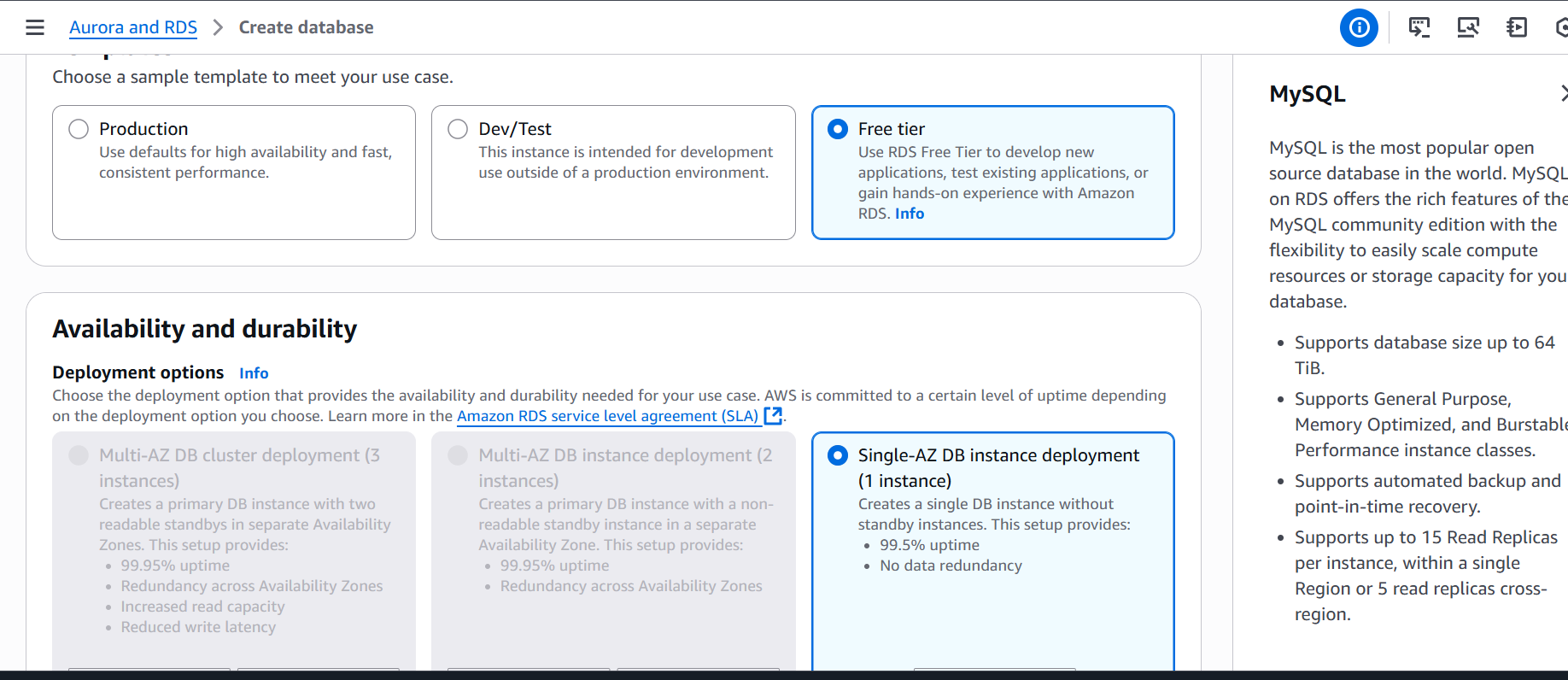
1. Retrieve endpoint from **RDS Console** → **Connectivity & Security**.
2. Use MySQL client:

mysql -h <RDS\_ENDPOINT> -u <USERNAME> -p

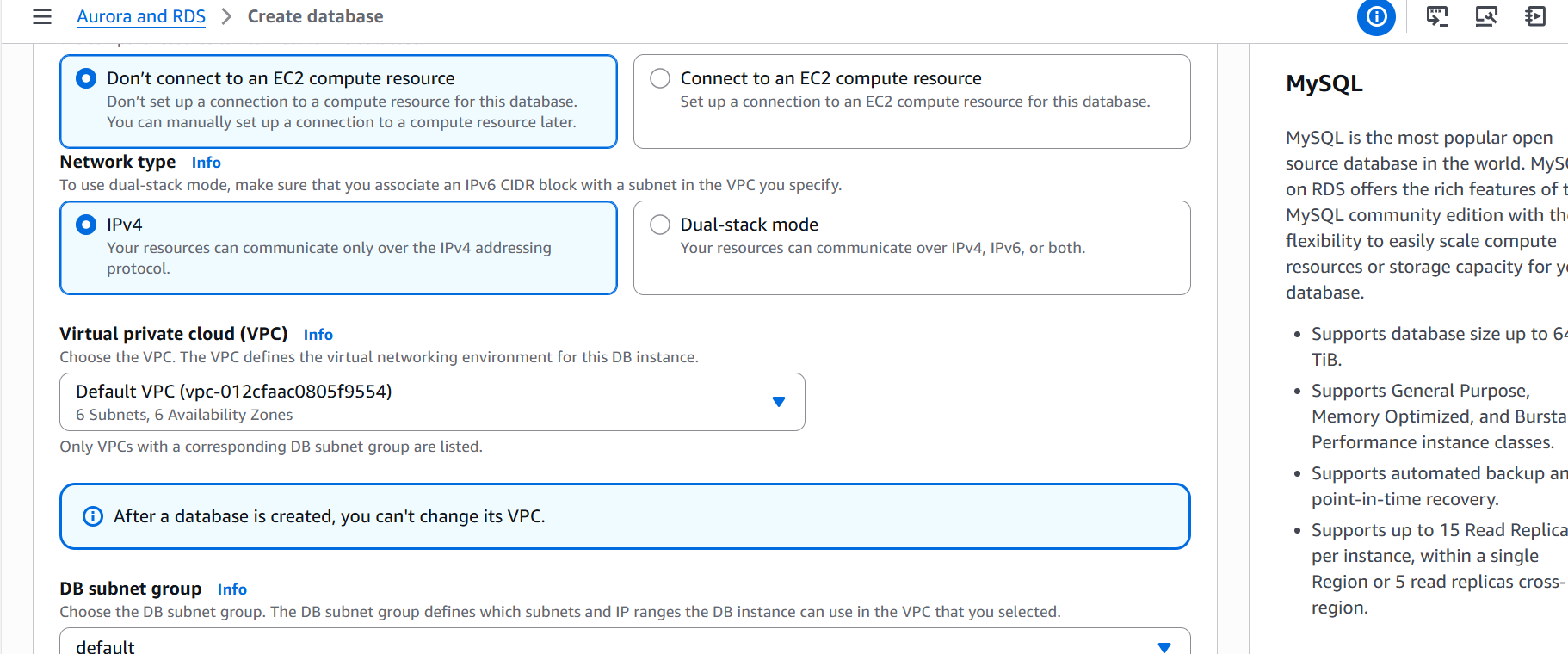
3. Verify connection and run:

