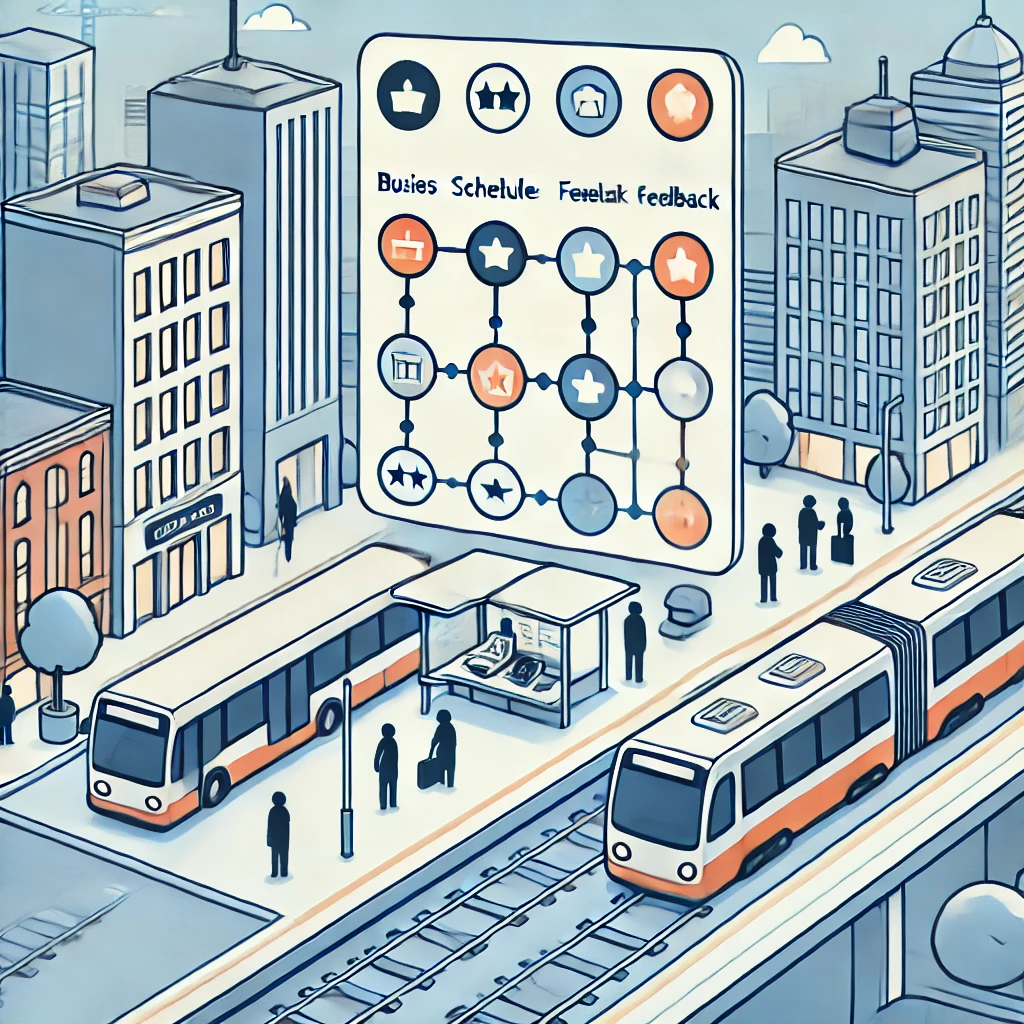
**Streamlining Public Transport Schedules for a Smarter City**

**Problem Statement:**

Hyderabad Urban Mobility Solutions (HUMS) aims to enhance the city's public transportation system by developing a centralized MySQL database using AWS RDS to store and manage bus and train schedules. The database will include route numbers, timings, starting and ending locations, and service frequency, providing users with easy access to accurate, real-time information. This initiative will also support the collection of feedback and complaints, helping improve the quality and punctuality of public transport services. By streamlining schedule management, HUMS seeks to improve commuter experiences and enable data-driven decisions for future transit improvements.



**Pre-requisites:**

### 1. AWS Account Setup: [https://youtu.be/CjKhQoYeR4Q?si=ui8Bvk\_M4FfVM-D](https://youtu.be/CjKhQoYeR4Q?si=ui8Bvk_M4FfVM-Dh)h

### 2. Understanding of IAM: <https://youtu.be/gsgdAyGhV0o?si=3qg-bULgkD4LXNvR>

### 3. Basic Knowledge of Amazon S3: <https://youtu.be/bktTomENEX8?si=dsJImPcPJrZY50CE>

4. Introduction to Amazon RDS: <https://youtu.be/ovK5JL_JyUg?si=lqBkJVpBuc-3M6OD>

**Objective:**

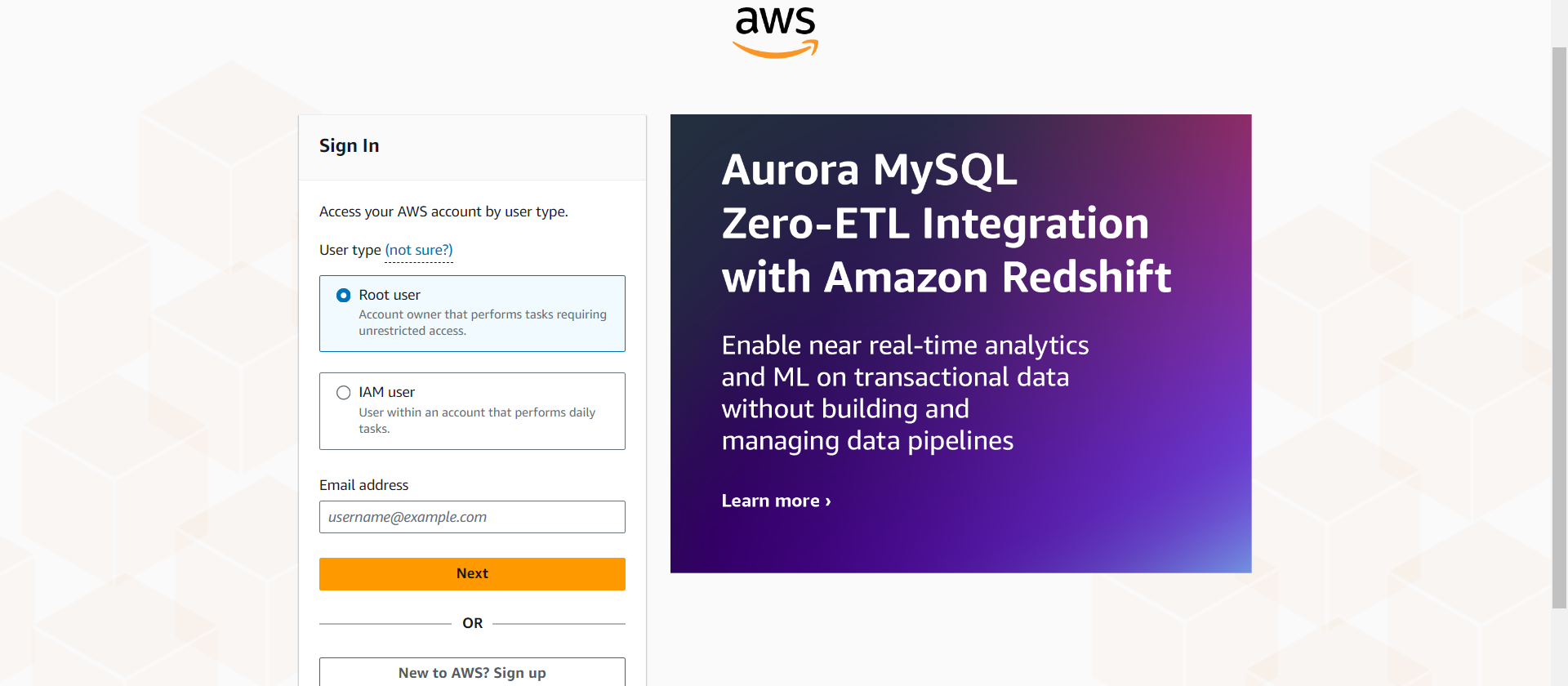
The objective of the *Hyderabad Urban Mobility Solutions (HUMS)* project is to create a centralized MySQL database using AWS RDS to manage bus and train schedules, providing commuters with real-time route information. The system will also allow users to submit feedback and complaints, helping improve service quality. Key services include AWS RDS for the database, AWS EC2 for hosting the web application, and MySQL Workbench for managing the schedules. The project aims to streamline public transport operations, making it easier for city planners and enhancing the commuter experience.

