



Connect a Web App to Amazon Aurora

M

Mohammed Dawoud Mota

Create database [Info](#)

Choose a database creation method

Full configuration
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 	<input type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2

Engine version

Aurora MySQL 3.10.3 (compatible with MySQL 8.0.42) - default for major version 8.0

Enable RDS Extended Support [Info](#)
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the Amazon Aurora documentation.

Templates

Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input checked="" type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.
--	---

Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a relational database that's built for high performance and large scale applications. It uses clusters with a writer and reader instance, which keeps your data available and gives you automatic failover and backups. It's designed for big workloads that need speed, reliability and strong uptime.

How I used Amazon Aurora in this project

I used Amazon Aurora by creating a brand-new Aurora MySQL database cluster and configuring it from scratch. I set the engine type to Aurora MySQL, chose the Dev Test template, created the cluster identifier and admin credentials, and launched the database with a writer and reader instance. I also connected the Aurora database to the EC2 instance I created so it could act as the web app server.

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

One thing I didn't expect in this project was seeing how Aurora automatically creates a whole database cluster with a writer and reader instance. I didn't realize it would show multiple databases in the list right away, and it was interesting to learn that Aurora uses clusters to keep everything highly available and ready for failover.



M

Mohammed Dawoud M...

NextWork Student

nextwork.org

This project took me...

This project took me around an hour to complete.



Mohammed Dawoud M...

NextWork Student

nextwork.org

In the first part of my project...

Creating an Aurora Cluster

A relational database stores data in tables with rows and columns, allowing relationships between data points. It uses SQL to easily manage, query, and organize information efficiently.

Aurora is a good choice when you need a relational database to support something large-scale with peak performance and uptime.



Mohammed Dawoud M...

NextWork Student

nextwork.org

Create database [Info](#)

Choose a database creation method

Full configuration
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 	<input type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2 

Engine version

Aurora MySQL 3.10.3 (compatible with MySQL 8.0.42) - default for major version 8.0

Enable RDS Extended Support [Info](#)
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the Amazon Aurora documentation.

Templates

Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input checked="" type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.
--	---

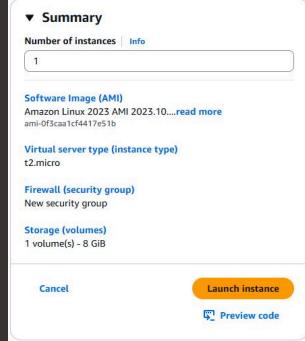
Halfway through I stopped!

I stopped creating my Aurora database because I needed to create an EC2 instance so that I could connect the web app to the database.

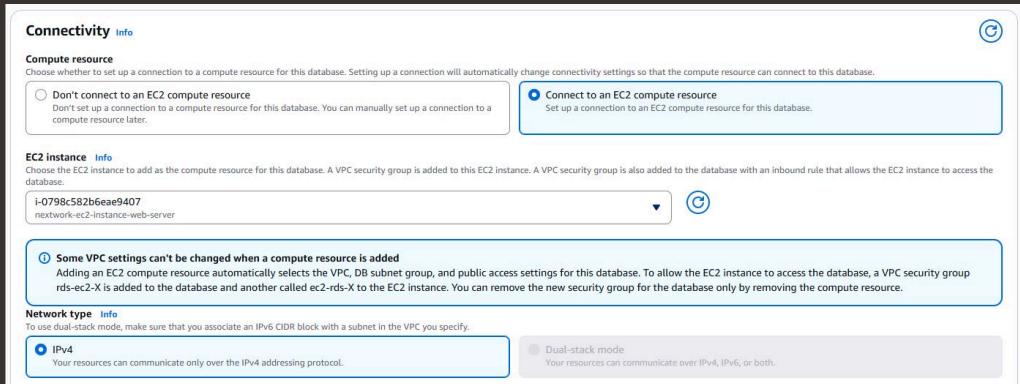
Features of my EC2 instance

I created a new key pair for my EC2 instance because this is what will allow me to access my instance and perform any of the needed modifications to the instance.

When I created my EC2 instance, I took particular note of the IPv4 address and the Key pair name. These are important because IPv4 tells you where the server is located, and the key pair is what allows you into the server. Without both, you can not get into the server.



Then I could finish setting up my database



Aurora Database uses clusters because a database cluster in Aurora is a group of database copies that work together, so your data is always available. Each cluster consists of a primary instance (where all write operations occur) and multiple read replicas as back-ups. If your database's primary instance fails, one of the replicas can be promoted to primary automatically.



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