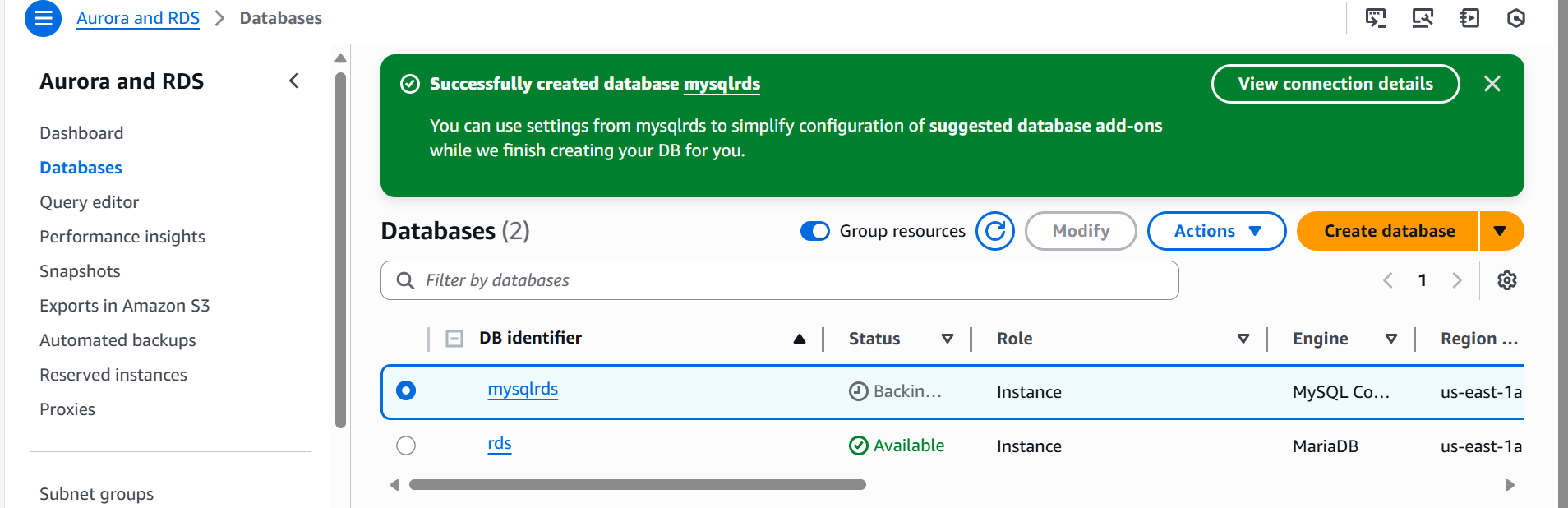
SHOW DATABASES;

