**😊 Amazon Aurora Serverless**

Amazon Aurora Serverless is a configuration option for the Amazon Aurora relational database offered by Amazon Web Services (AWS). Unlike traditional Aurora instances, which require you to provision and manage compute capacity, Aurora Serverless automatically adjusts database capacity based on actual usage.

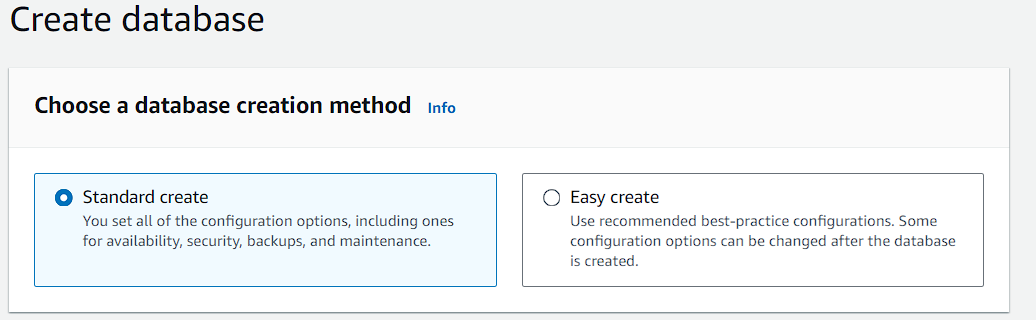
The key features of Amazon Aurora Serverless include:

1. **Auto-scaling:** Aurora Serverless automatically adjusts database capacity up or down based on application workload. This means you don't need to manually provision or manage database instances, leading to cost savings and simplified management.
2. **Pause and Resume:** Aurora Serverless allows you to pause your database during periods of inactivity, reducing costs further by eliminating charges for unused capacity. You can then resume the database when needed, with capacity scaling up automatically as the workload increases.
3. **High Availability:** Just like standard Aurora, Aurora Serverless provides high availability with data replicated across multiple Availability Zones within a region, ensuring durability and fault tolerance.
4. **Pay-per-Use Pricing:** With Aurora Serverless, you pay only for the resources consumed by your database, based on the capacity and duration of usage. This flexible pricing model aligns costs with actual application demand, making it cost-effective for variable workloads.
5. **Compatibility:** Aurora Serverless is compatible with MySQL and PostgreSQL, offering the same performance, scalability, and reliability benefits as standard Aurora databases.

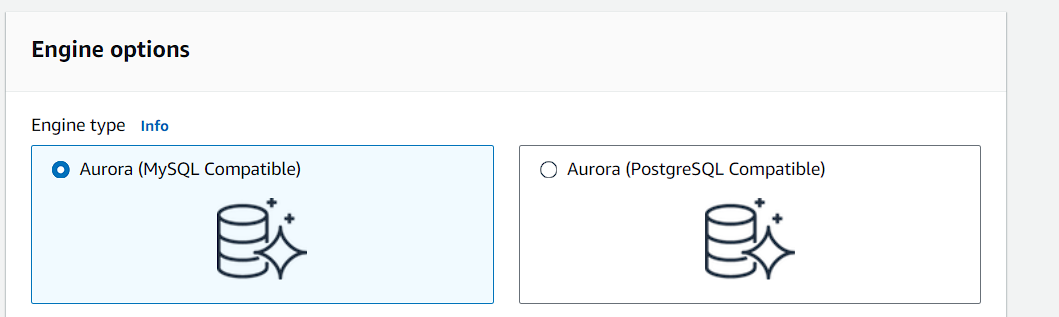
**In this lab, we're learning how to launch an Amazon Aurora Serverless instance in the AWS console. The end goal is to gain hands-on experience in setting up a serverless relational database with Amazon Aurora, allowing for auto-scaling, cost-efficient resource utilization, and seamless connectivity. We'll create a database instance, establish a connection using MySQL Workbench, and perform basic operations like creating a database, defining tables, inserting data, and querying data. Finally, we'll clean up by deleting the database instance to avoid ongoing costs.**

**😄 To begin with the Lab:**

1. In this lab you are going to learn how to launch Amazon Aurora Serverless.
2. For that login to the AWS console and navigate to RDS then click on create database.
3. Choose a standard for creation method.



