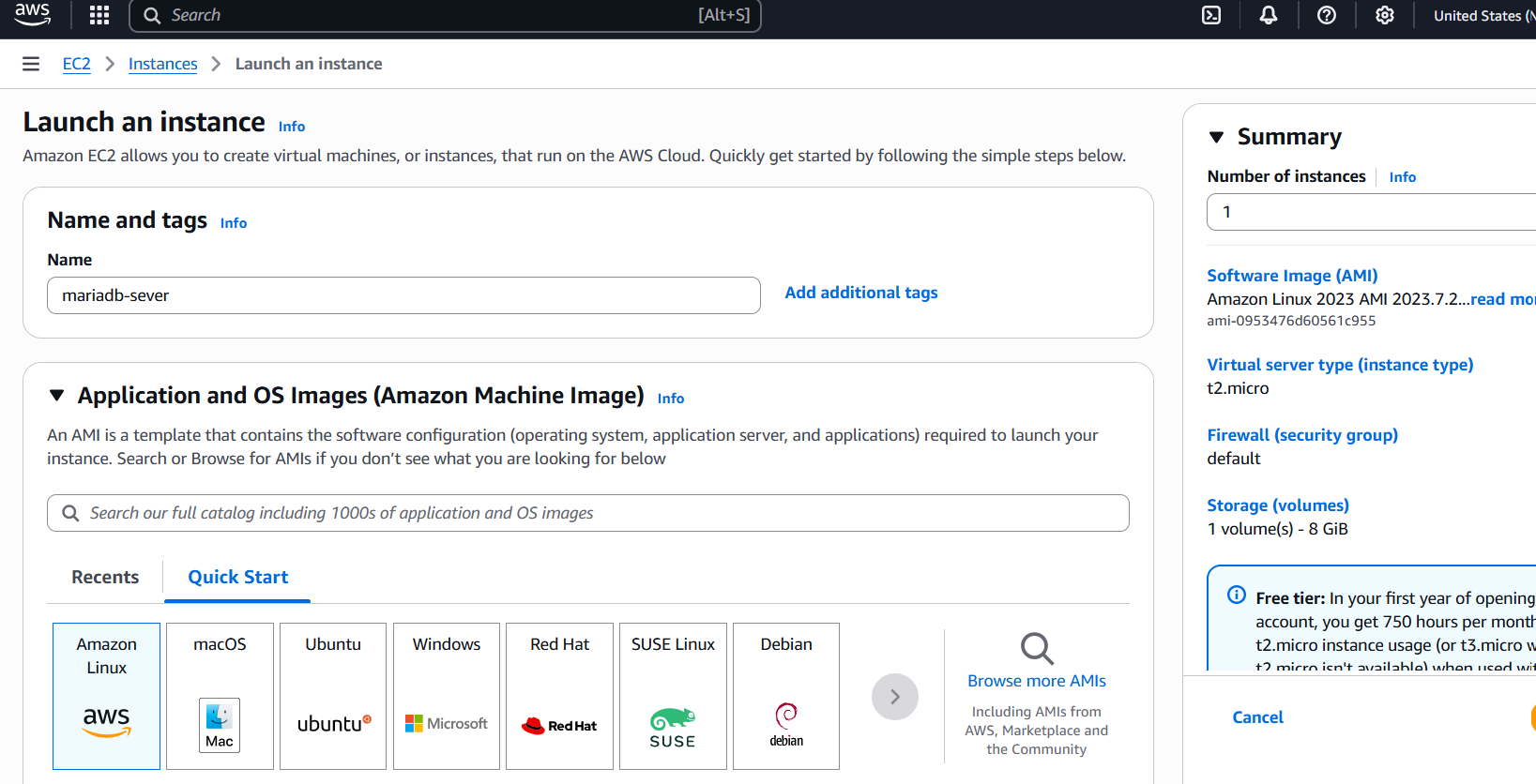
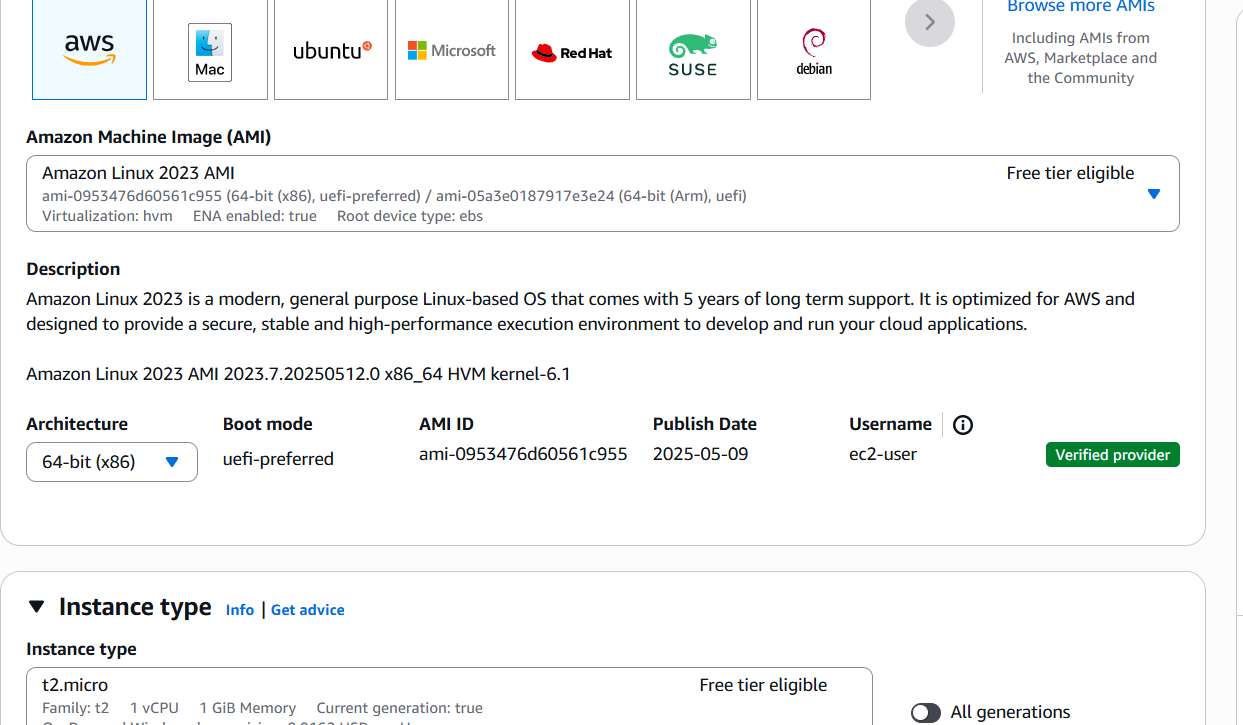
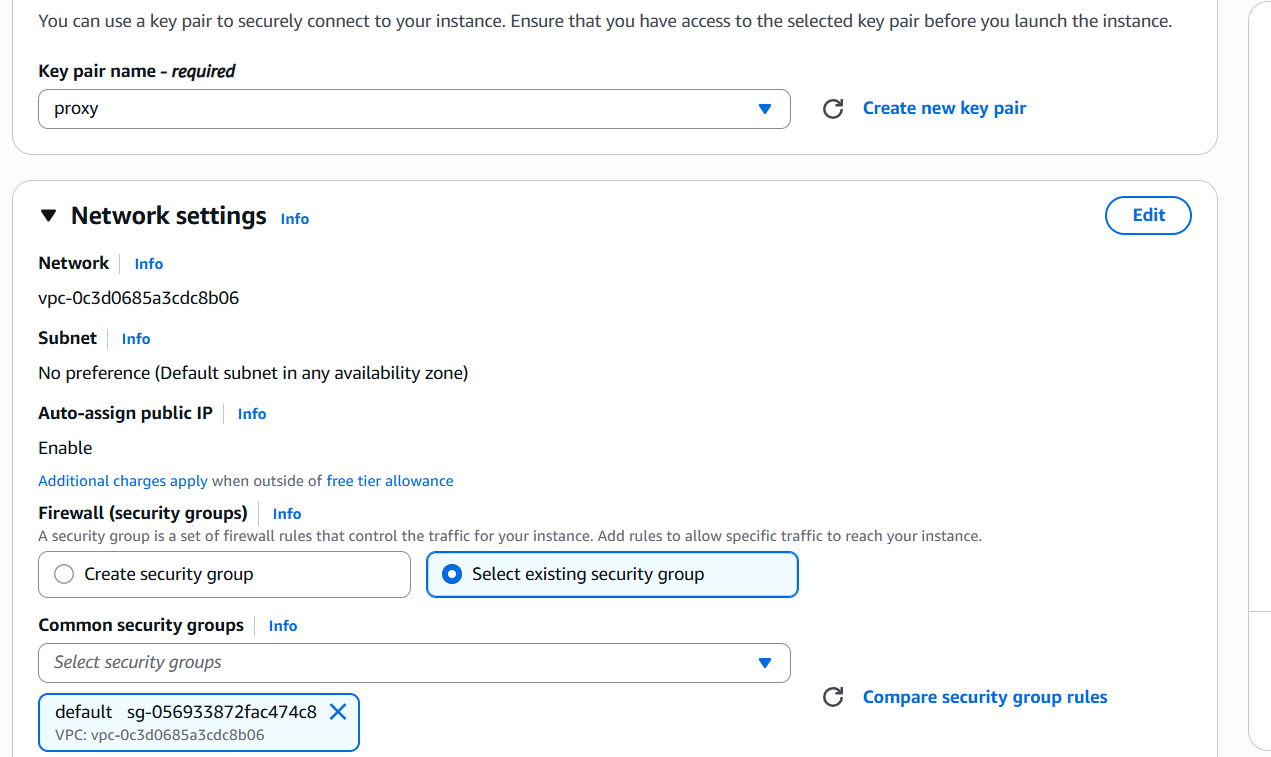
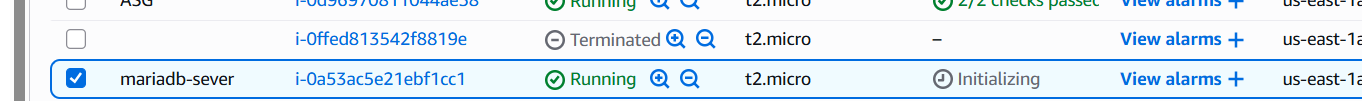
1. Create mariadb db on ec2.