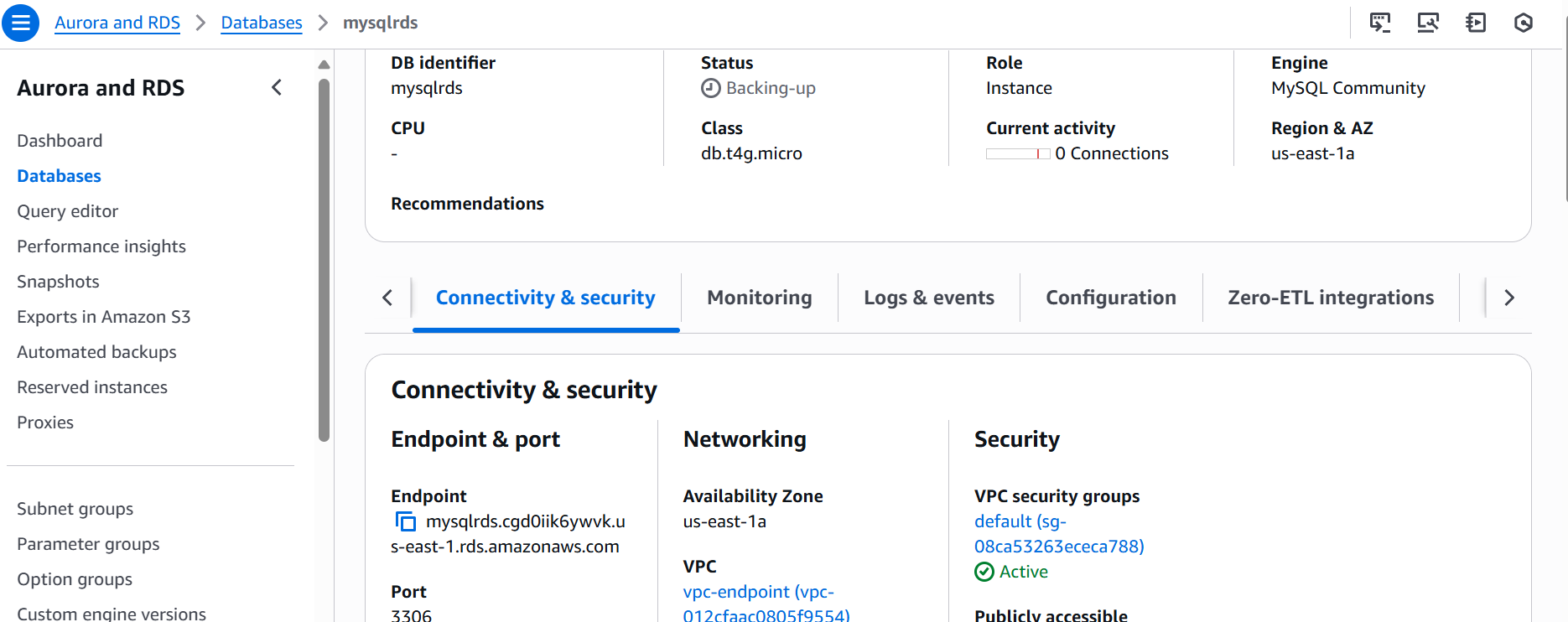


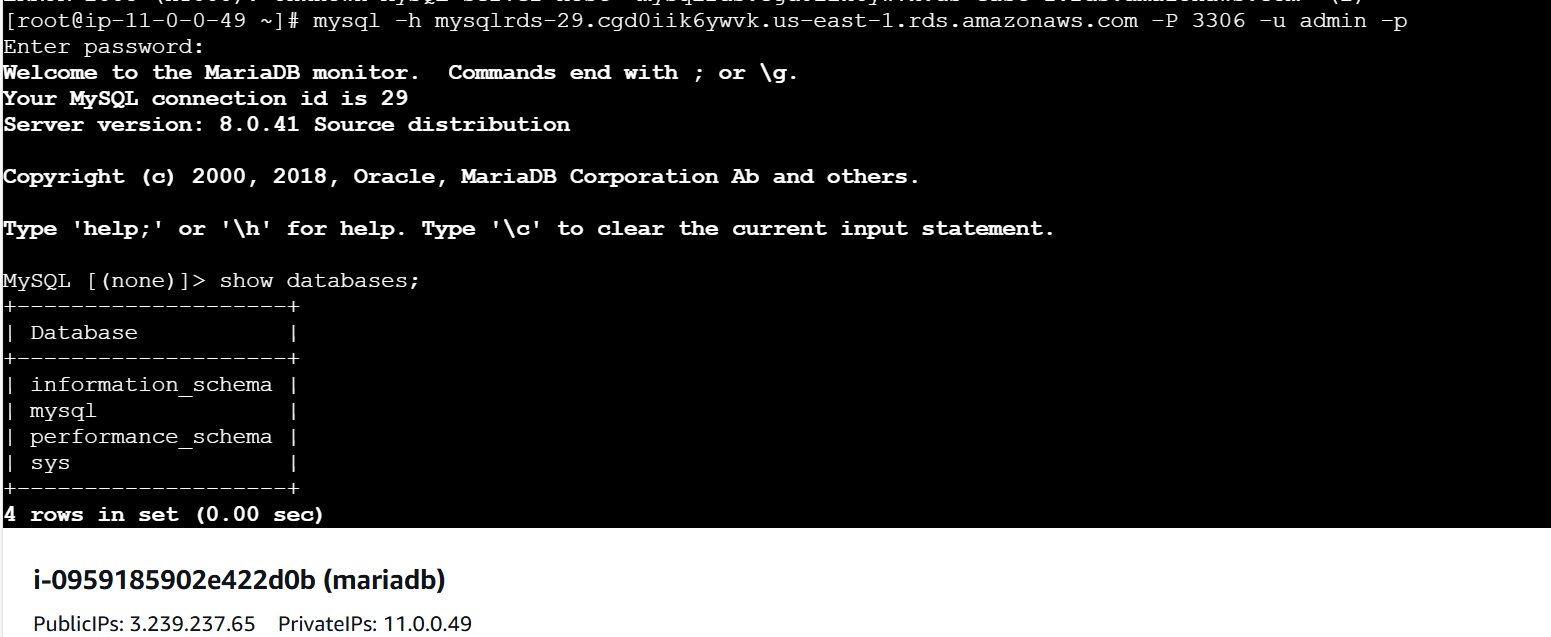


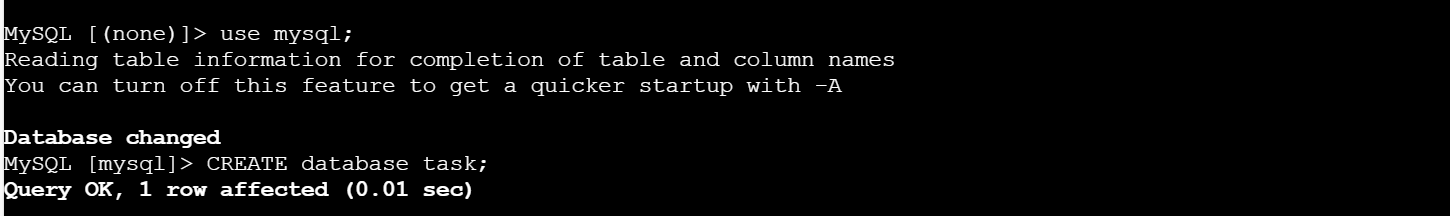












8) Configure multi-AZ

**Configuring Multi-AZ Deployment for Amazon RDS**

**Objective:**

Set up a Multi-AZ deployment for an Amazon RDS database to improve high availability and failover capabilities.

