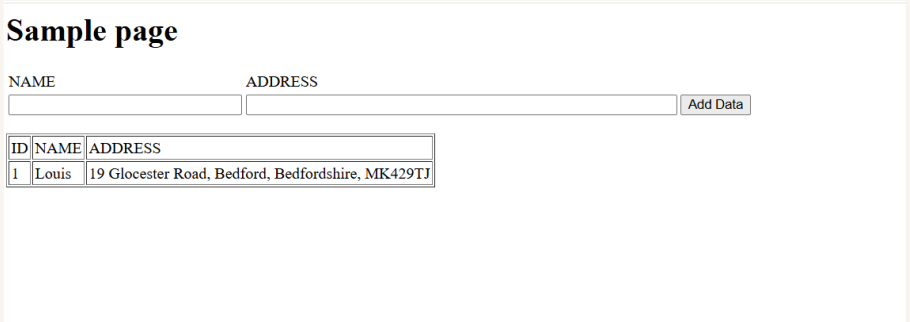
nextwork.org

Connect a Web App to Amazon Aurora

Louis Moyo



Louis Moyo
NextWork Student

Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a managed relational database from AWS that's compatible with MySQL/PostgreSQL. It's useful because it's fast, highly available, and scales easily while AWS handles heavy lifting like backups, patches, and failover. That lets you focus on your app instead of database ops.

How I used Amazon Aurora in this project

n today's project, I created an Aurora MySQL cluster and connected it to my EC2 web server. My PHP page used the cluster's writer endpoint, username, password, and database name to insert and read data, turning the site into a simple data-driven app.

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

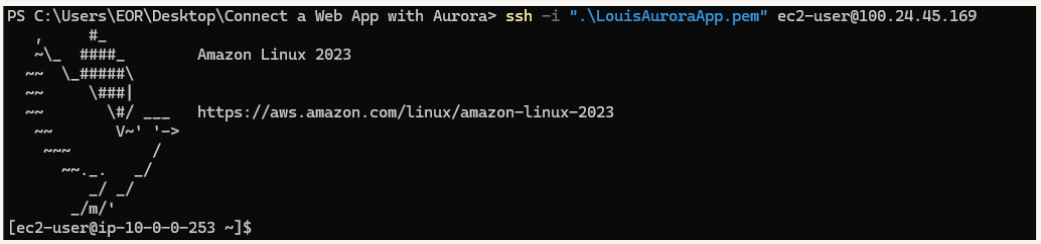
I didn't expect that Aurora shows a cluster plus instances with different roles (writer vs reader). A reader isn't created automatically - I had to add it via Multi-AZ/replica or "Add reader" later.

Louis Moyo
NextWork Student

This project took me...

About 90 minutes, including waiting for the EC2/Aurora resources to create, setting up SSH, installing packages, and testing the database connection from the app.

Creating a Web App



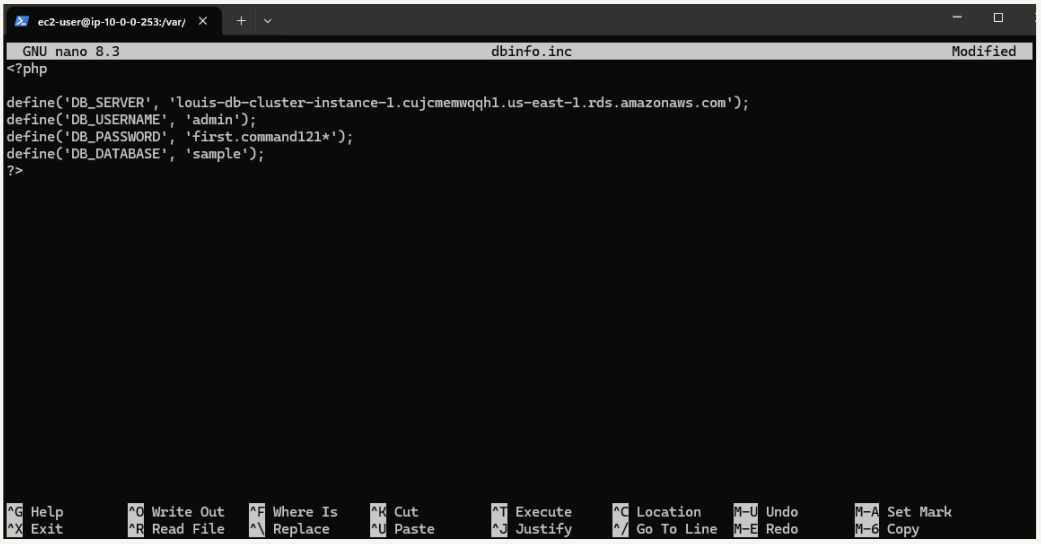
To connect to my EC2 instance, I used the SSH command in my terminal with the .pem key file and the Public IPv4 DNS of the instance. This gave me secure access to the server.