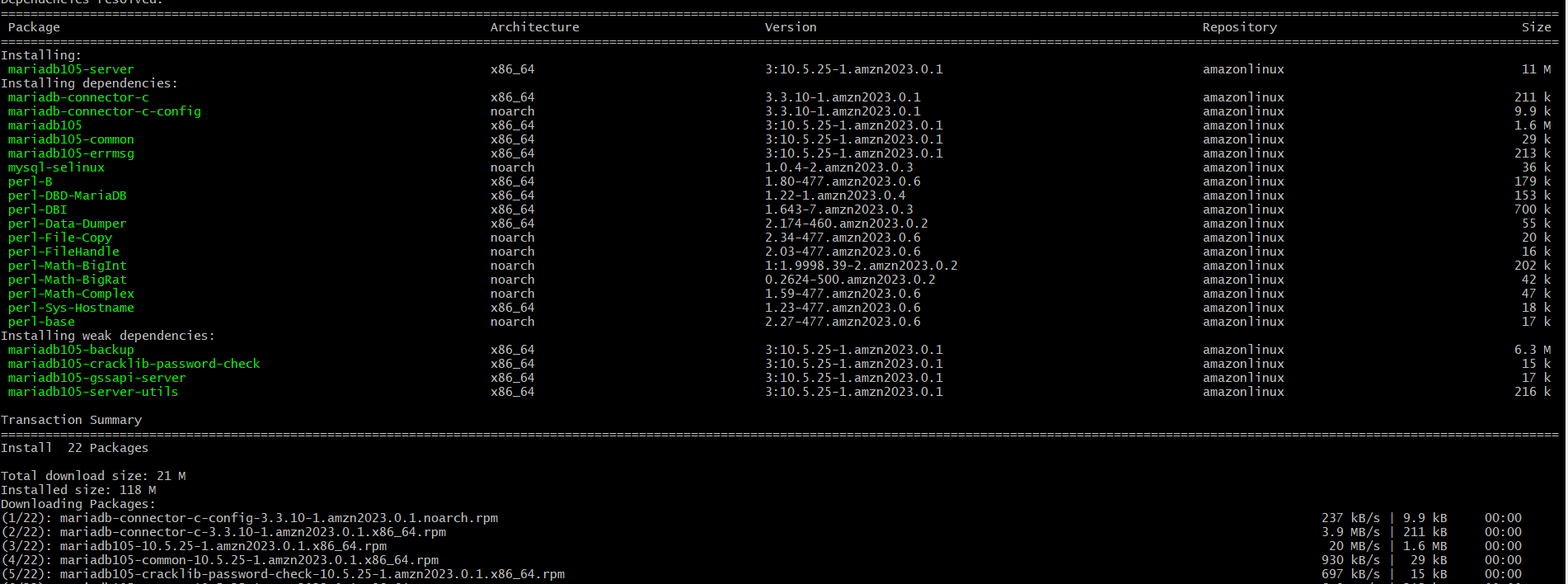
Here's a complete, step-by-step guide to creating a MariaDB database   
on an Amazon EC2 instance, including both setup and explanations.   
This approach is great if you want full control over your database server   
(as opposed to using Amazon RDS).  
 Overview  
You'll:  
1. Launch an EC2 instance  
2. SSH into it  
3. Install and configure MariaDB  
4. Secure MariaDB  
5. Open network access (with caution)  
6. Connect and test  
Step 1: Launch an EC2 Instance  
1. Go to EC2 Dashboard in AWS Console.  
2. Click "Launch Instance"  
3. Configure:  
Name: mariadb-server  
AMI: Choose Amazon Linux 2023 or Ubuntu 22.04 LTS  
Instance Type: Use t2.micro (Free tier) or higher if needed  
Key pair: Create or choose an existing key pair  
Network settings:  
Allow SSH (port 22) from your IP  
Add a custom TCP rule for port 3306 (MariaDB) –   
restrict it to your IP for security  
4. Click Launch Instance

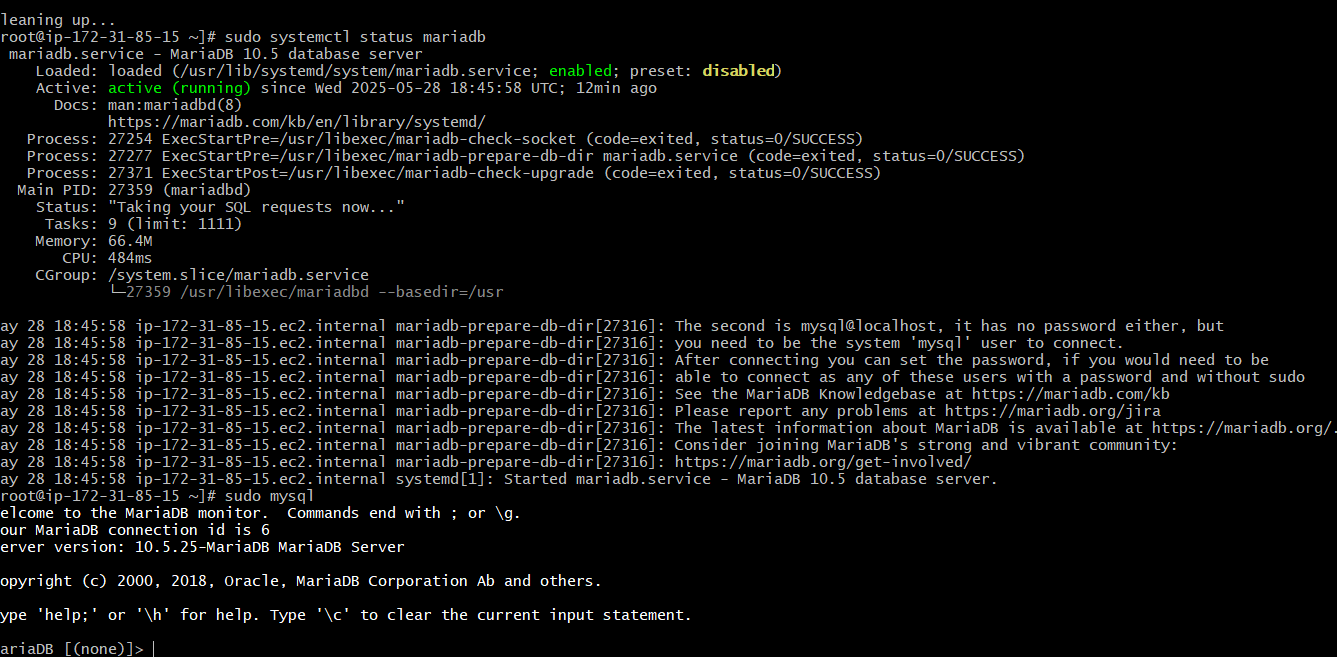












1. Insert some dummy data

**Step 1: Create a Database and Table**

First, log into MariaDB and create a new database and table:

CREATE DATABASE mydb;

USE mydb;

CREATE TABLE employees (

id INT AUTO\_INCREMENT PRIMARY KEY,

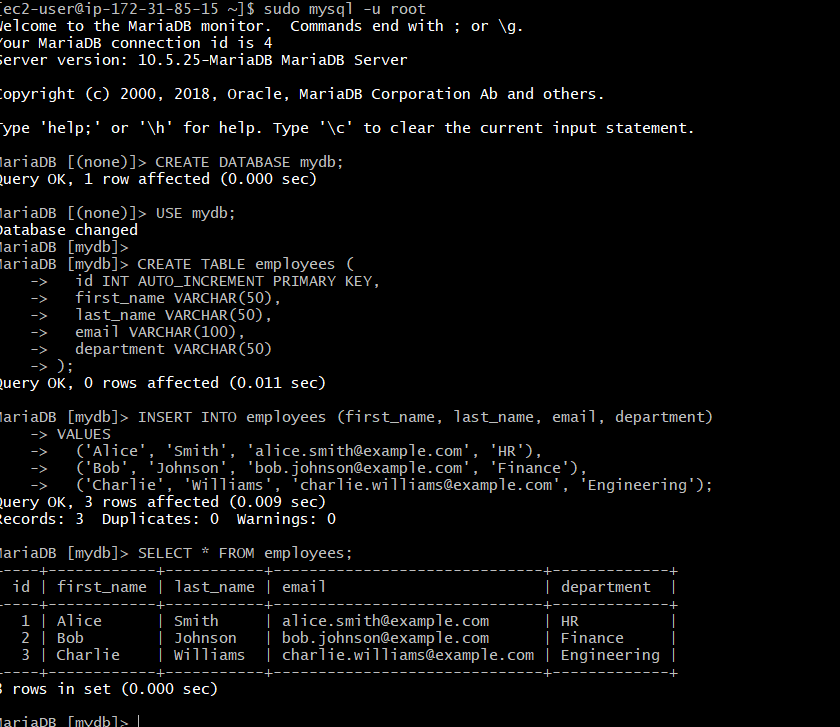
first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

department VARCHAR(50)

);



1. Take the backup of dummy data on ec2

check the available users by logging into MySQL as root using sudo

sudo mysql

Then run this command in the MySQL shell:

sql

Copy

Edit

SELECT user, host FROM mysql.user;

You'll see something like:

pgsql

Copy

Edit

+-----------+-----------+

| user | host |

+-----------+-----------+

| root | localhost |

| ec2user | localhost |

| admin | % |

+-----------+-----------+

Pick the correct user (for example ec2user or admin) and exit:

sql

EXIT;

Then use that in the mysqldump command:

bash

mysqldump -u ec2user -p mydb > /home/ec2-user/mydb\_backup.sql



1. launch Mariadb RDS instance.

**Step 1: Log in to AWS Console**

* Go to <https://console.aws.amazon.com/>
* Sign in with your AWS credentials.

**Step 2: Open Amazon RDS Dashboard**

1. In the **search bar**, type RDS and select **RDS**.
2. Click on **"Create database"**.

**Step 3: Configure Database Settings**

**1. Choose a Database Creation Method**

* Select **Standard Create** (recommended for full control).

**2. Engine Options**

* Choose **MariaDB** as the engine.
* Choose the **version** you want (e.g., 10.6.14).

**3. Templates**

* Choose:
  + **Free tier** (if eligible),
  + or **Dev/Test** or **Production** as needed.

