



[nextwork.org](http://nextwork.org)

# Connect a Web App to Amazon Aurora



Louis Moyo

## Sample page

NAME	ADDRESS	
<input type="text"/>	<input type="text"/>	<input type="button" value="Add Data"/>
ID	NAME	ADDRESS
1	Louis	19 Gloucester Road, Bedford, Bedfordshire, MK429TJ



# 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

In 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

[nextwork.org](http://nextwork.org)

## 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

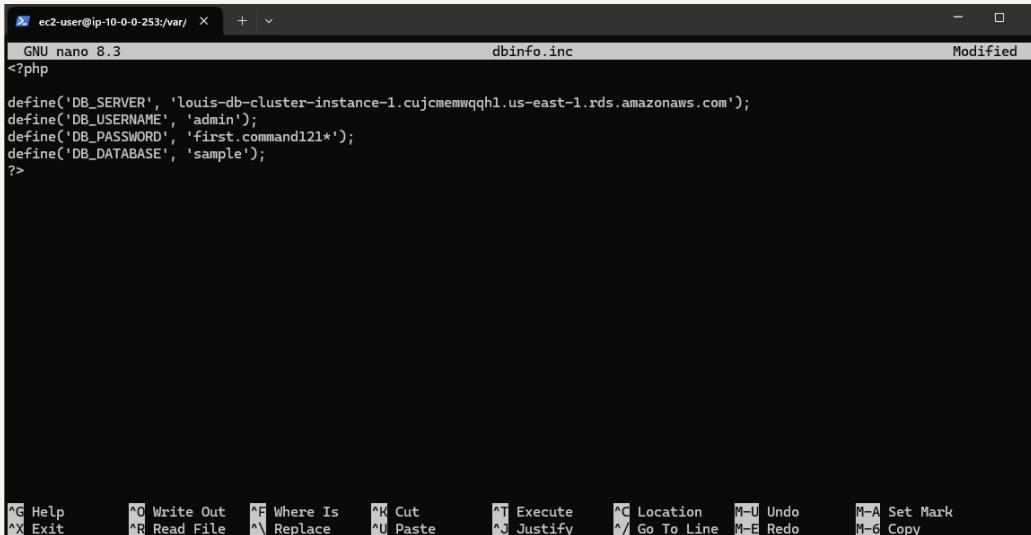
```
PS C:\Users\EOR\Desktop\Connect a Web App with Aurora> ssh -i ".\LouisAuroraApp.pem" ec2-user@100.24.45.169
      _#
     /###_          Amazon Linux 2023
    /##\ \###\_
   /##\ \###|_
  /##\ \#/ /-- https://aws.amazon.com/linux/amazon-linux-2023
 /##\ \#/\ \-\>
 /##\ \#/\ \-
 /##\ \#/\ \-
 [ec2-user@ip-10-0-0-253 ~]$
```

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.

# 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.



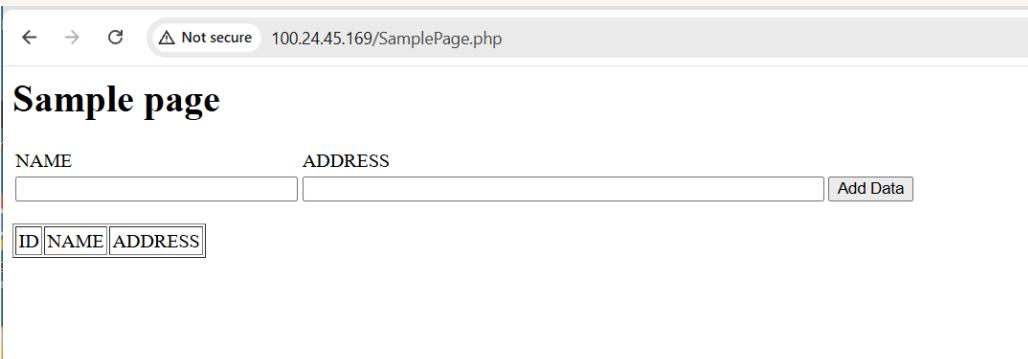
The screenshot shows a terminal window titled "ec2-user@ip-10-0-0-253:/var/" with the file "dbinfo.inc" open in nano editor. The file contains the following PHP code:

```
<?php
define('DB_SERVER', 'louis-db-cluster-instance-1.cujcmemwqqh1.us-east-1.rds.amazonaws.com');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'first.command121*');
define('DB_DATABASE', 'sample');
?>
```

The terminal window has a dark background with light-colored text. The bottom of the window shows a menu bar with various keyboard shortcuts for file operations like Help, Exit, Write Out, Read File, etc.

# 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.



A screenshot of a web browser window displaying a sample page. The address bar shows the URL `100.24.45.169/SamplePage.php`. The page title is "Sample page". Below the title, there is a form with two input fields labeled "NAME" and "ADDRESS", and a button labeled "Add Data". Below the form, there is a table with three columns labeled "ID", "NAME", and "ADDRESS". The table currently contains one row of data.

ID	NAME	ADDRESS
1	John Doe	123 Main Street

# 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.

```
MySQL [sample]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sample |
| sys |
+-----+
5 rows in set (0.002 sec)

MySQL [sample]> USE sample;
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 | Louis | 19 Gloucester Road, Bedford, Bedfordshire, MK429TJ |
+-----+-----+
1 row in set (0.001 sec)

MySQL [sample]> |
```



[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