**Prerequisites:**

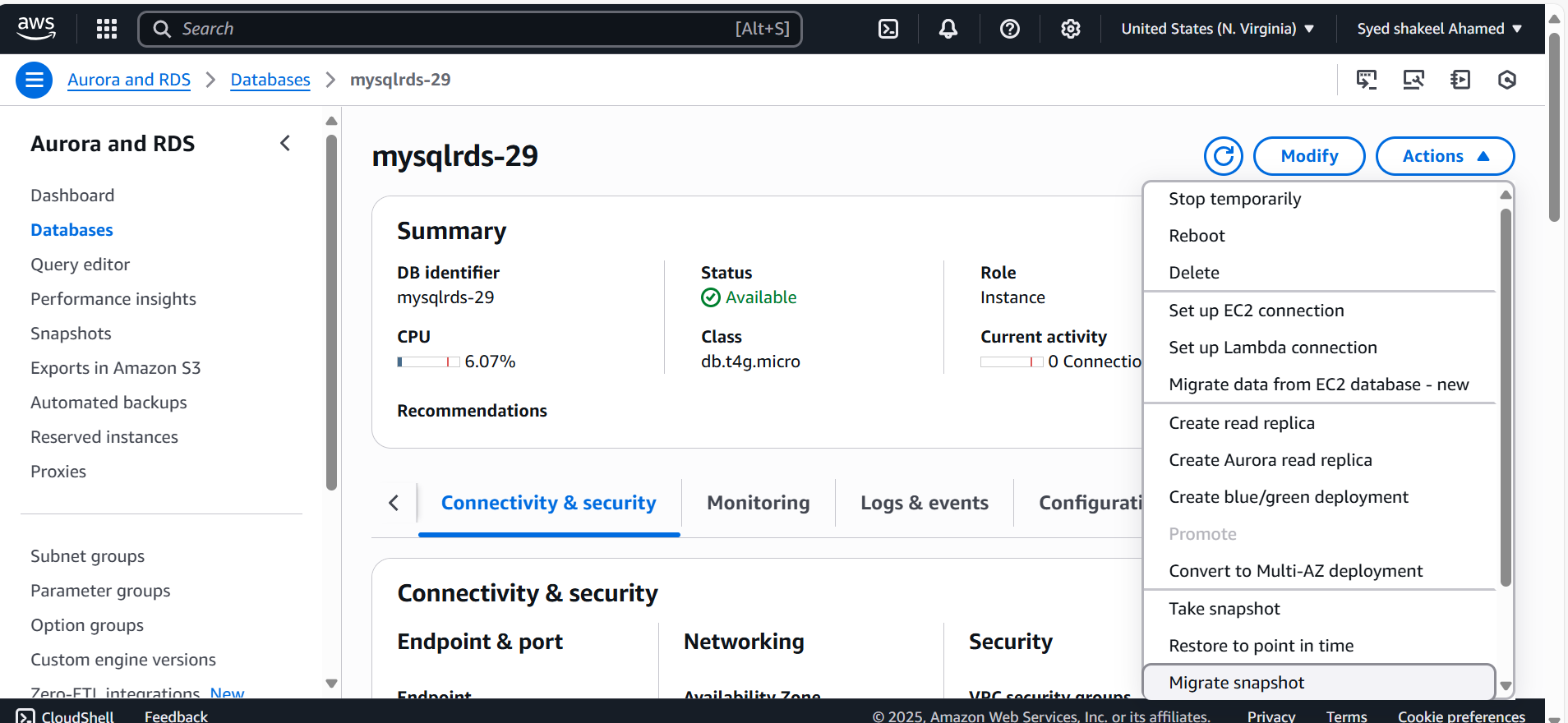
* An existing Amazon RDS instance.
* The database engine must support Multi-AZ (e.g., MySQL, PostgreSQL, MariaDB, or SQL Server).
* Ensure adequate IAM permissions to modify RDS settings.

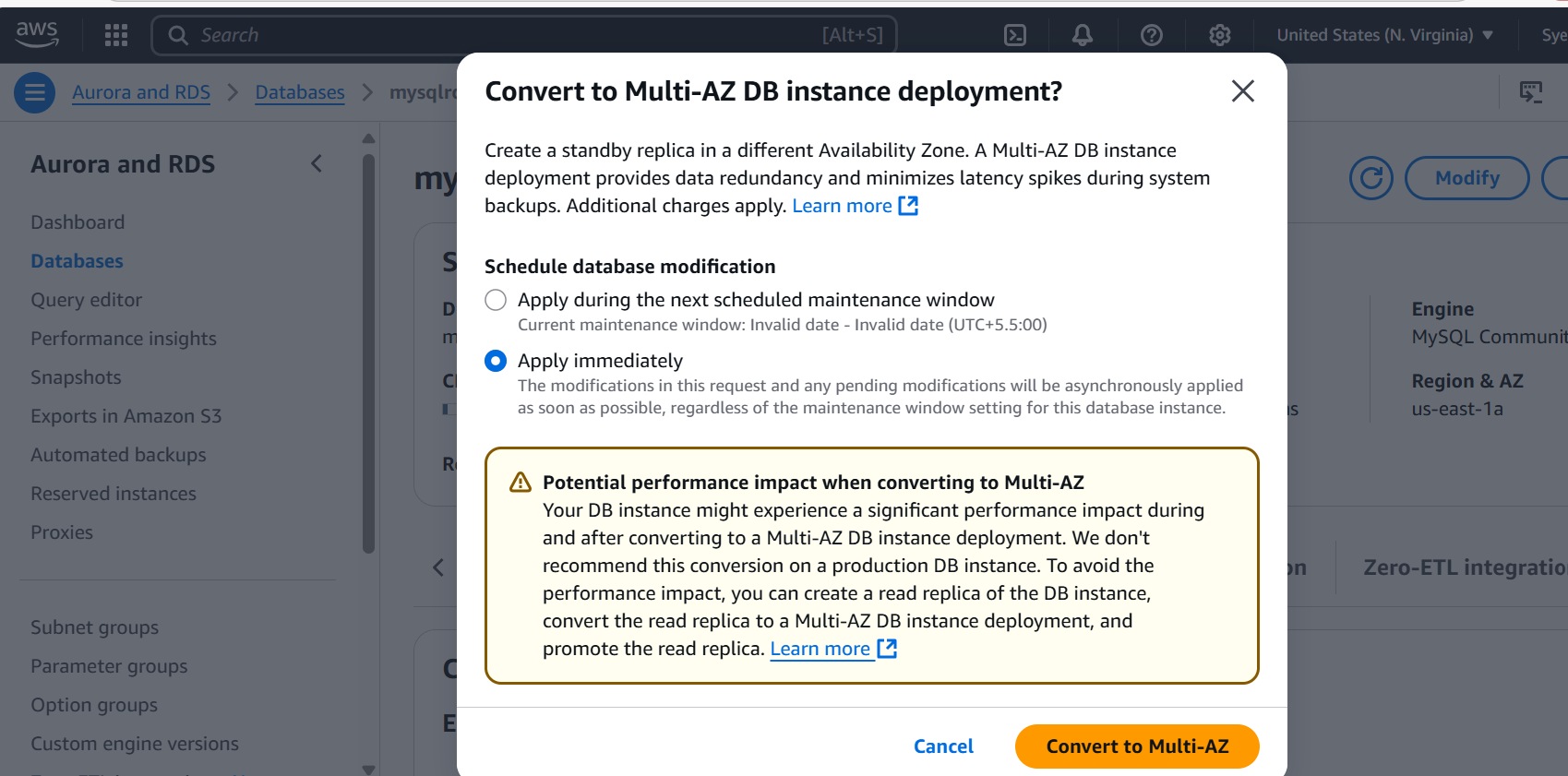
**Step 1: Verify Multi-AZ Support for Your Instance**