**Step 4: Settings**

* **DB instance identifier**: mariadb-instance
* **Master username**: admin (or your preferred username)
* **Master password**: Set a strong password and confirm it

**Step 5: DB Instance Class**

* For **free tier**, select:
  + **DB instance class**: db.t3.micro
  + **Storage**: 20 GiB, General Purpose (SSD)

**Step 6: Connectivity**

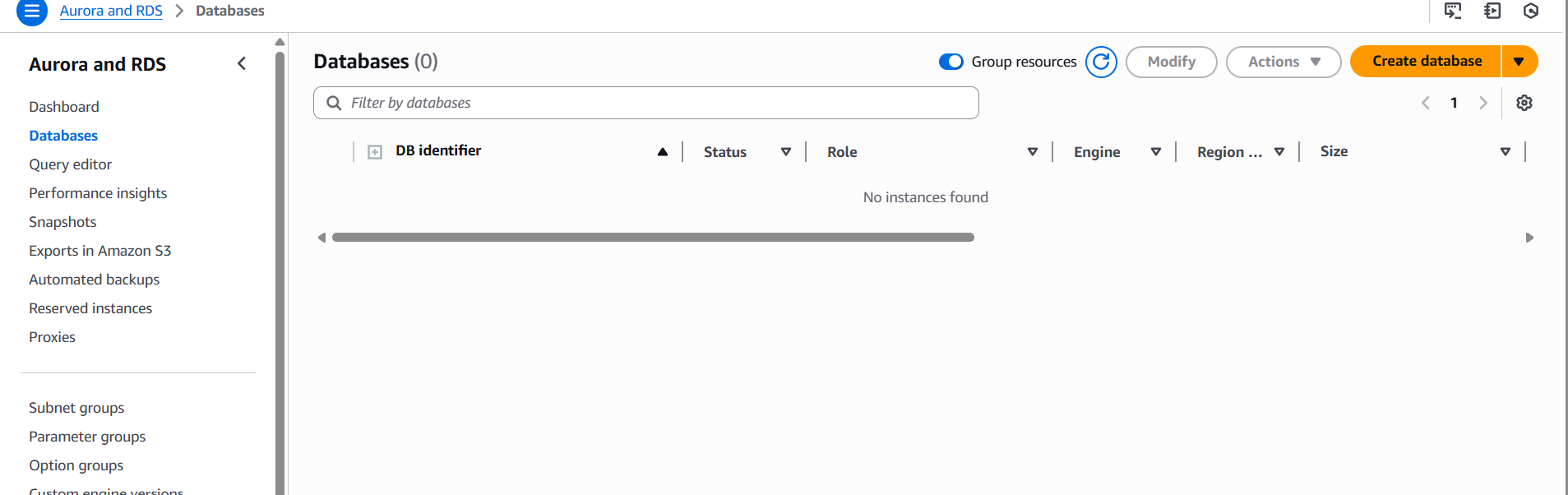
* **Virtual Private Cloud (VPC)**: Choose default or create a custom one.
* **Subnet group**: Default is okay.
* **Public access**:
  + Choose **Yes** if you want to connect over the internet.
  + **No** if you'll use it internally (more secure).
* **VPC security group**:
  + Choose **Create new** or **Choose existing**.
  + **Allow port 3306 (MariaDB default port)** from your IP or EC2.

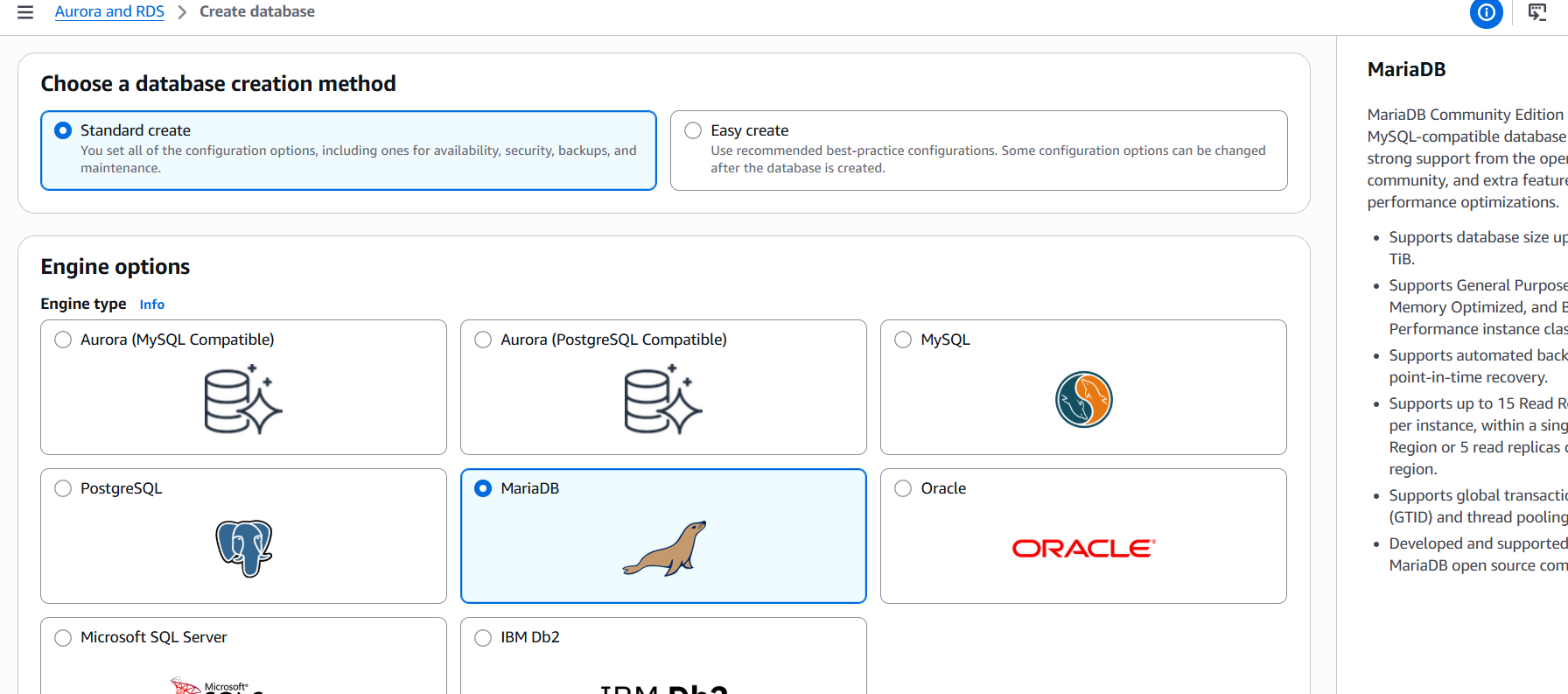
**Step 7: Additional Configuration**

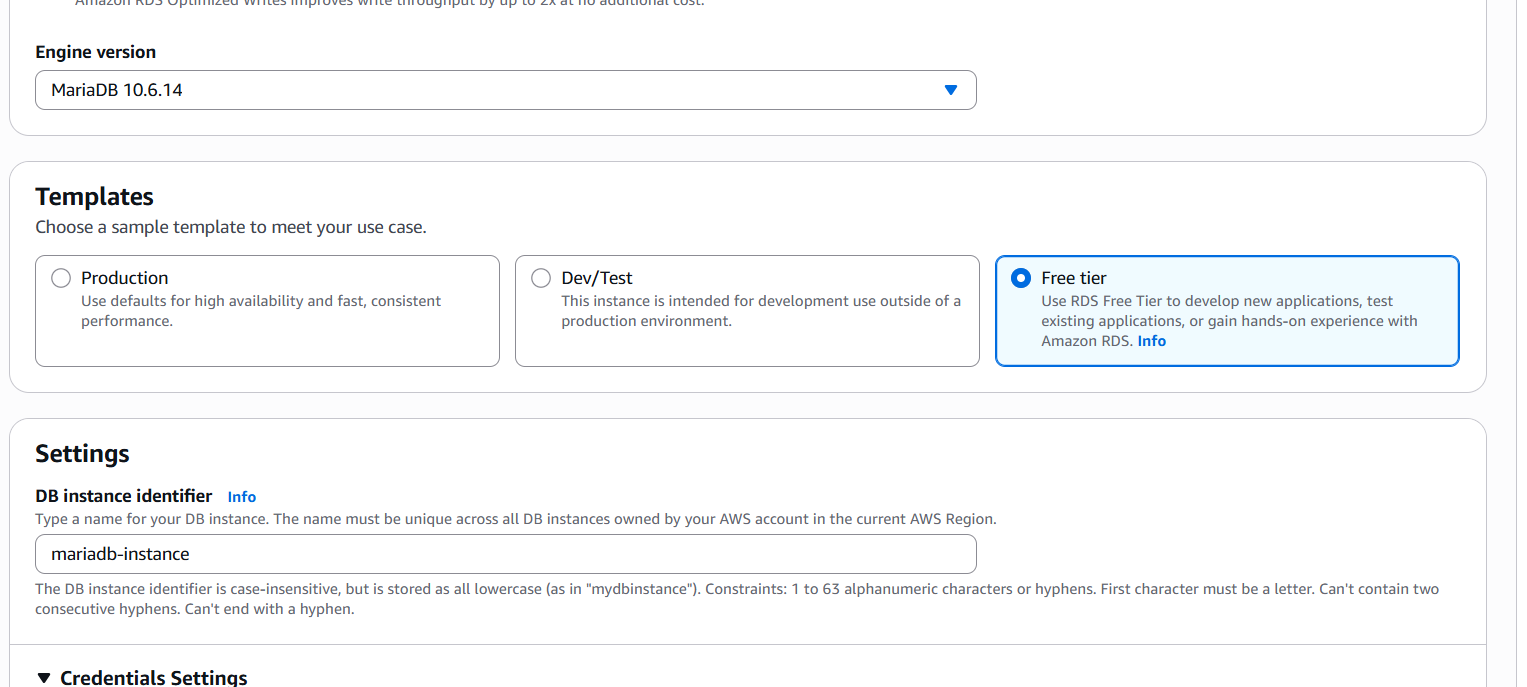
* **Initial database name**: mydb (or your choice)
* **DB parameter group** and **Option group**: use default
* **Backup**: Enable automatic backups (default 7 days)
* **Monitoring**: Optional — enable Enhanced Monitoring if needed
* **Maintenance**: Enable auto minor version upgrades

