



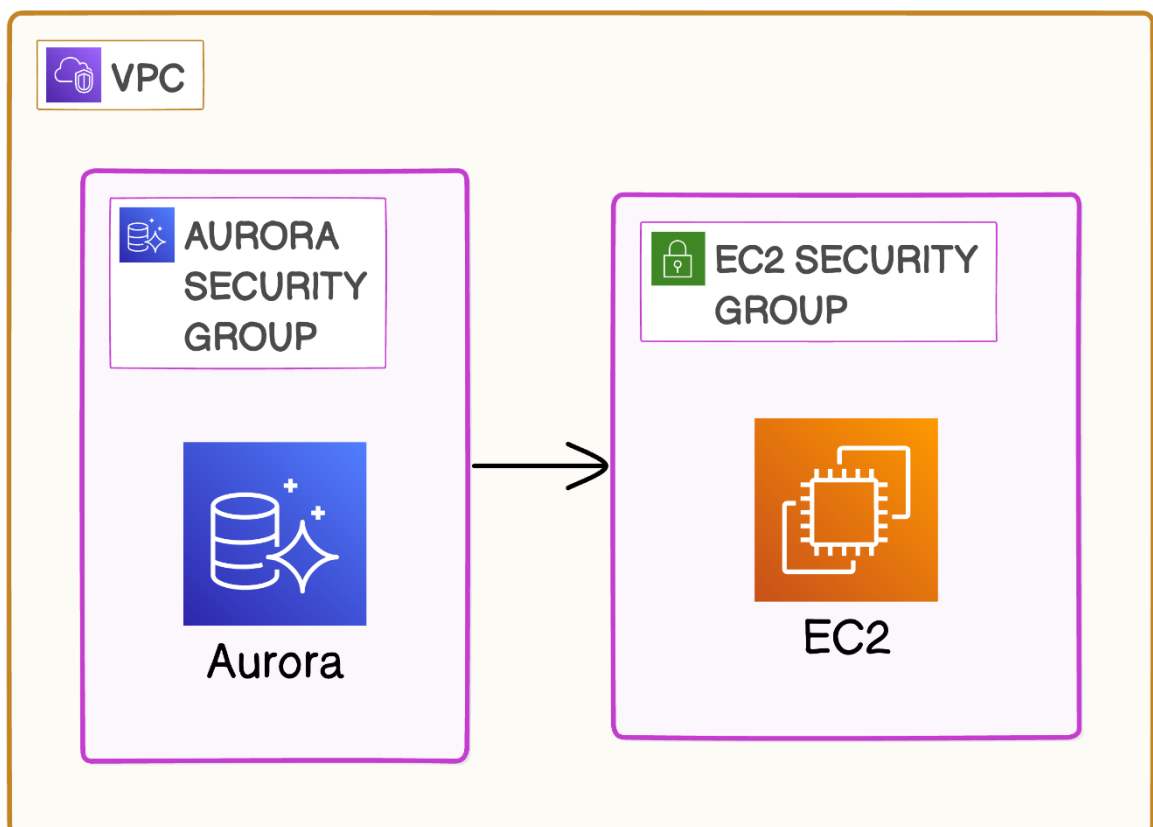
Aurora Database with EC2 Instance



Prajit Venkatachalam

<https://www.linkedin.com/in/prajit-venkatachalam/>

Aurora to Web App





Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a relational database service on AWS designed for handling large-scale databases that demand high performance and reliability. It's ideal for applications where uptime and speed are critical.

How I used Amazon Aurora in this project

In this project, I deployed an Amazon Aurora database, configured an EC2 instance independently, and linked the Aurora database to the EC2 instance, which served as a web server.

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

One unexpected aspect of this project was that I had to set up my EC2 instance first before I could complete the configuration of my Amazon Aurora database.

This project took me...

This project took me 1.5 hours to complete include report writing.



In the first part of my project...

Creating an Aurora Cluster

A relational database organizes data into tables with rows and columns, similar to a spreadsheet. It's called "relational" because the rows correspond to specific columns, and the columns describe data in the rows.

Aurora is an ideal choice for large-scale relational databases that demand high performance and minimal downtime, offering enhanced speed, availability, and scalability compared to standard database solutions.

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**

☐ **MariaDB**

☐ **PostgreSQL**

☐ **Oracle**

☐ **Microsoft SQL Server**

☐ **IBM Db2**



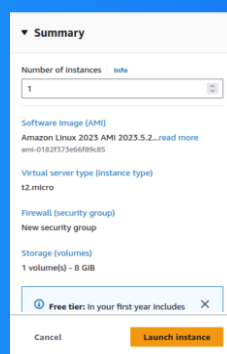
Halfway through I stopped!

I paused the creation of my Aurora database because I need to connect it to an EC2 instance first before completing the database setup, so I had to launch an EC2 instance first.

Features of my EC2 instance

I created a new key pair for my EC2 instance to enhance its security and provide direct access. This will allow me to configure the instance, develop a web app, and connect it to the Aurora database.

When I created my EC2 instance, I paid close attention to the instance's public IPv4 DNS and the key pair name. These two elements work together to indicate the location of the EC2 instance (my web server) and the keys that will provide access to it.





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
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.

☒ Connect to an EC2 compute resource
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-0aaec44331b7ca460 ▼ ↻
nextwork-ec2-instance-web-server

Aurora Database utilizes clusters to ensure high availability. Clusters consist of multiple copies of the same database, so if the primary instance fails, read replicas remain operational, providing backup and maintaining access to the data.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