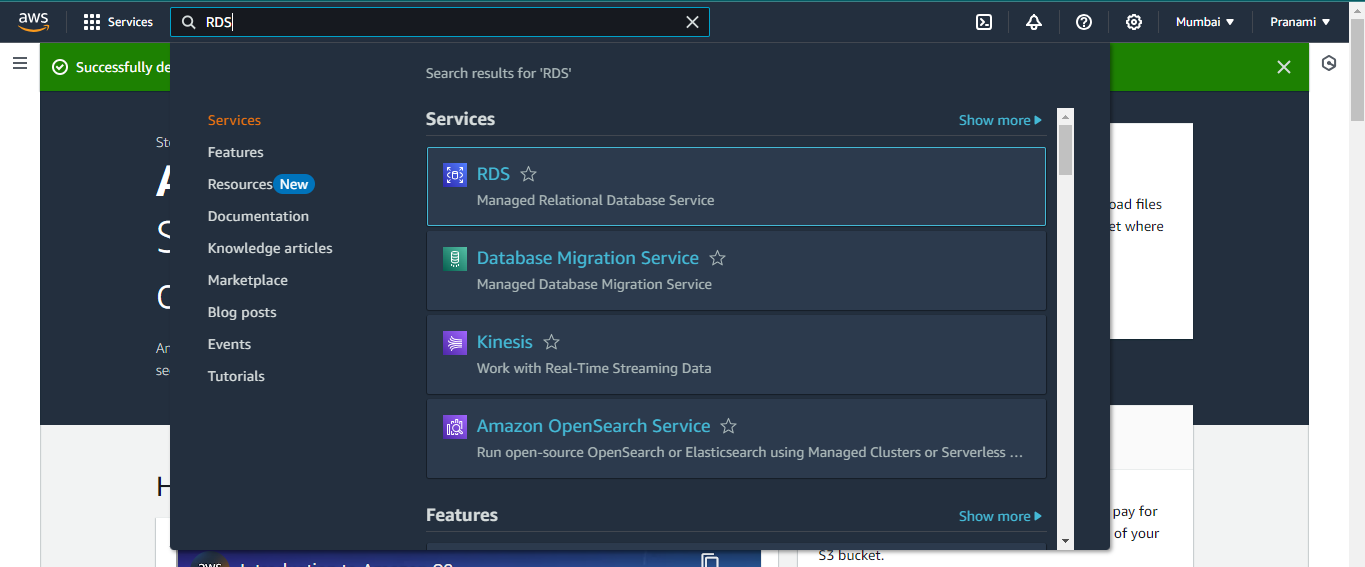
**Tasks:**

1. Login to AWS Management Console.
2. Launch an Amazon RDS Instance
3. Configure Database Settings
4. Configure Networking and Security
5. Launch the RDS Instance
6. Connect to the RDS Database
7. Perform Basic Database Operations in MySQL Workbench

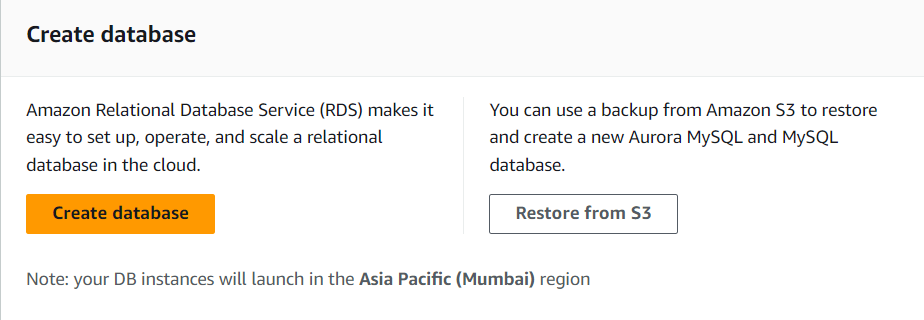
**Solution Development Procedure:**

1. Log in to the AWS Management Console and navigate to the S3 service.



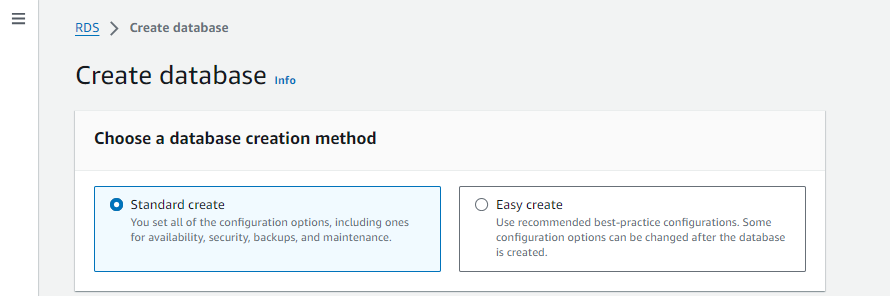


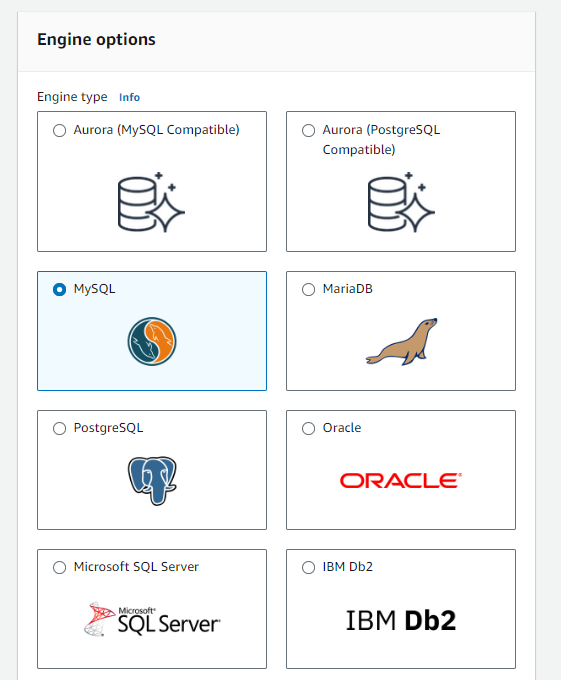
1. Launch an Amazon RDS Instance



Create a Database:

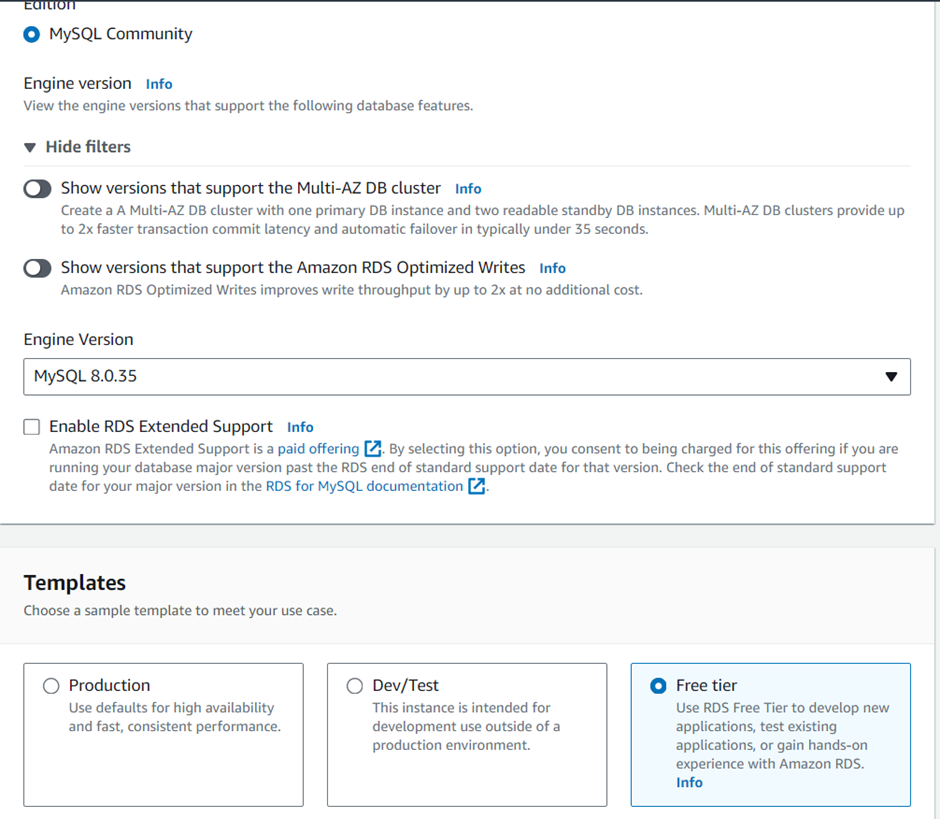
* Click on Create database.
* Select a Standard Create option for more customization options.
* Choose Database Engine
* Select the engine for your database, such as MySQL, PostgreSQL, MariaDB, or Amazon Aurora.





Version and Templates:

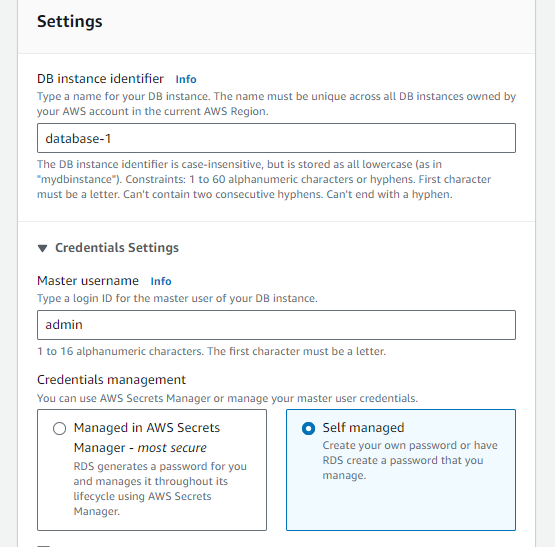
* Choose the database version (latest stable version recommended).
* Select the Free tier template for minimal costs (or Production/Dev depending on your use case).

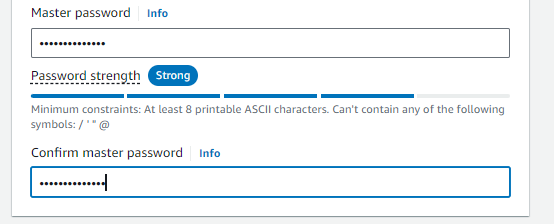


1. Configure Database Settings

DB Instance Settings:

* Set the DB instance identifier (name for your database).
* Choose a Master username and set a Master password to secure your database.





Instance Class:

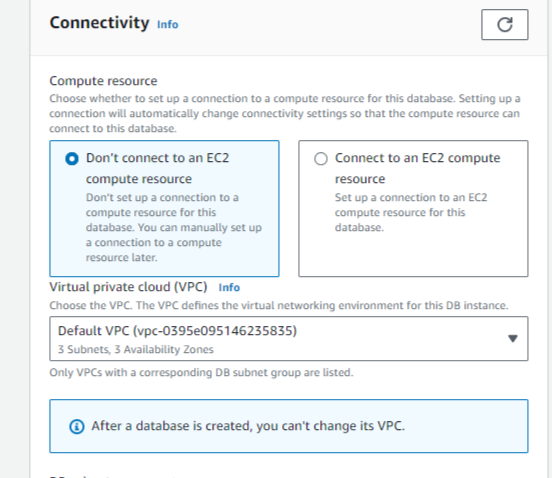
* Choose an instance class such as db.t3.micro (free tier eligible).
* Set up Storage options such as allocated storage (start with 20 GB, which is free-tier eligible).



1. Configure Networking and Security

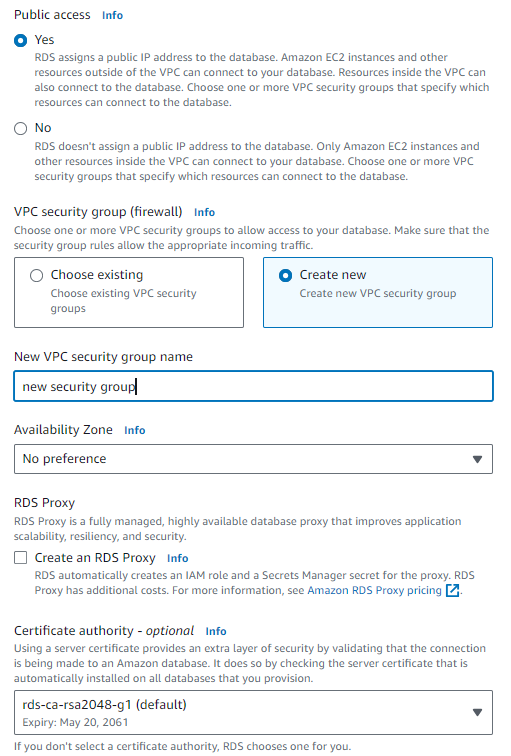
VPC & Subnet Group:

* Ensure that the database is launched in your desired VPC.
* Choose a subnet group for defining which subnets the database will be launched into.