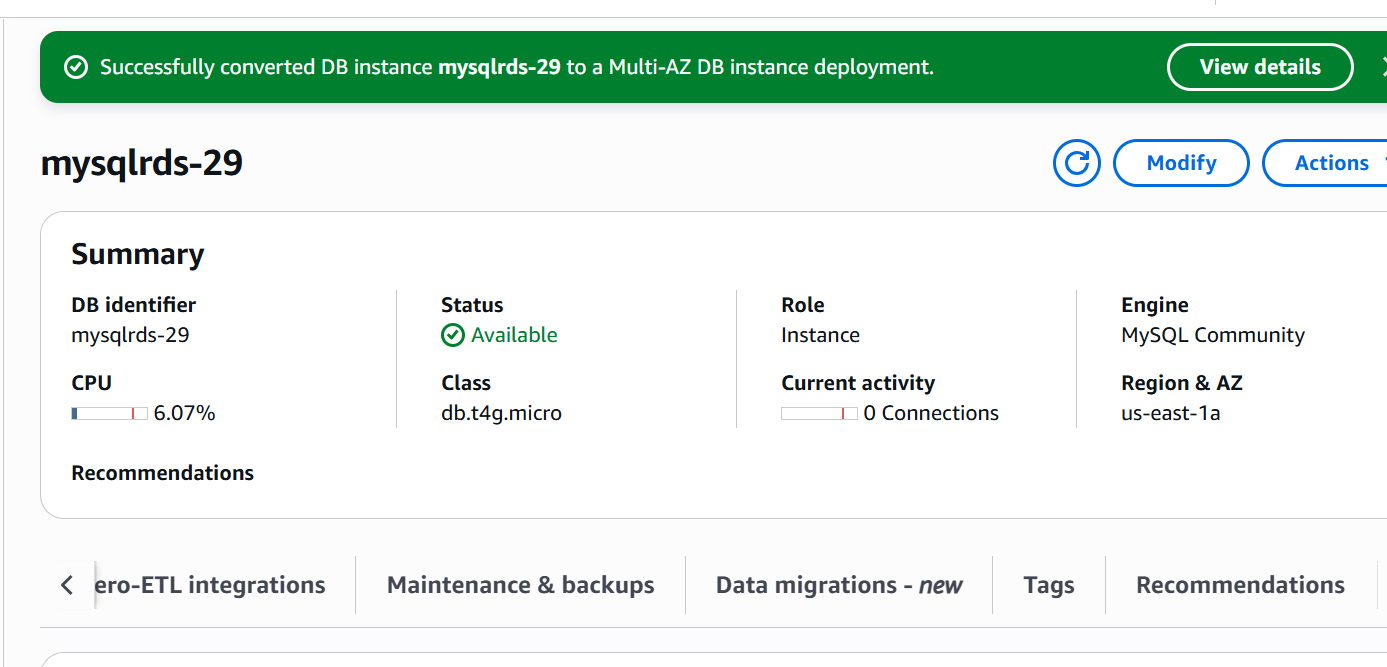
1. Go to **AWS Management Console** → **RDS**.
2. Select the **existing RDS instance**.
3. Under **Configuration**, check the **Multi-AZ deployment** setting.
4. If Multi-AZ is disabled, proceed with enabling it.

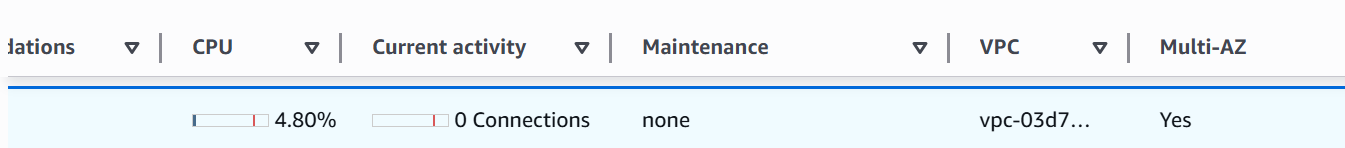
**Enable Multi-AZ Deployment (For an Existing Instance)**

1. In the **RDS Console**, select the database instance.
2. Click **Modify** → Scroll to the **Availability & Durability** section.
3. Under **Multi-AZ deployment**, choose **Create a standby instance**.
4. Click **Continue** and review changes.
5. Select **Apply Immediately** (for immediate updates) or **During the next maintenance window** (recommended).
6. Click **Modify DB Instance**.