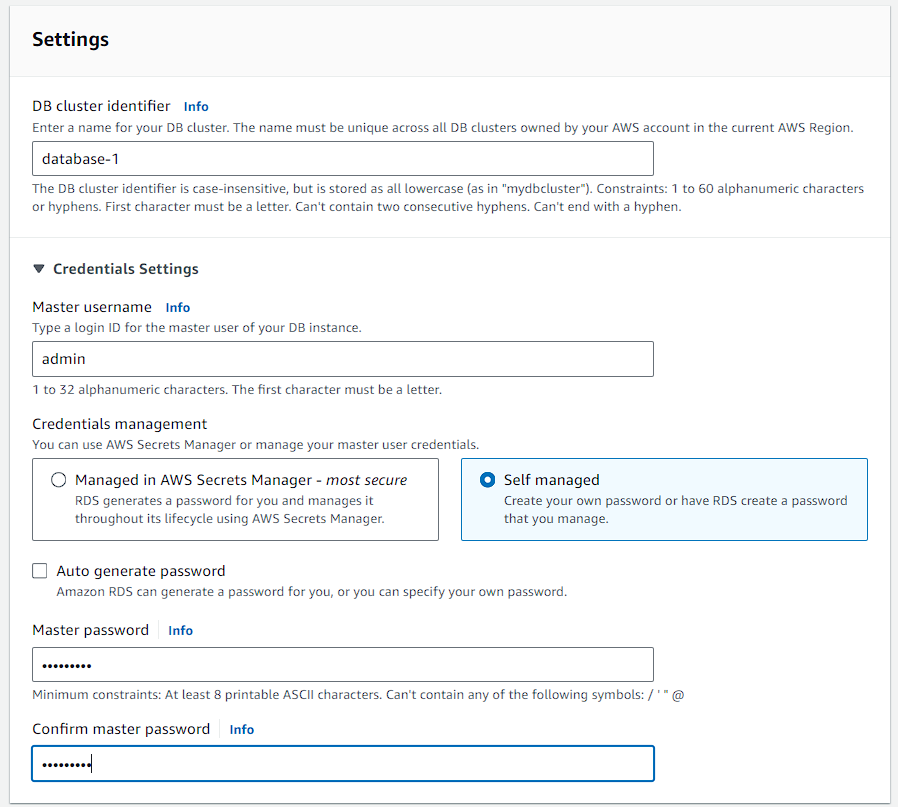
1. Then in the engine options choose Amazon Aurora and select the latest engine version which is available.



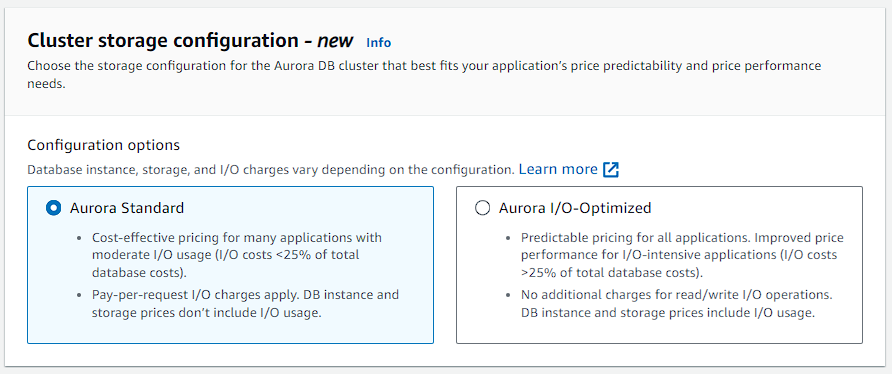
1. Choose Dev/Test for your template.