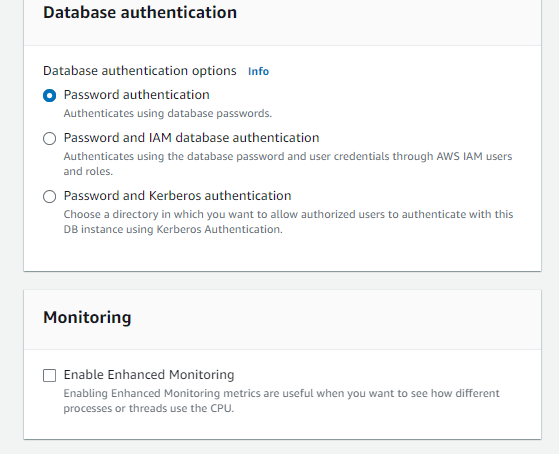
Public Access:

* Enable Public access only if you want the database to be accessible from the internet.
* For more security, it's best to disable public access and connect via an EC2 instance in the same VPC.



Database Authentication:

* Choose the default password authentication, or you can use IAM authentication for additional security.

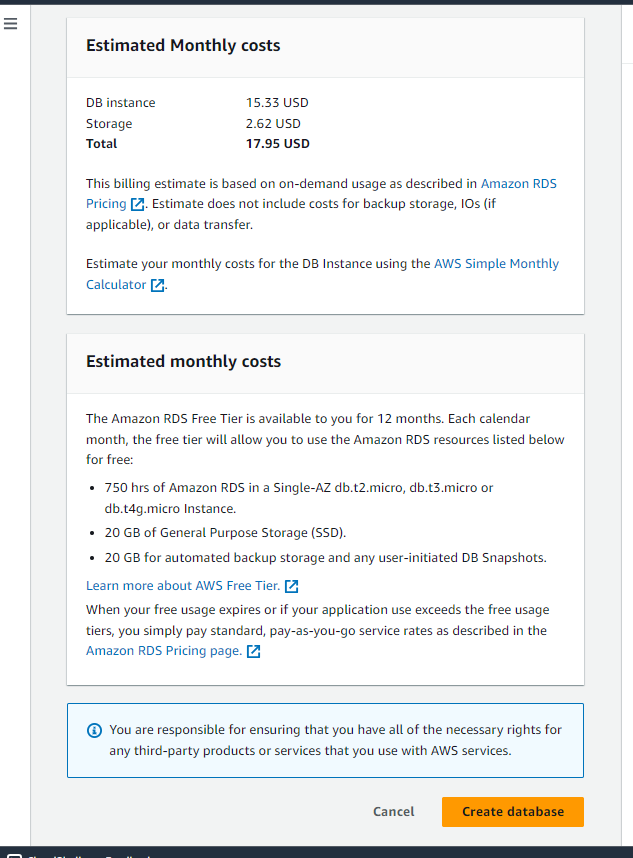


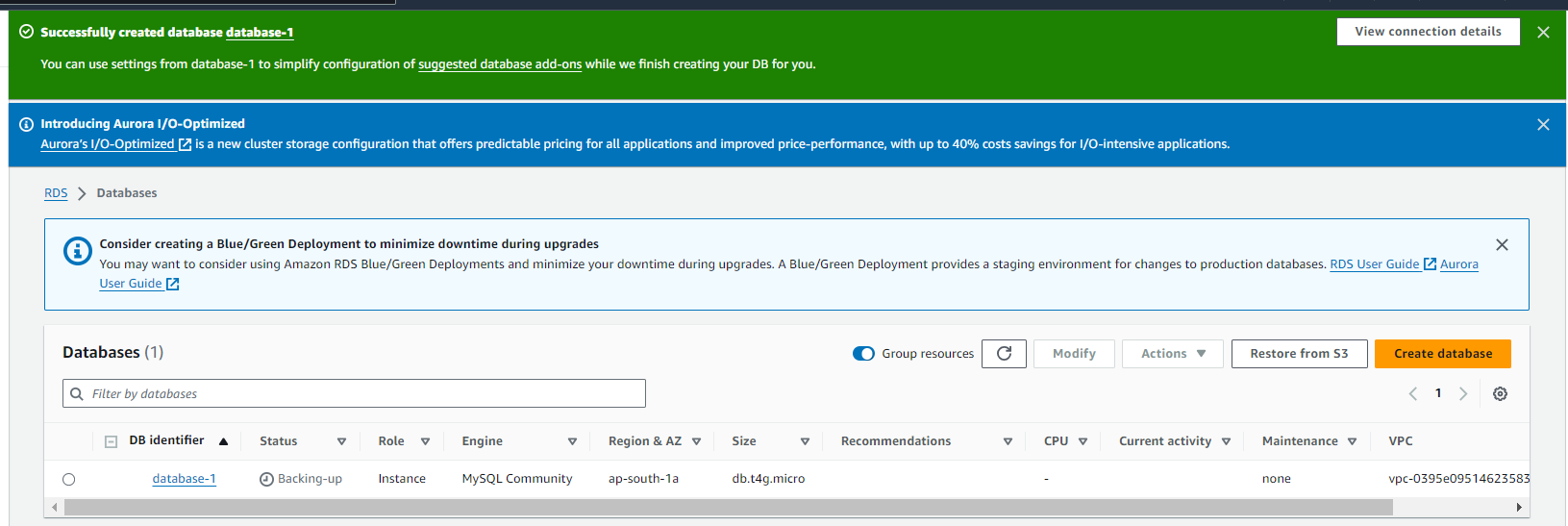
1. Launch the RDS Instance

Review and Launch:

* Review all the settings and configurations.
* Click on Create database to launch the RDS instance.

