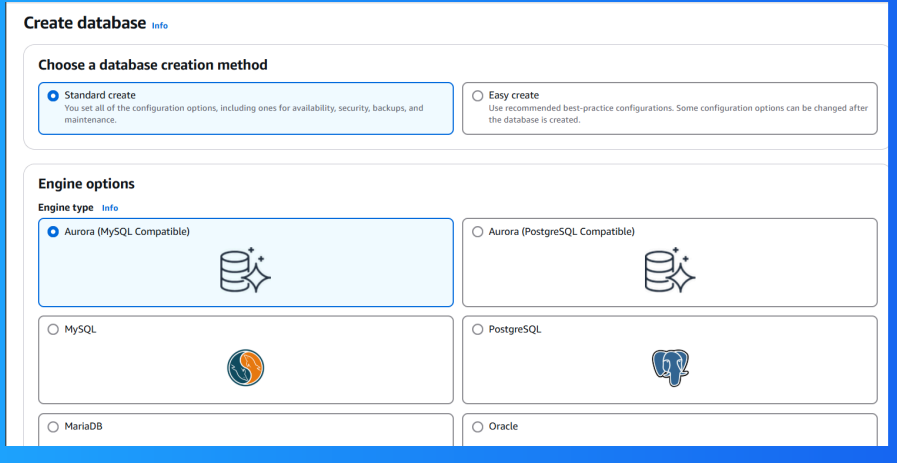
Connect a Web App to Amazon Aurora

Unmilan Mukherjee



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...

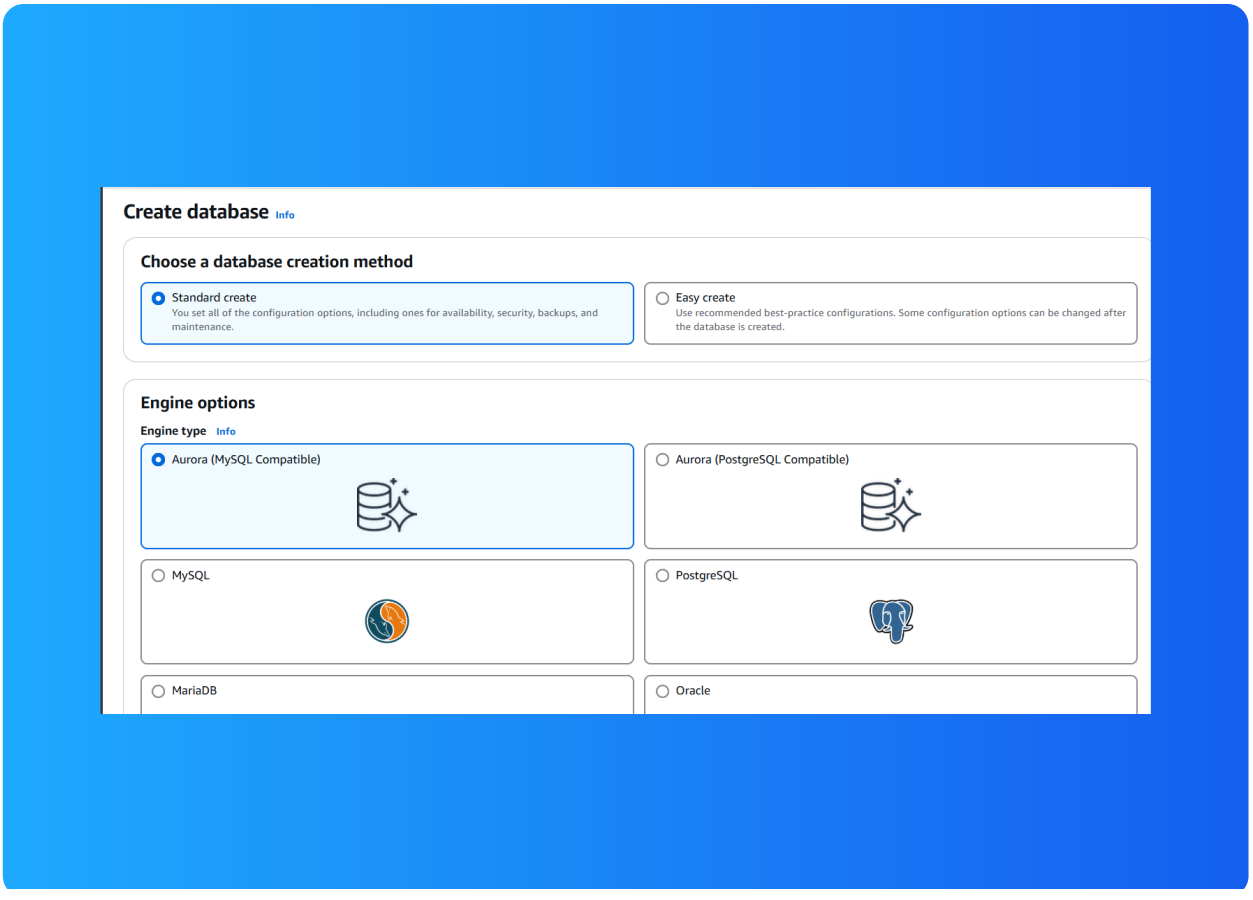
This project took me roughly half an hour.

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.