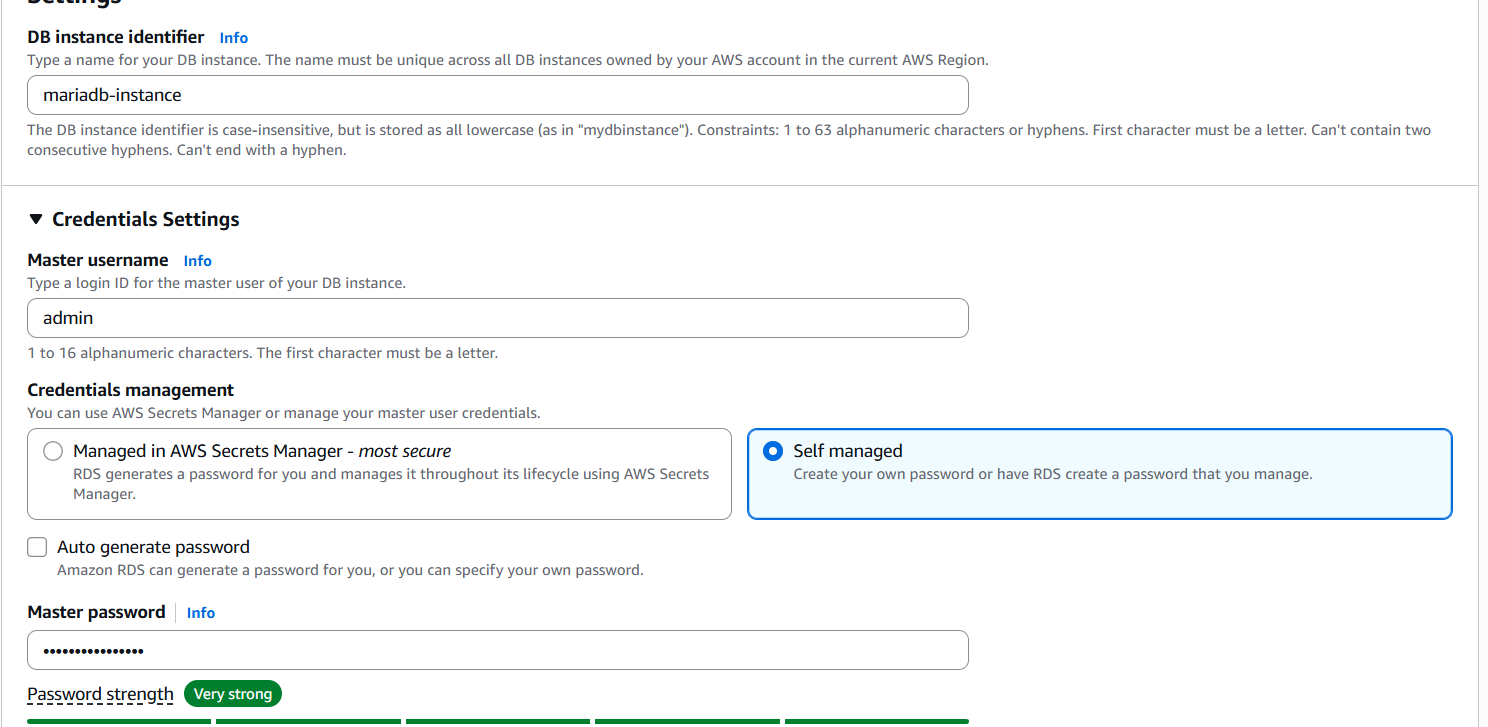
**Step 8: Create Database**

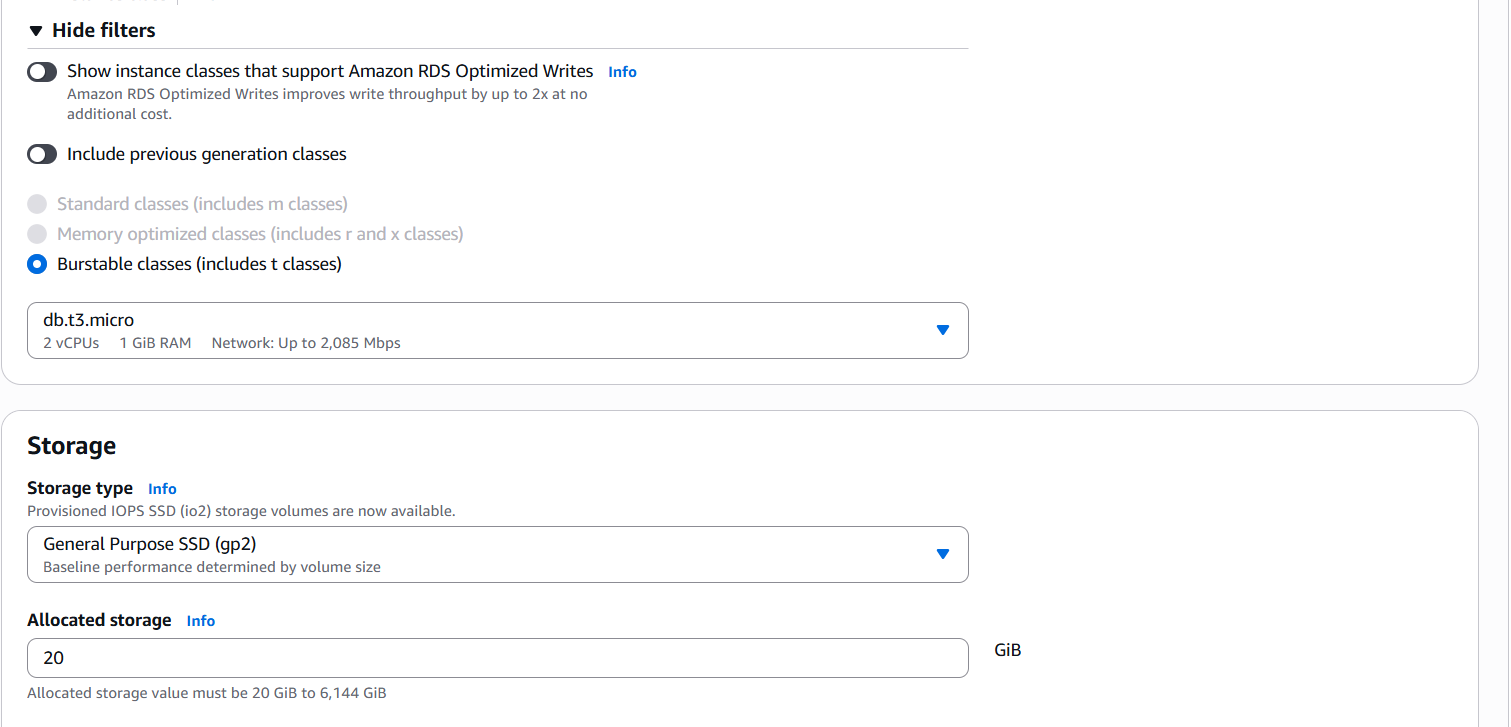
* Click **Create database**.
* Wait a few minutes for the DB instance status to become **“Available”**.

