



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

# Connect a Web App to Amazon Aurora



ampahben3@gmail.com

Choose a database creation method

Standard create  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

Engine type: [Info](#)

<input checked="" type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 
<input type="radio"/> MySQL 	<input type="radio"/> PostgreSQL 
<input type="radio"/> MariaDB 	<input type="radio"/> Oracle 

**Aurora MySQL-Compatible Edition** >

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

# Introducing Today's Project!

## What is Amazon Aurora?

Amazon Aurora is a fast, scalable, and reliable database service supporting MySQL and PostgreSQL. It offers automated backups, replication, and strong security while minimizing downtime and costs ideal for efficiently managing large-scale apps.

## How I used Amazon Aurora in this project

In today's project, Amazon Aurora was used to set up a high-performance, scalable database to support our application. It provided automated backups, replication, and strong security, ensuring reliability and efficiency.

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

I had to create an EC2 instance first, thinking I could set up the database right away. Setting up EC2 first ensures proper hosting and integration, but it was an unexpected step in the process.

## This project took me...

This project took 30 minutes since I was familiar with databases, but learning something new added time. The unexpected need to create an EC2 instance first helped me understand how database hosting integrates with AWS services.

# In the first part of my project...

## Creating an Aurora Cluster

A relational database is a system that organizes data into structured tables with rows and columns. It links related data using keys, ensuring efficient storage, retrieval, and integrity. This model is widely used for managing structured information.

Aurora is a good choice when you need high-performance, scalable databases. It offers fast replication, automated backups, strong security, and cost savings over traditional databases. Companies like Coca Cola uses Aurora to store their customers data

**Choose a database creation method**

Standard create  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create  
Use the recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

**Engine type** [Info](#)

Aurora (MySQL Compatible) 

MySQL 

MariaDB 

Aurora (PostgreSQL Compatible) 

PostgreSQL 

Oracle   
**ORACLE**

**Aurora MySQL-Compatible Edition** >

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

# Halfway through I stopped!

I stopped creating my Aurora database because I want to connect to EC2, but I haven't set up a web app server for my project yet. Without it, the database connection wouldn't be useful for my application. I needed to create the web server first.

## Features of my EC2 instance

I created a new key pair for my EC2 instance because it serves as my login credentials, allowing secure access to my instance. With this key pair, I can connect, modify settings, and manage my EC2 environment with full control.

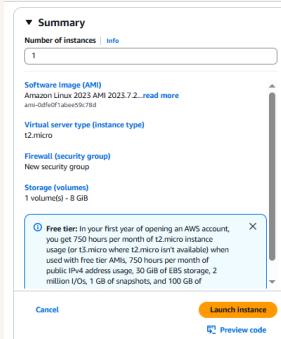
When I created my EC2 instance, I noted the Public IPv4 DNS, which defines its location, and the Key Pair, which serves as my login credentials giving secure access and control over my instance's configurations and operations. These two are vital.

AM

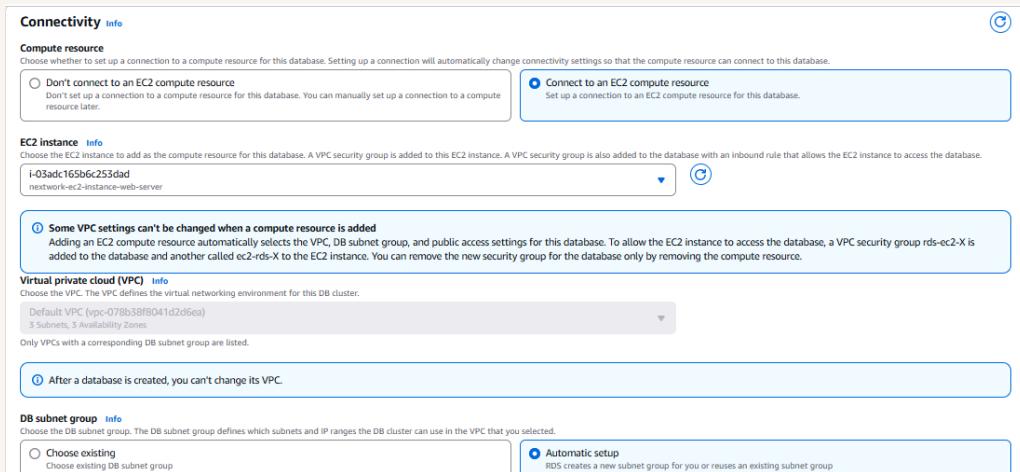
**ampahben3@gmail.com**

NextWork Student

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



# Then I could finish setting up my database



Aurora Database uses clusters because they allow data segmentation, replication, and efficient scaling. Clusters enhance availability, performance, and fault tolerance, ensuring your database remains reliable and responsive even under heavy workloads



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

# The place to learn & showcase your skills

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