To help me create my web app, I first installed Apache (httpd) to serve web pages, PHP to run the app, the php-mysqli library so PHP can talk to MySQL databases, and MariaDB so my EC2 instance can connect with the Aurora database.

Louis Moyo
NextWork Student

Connecting my Web App to Aurora

I created a dbinfo.inc file on the EC2 server and added my Aurora writer endpoint, database username, password, and DB name as constants (e.g., DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_DATABASE). My PHP page includes this file to open the MySQL connection. I also made sure the security groups allow EC2, Aurora on port 3306 so the connection works.

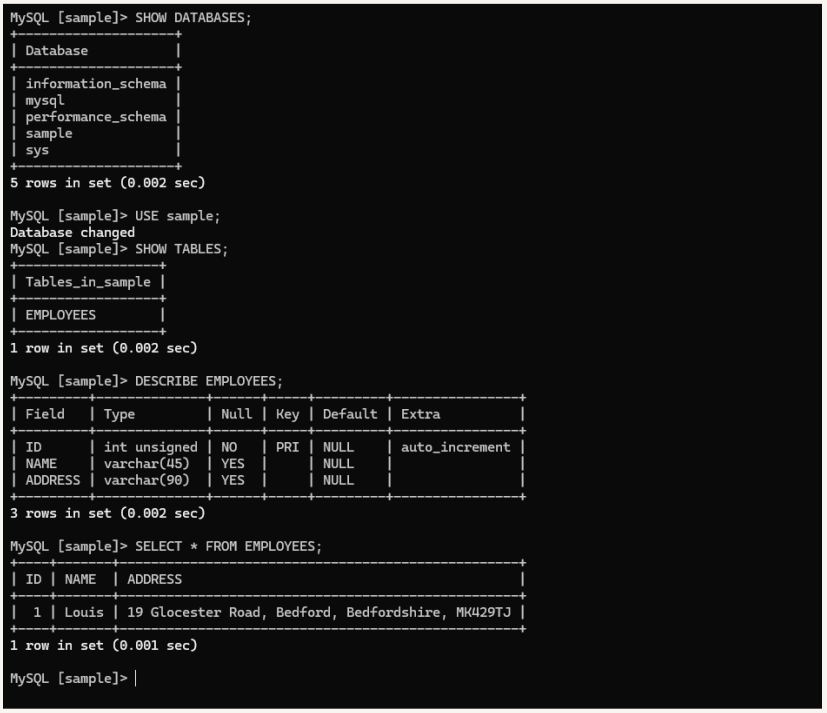


My Web App Upgrade

Next, I upgraded my web app by connecting it to Aurora and adding a simple page that can write and read data. I set the app's database connection, added a small form (Name/Address) to save entries, and a list to show stored records. This turned the EC2 site from a static page into a dynamic app with persistent data.

Testing my Web App

I installed the MySQL CLI on my EC2 instance, connected to the Aurora writer endpoint with: mysql -h <YOUR_ENDPOINT> -P 3306 -u admin -p then ran: SHOW DATABASES; USE sample; SHOW TABLES; DESCRIBE employees; SELECT * FROM employees;. I first added a record via http://<EC2-address>/SamplePage.php, and seeing the same row returned by SELECT * FROM employees; confirmed the web app successfully wrote to Aurora.



The place to learn & showcase your skills

Check out nextwork.org for more projects