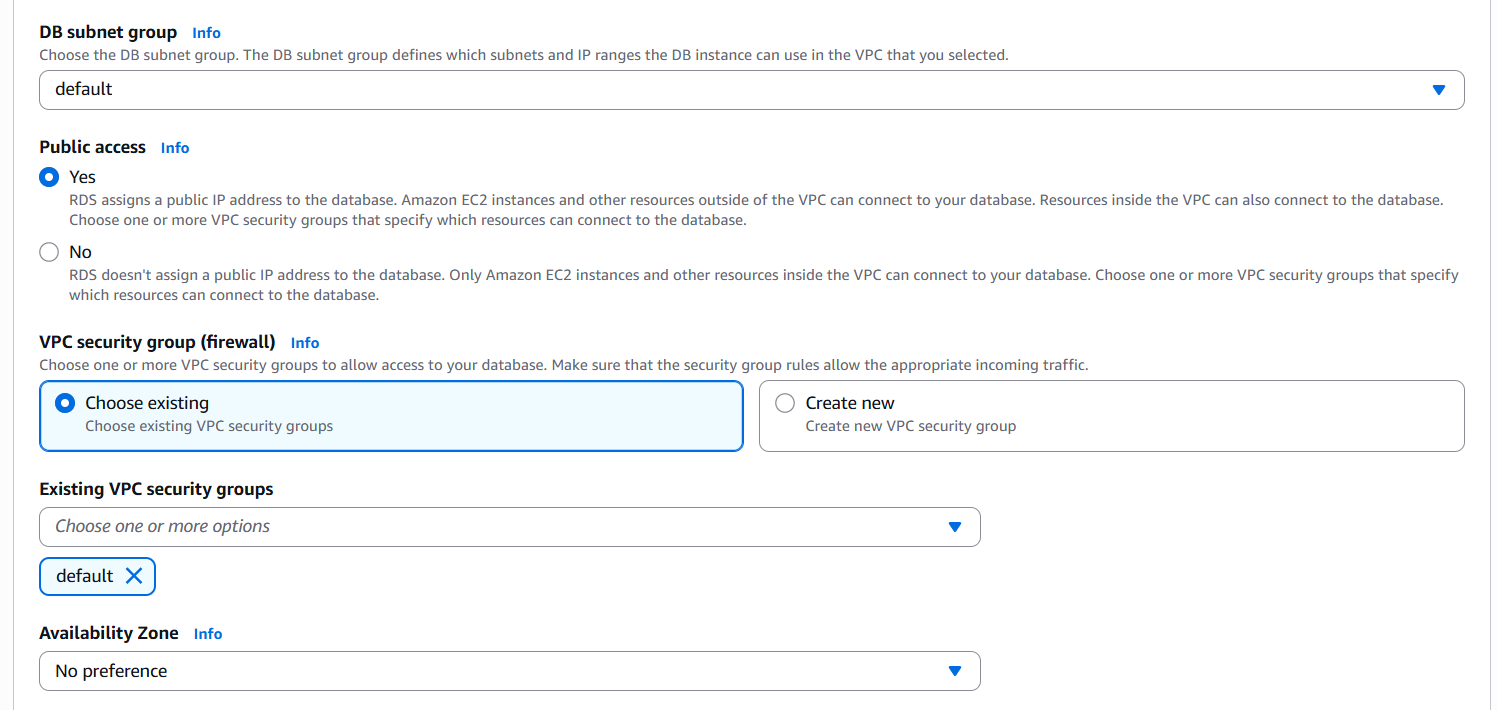


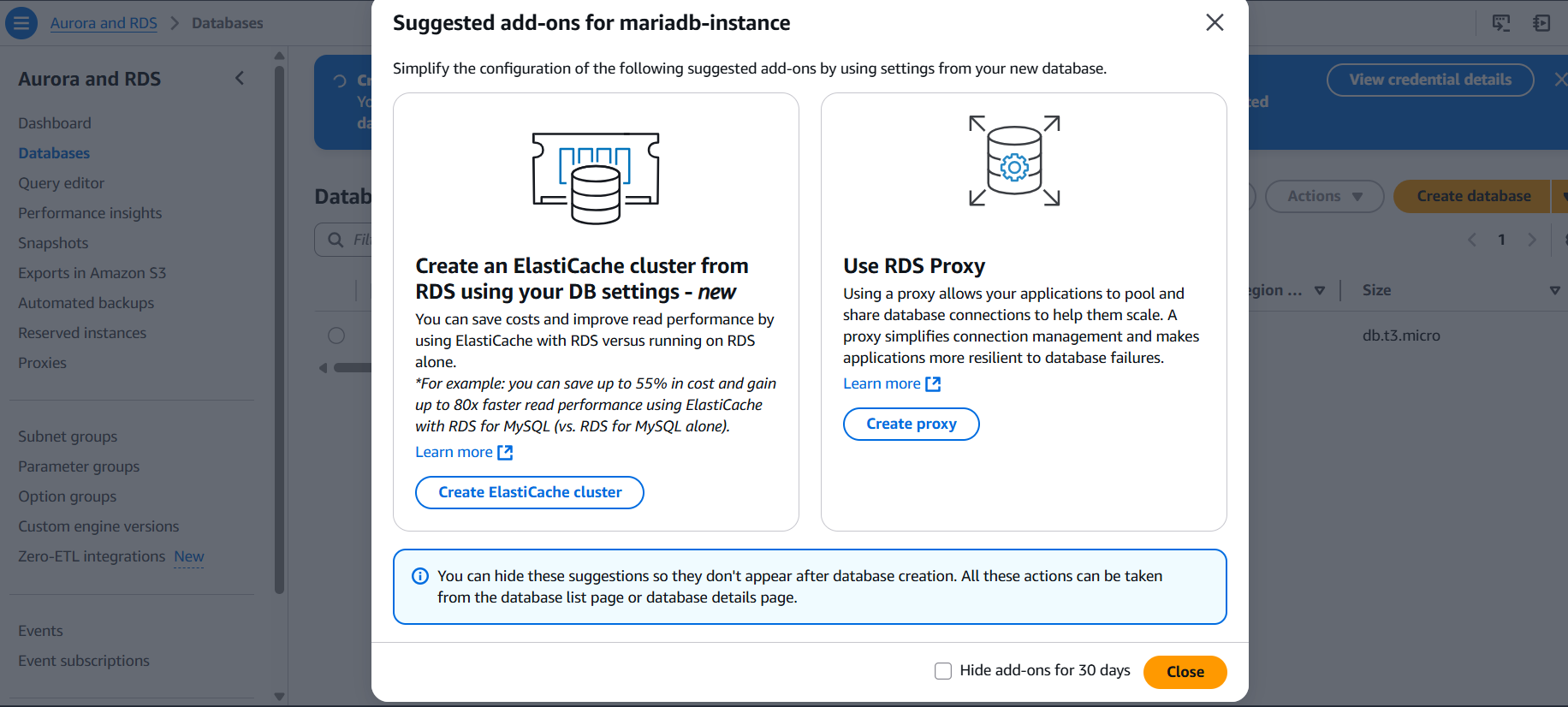




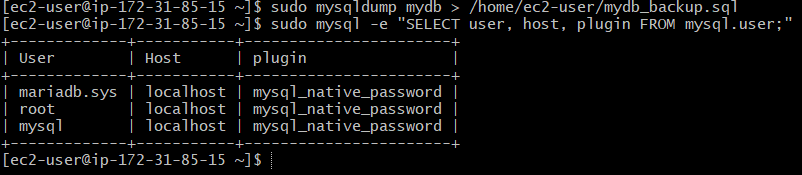








1. Migrate database from ec2 to RDS

Creating a backup dump of your MariaDB database on EC2:   
sudo mysqldump -u root -p mydb > /home/ec2-user/mydb\_backup.sql   
Importing that dump file into your RDS MariaDB instance:   
mysql -h <RDS\_ENDPOINT> -u admin -p mydb < /home/ec2-user/mydb\_backup.sql   
   
   
 

1. Install mysql db on ec2

**1. Verify OS (optional but helpful)**

Run:

bash

cat /etc/os-release

You’ll see something like:

NAME="Amazon Linux"

VERSION="2023"

**2. Install MariaDB Server**

Run the following:

sudo dnf install -y mariadb105-server

If it fails, try the generic:

sudo dnf install -y mariadb-server

**3. Start MariaDB Service**

sudo systemctl start mariadb

sudo systemctl enable mariadb

1. **Secure the Installation**

Run this to set root password and secure defaults:

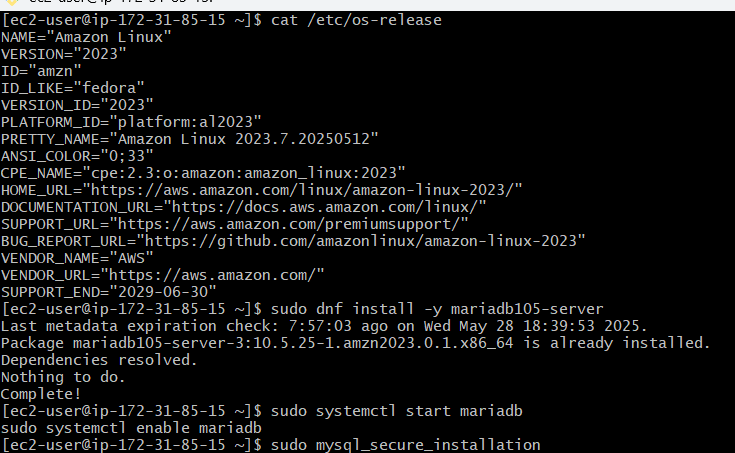
sudo mysql\_secure\_installation

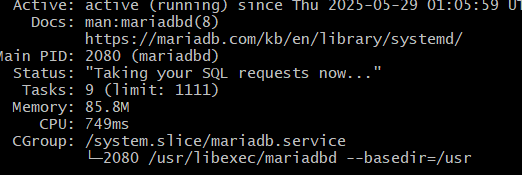
**Log into MySQL**

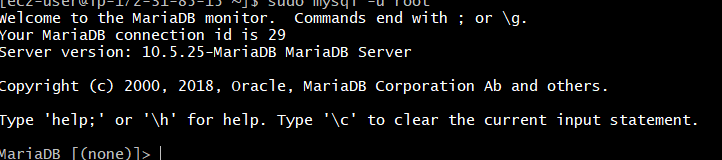
sudo mysql -u root

Or if you set a password:

1. bash
2. CopyEdit
3. mysql -u root -p







7) Launch mysql RDS image

**Step 1: Sign in to AWS and Open RDS Console**

1. Log in to your [AWS Management Console](https://console.aws.amazon.com/).
2. In the search bar, type **RDS** and select **RDS** to open the Amazon RDS console.

**Step 2: Create a MySQL DB Instance**

1. In the RDS console, click **Databases** in the left navigation pane.
2. Click **Create database**.
3. Under **Engine options**, select **MySQL**.
4. Choose the **Free tier** template if you're eligible.
5. In **Settings**, provide:
   * **DB instance identifier**: A unique name for your DB instance.
   * **Master username**: Your desired admin username.
   * **Master password**: A strong password.
6. Under **DB instance size**, select **db.t2.micro** (eligible for free tier).
7. In **Storage**, keep the default 20 GiB.
8. Under **Connectivity**, ensure:
   * **Public access** is set to **Yes** if you want to connect from outside the VPC.
   * Choose an existing VPC security group or create a new one.
9. Click **Create database**.

*Note: It may take a few minutes for the DB instance to become available.*

**Step 3: Configure Security Group**

To allow connections to your DB instance:

1. In the AWS console, navigate to **EC2 > Security Groups**.
2. Find the security group associated with your RDS instance.
3. Select the security group and go to the **Inbound rules** tab.
4. Click **Edit inbound rules** and add a rule:
   * **Type**: MySQL/Aurora
   * **Protocol**: TCP
   * **Port Range**: 3306
   * **Source**: Your IP or a specific IP range.
5. Click **Save rules**.

**✅ Step 4: Connect to Your MySQL DB Instance**

You can connect using the MySQL command-line client or MySQL Workbench.

**Using MySQL Workbench:**

1. Download and install [MySQL Workbench](https://dev.mysql.com/downloads/workbench/).
2. Open MySQL Workbench and click the **+** icon to create a new connection.
3. In the **Setup New Connection** window:
   * **Connection Name**: Any name you prefer.
   * **Hostname**: The **Endpoint** of your RDS instance (found in the RDS console under **Connectivity & security**).
   * **Port**: 3306
   * **Username**: Your master username.
4. Click **Test Connection**.
5. Enter your password when prompted.
6. If the connection is successful, click **OK** to save the connection.

**Using MySQL Command-Line Client:**

Install the MySQL client on your local machine and run:

bash

CopyEdit

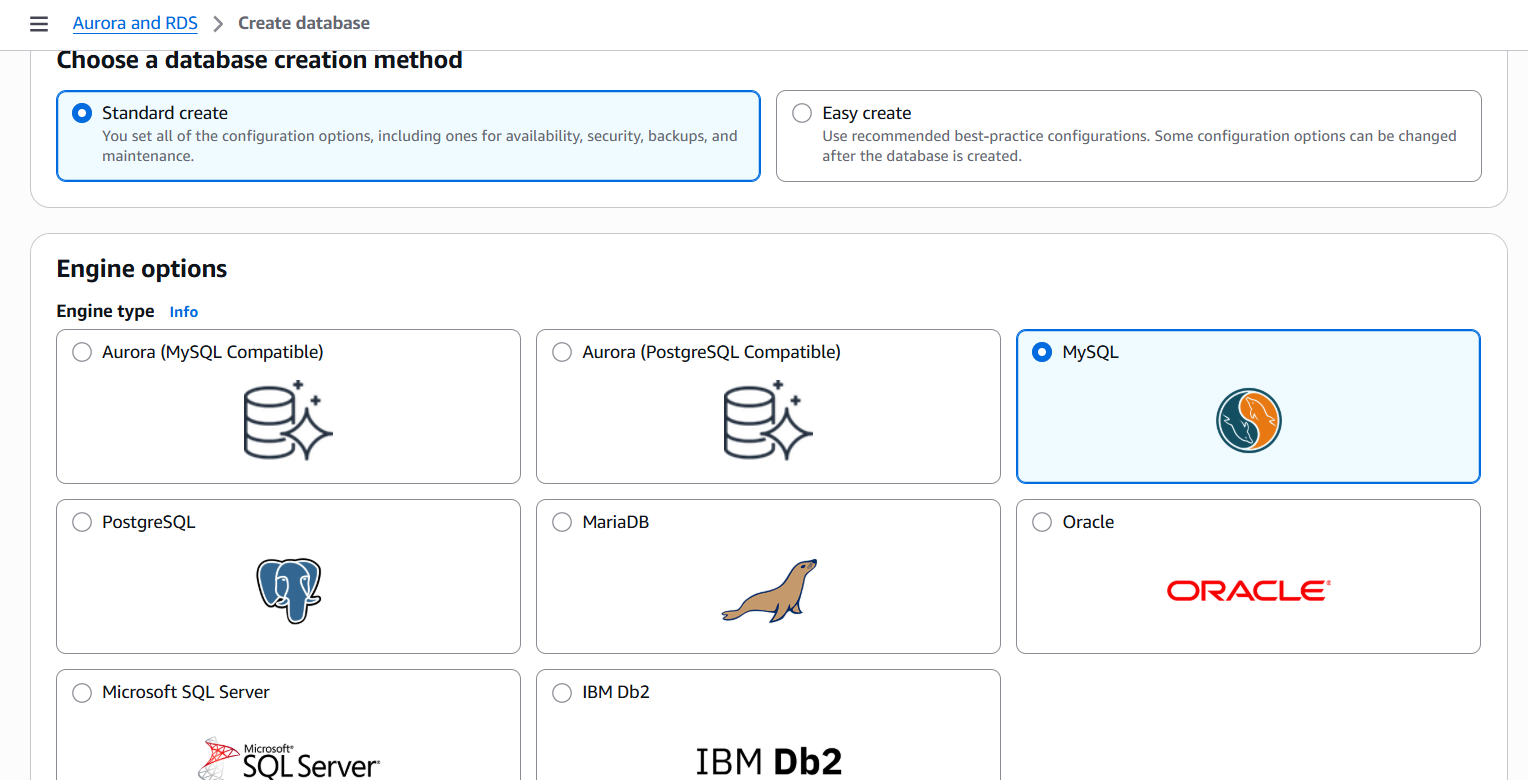
mysql -h your-db-endpoint -P 3306 -u your-username -p

