



Connect a Web App to Amazon Aurora



Kehinde Abiuwa

The screenshot shows a web browser window with the following details:

- Address Bar:** Not Secure — 13.60.58.27
- Content Area:**
 - Form:** A horizontal input field labeled "NAME" and another labeled "ADDRESS". To the right of the ADDRESS field is a button labeled "Add Data".
 - Table:** A data grid with columns "ID", "NAME", and "ADDRESS". It contains two rows of data:

ID	NAME	ADDRESS
1	Anthonia	Flat 203 Chapman house, London England
2	James	Kings cross london, England



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.



K

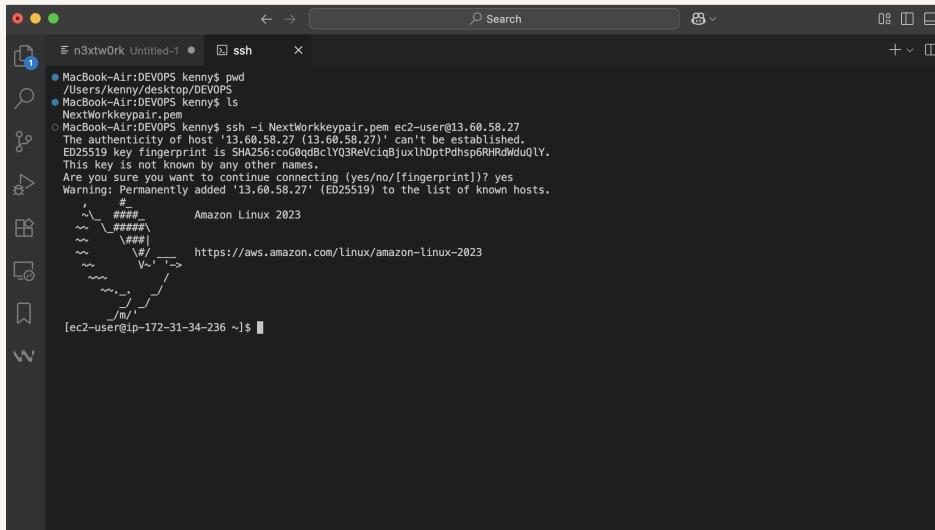
Kehinde Abiuwa
NextWork Student

nextwork.org

This project took me...

This project took me 35 minutes

Creating a Web App



```
n3xtw0rk Untitled-1 ● ssh x
MacBook-Air:DEVOPS kenny$ pwd
/Users/kenny/Desktop/DEVOPS
MacBook-Air:DEVOPS kenny$ ls
NextWorkkeypair.pem
MacBook-Air:DEVOPS kenny$ ssh -i NextWorkkeypair.pem ec2-user@13.60.58.27
The authenticity of host '13.60.58.27 (13.60.58.27)' can't be established.
ED25519 key fingerprint is SHA256:cod0qdBclYQ3ReVciqBjuxlhptDhsP6RHRdWduQly.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.60.58.27' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

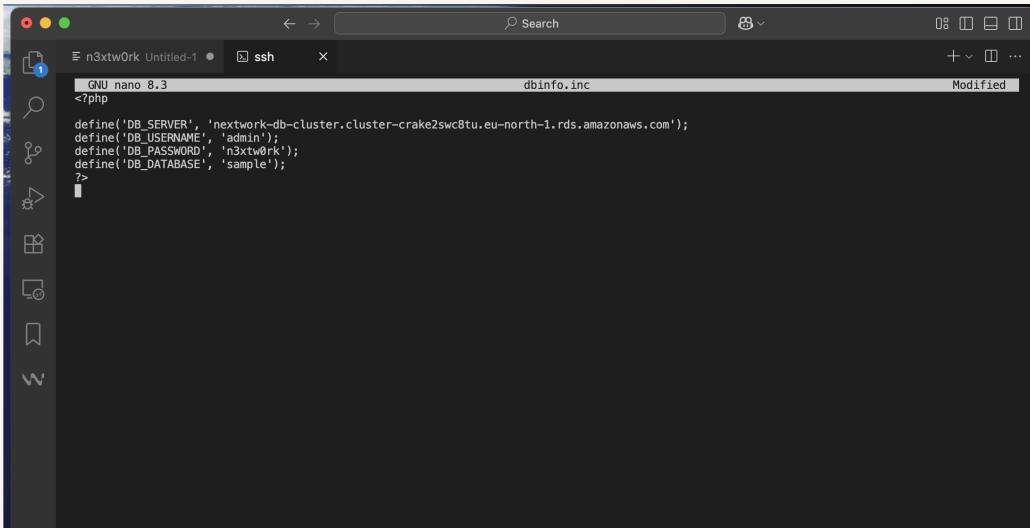
[ec2-user@ip-172-31-34-236 ~]$
```

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.

