Provisioning Amazon RDS database Using AWS Academy Learner Lab

Table of Contents

Introduction

In this article, you will learn how to create a MySQL database system, connect to the database, run queries against the database, and delete the database instance. We will do this using the Amazon Relational Database Service (Amazon RDS) within the AWS Academy Learner Lab (LL).

Tasks

In this article, you will:

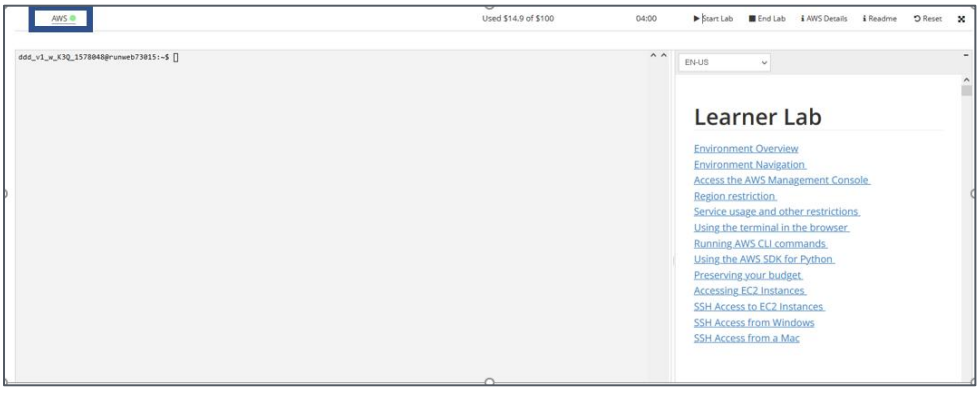
- Create an environment to run a MySQL database
- Connect to the database using the MySQL Workbench
- Create tables, insert data, and run queries
- Delete the database instance

Prerequisites

Before continuing with the following activities, you will need:

- **Access to the AWS Academy Learner (LL) course**: From the Student View of the LL, click the Start Lab button as shown in the screenshot below:

- After starting the lab, wait until the AWS icon turns green as shown below:

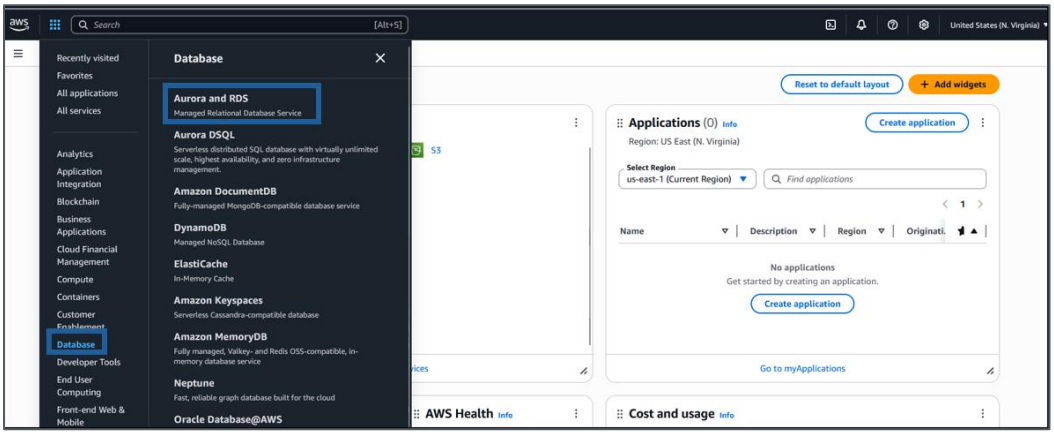


Implementation

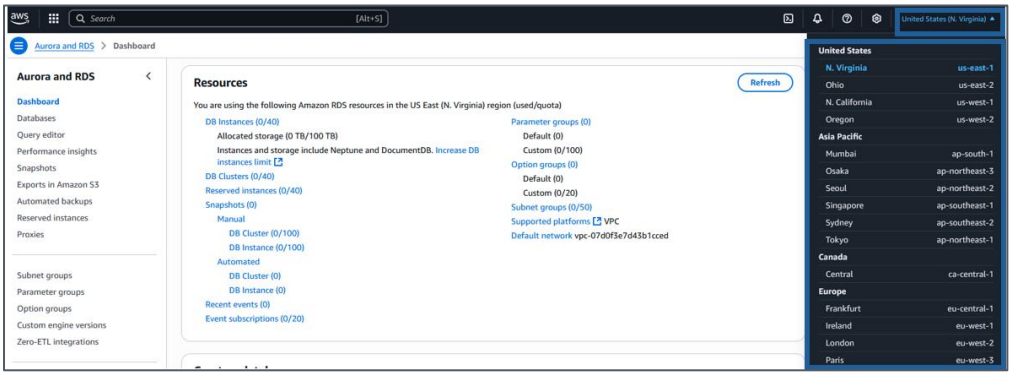
Create a MySQL DB Instance

In this step, we will use Amazon RDS to create a MySQL DB Instance with db.t3.micro DB instance class, 20 GB of storage, and automated backups disabled. All of these are Free Tier eligible.

