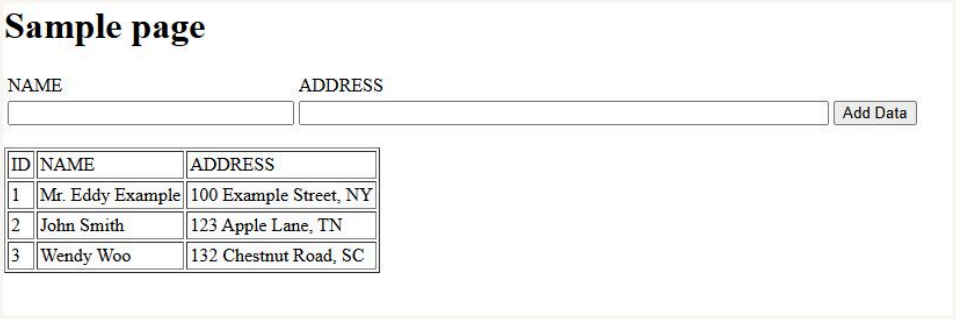
nextwork.org

Connect a Web App to Amazon Aurora

Mohammed Dawoud Mota



Sample page

NAME ADDRESS

Add Data

ID	NAME	ADDRESS
1	Mr. Eddy Example	100 Example Street, NY
2	John Smith	123 Apple Lane, TN
3	Wendy Woo	132 Chestnut Road, SC

Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a fully managed relational database service provided by Amazon Web Services that is compatible with MySQL and PostgreSQL. It is built for high performance, scalability, and availability in the cloud. Aurora uses a cluster architecture with a primary writer instance and multiple reader instances, which allows it to handle large amounts of traffic while keeping data highly available. Aurora is useful because it automatically handles backups, replication, and failover, which reduces the amount of manual database management required. It is designed to be more reliable and scalable than standard MySQL databases, making it a strong choice for modern web applications. In this project, Aurora allowed me to store and manage my web app data securely while ensuring that the application could read and write data efficiently.

How I used Amazon Aurora in this project

In today’s project, I used Amazon Aurora to create a MySQL compatible relational database that stored data from my web app. I created a DB cluster, set up the master credentials, and connected it to my EC2 instance. I then used the writer endpoint in my dbinfo.inc file so my PHP web app could connect to the database. When I added data through the web app, it was stored in the Aurora database. I verified it by connecting with MySQL CLI and running SQL commands to check the data.

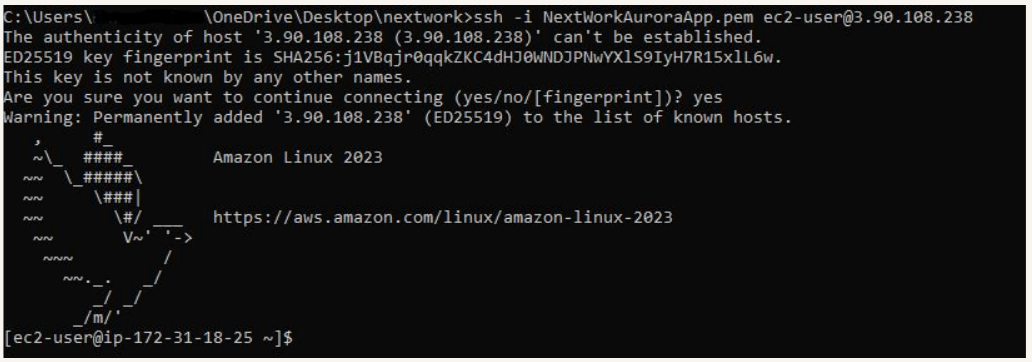
One thing I didn't expect in this project was...

One thing I did not expect in this project was how important file permissions were on the EC2 instance. I ran into permission denied errors when trying to create folders and edit files inside the web directory, and I had to change ownership using chown before I could continue.

This project took me...

This project took me about one hour to complete. Most of the time went into setting up the EC2 instance, configuring the Aurora database, and making sure the connection between the web app and the database was working correctly. Troubleshooting permissions and verifying everything with MySQL CLI also added a bit of extra time.

