



Connect a Web App to Amazon Aurora

 Gloria



The screenshot shows a web browser window with the URL `ec2-107-20-77-236.compute-1.amazonaws.com/SamplePage.php`. The page title is "Sample page". It features two input fields labeled "NAME" and "ADDRESS". Below these is a table with three rows, each containing an ID, a name, and an address.

ID	NAME	ADDRESS
1	MISS GLORIA LARBI	ROYAL STREET ACCRA
2	MR ELDRIGE ACKOM	234 GISE STREET
3	MRS FRANCISA PRAH	PALM STREET , KUMASI

An "Add Data" button is located to the right of the ADDRESS field.



Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a fast and reliable database service compatible with MySQL and PostgreSQL. It handles large amounts of data, takes care of backups, scales easily, and keeps data secure, making it ideal for apps that need strong performance

How I used Amazon Aurora in this project

In today's project, I used Amazon Aurora to create a relational database to store user data from my web app. I set up the database, connected it to my EC2 instance, and verified that the app fully stored and retrieved data from the Aurora database

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

One thing I didn't expect in this project was how smoothly Amazon Aurora integrated with the EC2 instance and the web app. Setting up the connection and seeing the data flow from the app to the database in real time was easier than I thought.

This project took me...

This project took me 30 minutes to complete. Setting up the Aurora database, connecting it to the EC2 instance, and verifying the web app's functionality with the database was very straightforward.



