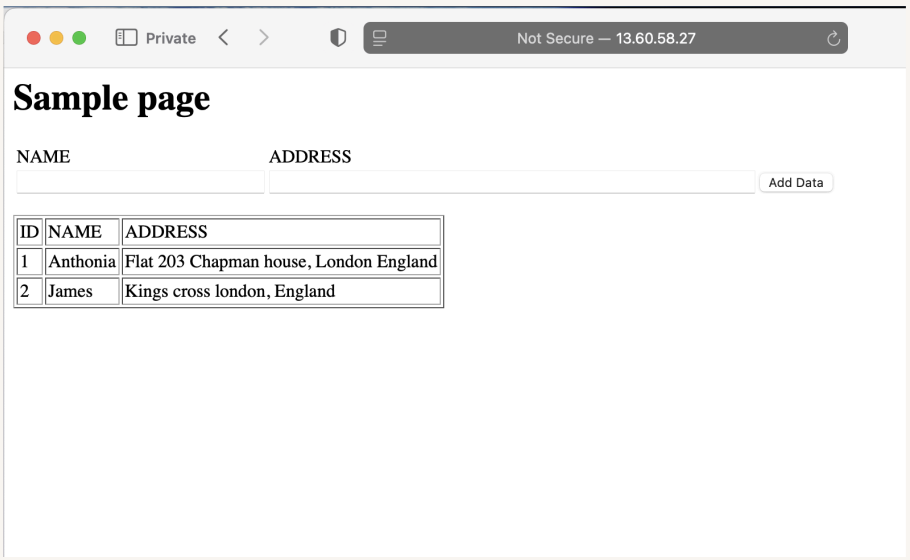
nextwork.org

Connect a Web App to Amazon Aurora

Kehinde Abiuwa



Kehinde Abiuwa
NextWork Student

Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a fully managed relational database service from AWS that is compatible with MySQL and PostgreSQL. It is useful because it combines the scalability and high availability of enterprise-grade databases with the cost-effectiveness and simplicity of open-source engines, automatically handling tasks like backups, replication, and failover while delivering faster performance and seamless integration with other AWS services.

How I used Amazon Aurora in this project

In today's project, I used Amazon Aurora to create a relational database that stores application data, connect it to my EC2-hosted web app, and verify that data entered through the web app is successfully saved and retrieved from the database.

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

One thing I didn't expect in this project was how much configuring permissions and file ownership on the EC2 instance mattered — without adjusting them, I couldn’t create or edit the necessary files for my web app to connect to Aurora.

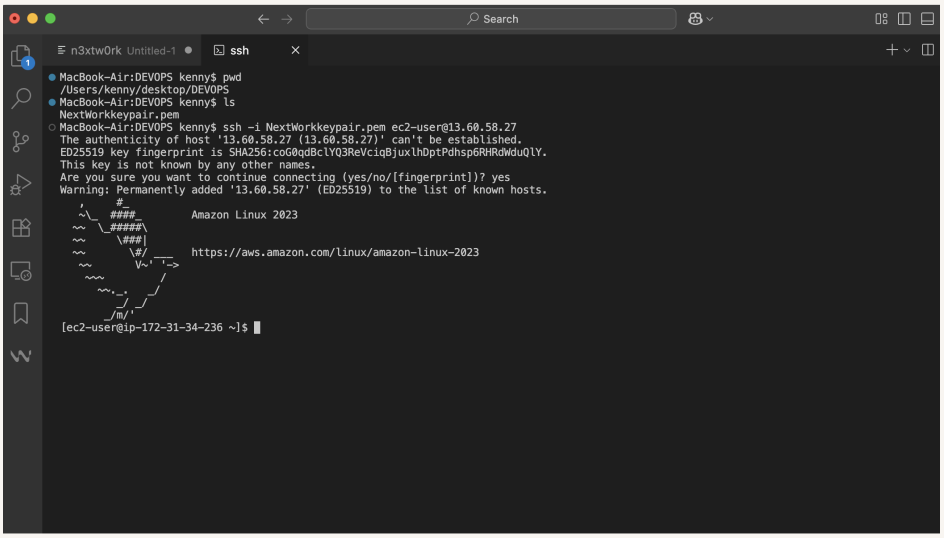
Kehinde Abiuwa
NextWork Student

This project took me...