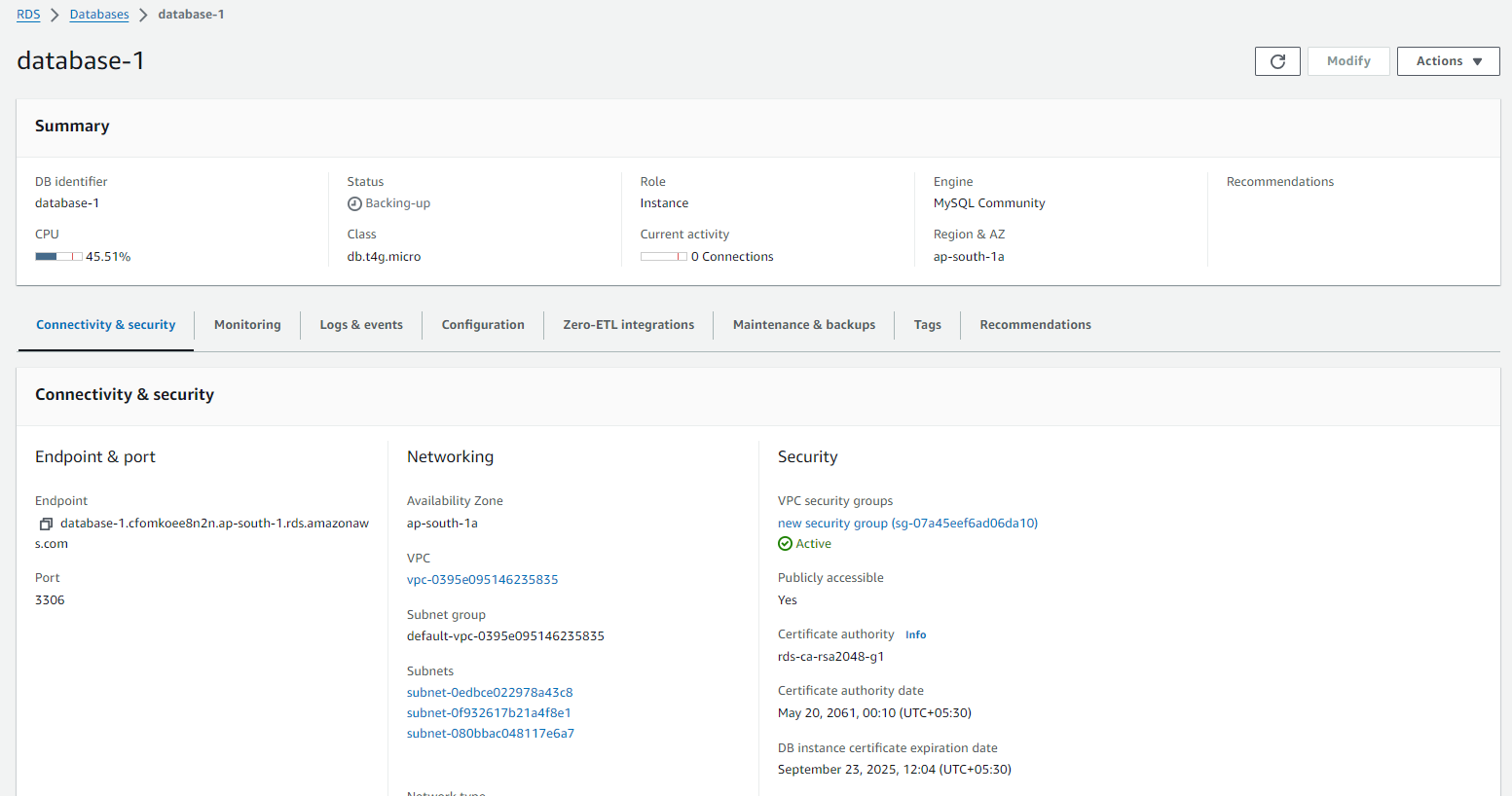
This process may take a few minutes as the RDS instance is being created and provisioned.





Obtain the Endpoint:

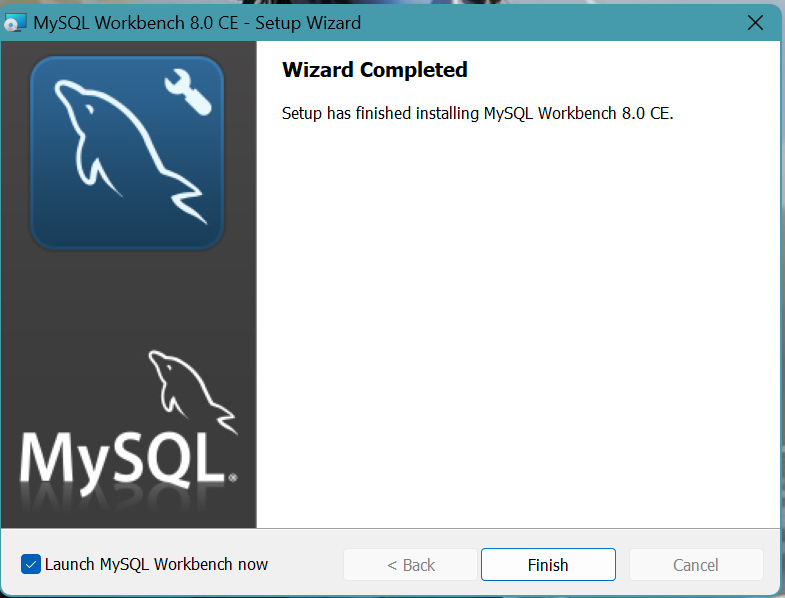
* After the database is created, go to the RDS Instances dashboard and find your DB instance.
* Copy the endpoint (the connection string for your database).