Creating a Web App

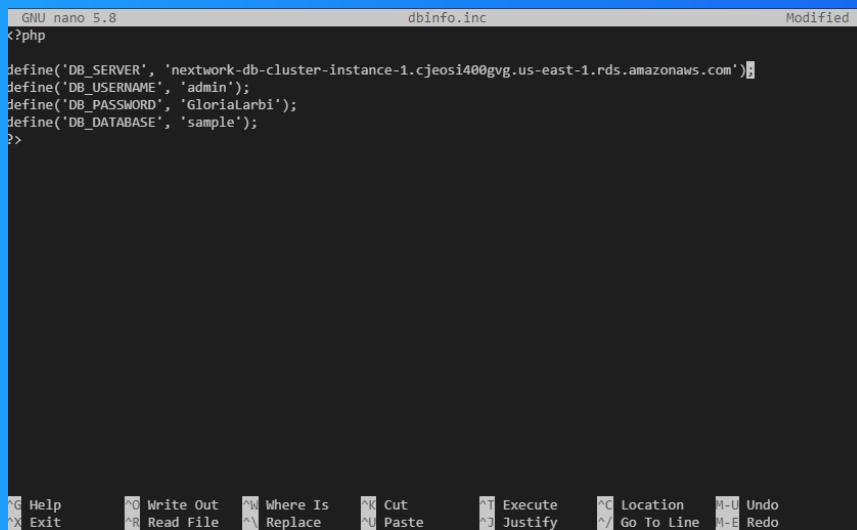
```
ALINCO TECH@GLORIA-LARBI MINGW64 ~/Downloads$ ssh -i "NextWorkAuroraApp.pem" ec2-user@ec2-107-20-77-236.compute-1.amazonaws.com
The authenticity of host 'ec2-107-20-77-236.compute-1.amazonaws.com (107.20.77.236)' can't be established.
ED25519 key fingerprint is SHA256:GZMfCr2MOrEQoe650TL+S2YTiVDoH54tWKf5A5wSvXA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-107-20-77-236.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

          _#
 ~\_\_ #####_      Amazon Linux 2023
 ~~ \#####\_
 ~~  \|##\|_
 ~~   \|#/ _--> https://aws.amazon.com/linux/amazon-linux-2023
 ~~    V~' _-'
 ~~~   /
 ~~: _- _/
 /m/ ,_/
 [ec2-user@ip-172-31-22-212 ~]$
```

To help me create my web app, I first updated the software on my EC2 instance and then installed Apache for running the web server, PHP for creating app pages, php-mysqli for connecting to the Aurora database, and MariaDB for managing database.

To connect to my EC2 instance, I used SSH by running a command in the terminal. I specified the private key from my key pair, the username, and the instance's Public IPv4 DNS to securely access the server and manage it remotely.

Connecting my Web App to Aurora



The screenshot shows a terminal window with the nano 5.8 text editor open. The file being edited is named 'dbinfo.inc'. The content of the file is a PHP configuration snippet:

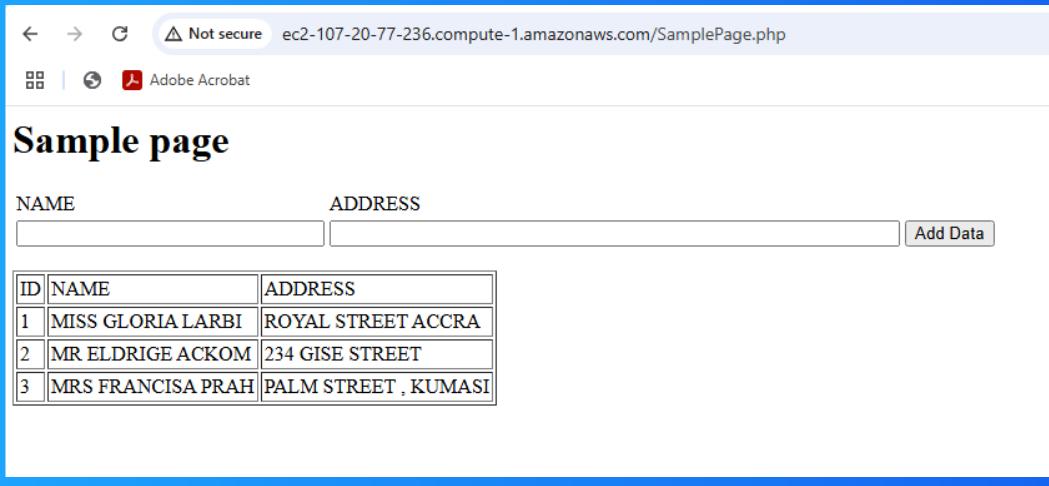
```
GNU nano 5.8                               dbinfo.inc                                         Modified
<?php
define('DB_SERVER', 'nextwork-db-cluster-instance-1.cjeosi400gvg.us-east-1.rds.amazonaws.com');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'GloriaLarbi');
define('DB_DATABASE', 'sample');
?>
```

At the bottom of the terminal window, there is a menu bar with various options like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, and Redo.

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

To connect to my EC2 instance, I used SSH by running a command in the terminal. I specified the private key from my key pair, the username, and the instance's Public IPv4 DNS to securely access the server and manage it remotely.

My Web App Upgrade



Next, I upgraded my web app by creating a `SamplePage.php` file with a PHP script. The script connects the web app to the Aurora database, lets users add data through a form, and dynamically displays the data from the database on the web page.



Testing my Web App

To make sure my web app was working correctly, I connected to my Aurora database using MySQL CLI. I checked the database with SHOW DATABASES; and SHOW TABLES; then ran SELECT * FROM EMPLOYEES; to confirm the data entered through the app was stored.

```
Database changed
MySQL [sample]> SHOW TABLES;
+-----+
| Tables_in_sample |
+-----+
| EMPLOYEES         |
+-----+
1 row in set (0.002 sec)

MySQL [sample]> DESCRIBE EMPLOYEES;
+-----+-----+-----+-----+-----+
| Field    | Type      | Null | Key | Default | Extra       |
+-----+-----+-----+-----+-----+
| ID       | int unsigned | NO   | PRI  | NULL    | auto_increment |
| NAME     | varchar(45)  | YES  |      | NULL    |               |
| ADDRESS  | varchar(90)   | YES  |      | NULL    |               |
+-----+-----+-----+-----+-----+
3 rows in set (0.002 sec)

MySQL [sample]> SELECT * FROM EMPLOYEES;
+-----+-----+
| ID | NAME          | ADDRESS        |
+-----+-----+
| 1  | MISS GLORIA LARBI | ROYAL STREET ACCRA |
| 2  | MR ELDRIGE ACKOM  | 234 GISE STREET   |
| 3  | MRS FRANCISA PRAH | PALM STREET , KUMASI |
+-----+-----+
3 rows in set (0.001 sec)
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