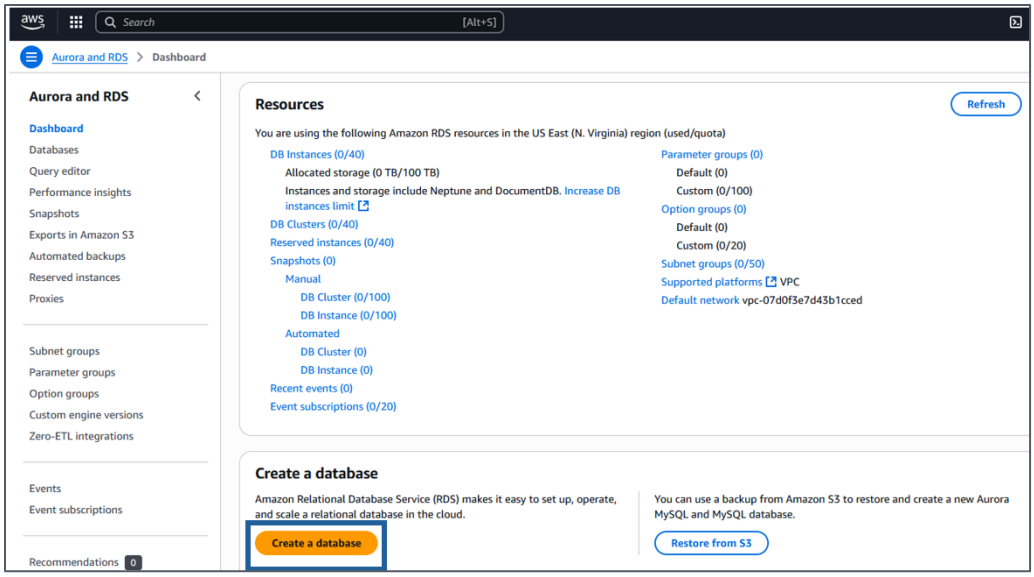
a. Click the AWS icon as shown above to open the *AWS Management Console* in a new browser tab. When the console opens, click on **Services (square icon)** to select **Database** from the left navigation pane and choose **Aurora and RDS** to open the **Amazon Aurora and RDS console**.



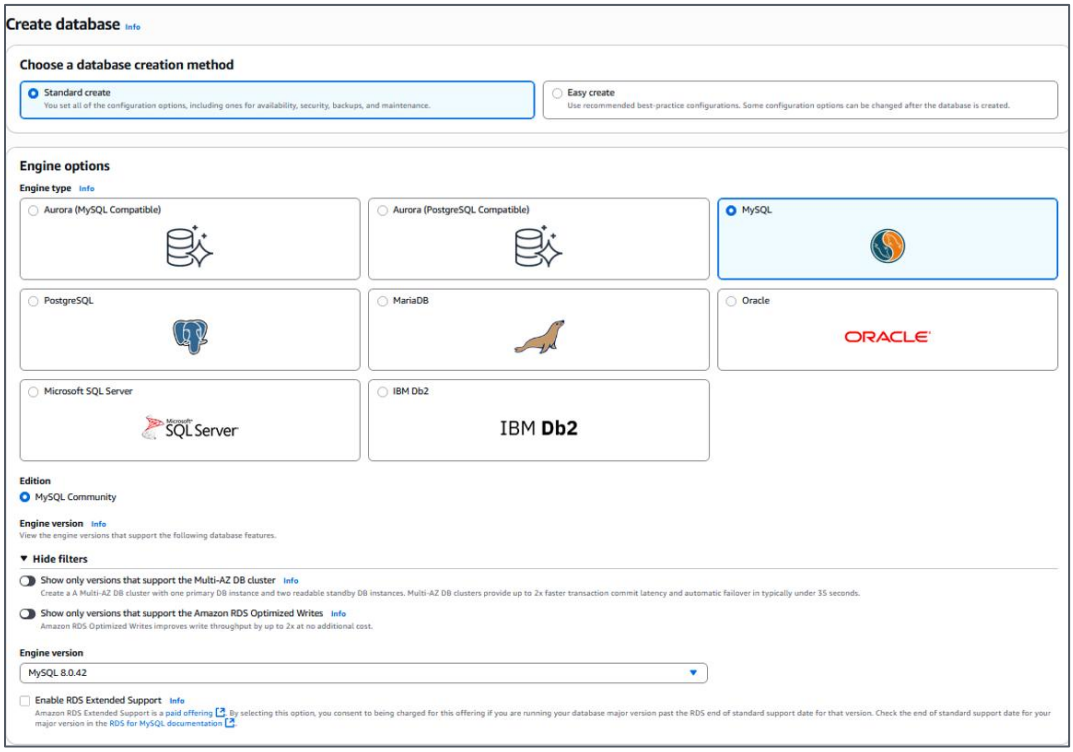
b. In the top right corner of the Amazon RDS console, select the Region in which you want to create the DB instance.