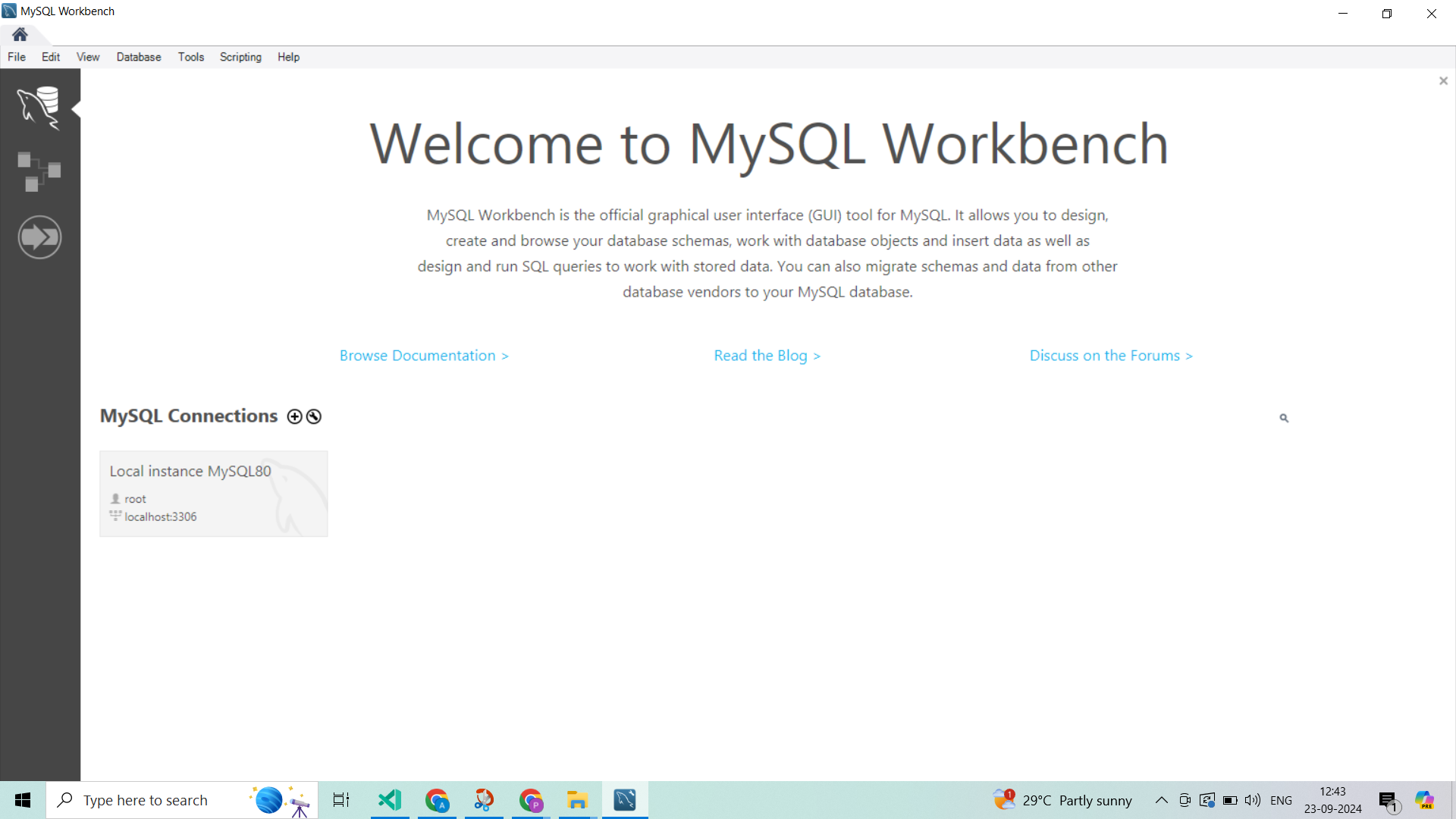
1. Connect to the RDS Database

Connect Using a MySQL Workbench:

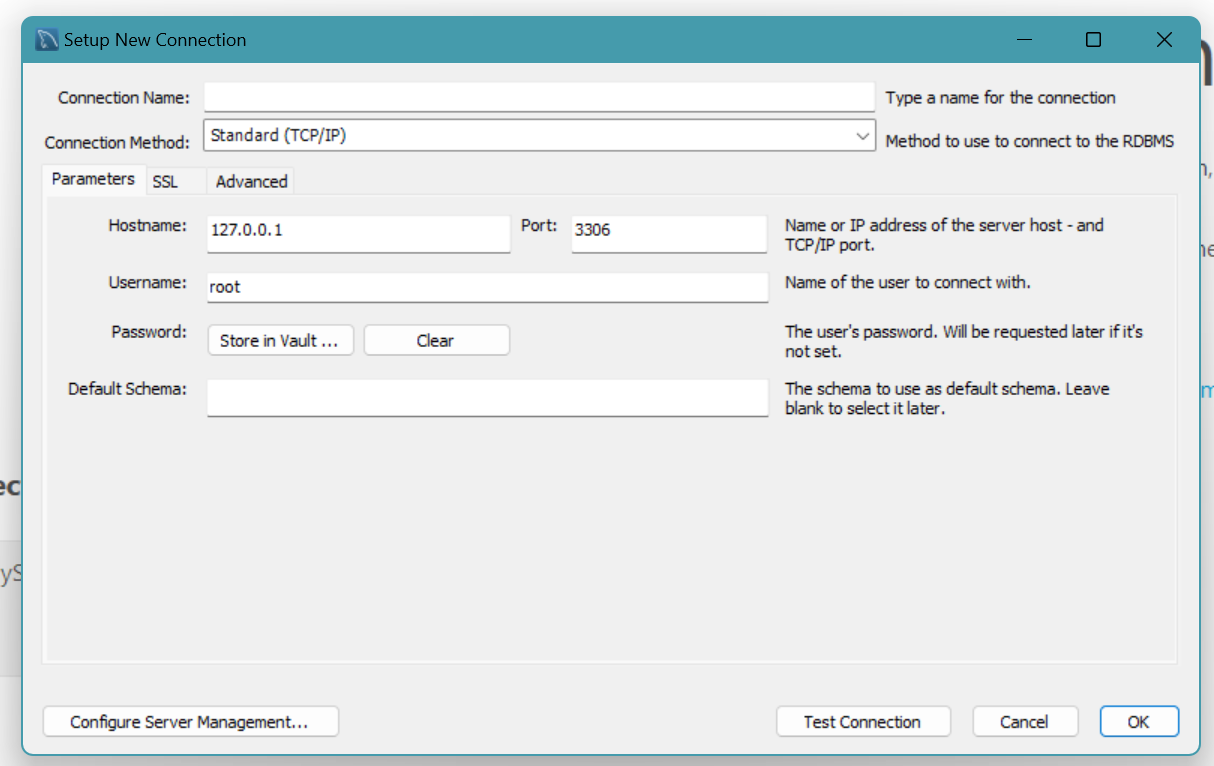
* Open your MySQL Workbench or SSH into an EC2 instance with database client tools installed.