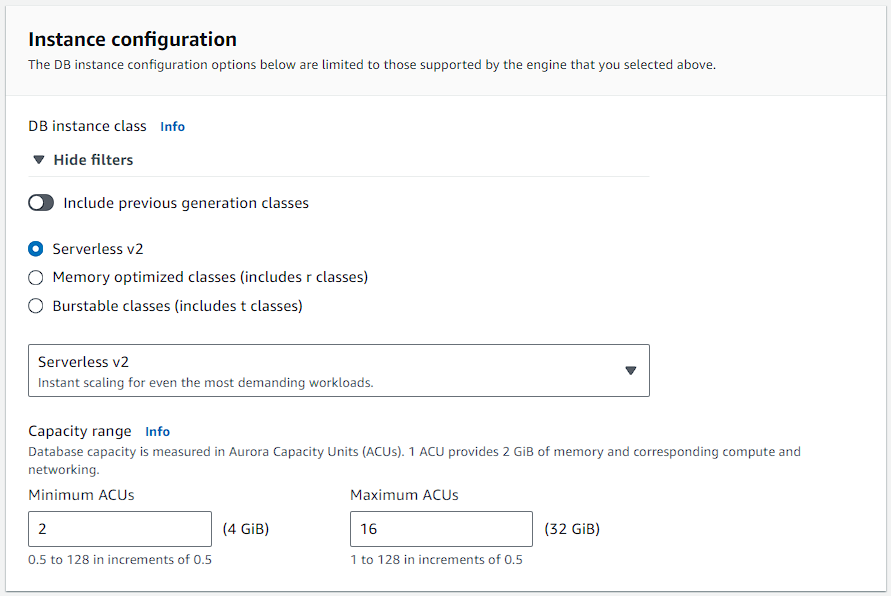
1. Then in the settings you can give your cluster identifier a name or you can leave it to default.
2. Then give it a master username and then select self-managed password and give a password of your choice.



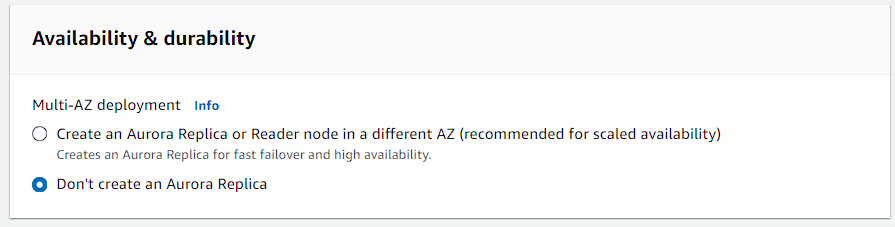
1. For cluster storage choose the default Aurora standard.



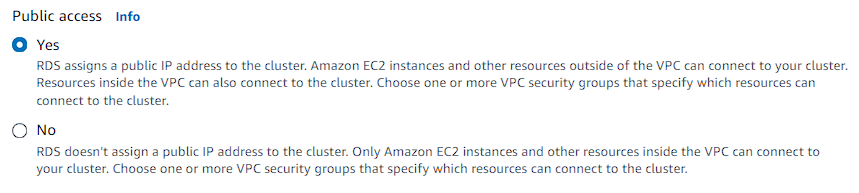
1. In the instance configuration choose serverless and keep it to default for capacity range.



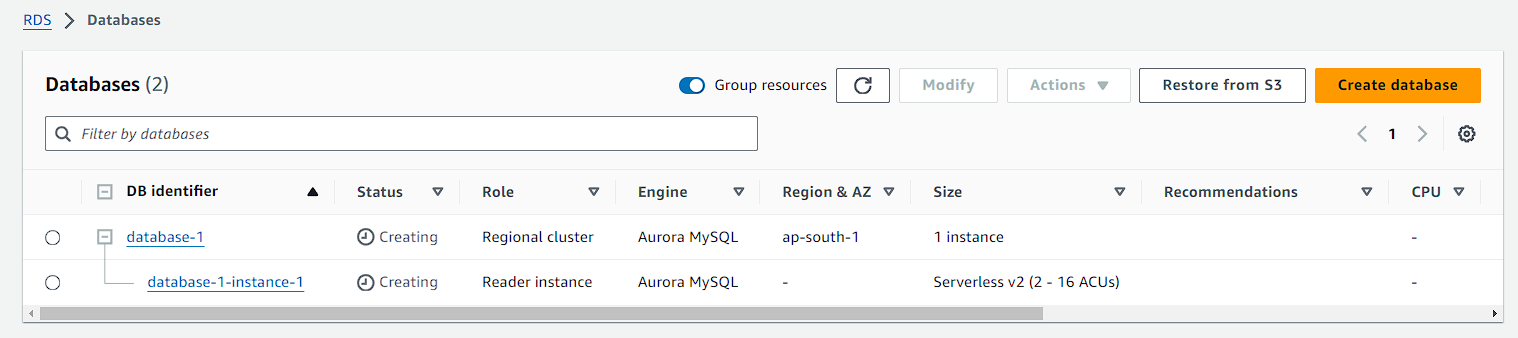
1. Keep it to default in availability and durability as we don’t need any replica for now.