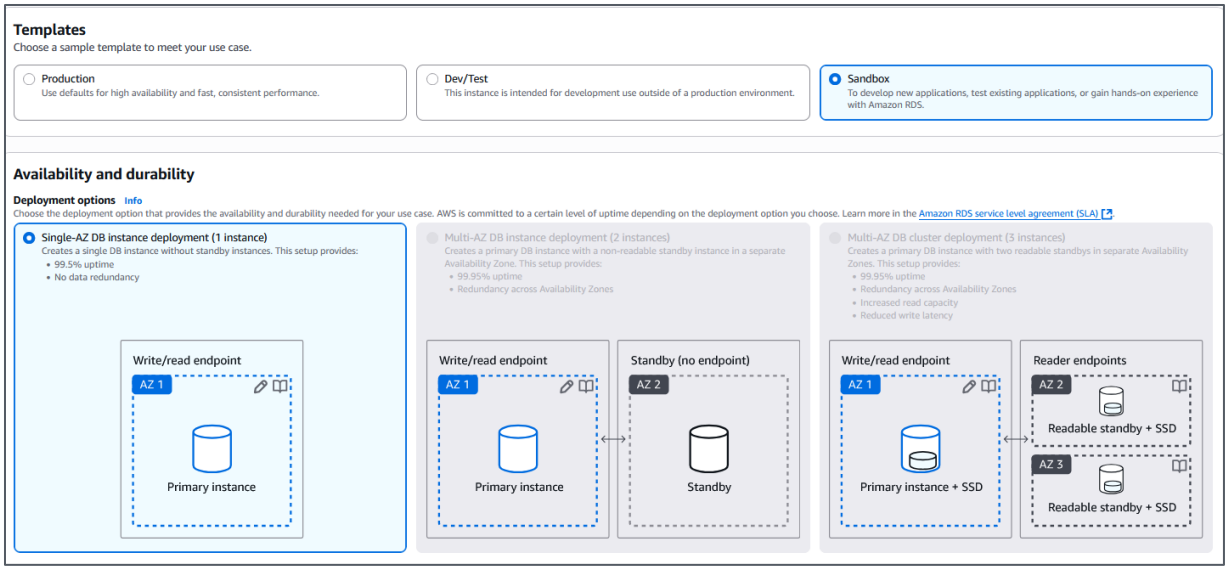


c. In the **Create a database** section, choose **Create a database**.



d. You now have options to select your engine. For this article, choose the MySQL icon, leave the default value of edition and engine version, and select the **Sandbox** template.

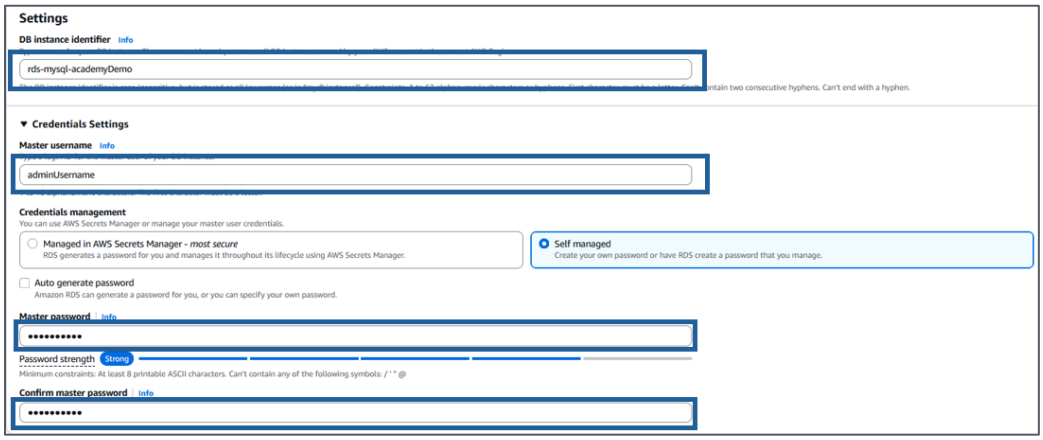


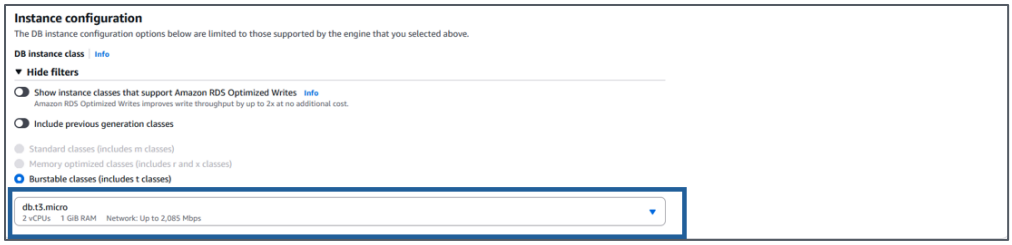


e. You will now configure your DB instance. The list below shows the example settings you can use for this article:

Settings:

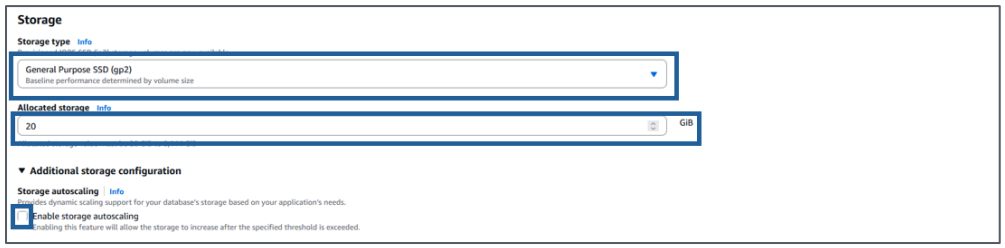
- **DB instance identifier:** Type a name for the DB instance that is unique for your account in the Region that you selected. For this article, we will name it **rds-mysql-academyDemo.**
- **Master username**: Type a username that you will use to log in to your DB instance. We will use adminUsername in this example.
- **Credentials management:** Choose Self managed option**,** then specify your
    - **Master password:** Type a password that contains from 8 to 41 printable ASCII characters (excluding /,", and @) for your master user password.
    - **Confirm password:** Retype your password





Instance specifications:

- **DB instance class**: Select **db.t3.micro** — 2 vCPU, 1 GiB RAM. This equates to 1 GB memory and 2 vCPU.
- **Storage type:** Select General Purpose SSD (gp2).
- **Allocated storage**: Select 20 to allocate 20 GB of storage for your database. Please check the LL service restrictions for Amazon RDS before you scale up.
- **Enable storage autoscaling**: If your workload is cyclical or unpredictable, you would enable storage autoscaling to enable Amazon RDS to automatically scale up your storage when needed. This option does not apply to this demo.
- **Multi-AZ deployment:** Using a Multi-AZ deployment will automatically provision and maintain a synchronous standby replica in a different Availability Zone. This feature is not available for **Sandbox** template.



f. You are now in the **Connectivity** section where you can provide information that Amazon RDS needs to launch your MySQL DB instance. The list below shows settings for our demoDB instance.