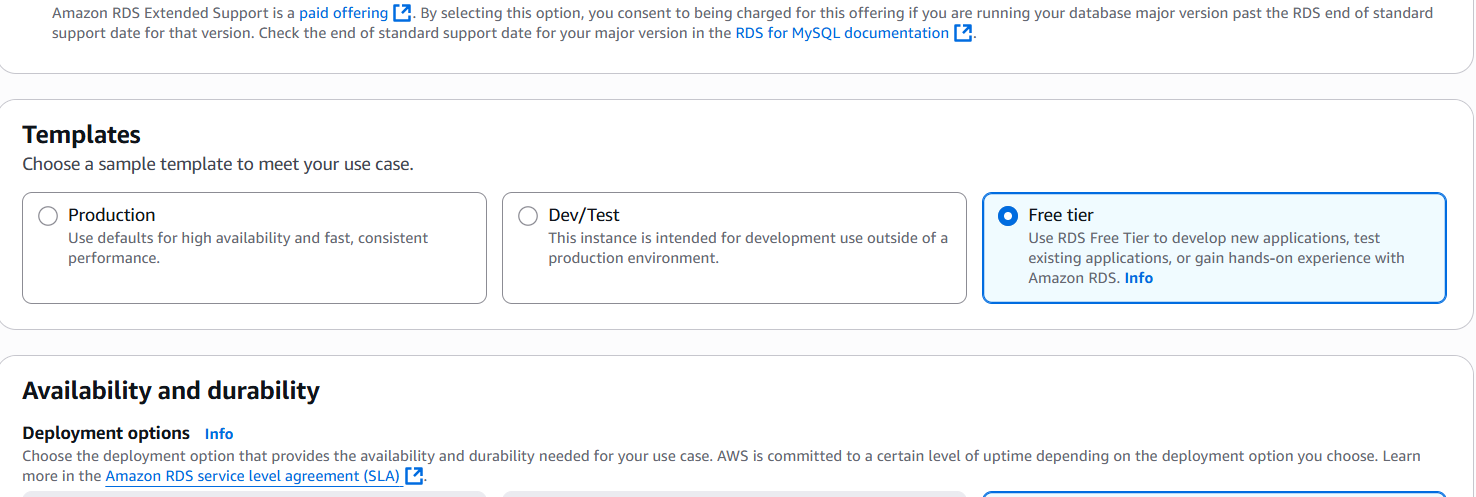
Replace your-db-endpoint with your RDS instance's endpoint, and your-username with your master username. Enter your password when prompted.

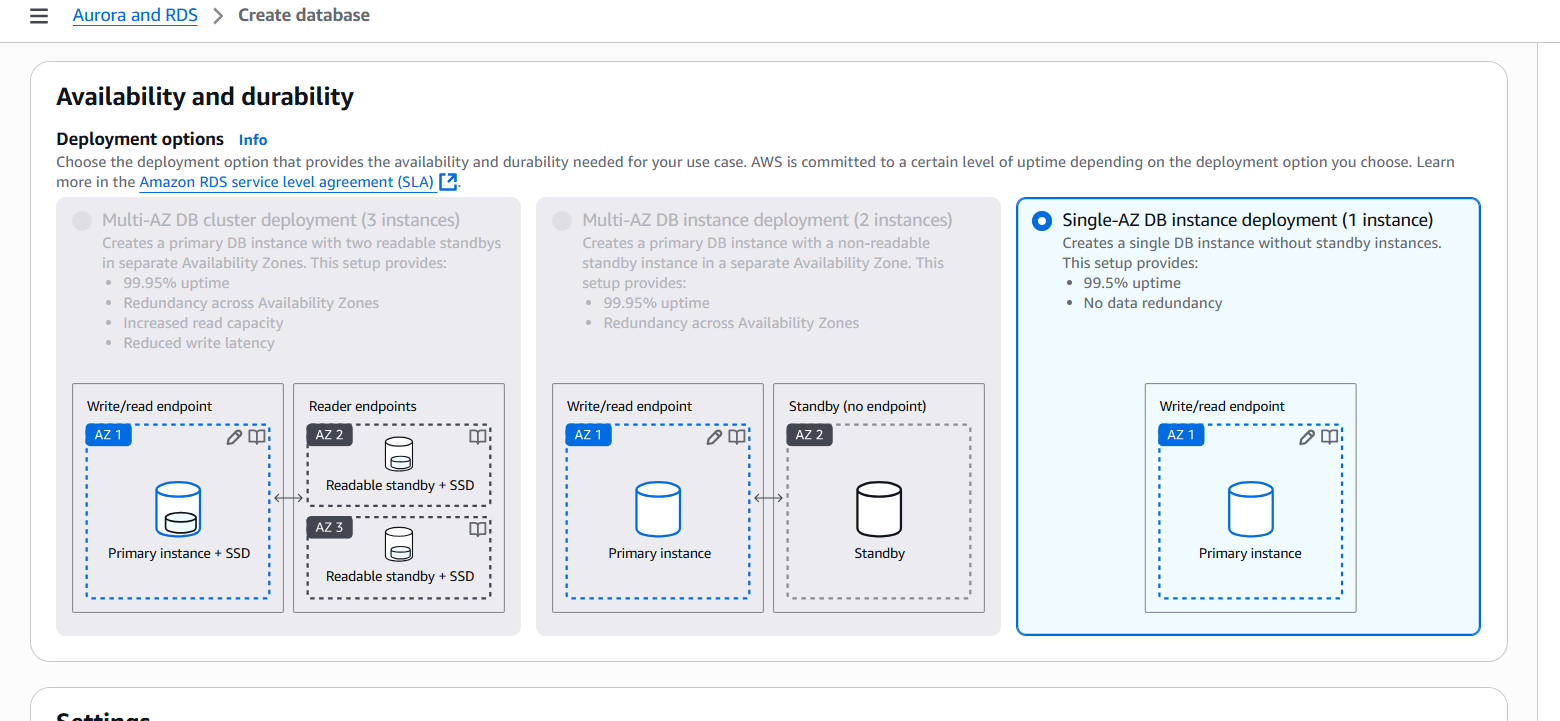
**✅ Step 5: Delete the DB Instance (Optional)**

If you no longer need the DB instance:

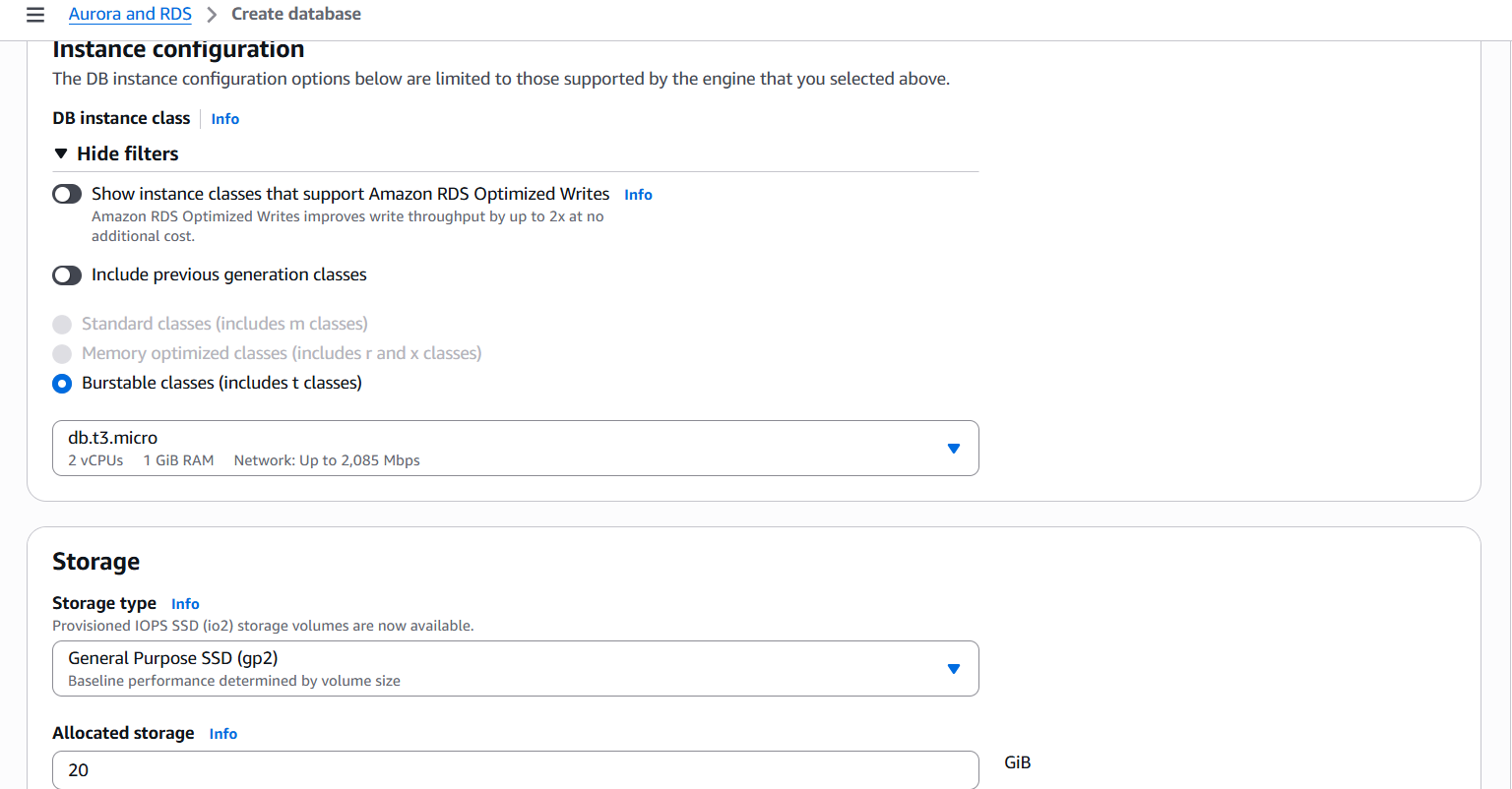
1. In the RDS console, select your DB instance.
2. Click **Actions > Delete**.
3. Choose whether to create a final snapshot.
4. Type **delete me** in the confirmation box and click **Delete**.