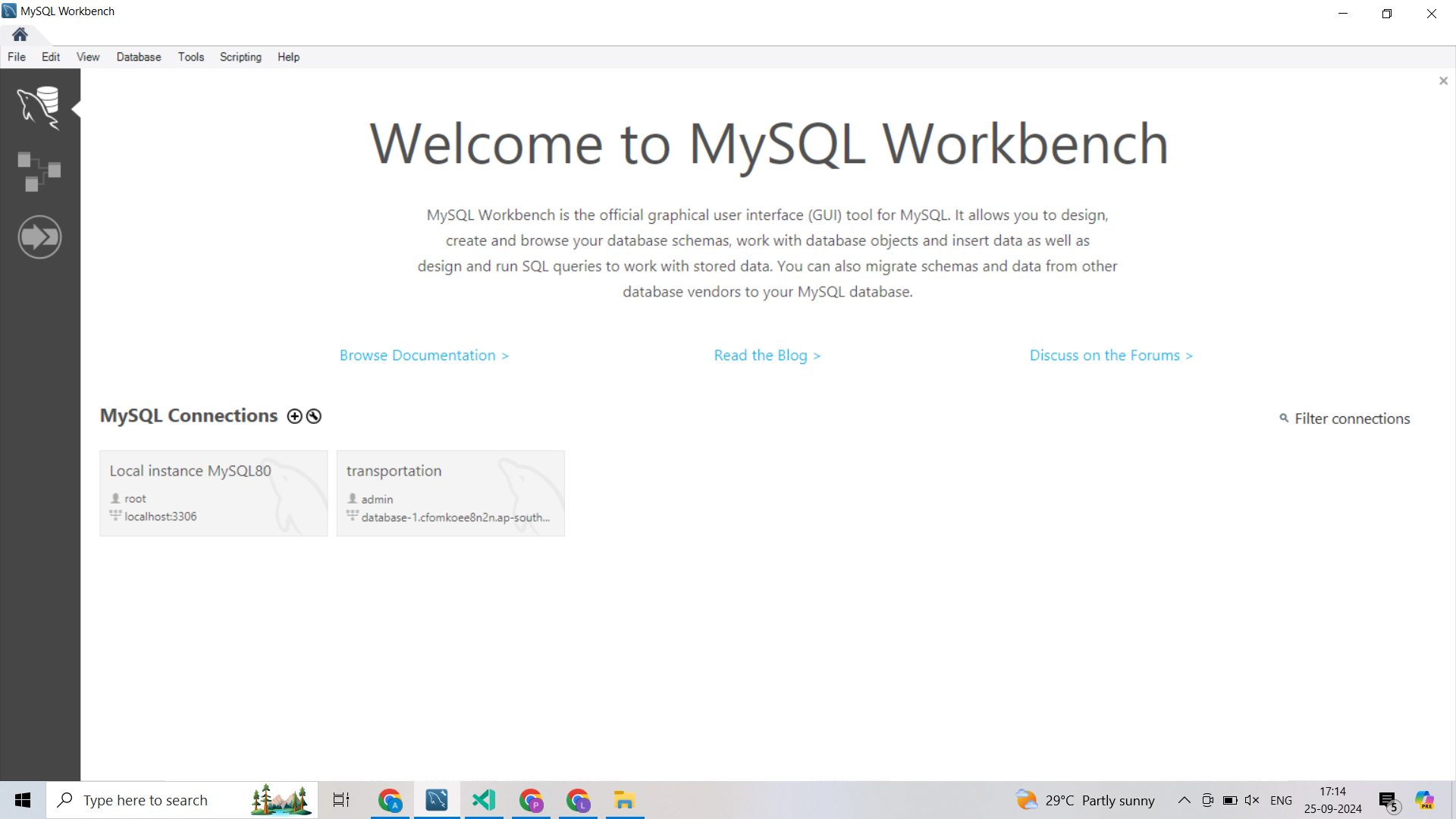


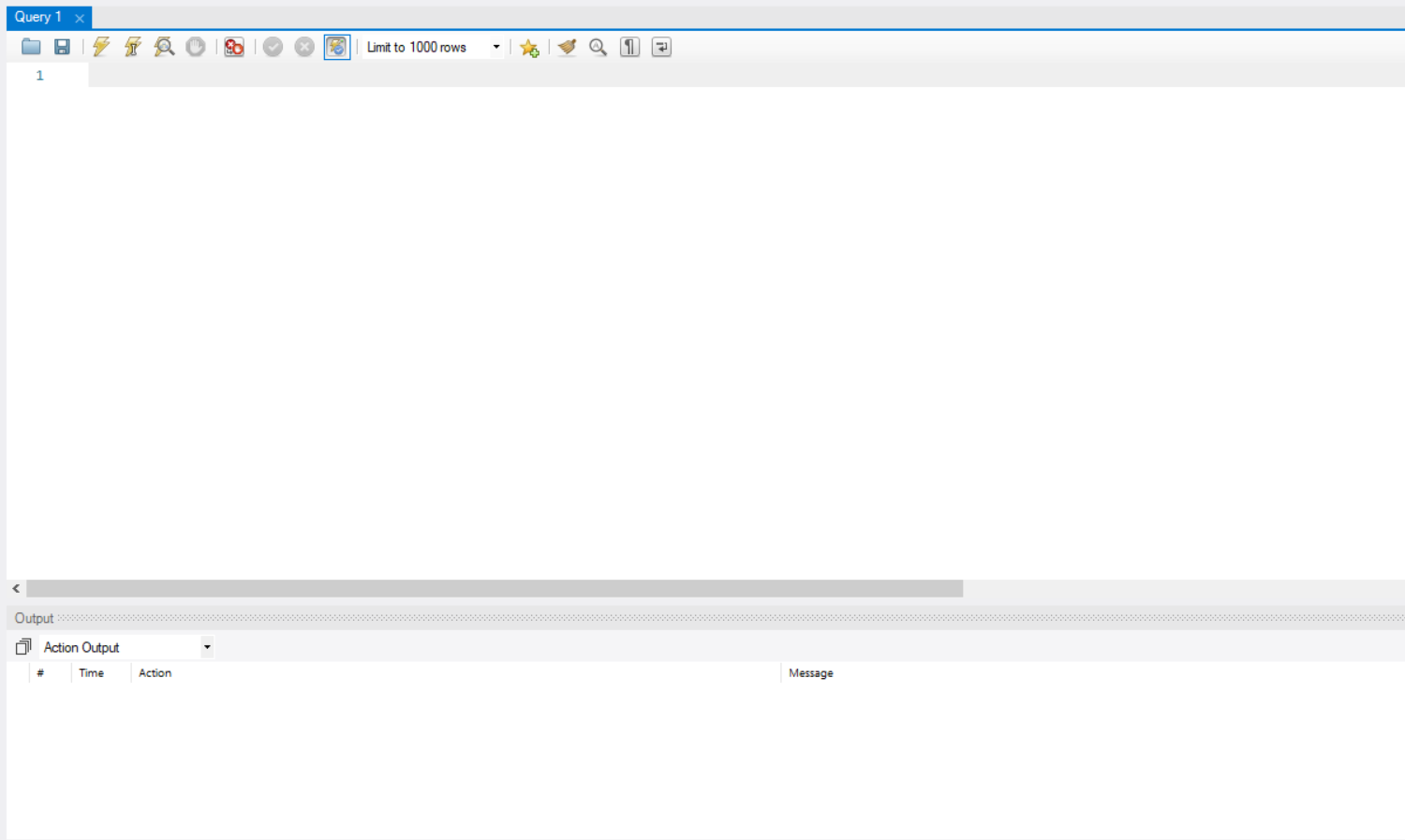
* Click on ‘+’ symbol to connect RDS with MySQL Workbench