9) Take Backup of db and restore the DB

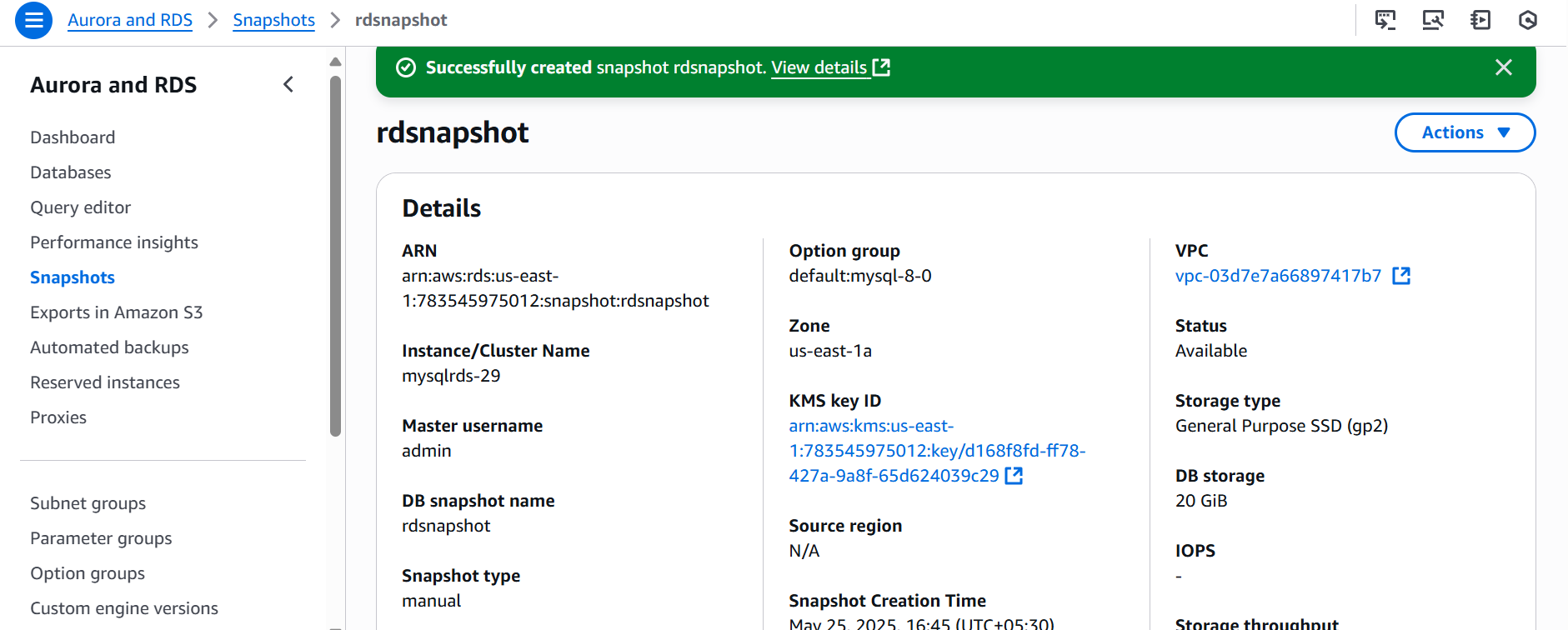
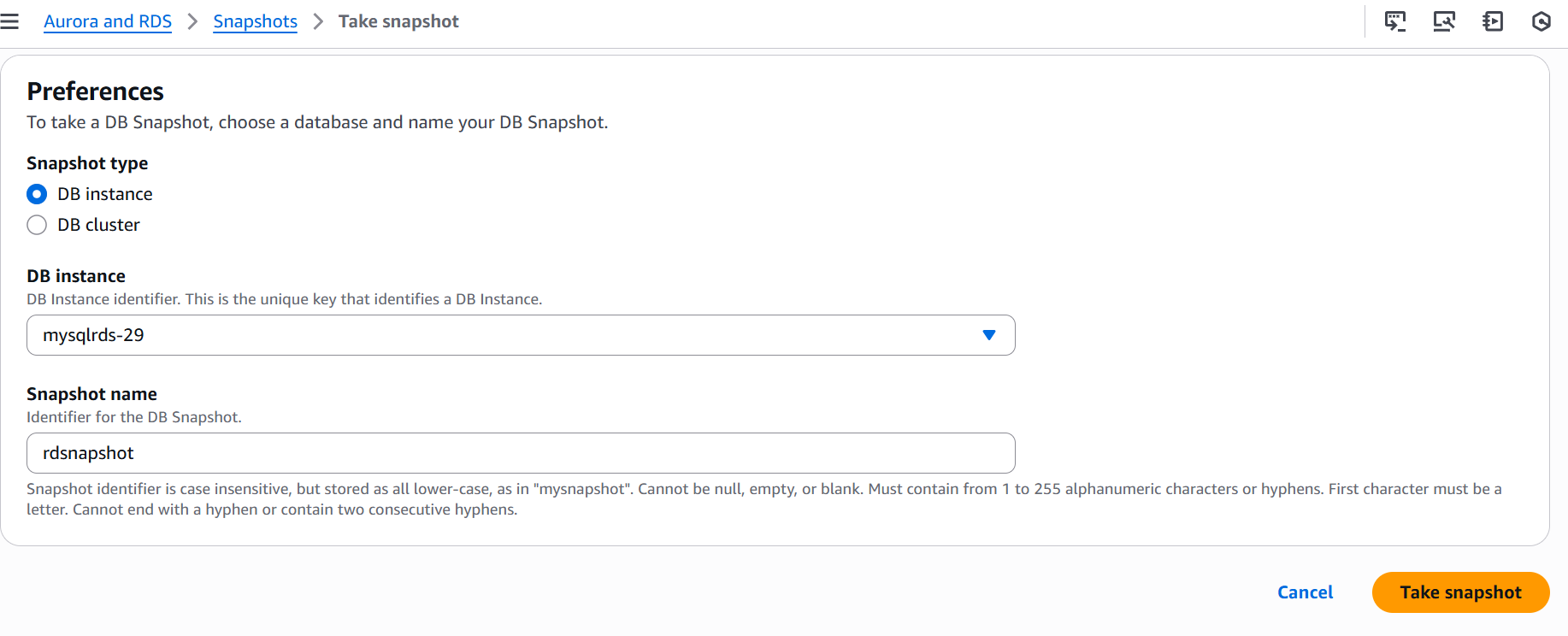
**1️⃣ Take a Manual Backup (Snapshot)**

1. Open the **AWS RDS Console**.
2. Select your **RDS instance**.
3. Click **"Actions" → "Take Snapshot"**.
4. Enter a **snapshot name** and confirm.