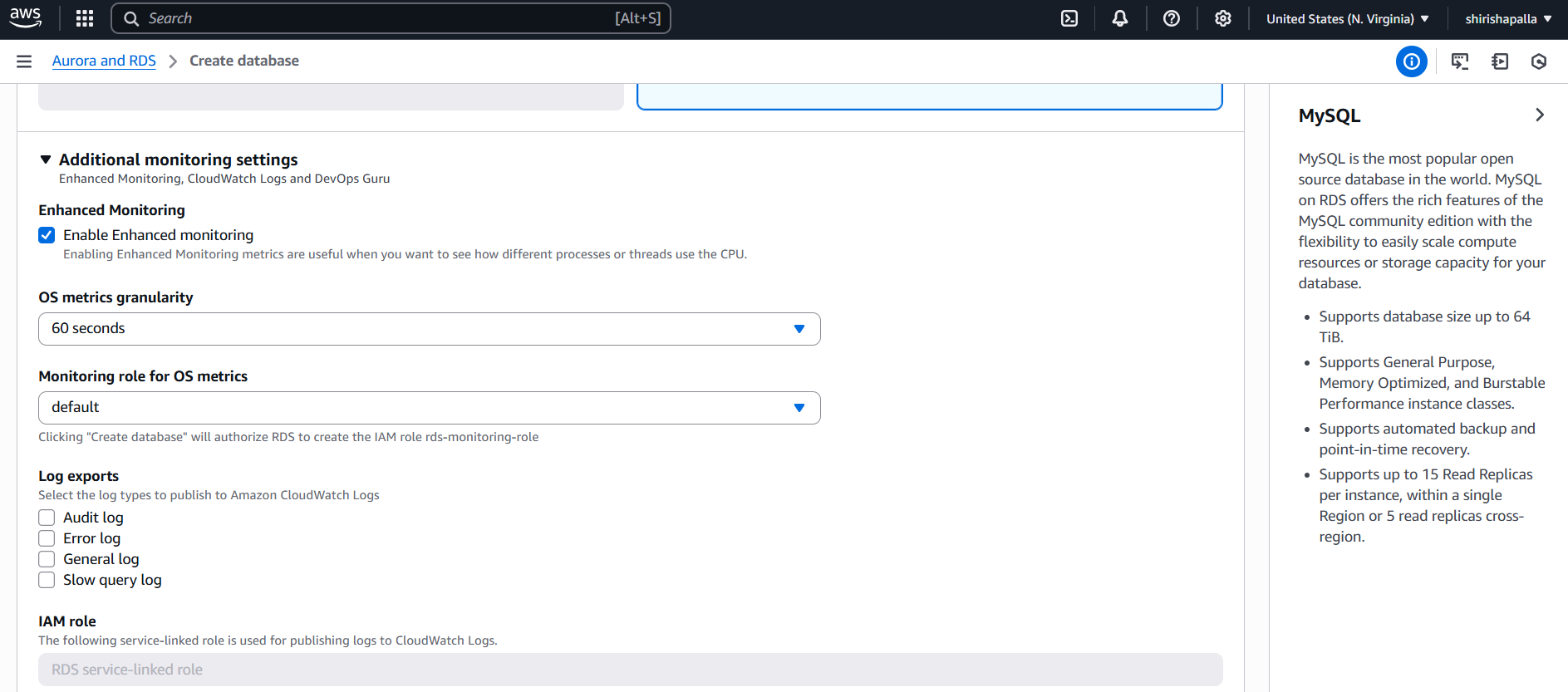


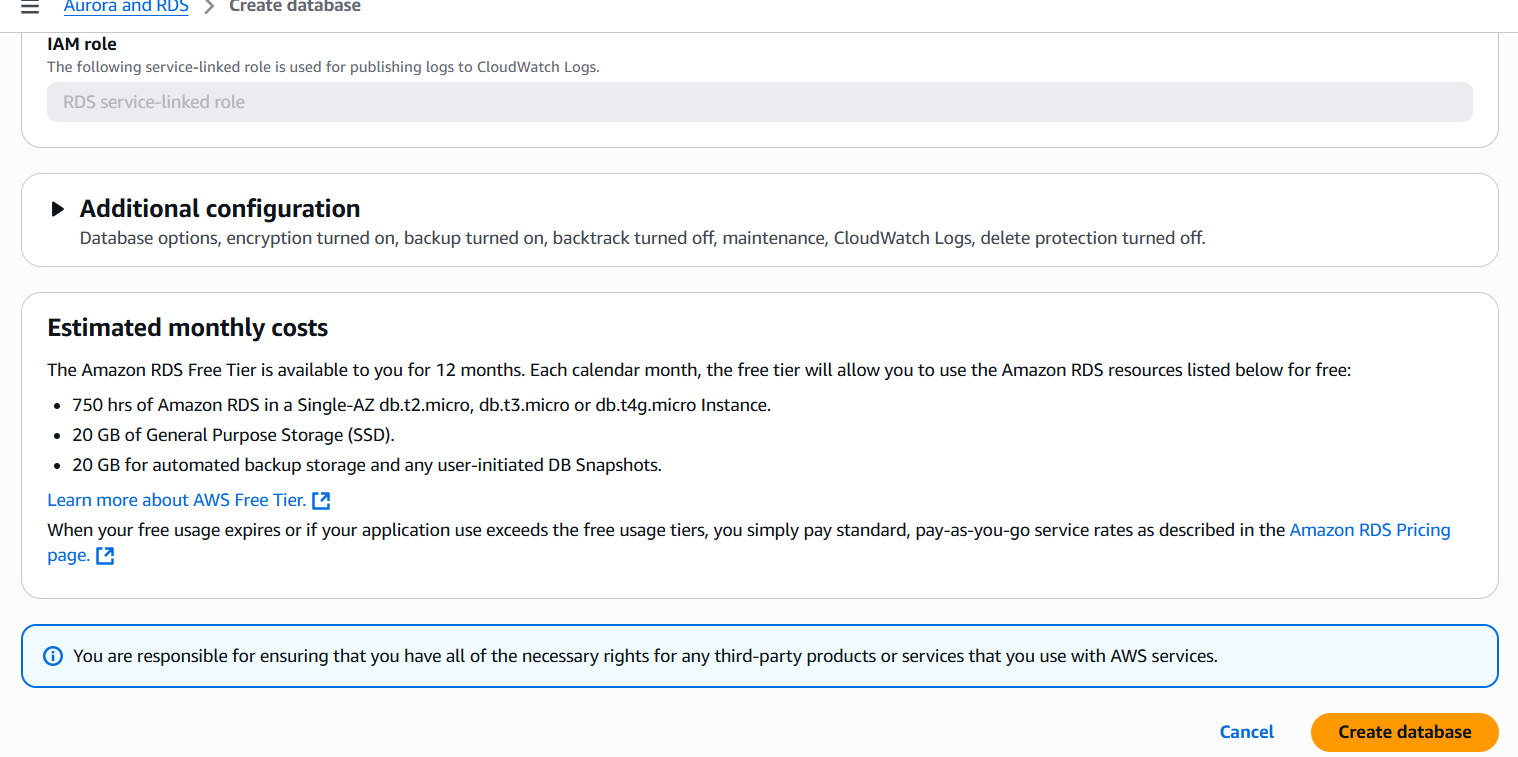




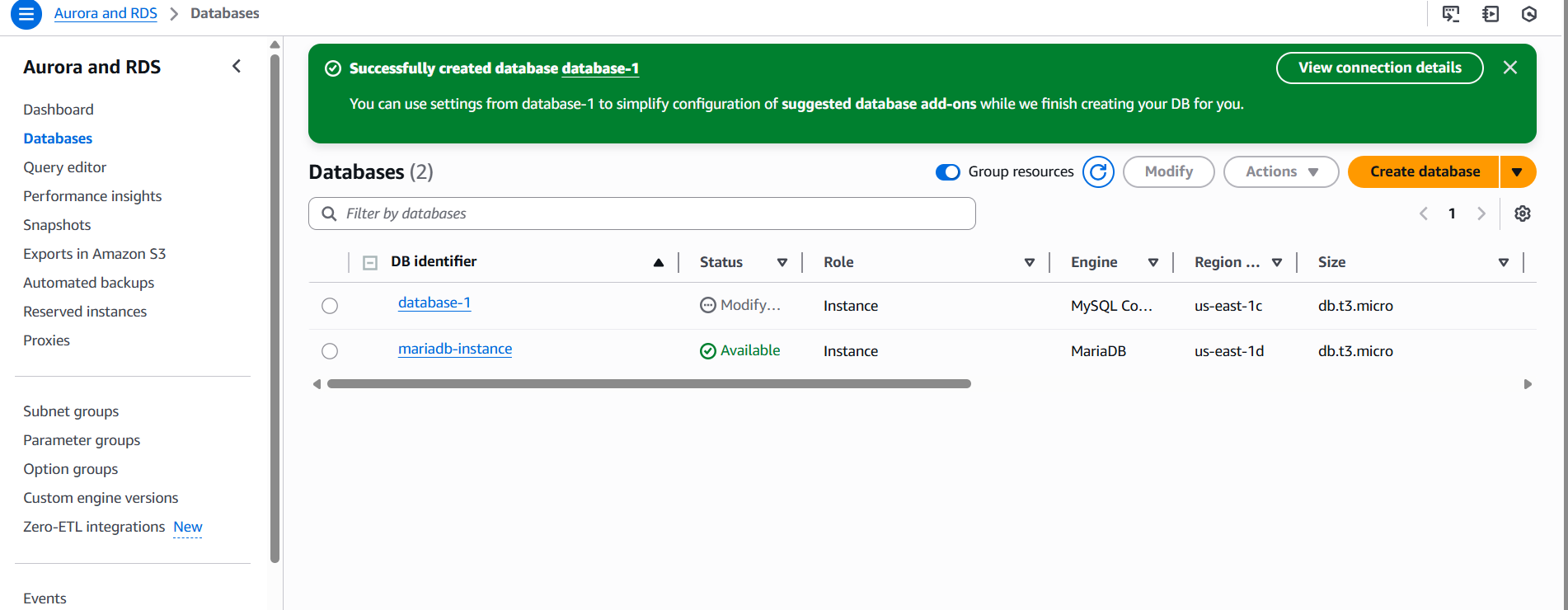












8) COnfigure multi AZ

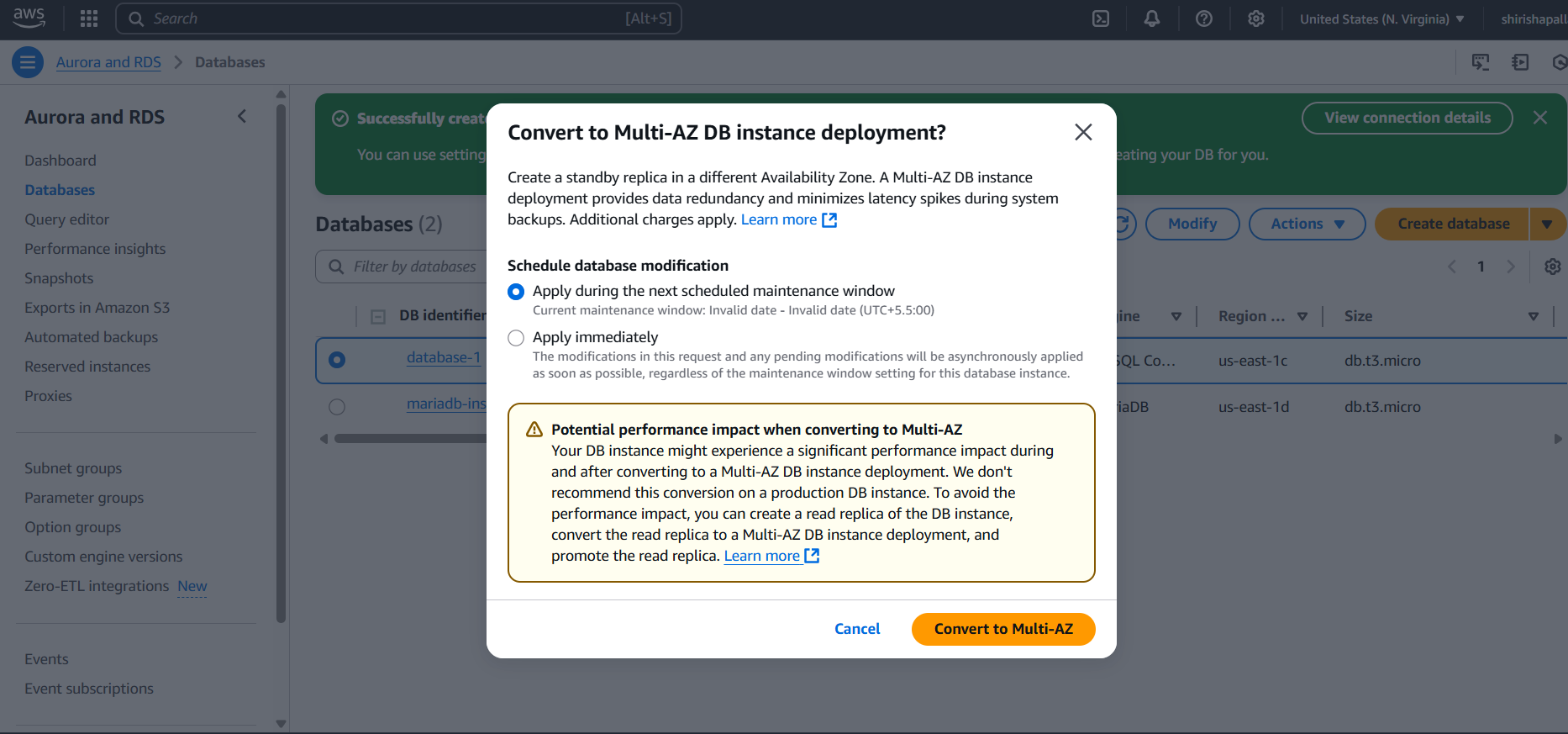
**Option 1: Create a New Multi-AZ MySQL DB Instance**

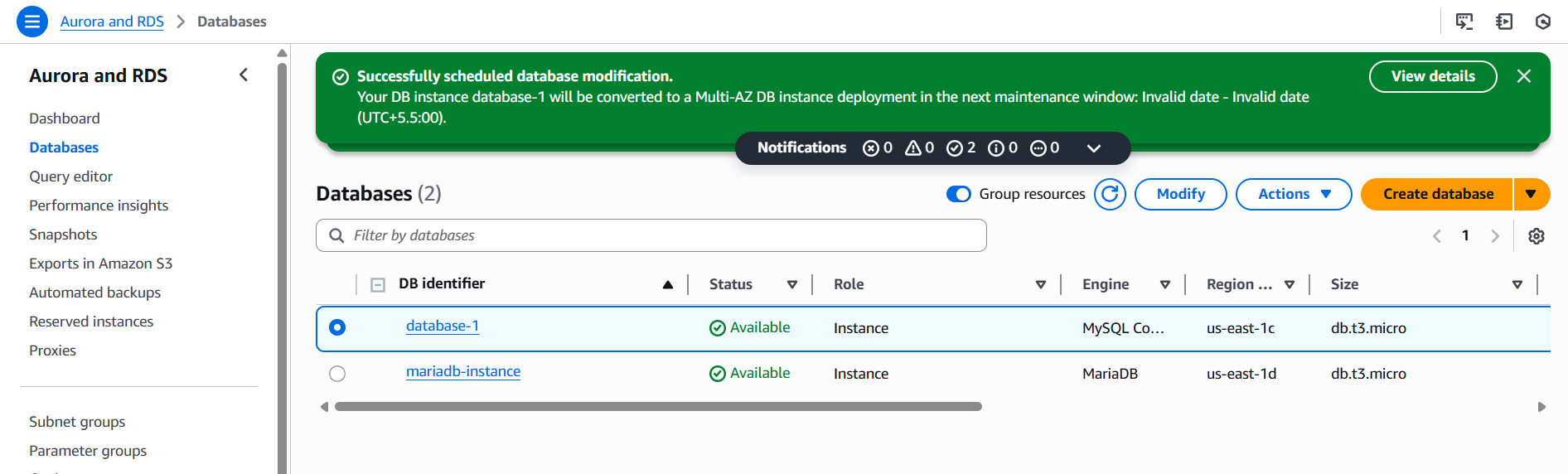
1. **Sign in to AWS Console**:
   * Navigate to the [Amazon RDS Console](https://console.aws.amazon.com/rds/).
2. **Initiate Database Creation**:
   * Click on **Databases** in the left pane.
   * Select **Create database**.
3. **Select Database Creation Method**:
   * Choose **Standard Create**.
4. **Configure Engine Options**:
   * Under **Engine options**, select **MySQL**.
   * Choose the desired **Version**.
5. **Specify Templates**:
   * Under **Templates**, select the appropriate option (e.g., **Production**).
6. **Set DB Instance Identifier and Credentials**:
   * Provide a unique **DB instance identifier**.
   * Set the **Master username** and **Master password**.
7. **Configure DB Instance Class**:
   * Choose an instance class suitable for your workload (e.g., **db.m5.large**).
8. **Enable Multi-AZ Deployment**:
   * In the **Availability & durability** section, select **Create a standby instance (Multi-AZ)**.
9. **Set Storage Options**:
   * Configure storage type and allocated storage as per requirements.
10. **Configure Connectivity**:
    * Select the appropriate **Virtual Private Cloud (VPC)**.
    * Choose or create a **DB Subnet Group** that spans multiple Availability Zones.
    * Set **Public access** to **Yes** or **No** based on your access needs.
    * Select or create a **VPC security group** to manage access.
11. **Additional Configuration**:
    * Set backup retention periods, maintenance windows, and other configurations as needed.
12. **Create Database**:
    * Review all settings and click **Create database**.

*Note: The creation process may take several minutes. Once available, your primary DB instance will have a synchronous standby in a different Availability Zone.*

**Option 2: Convert an Existing Single-AZ DB Instance to Multi-AZ**

1. **Access the RDS Console**:
   * Navigate to the [Amazon RDS Console](https://console.aws.amazon.com/rds/).
2. **Select the DB Instance**:
   * In the **Databases** section, choose the DB instance you want to modify.
3. **Modify the DB Instance**:
   * Click on **Modify**.
4. **Enable Multi-AZ Deployment**:
   * In the **Availability & durability** section, select **Create a standby instance (Multi-AZ)**.
5. **Apply Changes**:
   * Choose to apply changes **immediately** or during the next maintenance window.
   * Click **Continue**, review the summary, and then click **Modify DB Instance**.





1. Take Backup of db and restore the DB

**1. Automated Backups (Point-in-Time Recovery)**

Amazon RDS automatically performs daily backups of your DB instance during a specified backup window.

**Steps to Enable or Modify Automated Backups:**

1. **Sign in to AWS Console**:
   * Navigate to the [Amazon RDS Console](https://console.aws.amazon.com/rds/).
2. **Select Your DB Instance**:
   * In the **Databases** section, choose the DB instance you want to configure.
3. **Modify Backup Settings**:
   * Click on **Modify**.
   * In the **Backup** section, set the **Backup retention period** (up to 35 days).
   * Specify the **Backup window** if desired.
4. **Apply Changes**:
   * Choose to apply changes **immediately** or during the next maintenance window.
   * Click **Continue**, review the summary, and then click **Modify DB Instance**.

*Note: Automated backups allow you to restore your DB instance to any point in time within the retention period.*

**2. Manual Snapshots**

Manual snapshots are user-initiated backups that persist until explicitly deleted.[AWS Documentation](https://docs.aws.amazon.com/AmazonRDS/latest/gettingstartedguide/managing-backup-restore.html?utm_source=chatgpt.com)

**Steps to Create a Manual Snapshot:**

1. **Access the RDS Console**:
   * Navigate to the [Amazon RDS Console](https://console.aws.amazon.com/rds/).
2. **Select the DB Instance**:
   * In the **Databases** section, choose the DB instance you want to back up.
3. **Initiate Snapshot**:
   * Click on **Actions** and select **Take snapshot**.
   * Provide a name for the snapshot and confirm.[AWS Documentation](https://docs.aws.amazon.com/AmazonRDS/latest/gettingstartedguide/managing-backup-restore.html?utm_source=chatgpt.com)

**3. On-Demand Backups Using AWS Backup**

AWS Backup provides centralized backup management for AWS services, including RDS.

**Steps to Create an On-Demand Backup:**

1. **Open AWS Backup Console**:
   * Navigate to the [AWS Backup Console](https://console.aws.amazon.com/backup/).
2. **Create Backup Job**:
   * Click on **Create on-demand backup**.
   * Select **Resource type** as **RDS**.
   * Choose the specific DB instance.
   * Set the **Retention period** and **Backup vault**.
   * Choose the IAM role (e.g., **AWSBackupDefaultRole**).
   * Click **Create on-demand backup**.[Amazon Web Services, Inc.](https://aws.amazon.com/getting-started/hands-on/amazon-rds-backup-restore-using-aws-backup/?utm_source=chatgpt.com)

*Note: AWS Backup allows you to automate and manage backups across AWS services.*

**4. Command-Line Backup Using mysqldump**

For granular control, you can use the mysqldump utility to export your database.

**Steps to Backup Using mysqldump:**

1. **Run the Following Command**:

bash

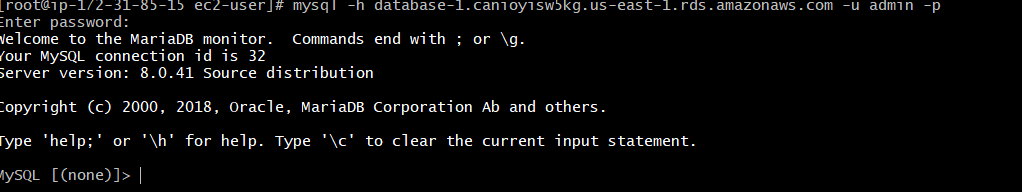
CopyEdit

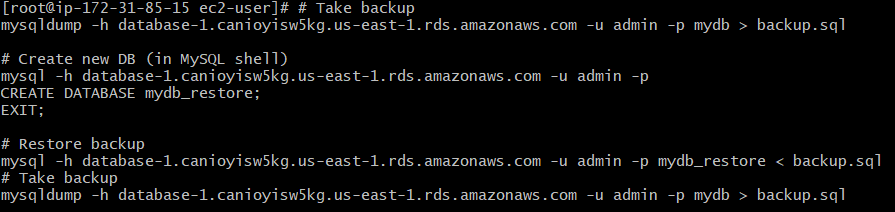
mysqldump -h your-rds-endpoint -u your-username -p your-database-name > backup.sql

* Replace your-rds-endpoint, your-username, and your-database-name with your actual RDS endpoint, username, and database name.

1. **Enter Password**:
   * When prompted, enter your database password.

*Note: Ensure that your security group allows access from your IP address.*





1. Create ReadReplca