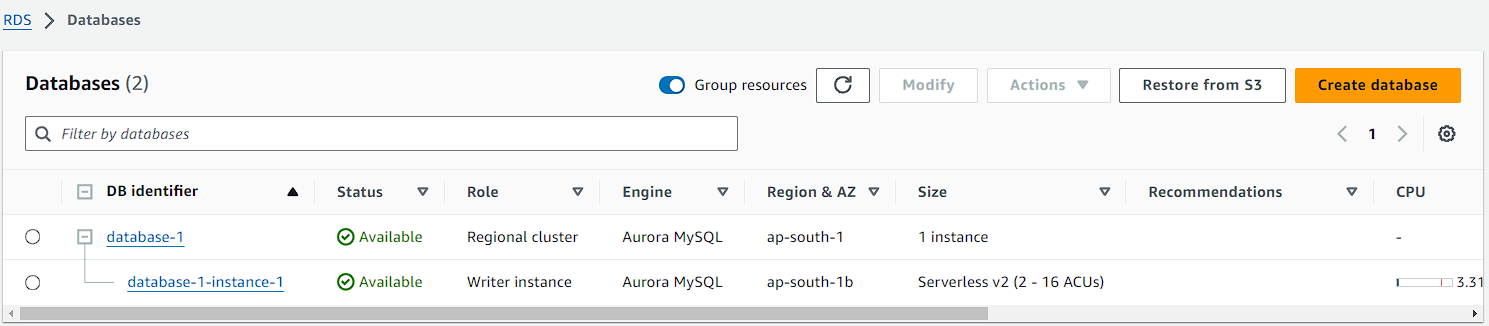
1. Now in the connectivity keep everything to default just make your RDS publicly accessible so that you can connect to it.



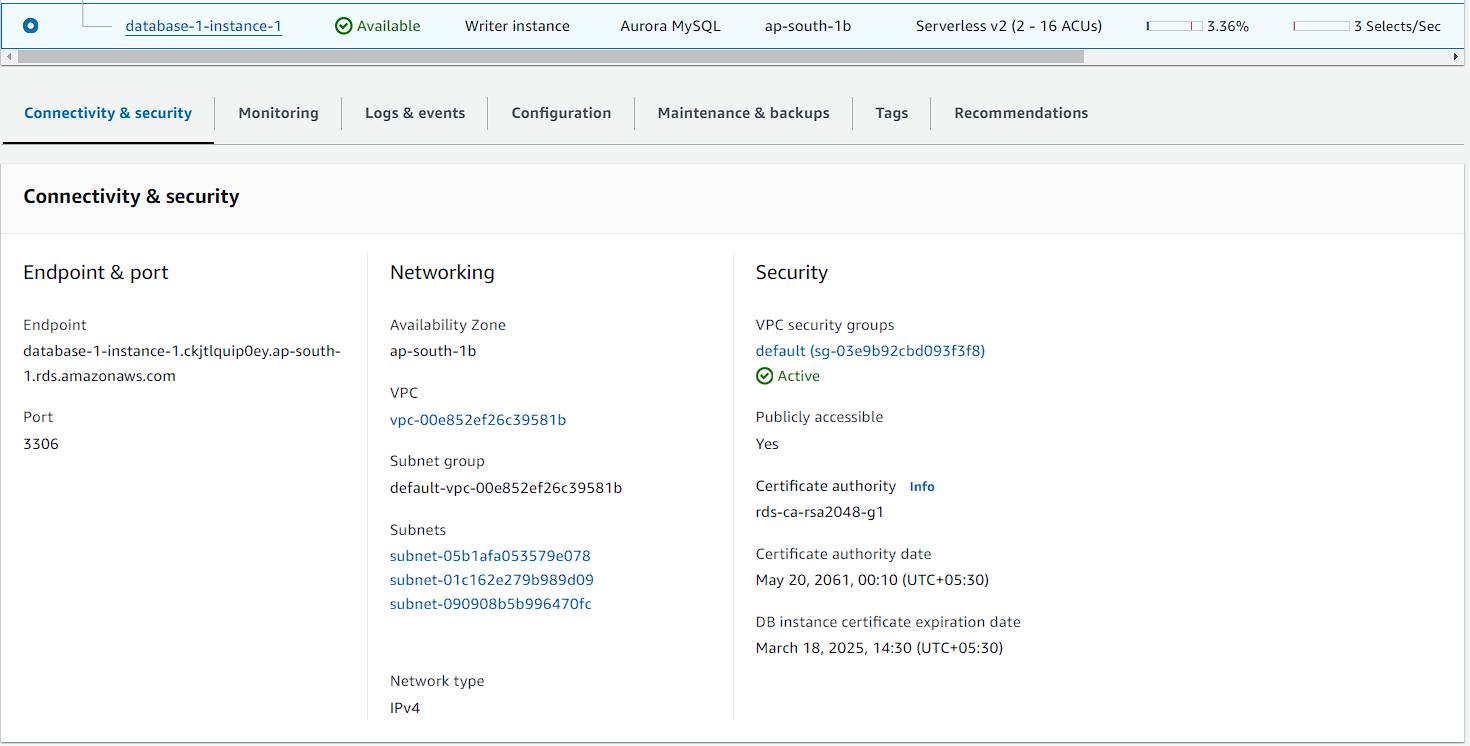
1. With that you can also disable enhanced monitoring and then create your instance.
2. Now you have to wait for some time because it will take several minutes to create your database.