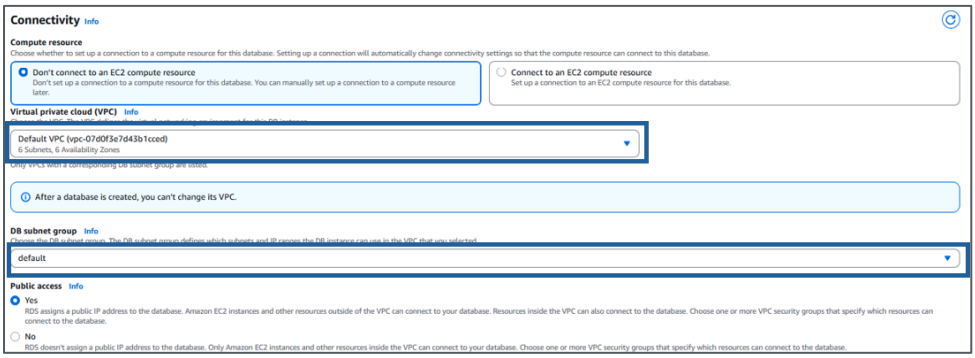
Connectivity

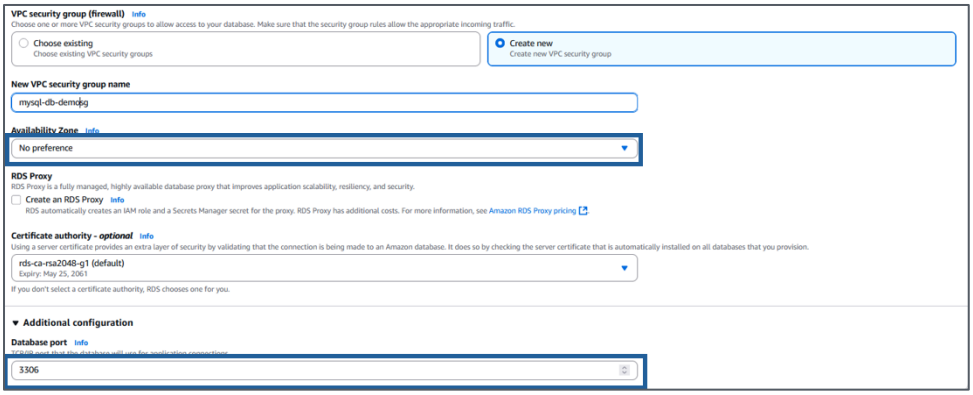
- **Compute resource**: Choose **Don't connect to an EC2 compute resource.** You can manually set up a connection to a compute resource later.
- **Virtual Private Cloud (VPC)**: Select **Default VPC**.
- **DB subnet group**: Choose the **default** subnet group.

- **Public accessibility**: Choose **Yes.** This will allocate an IP address for your database instance so that you can directly connect to the database from your own device.
- **VPC security group (firewall)**: Select **Create new.** Then specify a name for your security group. This will create a security group that will allow connection from the IP address of the device that you are currently using to the database created.
- **Availability Zone**: Choose **No preference.**
- **RDS Proxy**: By using Amazon RDS Proxy, you can allow your applications to pool and share database connections to improve their ability to scale. Leave the **RDS Proxy** unchecked.

Additional connectivity configuration

- **Port**: Leave the default value of 3306.



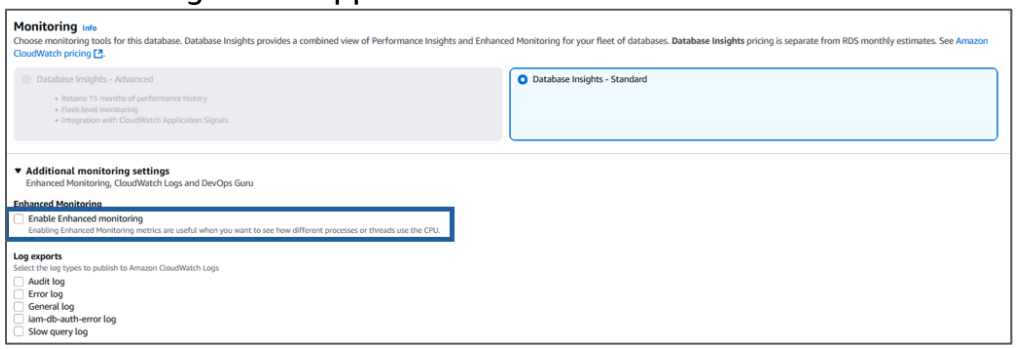


g. Amazon RDS supports several ways to authenticate database users. Select **Password authentication** from the list of options if it is not selected by default.

Monitoring

- The **Database Insight - Standard** is selected here for you by default because of the Sandbox template.

- **Enhanced monitoring**: Leave **Enable enhanced monitoring** unchecked as this setting is not supported within the LL.



In the **Additional configuration** section**:**

Database options

- **Database name**: Enter a database name that is 1 to 64 alphanumeric characters. If you do not provide a name, Amazon RDS will not automatically create a database on the DB instance you are creating.
- **DB parameter group**: Leave the default value.
- **Option group**: Leave the default value. Amazon RDS uses option groups to enable and configure additional features.
- **Encryption**: This option is not available in the LL. Therefore, uncheck the Enable encryption setting.