This project took me 35 minutes

Kehinde Abiuwa
NextWork Student

Creating a Web App



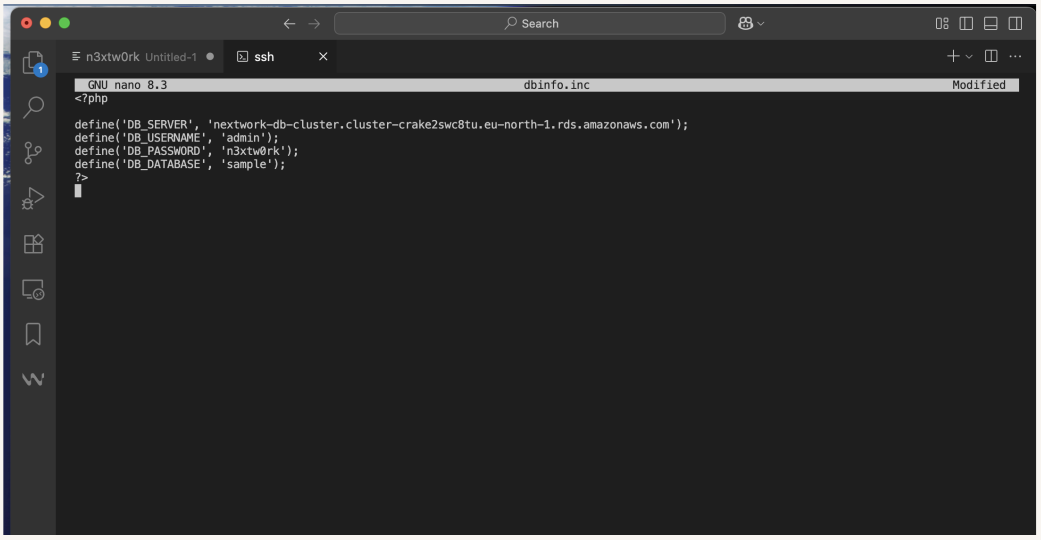
To connect to my EC2 instance, I used the SSH command with my key pair file (NextWorkkeypair.pem) and the public IPv4 address of the instance. This allowed me to securely log in as the ec2-user and access the Amazon Linux 2023 environment running on my server.

To help me create my web app, I first updated the server packages, installed Apache (httpd), PHP with the MySQLi extension, and the MariaDB client (mariadb105), then started the Apache service so the instance could serve PHP pages.

