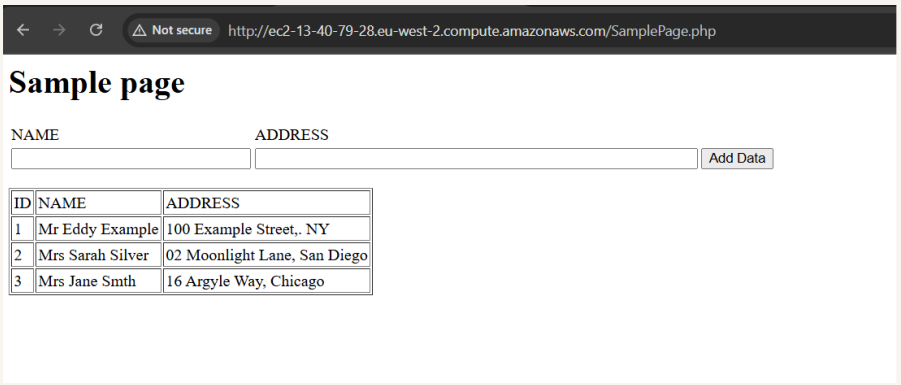
nextwork.org

Connect a Web App to Amazon Aurora

MA Marzena Pugo



Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a relational database service (RDS) provided by AWS. It's compatible with MySQL and PostgreSQL, which means you can use the same SQL queries, tools, and drivers you already know, but with Aurora's cloud-native performance and scalability. It is useful because Amazon Aurora gives you the power of an enterprise-grade database with the simplicity and flexibility of a fully managed AWS service. It's especially useful when you want reliability + performance without the headache of managing databases yourself.

How I used Amazon Aurora in this project

In today's n today's project, I used Amazon Aurora (MySQL-compatible) as the database layer for my web application. - I created an Aurora database cluster in AWS RDS. - I set up a schema and tables inside Aurora to store user data. - I connected to the Aurora database from my EC2 instance using the MySQL CLI to verify the connection. - Finally, I configured my web application (running on EC2) to connect to Aurora, so that any data entered in the app is stored in the database.project. Aurora database acted as the persistent storage for my app, allowing me to enter data through the web app and confirm it was saved in the database.

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

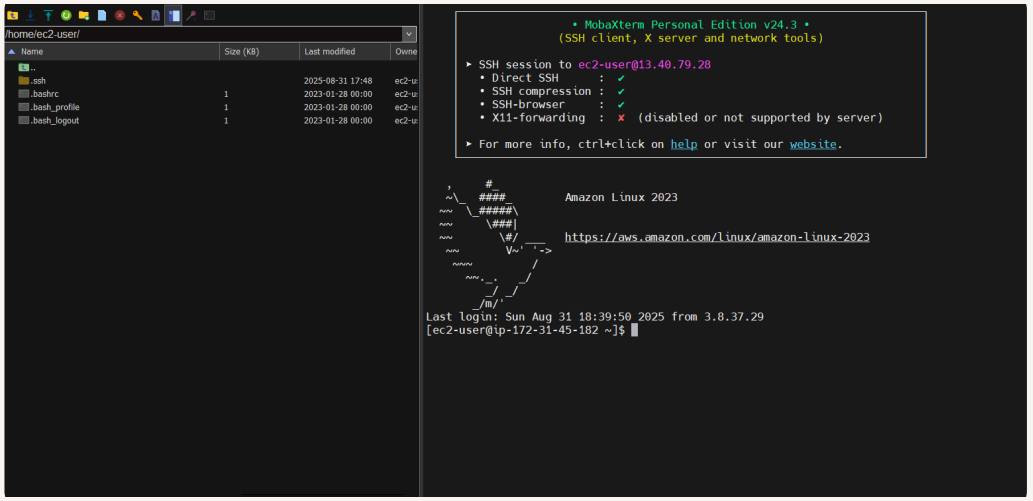
One thing I didn't expect was running into package compatibility issues when installing the MySQL client on my EC2 instance.

Instead of getting the standard MySQL client right away, I ended up with the MariaDB client, which still worked but was a surprise. This taught me that in cloud projects, you often need to troubleshoot unexpected issues and find alternative solutions, which is a valuable part of the learning process.

This project took me...

This project took me about 1.5–2 hours in total. Most of the time went into setting up the Aurora database, configuring the EC2 instance, and troubleshooting the MySQL client installation. The actual steps of connecting the web app to Aurora were quicker once everything was set up, but the process of learning and fixing issues added extra time , which was valuable for understanding how things really work behind the scenes.