Backup

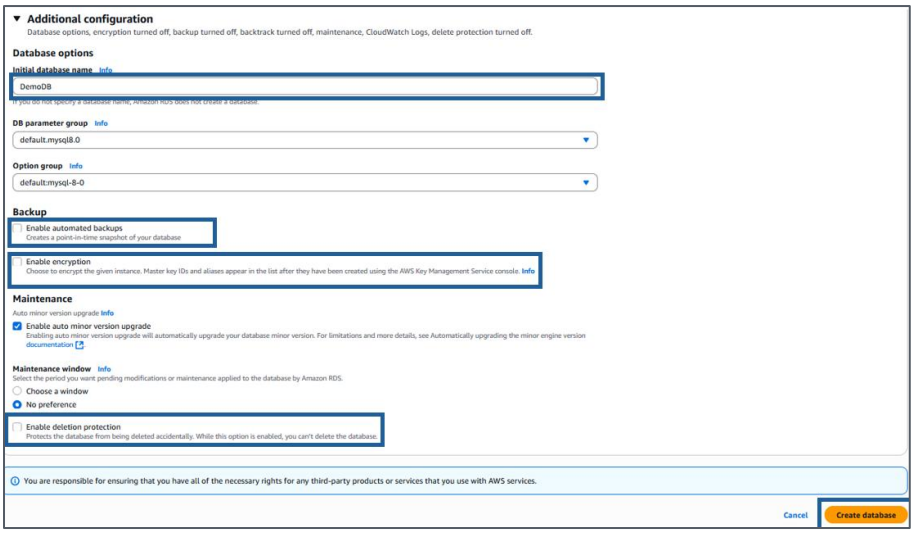
- **Enable automated backups**: For this article, we will uncheck this setting.

Maintenance

- **Auto minor version upgrade**: Select **Enable auto minor version upgrade** to receive automatic updates when they become available.
- **Maintenance Window**: Select **No preference.**

Deletion protection: Turn off **Enable deletion protection** for this article. When this option is enabled, you're prevented from accidentally deleting the database.

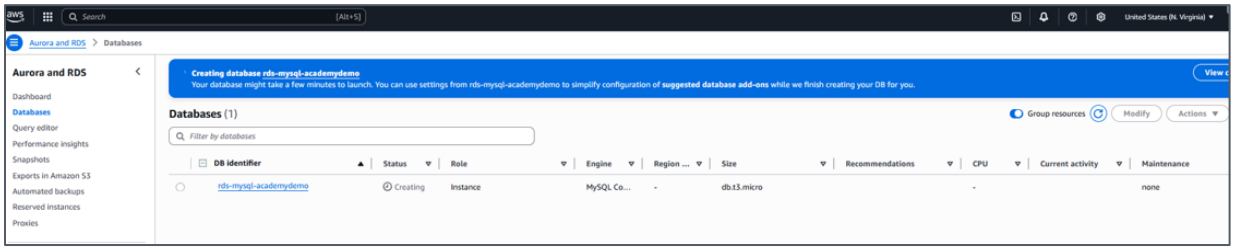
Choose **Create Database**.



h. Your DB instance is now being created.

Note: Depending on the DB instance class and storage allocated, it could take several minutes for the new DB instance to become available.

The new DB instance appears in the list of DB instances on the RDS console. The DB instance will have a status of **creating** until the DB instance is created and ready for use. When the state changes to **available,** you can connect to a database on the DB instance. Feel free to move on to the next step as you wait for the DB instance to become available.



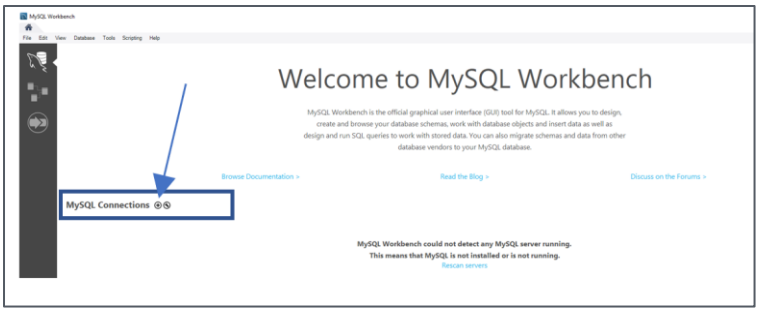
Connect to the Database using MySQL Workbench

Once the database instance creation is complete and the status changes to **available**, you can connect to the database on the RDS instance using any standard SQL client, such as MySQL

Workbench. If you do not have MySQL Workbench installed on your computer, please go to the Download MySQL Workbench page to download and install it.

In this step, we will connect to the database you created using MySQL Workbench.

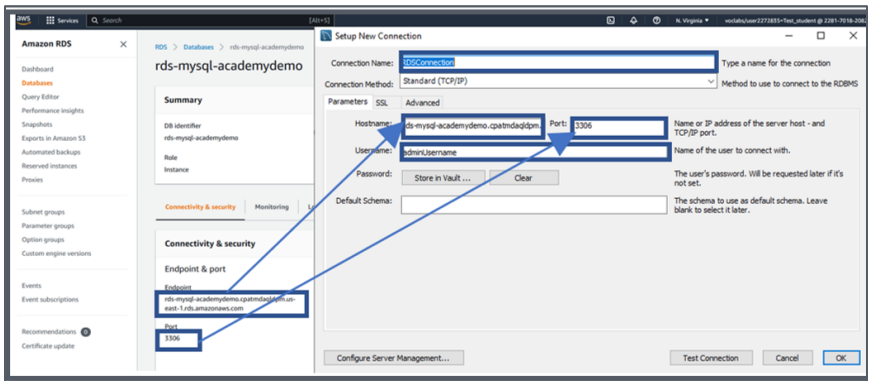
a. Launch the MySQL Workbench application and go to **MySQL Connections > +** from the home screen.



b. A dialog box appears. Enter the following:

- **Hostname**: You can find your hostname on the Amazon RDS console as shown in the screenshot.
- **Port**: The default value should be 3306.
- **Username**: Type in the username you created for the Amazon RDS database. In this article, it is 'adminUsername.'
- **Password**: Choose **Store in Vault** (or Store in Keychain on MacOS) and enter the password that you used when creating the Amazon RDS database.
- **Connection Name**: Please provide a suitable name for your connection (For example, RDSConnection). MacOS users might not need to specify a name for their connection.