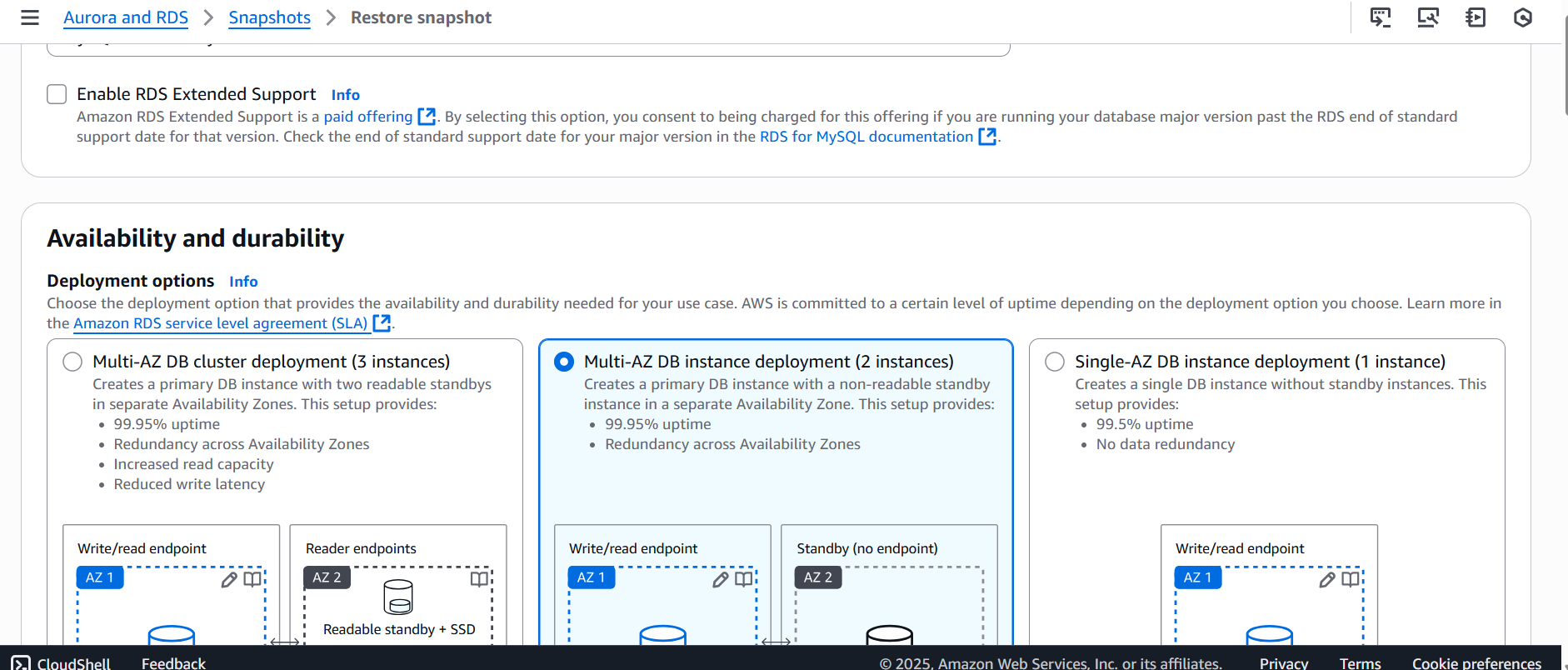
**🔹 2️⃣ Restore from a Snapshot**

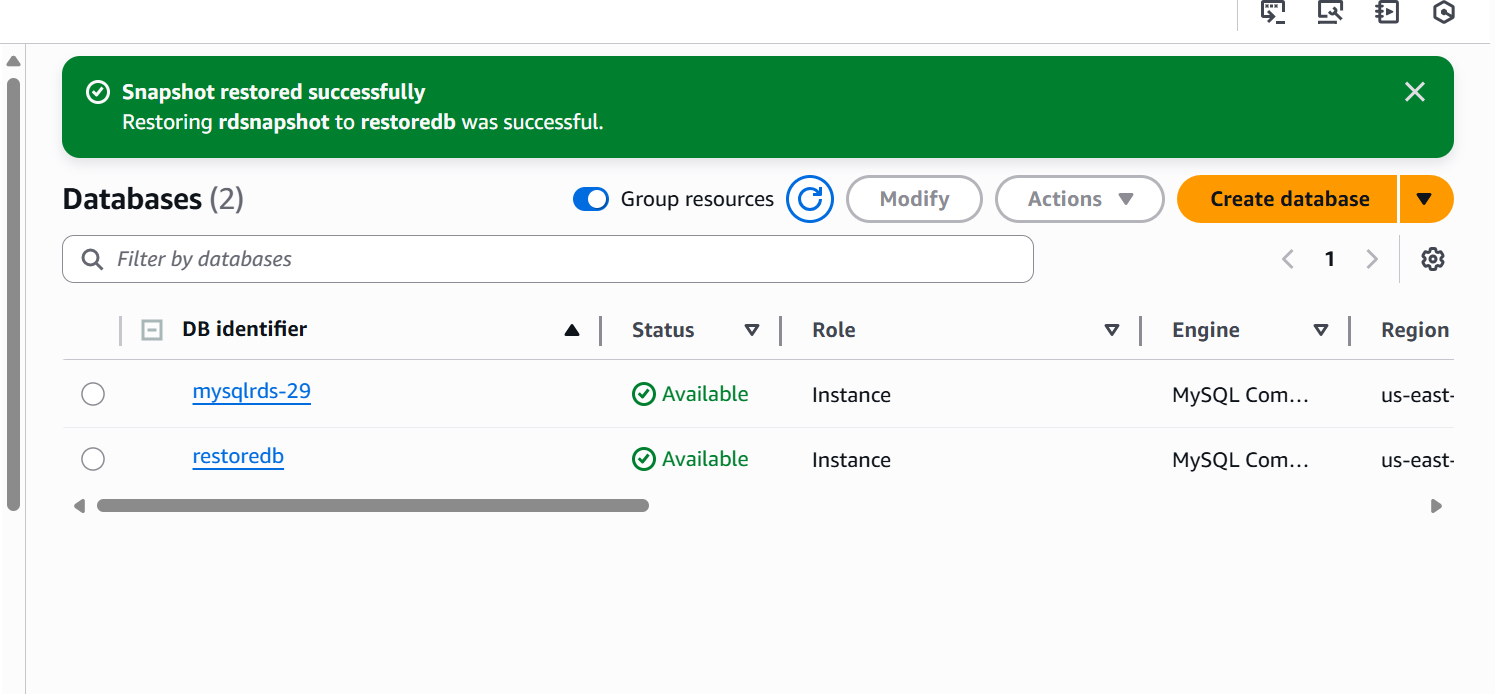
1. Go to **RDS Console → Snapshots**.
2. Select the snapshot you want to restore.
3. Click **"Restore Snapshot"**.
4. Enter a **new DB instance identifier**.
5. Configure instance settings and launch.