Creating a Web App



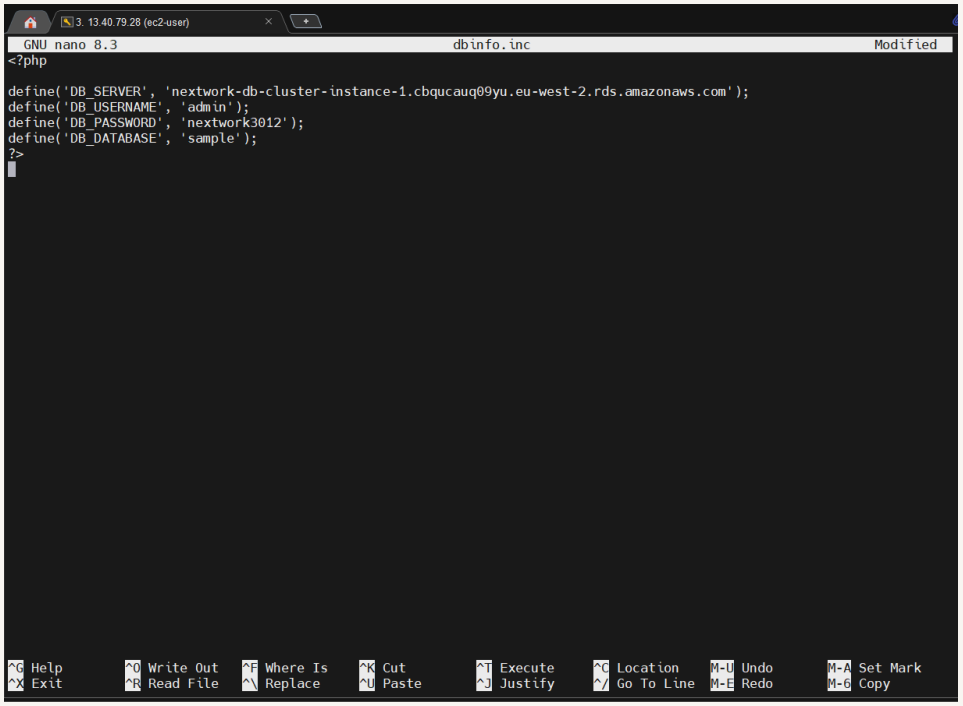
To connect to my EC2 instance, I had to change my permissions to access your .pem file. I have used mobaXterm to get connected via SSH with my EC2 instance.

To help me create my web app, first I have installed Apache HTTP Server package.

MA
Marzena Pugo
NextWork Student

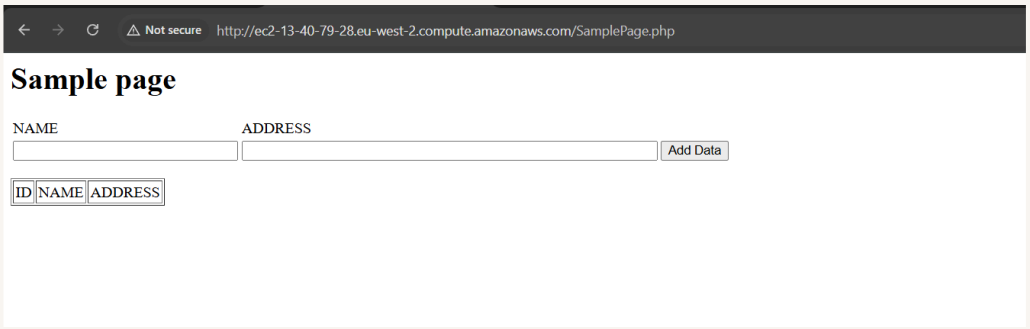
Connecting my Web App to Aurora

I set up my EC2 instance's connection details to my database by adding some code to dbinfo.inc file, that connects it to my AWS database.



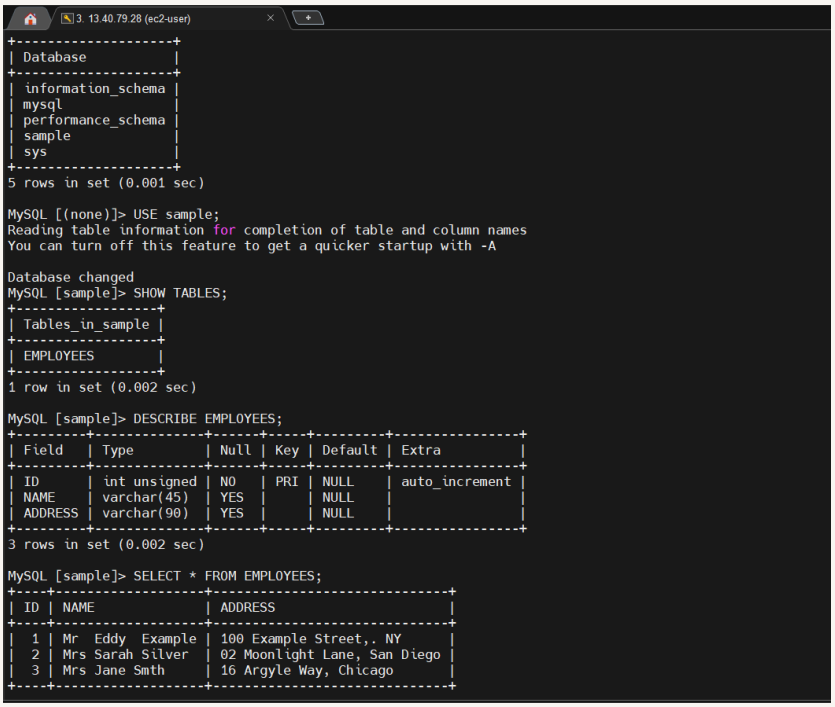
My Web App Upgrade

Next, I upgraded my web app by creating a new file in the html directory named SamplePage.php and copying and pasting a script into your SamplePage.php file. The bloc of code is pulling in the details from my dbinfo.inc file I created earlier and using it to display up-to-date changes directly from my website.



Testing my Web App

To make sure my web app was working correctly, I first tested the connection directly from the EC2 instance using the MySQL CLI. - I installed the MySQL client on my EC2 instance. - Connected to my Aurora MySQL database using the cluster endpoint, username, and password. - Ran the following commands to confirm access: SHOW DATABASES; USE sample; SHOW TABLES; DESCRIBE employees; SELECT * FROM employees; These commands allowed me to see the available databases, switch to the sample schema, view the tables, check the structure of the employees table, and display the data stored in it



The place to learn & showcase your skills

Check out nextwork.org for more projects