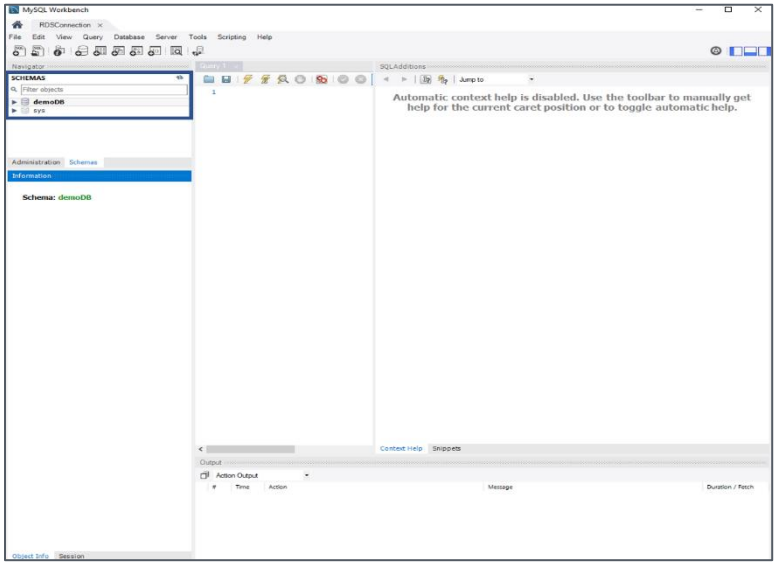
Choose **OK.**



Amazon RDS Console | MySQL Workbench

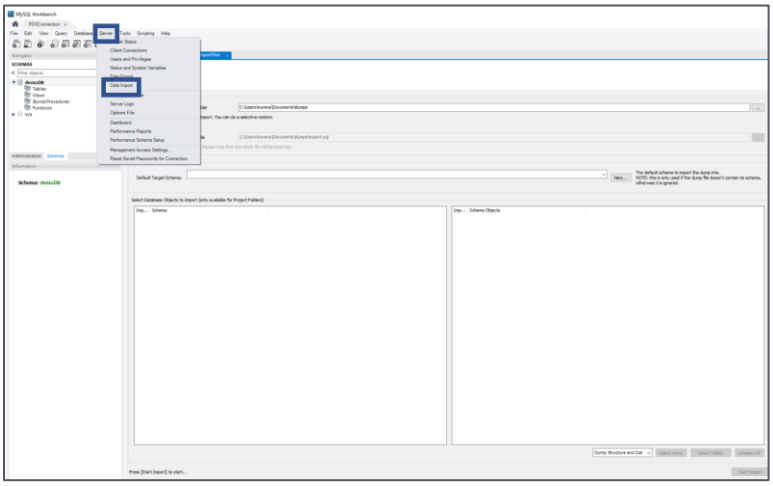
Create tables, insert data, and run queries

a. Congratulations!!! Now that you are connected to your database, you will see various schema objects available in the database. Now you can create tables, insert data, and run queries.

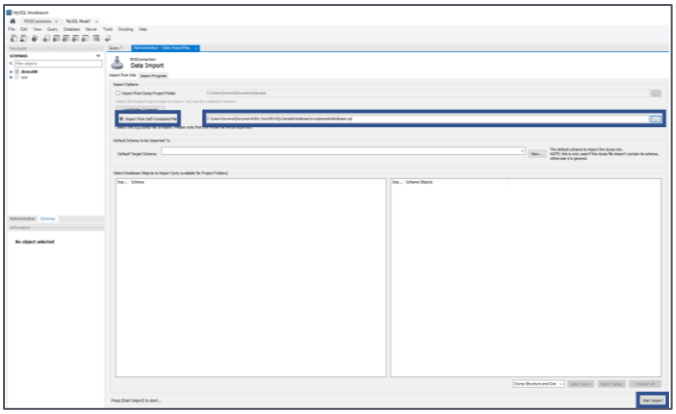


b. Let's create a new database, create some tables, and insert some data into the tables. The **classicmodels** database used in this demo is available on the [MySQL Sample Database website]. You can download the database and import it into your MySQL DB instance.

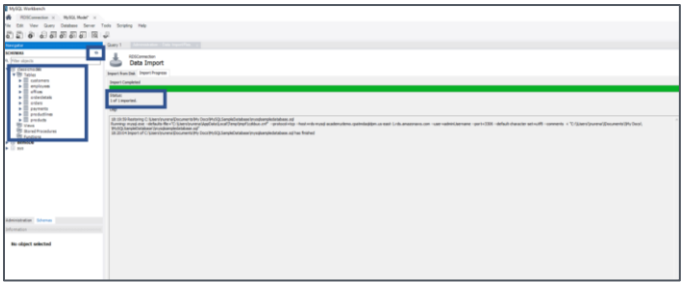
c. To import the **classicmodels** script into your database, Go to **Server** > **Data Import** from the menu bar.