Connecting my Web App to Aurora

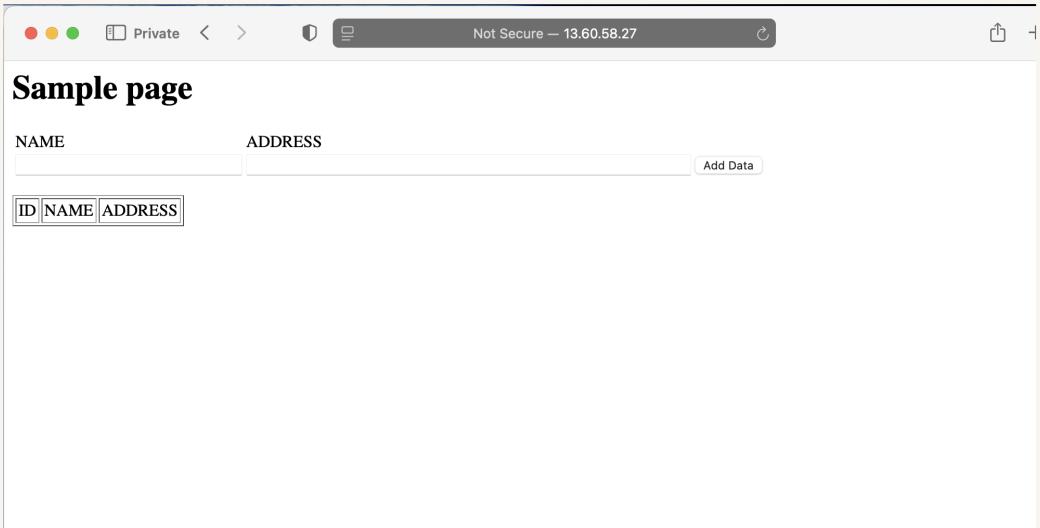
I set up my EC2 instance's connection details to my database by creating a new folder (inc) inside the /var/www directory, then adding a settings file called dbinfo.inc. In this file, I defined my Aurora database endpoint, username, password, and database name so that my web app can securely connect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



```
GNU nano 8.3
dbinfo.inc
Modified
<?php
define('DB_SERVER', 'nextwork-db-cluster.cluster-crake2swc8tu.eu-north-1.rds.amazonaws.com');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'n3xtw0rk');
define('DB_DATABASE', 'sample');
?>
```

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.

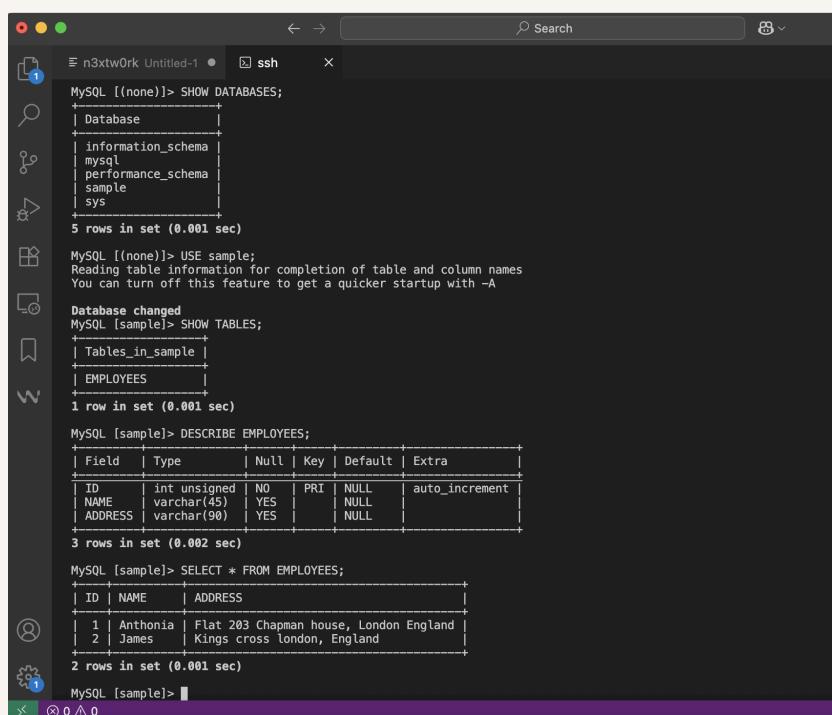


The screenshot shows a web browser window with the following details:

- Address Bar:** Not Secure — 13.60.58.27
- Content Area:**
 - Title:** Sample page
 - Form Fields:** NAME (input field), ADDRESS (input field), and a button labeled Add Data.
 - Data Table:** A table with three columns: ID, NAME, and ADDRESS. The first row contains the values 1, Kehinde, and Abiuwa.

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 screenshot shows a terminal window titled "n3xtw0rk Untitled-1" running an SSH session. The MySQL command-line interface is open, displaying the following commands and their results:

```
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sample |
| sys |
+-----+
5 rows in set (0.001 sec)

MySQL [(none)]> USE sample;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MySQL [sample]> SHOW TABLES;
+-----+
| Tables_in_sample |
+-----+
| EMPLOYEES |
+-----+
1 row in set (0.001 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 | Anthonia | Flat 203 Chapman house, London England |
| 2 | James | Kings cross london, England |
+-----+
2 rows in set (0.001 sec)

MySQL [sample]> 
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