Kehinde Abiuwa
NextWork Student

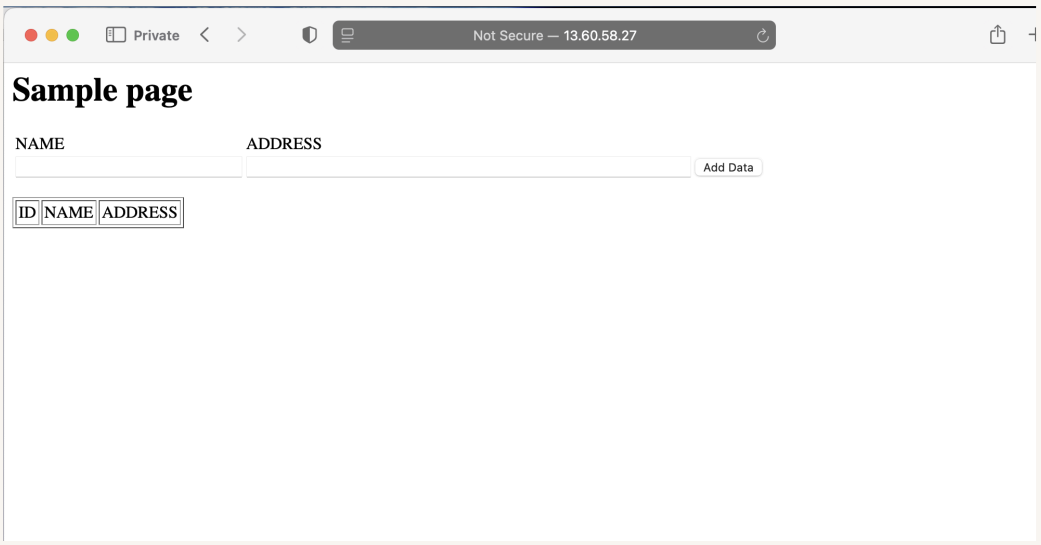
Connecting my Web App to Aurora

I set up my EC2 instance's connection details to my database by creating a new folder (inc) inside the /var/www directory, then adding a settings file called dbinfo.inc. In this file, I defined my Aurora database endpoint, username, password, and database name so that my web app can securely connect to the Aurora database.



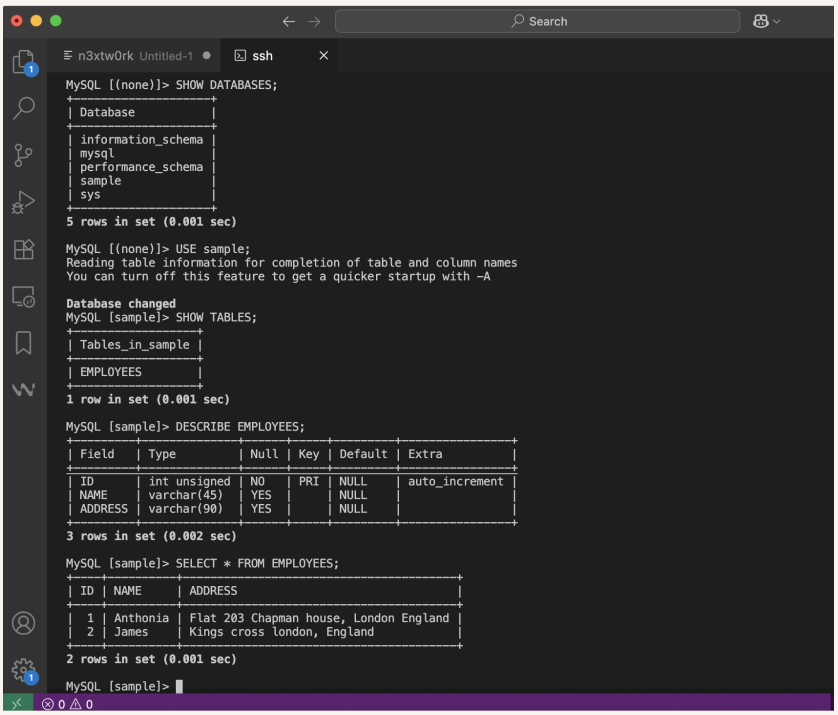
My Web App Upgrade

Next, I upgraded my web app by adding a SamplePage.php file that includes a simple HTML form and a table connected to my Aurora database. This upgrade lets me enter a Name and Address through the web page, click Add Data, and then see the information stored in and retrieved from the Aurora database—all directly in my browser.



Testing my Web App

To make sure my web app was working correctly, I tested it in the browser by adding sample data through the form and then verified that the same data appeared in my Aurora database using the MySQL CLI. I checked the sample database, confirmed the EMPLOYEES table structure, and ran a SELECT * FROM EMPLOYEES; query to ensure the records I entered (like Anthonia and James) were stored and retrieved successfully.



The place to learn & showcase your skills

Check out nextwork.org for more projects