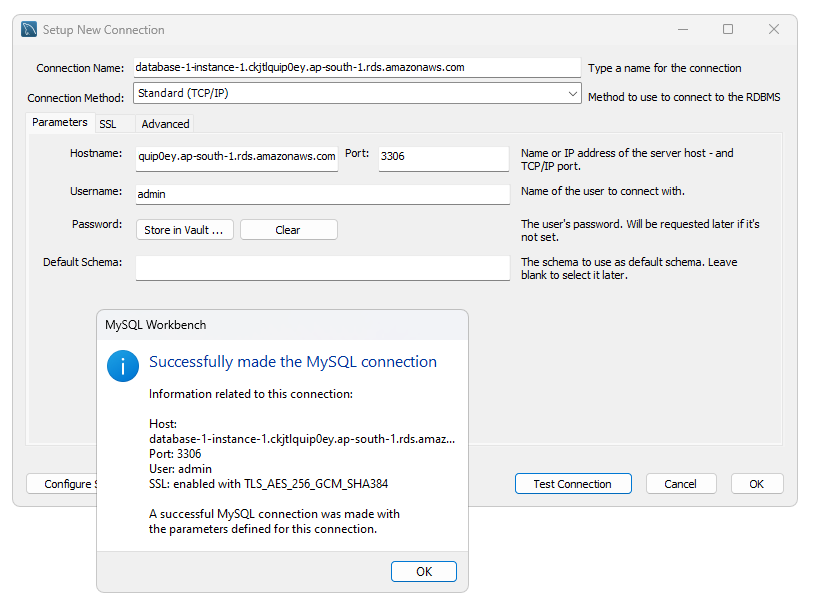
1. Once your database instance is launched open it.