Creating a Web App



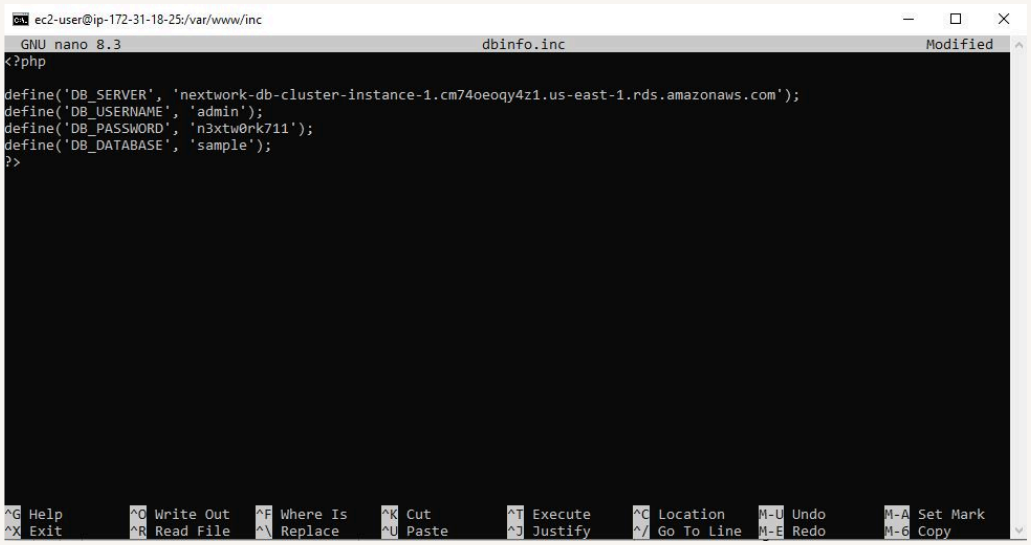
To connect to my EC2 instance, I used SSH and used the .pem file as the key to gain access.

I installed everything needed for a basic web app stack on the EC2 instance, including Apache (httpd) to serve the web pages, PHP to run the application code, the php-mysqli extension so PHP can talk to a MySQL-compatible database, and the MariaDB client libraries to let the instance communicate with my Aurora database, all using sudo dnf install -y httpd php php-mysqli mariadb105, and then I started the Apache service with sudo systemctl start httpd to bring the web app online.

Mohammed Dawoud M...
NextWork Student

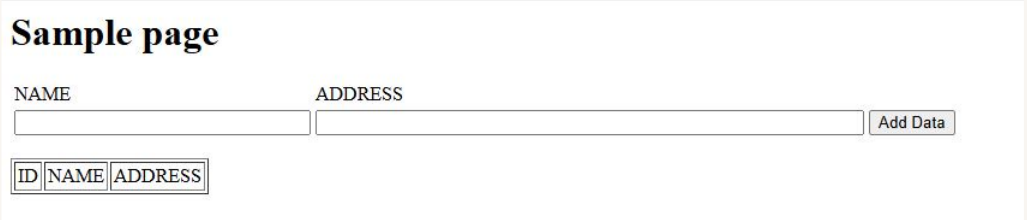
Connecting my Web App to Aurora

I set up my EC2 instance's connection details to my database by creating a file called dbinfo.inc, which includes the writer endpoint, the username, the password, and the database name.



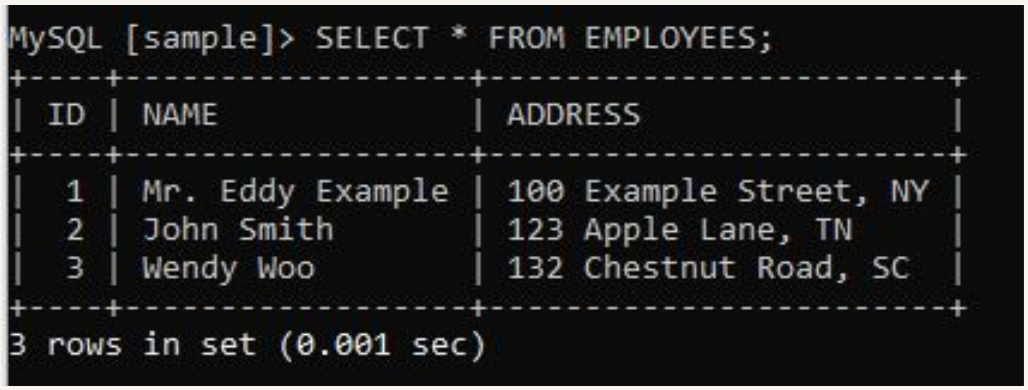
My Web App Upgrade

Next, I upgraded my web app by navigating to the HTML folder, creating a new file, and editing that file with nano. I then added an updated script that would allow user access to the database.



Testing my Web App

To make sure my web app was working correctly, I installed the MySQL CLI. I connected to the database and ran some SQL commands that allowed me to verify that my database was updated correctly.



The place to learn & showcase your skills

Check out nextwork.org for more projects