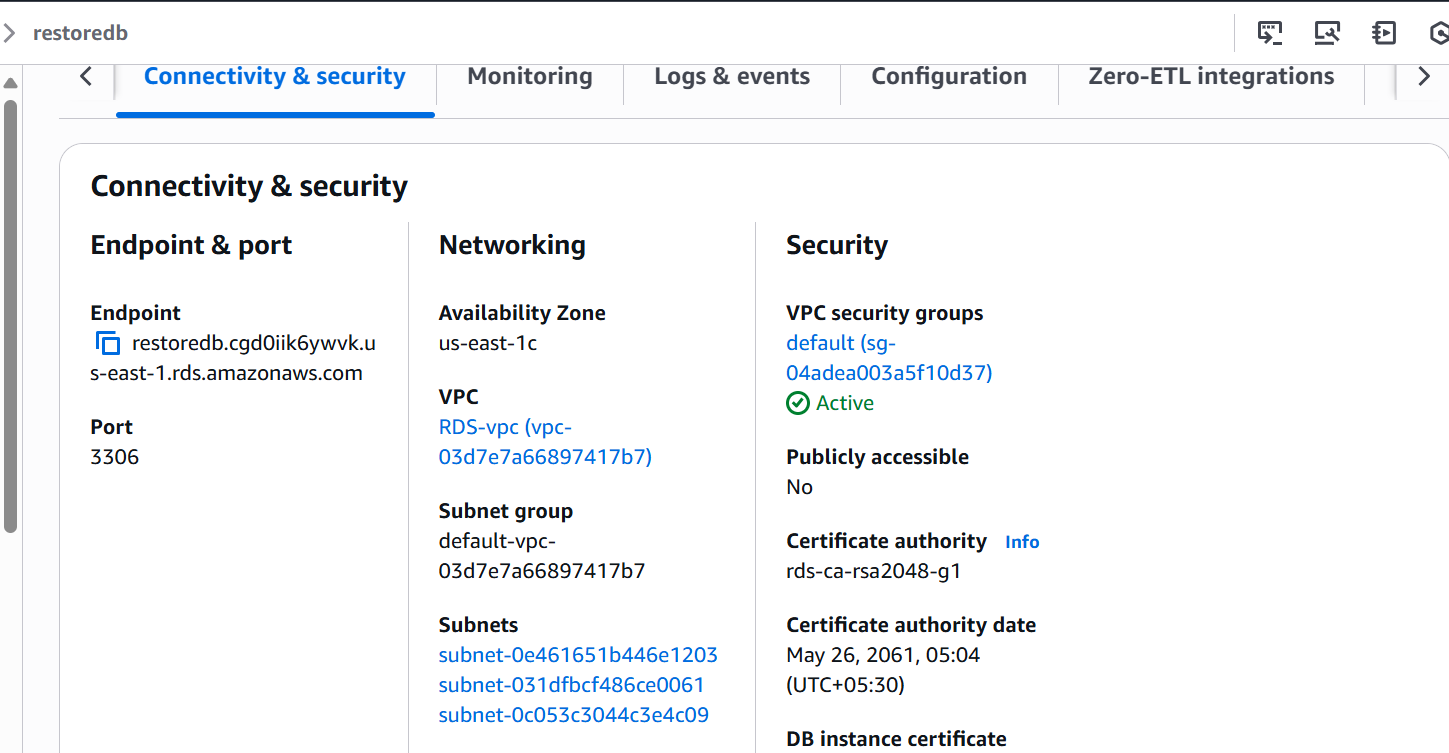




Restore db







10) Create ReadReplica

Set up a Read Replica for an Amazon RDS instance to improve database scalability and read performance.

**Prerequisites:**

* An existing Amazon RDS instance.
* The instance should be running MySQL, PostgreSQL, MariaDB, or SQL Server (Oracle Read Replicas are not supported).
* The instance must have automatic backups enabled.

**Enable Backups on the Source Database**

1. Go to the **AWS Management Console** → **RDS** service.
2. Select the **source database instance**.
3. Verify that automated backups are enabled (Backup Retention Period > 0).
4. If backups are disabled, modify the instance to enable them.

**Create a Read Replica**

1. In the **RDS Console**, select the **source instance**.
2. Click **Actions** → **Create Read Replica**.
3. Configure the Read Replica:
   1. **DB Instance Identifier:** Enter a unique name.
   2. **Instance Class:** Choose an appropriate instance type.
   3. **Storage Type:** Match the primary instance or optimize for read-heavy workload.
   4. **Multi-AZ Deployment:** Choose **No** (Read Replicas don’t support Multi-AZ by default).
   5. **Public Accessibility:** Define access level based on requirements.
   6. **Replication Option:** Enable encryption if needed.
   7. **Network & Security:** Ensure security groups allow traffic from relevant sources.
   8. **Monitoring:** Enable Enhanced Monitoring (optional).
4. Click **Create Read Replica**.

**Monitor the Replication Process**

1. Navigate to **RDS Console** → **Read Replica instance**.
2. Check the **Replication Status** under instance properties.
3. Ensure the replica is "Replicating" without errors.

**Redirect Read Queries to the Read Replica**

1. Modify application/database settings:

* Use **endpoint** of the Read Replica for read-only queries.

