

Connect a Web App to Amazon Aurora

DI

Dineshraj Dhanapathy

Create database [Info](#)

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

Engine version
Aurora MySQL 5.05.2 (compatible with MySQL 8.0.52) - 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](#).

Parallel query is off by default. To enable it, use a DB instance parameter group with the aurora_parallel_query parameter enabled. [Learn more](#)

Templates
Choose a sample template to meet your use case.

Production Use defaults for high availability and fast, consistent performance.

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 fully managed relational database offering high performance, scalability, and automatic failover. It's useful for cost-efficient, fault-tolerant applications needing MySQL or PostgreSQL compatibility.

How I used Amazon Aurora in this project

I used Amazon Aurora to create a highly available relational database cluster, ensuring fault tolerance and scalability. It connects to an EC2-hosted web app, providing efficient data storage and retrieval.

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

I didn't expect the ARN role permission issue when creating the EC2 instance for the database. It required additional IAM permissions to allow proper access, delaying the setup and needing troubleshooting.



DI

Dineshraj Dhanapathy
NextWork Student

NextWork.org

This project took me...

This project took me a few hours, including setting up the EC2 instance, configuring the Aurora database, troubleshooting IAM role issues, and ensuring proper connectivity between the web app server and database.

In the first part of my project...

Creating an Aurora Cluster

A relational database is a structured system that stores data in tables with rows and columns. It uses SQL for queries and maintains relationships between tables using keys, ensuring data integrity and efficiency.

Aurora is a good choice when you need a fully managed, high-performance relational database with auto-scaling, fault tolerance, and up to 5x MySQL or 3x PostgreSQL speed at lower costs than commercial DBs.

DI

Dineshraj Dhanapathy

NextWork Student

NextWork.org

Create database Info

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

Aurora (MySQL, Compatible) 

Aurora (PostgreSQL Compatible) 

MySQL 

PostgreSQL 

MariaDB 

Oracle 

Microsoft SQL Server 

IBM Db2 

Engine version

Aurora MySQL 3.05.2 (compatible with MySQL 8.0.52) - 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](#).

⚠ Parallel query is off by default. To enable it, use a DB instance parameter group with the aurora_parallel_query parameter enabled. [Learn more](#)

Templates

Choose a sample template to meet your use case.

Production
Use defaults for high availability and fast, consistent performance.

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 set up an EC2 instance first to act as a web app server, ensuring a seamless connection between the application and the database for smooth operation.

Features of my EC2 instance

I created a new key pair for my EC2 instance because it provides a secure way to connect via SSH, ensuring only authorized access while protecting the instance from unauthorized logins and security threats.

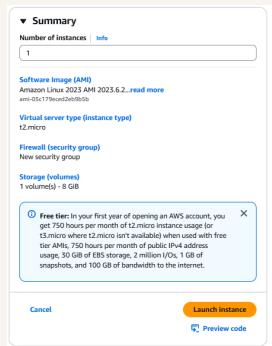
When I created my EC2 instance, I took particular note of the public IP, private IP, security group, instance ID, and key pair. These are crucial for secure access, network setup, and managing the instance.

DI

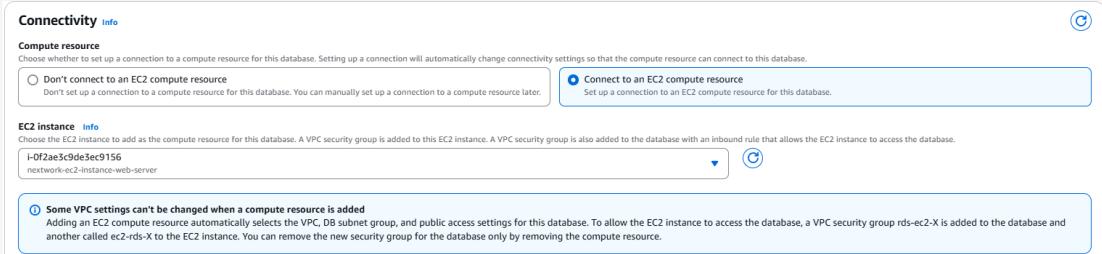
Dineshraj Dhanapathy

NextWork Student

NextWork.org



Then I could finish setting up my database





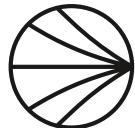
DI

Dineshraj Dhanapathy

NextWork Student

NextWork.org

Aurora Database uses clusters because they ensure high availability, fault tolerance, and scalability by using a primary instance for writes and multiple read replicas for backups and failover protection.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