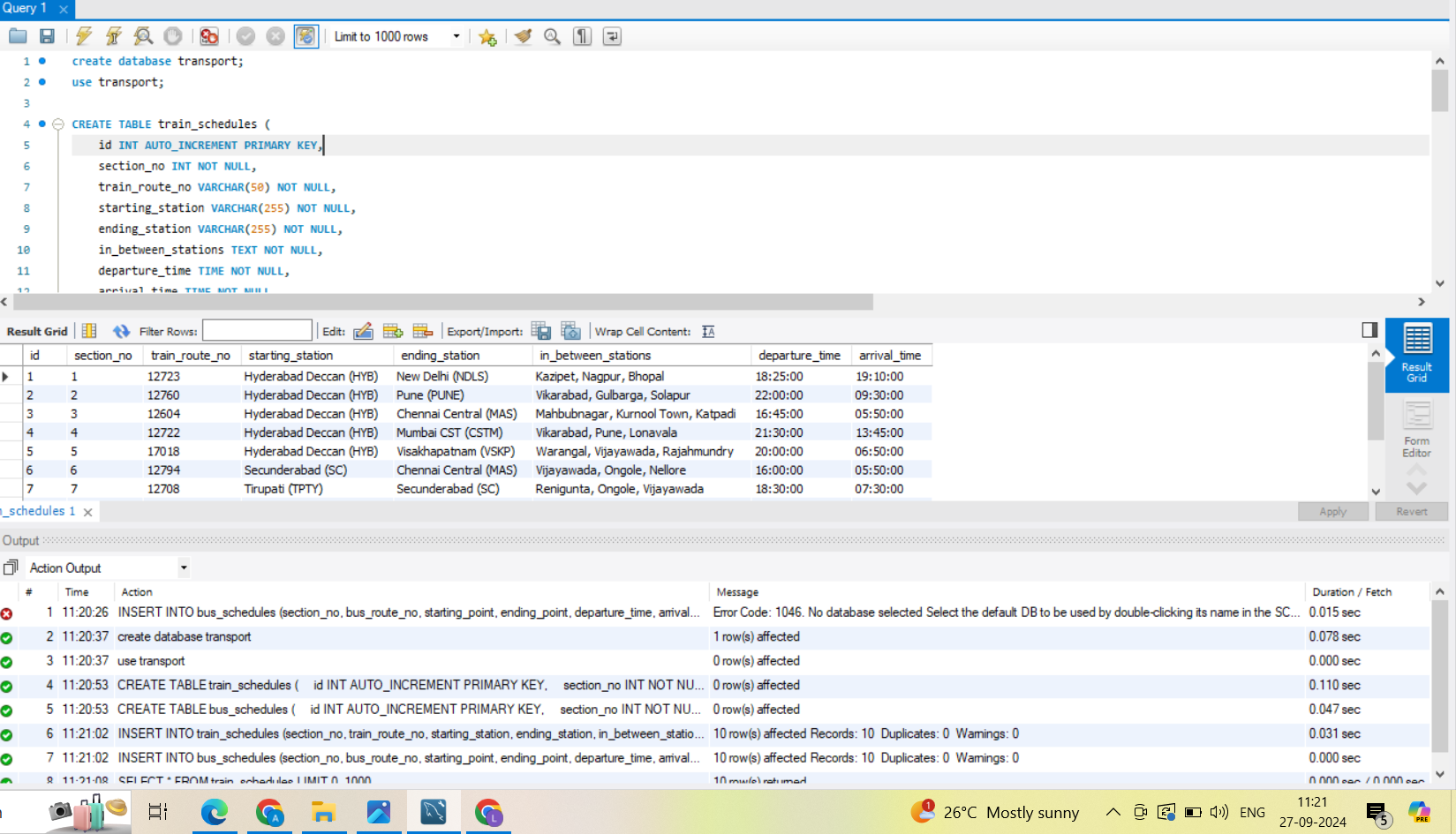
* Paste your Endpoint in place of Hostname, give an connection name and enter your Username and password which used in AWS RDS ,to connect the database





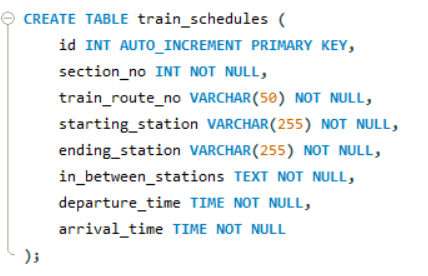


1. Create the Database and the tables which are required.

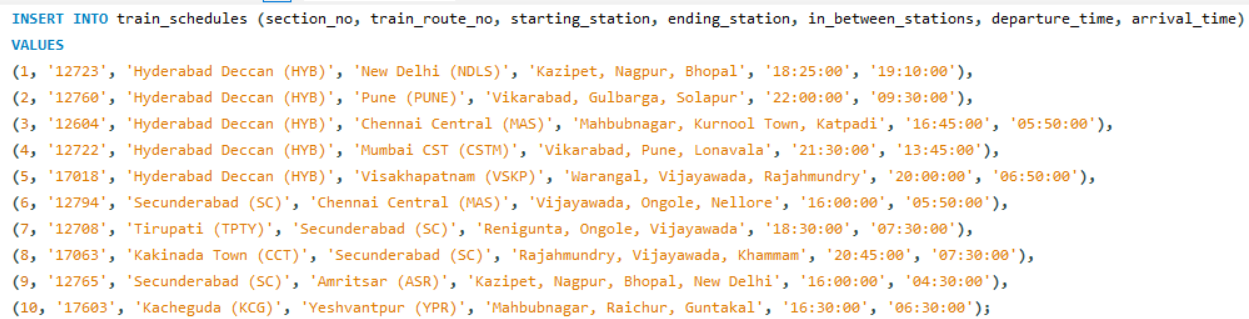
****

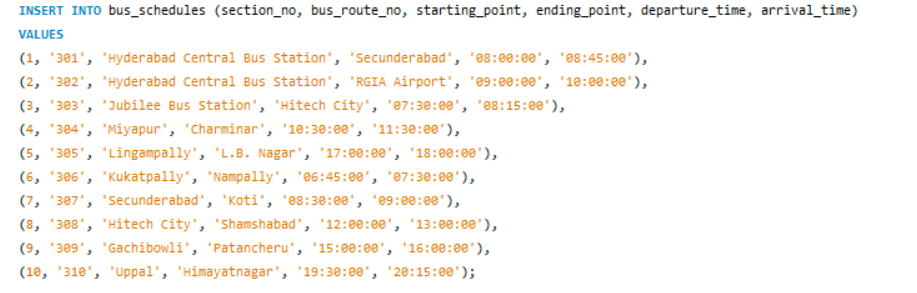
**Code:**

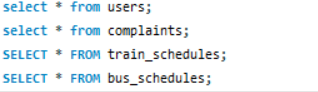




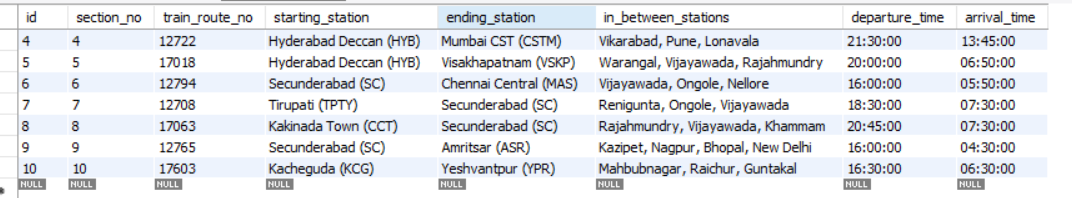


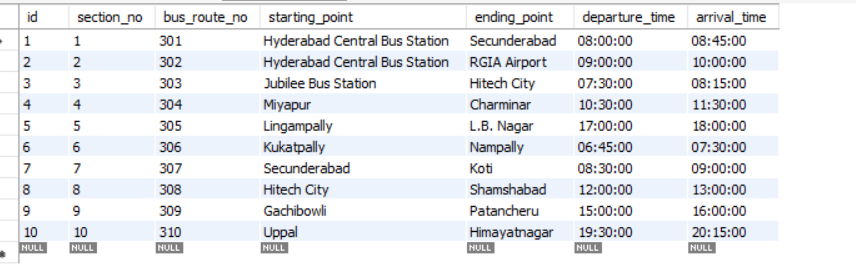






Output Tables:



****

**Conclusion:**

The *Hyderabad Urban Mobility Solutions (HUMS)* project successfully addresses the growing need for efficient public transportation management in Hyderabad by creating a centralized MySQL database hosted on AWS RDS. By providing real-time access to bus and train schedules, along with a feedback and complaint system, the project enhances the commuter experience and empowers city planners to make data-driven decisions for service improvements. The integration of AWS services ensures scalability and reliability, while the user-friendly interface streamlines access to crucial transit information, ultimately contributing to a smarter, more organized urban mobility system.