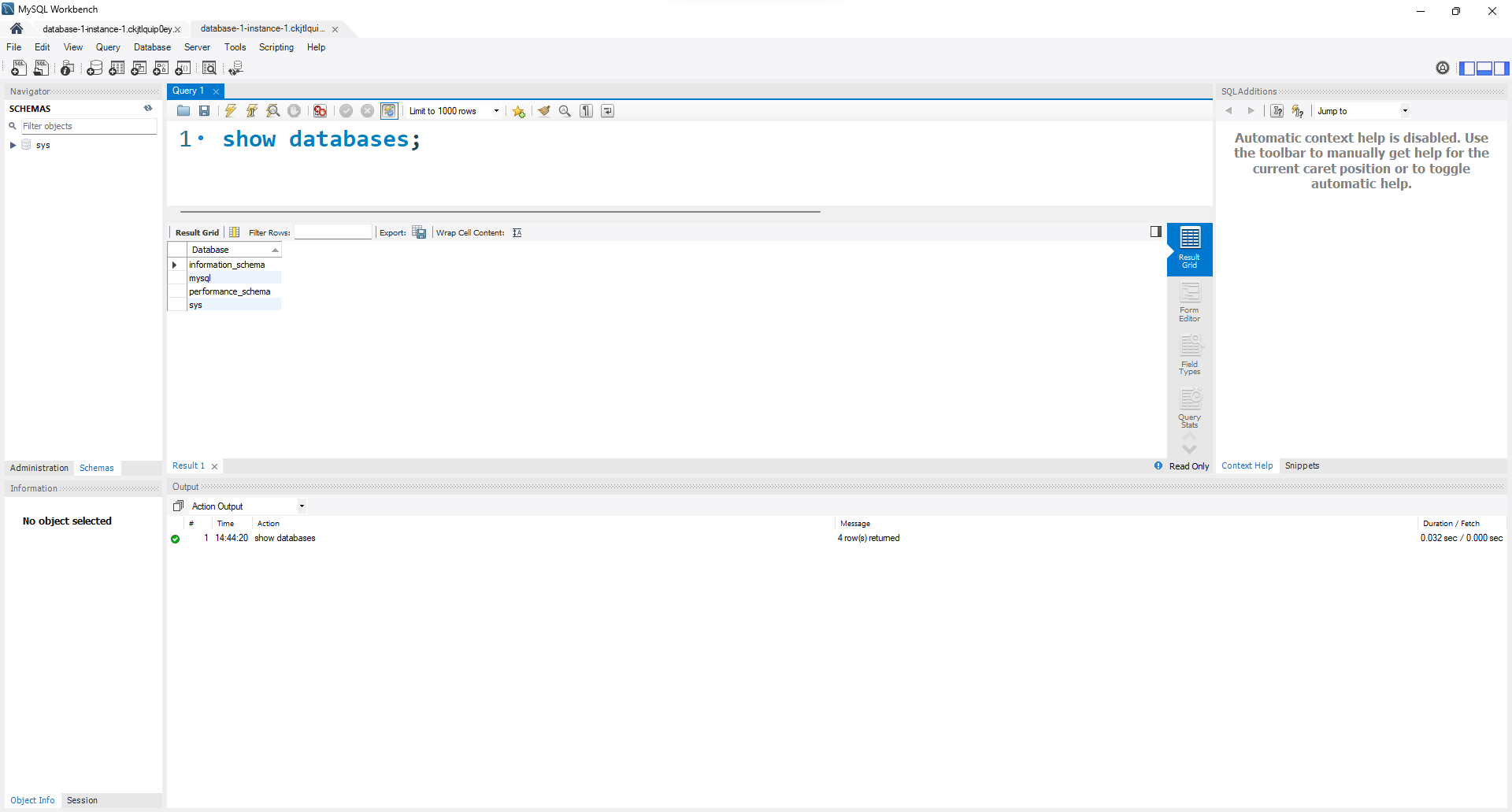
1. There you will be seeing the Endpoint to connect your database. Now copy the endpoint.



1. Open the MySQL Workbench and try to establish a connection with your database instance.



1. Once the connection is established successfully now open it and run the basic query to show your databases. You will be able to see a database.



1. Now you can also create a database and table from scratch. You can use the below SQL commands to create a table and see your output.

**CREATE DATABASE IF NOT EXISTS my\_database;**

**USE my\_database;**

**CREATE TABLE IF NOT EXISTS products (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**name VARCHAR(255),**

**price DECIMAL(10, 2)**

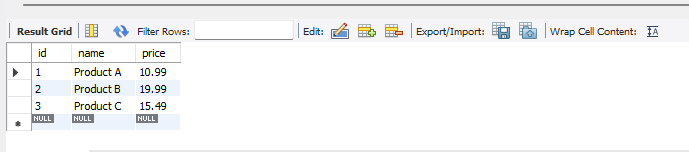
**);**

**INSERT INTO products (name, price) VALUES ('Product A', 10.99);**

**INSERT INTO products (name, price) VALUES ('Product B', 19.99);**

**INSERT INTO products (name, price) VALUES ('Product C', 15.49);**

**SELECT \* FROM products;**



1. Once you are done delete your database and instance.