Connect A Web App with Aurora

Hassan Gachoka

Introducing Today's Project!

What is Amazon Aurora?

AmazonAurora is a fully managed relational database compatible with MySQL and PostgreSQL. It offers high performance, scalability, and availability, automating backups, scaling, and patching while delivering faster performance and cost savings.
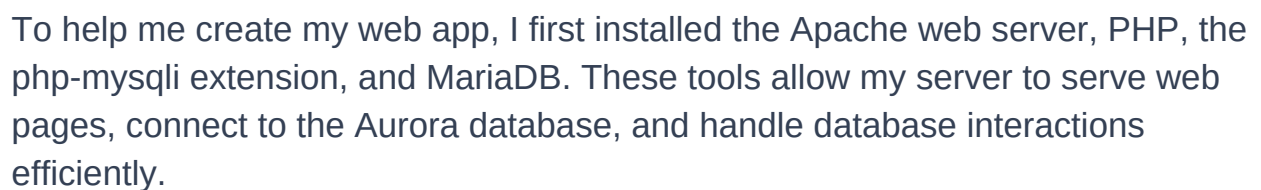
How I used Amazon Aurora in this project

Intoday's project, I usedAmazon Aurorabysetting up a relational database to store employee data. I connected this database to a web app on an EC2 instance, enabling the app to add and retrieve records interactively from the database.

One thing I didn't expect in this project was...

Onething I didn't expect in this project was how smoothly theintegration between the web app and the Aurora database went. The automation features of Aurora made setting up connections and managing data easier than anticip ated.
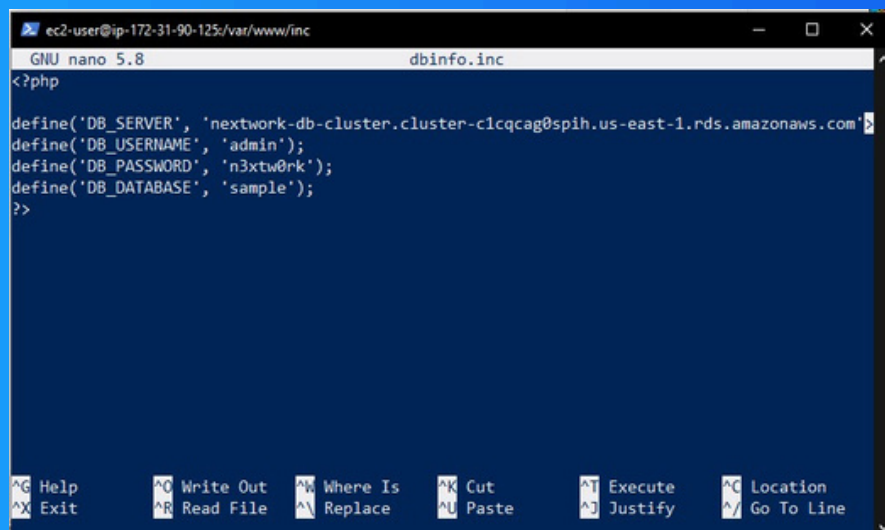
This project took me...

The projecttookabout 1 hourto complete, including setting up the EC2 instance, configuring the Aurora database, and developing the web app with the necessary functionalities.

Creating a Web App

To help me create my web app, I first installed the Apache web server, PHP, the php-mysqli extension, and MariaDB. These tools allow my server to serve web pages, connect to the Aurora database, and handle database interactions efficiently.

To connect to my EC2 instance, I used Windows PowerShell and ran the SSH command with my .pem key file, EC2 username, and the EC2 instance's Public IPv4 DNS address.

Connecting my Web App to Aurora

I set up my EC2 instance's connection details to my database by creating a dbinfo.inc file, adding the Aurora database endpoint, and defining the database username, password, and database name in the file using PHP constants.

To connect to my EC2 instance, I used Windows PowerShell and ran the SSH command with my .pem key file, EC2 username, and the EC2 instance's Public IPv4 DNS address.

Hassan Gachoka

linkedin.com/gachokahassan

My Web App Upgrade

I created a dbinfo.inc file to store the db connection details, then made a SamplePage.php file that connects to the Aurora db. I added a form for users to input employee names and addresses and display existing records from the "EMPLOYEES" table.

Hassan Gachoka
linkedin.com/gachokahassan

Testing my Web App

To make sure my web app was working correctly, I tested it in the browser by adding employee data. Then, I connected to the Aurora database using MySQL CLI and verified that the new data was successfully added to the "EMPLOYEES" table.