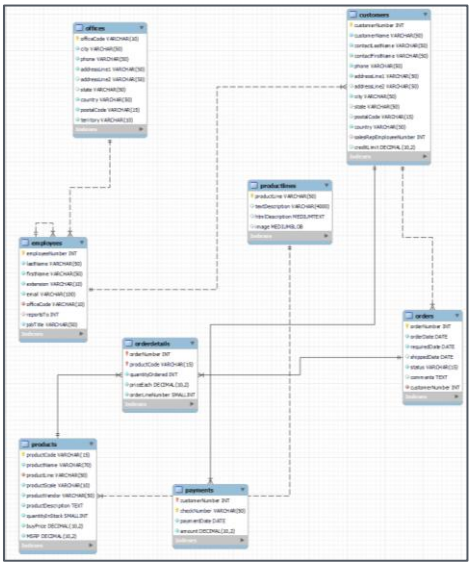
d. A **Data Import** box appears. Select **Import from self-contained File,** then click the button in front of it to choose the **classicmodels** script you downloaded to your computer from step b. Click **Start Import** to import the script.



e. Once the data has imported successfully, **refresh** the **SCHEMAS** and a new **schema object** (a database) will appear.

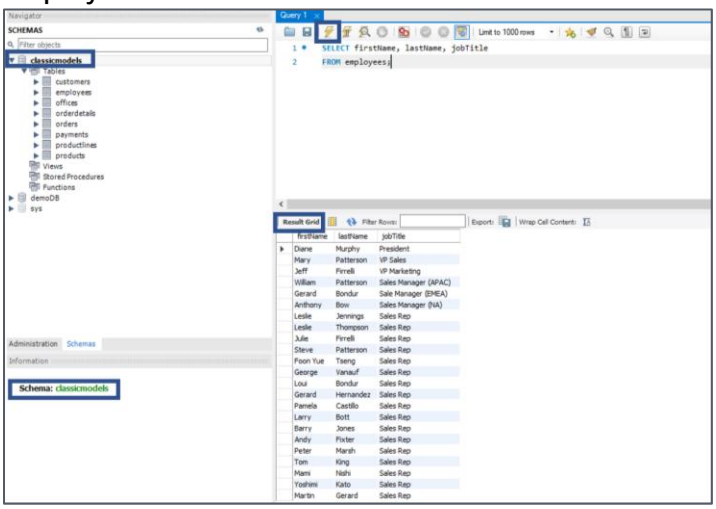


f. If you like, you can use the MySQL Workbench **Reverse Engineer Database** Wizard to create the Entity Relationship diagram (ERD) of your database. Go to **Database** menu **> Reverse Engineer Database > Select your Stored Connection** (for example**,** RDSConnection) **> Click Next > Click Next > Check your database name** (classicmodels) **> Click Next > Click Next > Click Execute > Click Next > Click Finish**.

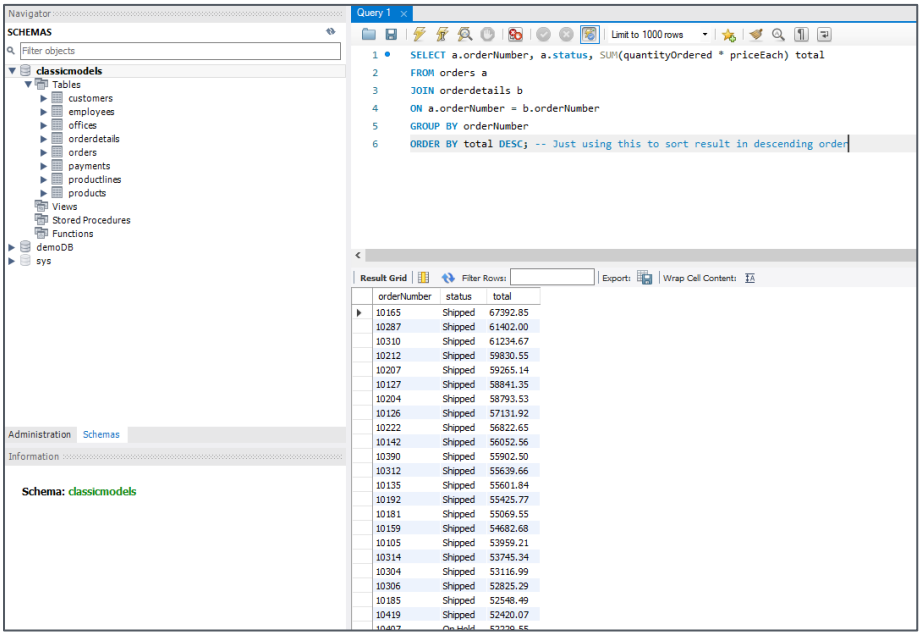


g. It is time to run some queries against your database.

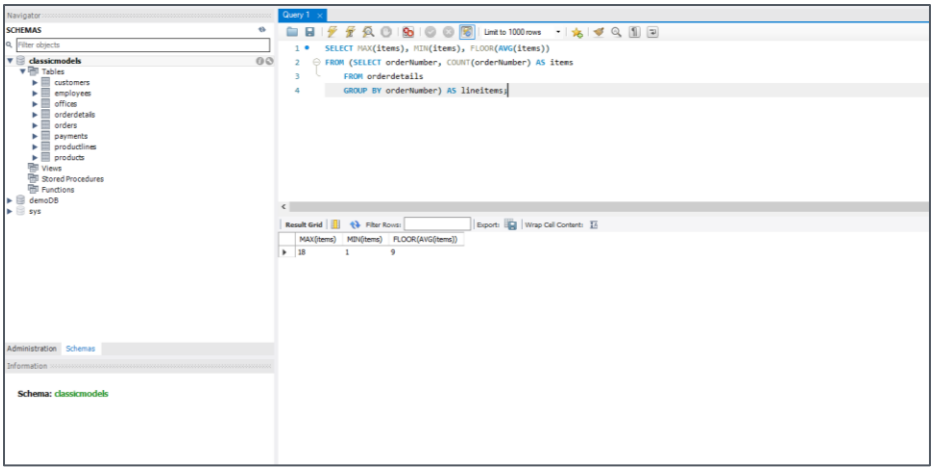
- **Query 1**: Write a query to get the first name, last name, and job title of all the employees:



- **Query 2:** Write a query that returns the order number, order status, and total sales per order. Note that there is no column named totals sales in either orders and orderdetails tables.



