

Connect a Web App to Amazon Aurora



Unmilan Mukherjee

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**

Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a type of database provided by AWS RDS. It allows us to create databases that can handle large amounts of traffic and huge scale.

How I used Amazon Aurora in this project

I used Amazon Aurora to create a database to store data for our web app.

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

I did not expect to see much of a difference between regular MySQL database and Aurora, but surprisingly there was a huge difference between the 2 (clusters being the main differentiator).

This project took me...

This project took me roughly half an hour.

In the first part of my project...

Creating an Aurora Cluster

A relational database is a type of database that organizes our data into tables, which is a collection of rows and columns. Similar to a spreadsheet. We navigate and use this type of database using a language called SQL.

Aurora is a good choice when we need a database that we are expecting to have a high uptime and peak performance. So Aurora is really good for bigger projects or companies that are expecting a high load.

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**

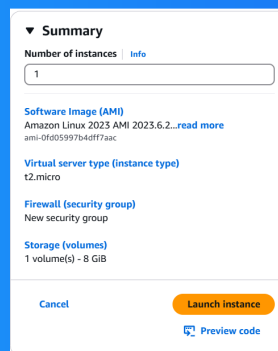
Halfway through I stopped!

I stopped creating my Aurora database because I still had not created an EC2 instance that I was going to connect my DB to for hosting the web app. We are going to be creating the EC2 instance first and come back to creating the DB after.

Features of my EC2 instance

I created a new key pair for my EC2 instance because we need to keep our web app's instance as secure as possible and to do that key pairs are important since they are basically like the login credentials to our instance.

When I created my EC2 instance, I took particular note of the Public IPv4 DNS(the address) and the Key pair name(the key) as those 2 are critical to access our EC2 instance.



The screenshot displays the 'Summary' section of the AWS Management Console for launching a new EC2 instance. The interface is clean with a white background and blue accents. The 'Number of instances' is set to 1. The 'Software image (AMI)' is 'Amazon Linux 2023.6.2' with a link to 'read more'. The 'Virtual server type (instance type)' is 't2.micro'. The 'Firewall (security group)' is 'New security group'. The 'Storage (volumes)' section shows '1 volume(s) - 8 GiB'. At the bottom, there are 'Cancel' and 'Launch instance' buttons, with a 'Preview code' link below the 'Launch instance' button.

▼ Summary	
Number of instances	Info
<input type="text" value="1"/>	
Software image (AMI)	
Amazon Linux 2023.6.2... read more	
ami-0f05997b4dfff7aac	
Virtual server type (instance type)	
t2.micro	
Firewall (security group)	
New security group	
Storage (volumes)	
1 volume(s) - 8 GiB	
Cancel	Launch instance
Preview code	

Then I could finish setting up my database

Connectivity Info

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

☐ Don't connect to an EC2 compute resource

☒ Connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Set up a connection to an EC2 compute resource for this database.

EC2 instance Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

I-0c96ed54f4ab1c05d

network-ec2-instance-web-server

Some VPC settings can't be changed when a compute resource is added

Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group rds-ec2-X is added to the database and another called ec2-rds-X to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

Network type Info

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

☒ IPv4

☐ Dual-stack mode

Your resources can communicate only over the IPv4 addressing protocol.

Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) Info

Aurora Database uses clusters to enhance scalability, availability, and performance. Clusters provide fault tolerance with multi-AZ replication, enable read scalability via replicas, auto-scale storage, ensure fast failovers, and optimize resources.