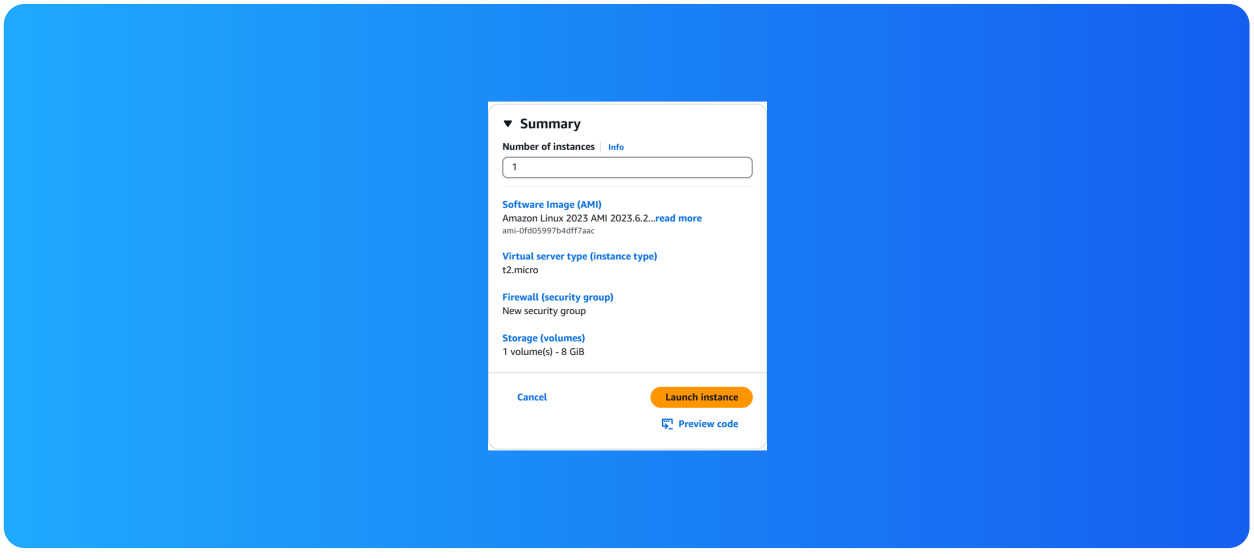
Halfway through I stopped!

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.

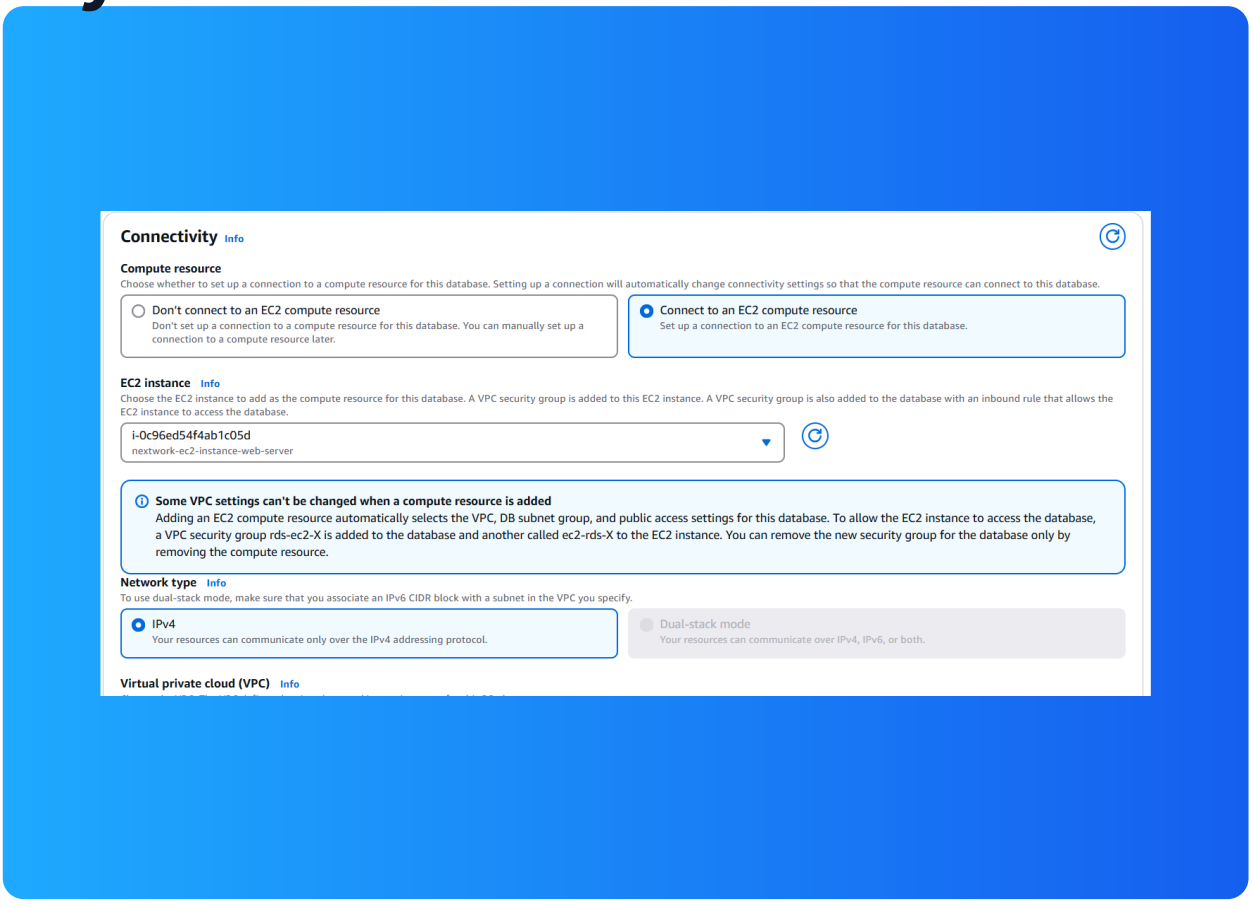
Features of my EC2 instance

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.



Then I could finish setting up my database



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.