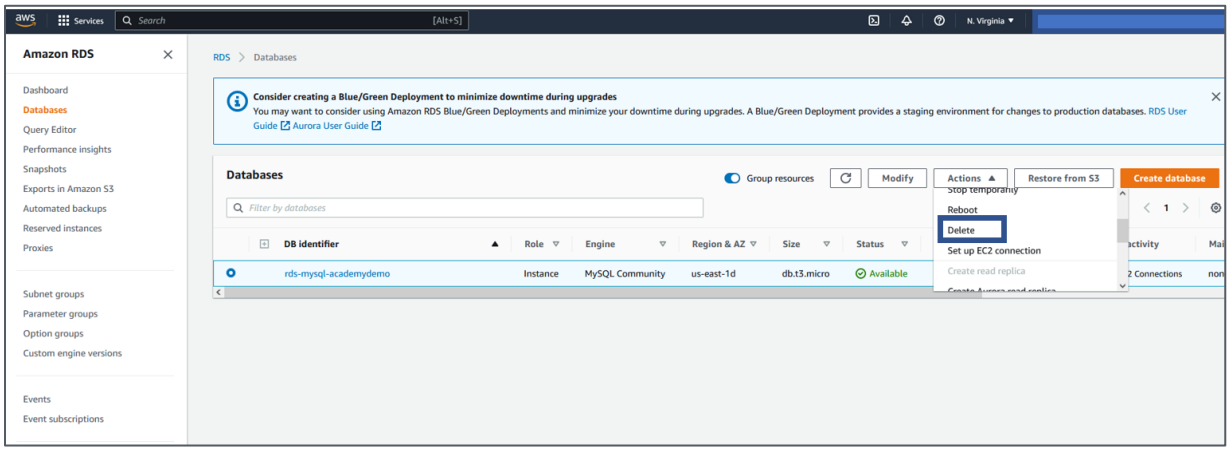
- **Query 3:** Write a query that returns the maximum, minimum, and average number of items in sales orders.



Delete the database instance

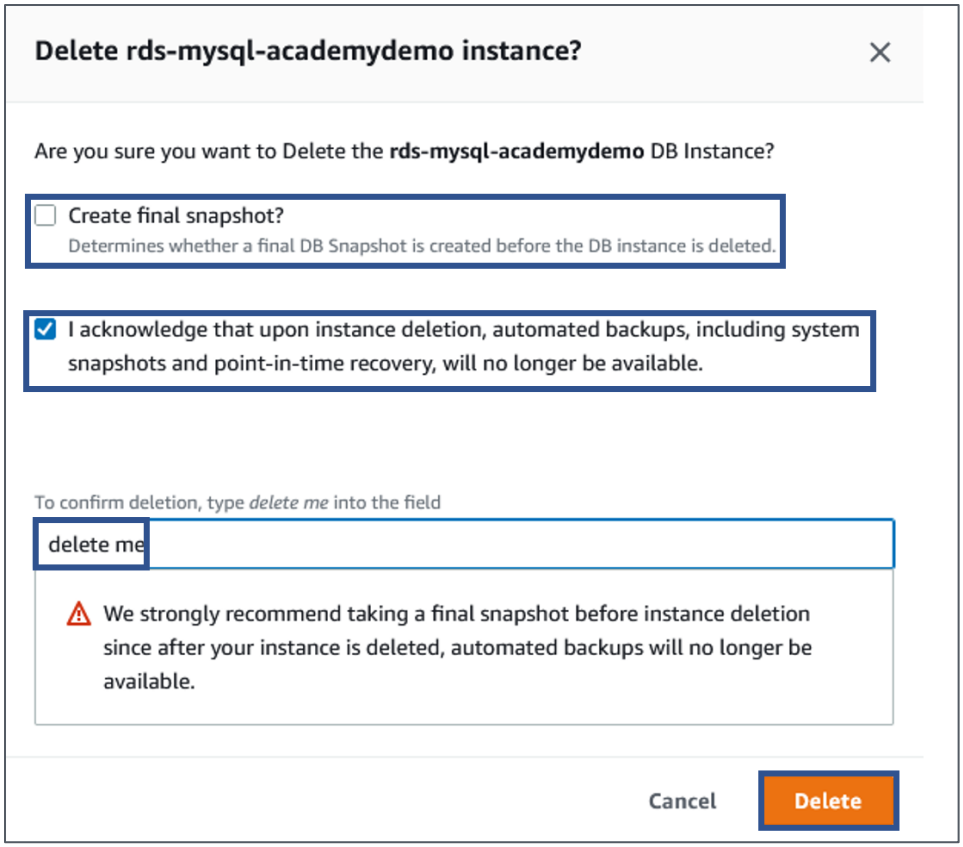
It is a best practice to delete instances that you are no longer using so that you don't keep getting charged for them.

a. Go back to the Amazon RDS console. Select **Databases**, choose the instance that you want to delete, and then select **Delete** from the **Actions** dropdown menu.



b. You are asked to create a final snapshot and to confirm the deletion. In this case, do not create a final snapshot, acknowledge that you want to delete the instance, and then choose Delete.

Note: Deleting your DB instance may take a few minutes.


Delete rds-mysql-academydemo instance?
Are you sure you want to Delete the rds-mysql-academydemo DB Instance?
Create final snapshot?
Determines whether a final DB Snapshot is created before the DB instance is deleted.
I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.
To confirm deletion, type delete me into the field
delete me
We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available.
Cancel
Delete