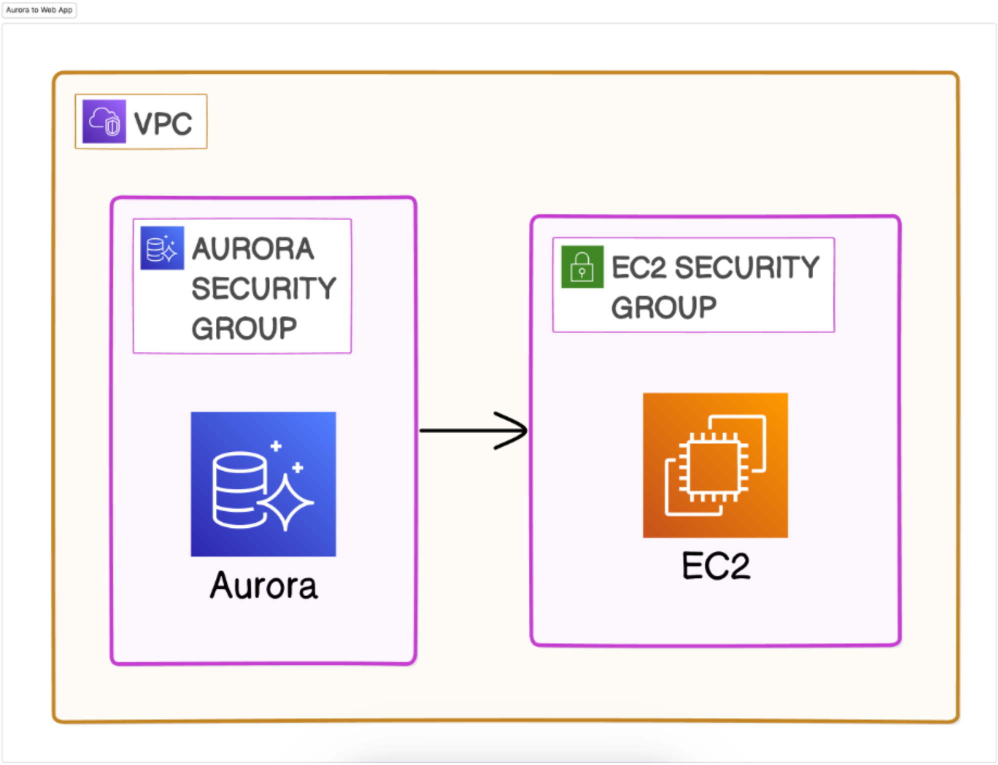
Connect a Web App to Amazon Aurora



Introducing Today's Project!

What is Amazon Aurora?

Amazon Aurora is a fully managed relational database compatible with MySQL and PostgreSQL. It provides high performance, scalability, automatic backups, and strong security, making it ideal for cloud applications with demanding workloads.

How I used Amazon Aurora in this project

In today's project, I used Aurora to set up a high-performance, scalable relational database. Its compatibility with MySQL, automatic backups, and strong security features allowed efficient data management and quick scaling to meet growing demands.

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

One thing I didn't expect in this project was how quickly the database scaling happened with Aurora. The automatic scaling feature handled increased traffic seamlessly without any manual intervention, making it much easier to manage.

This project took me...

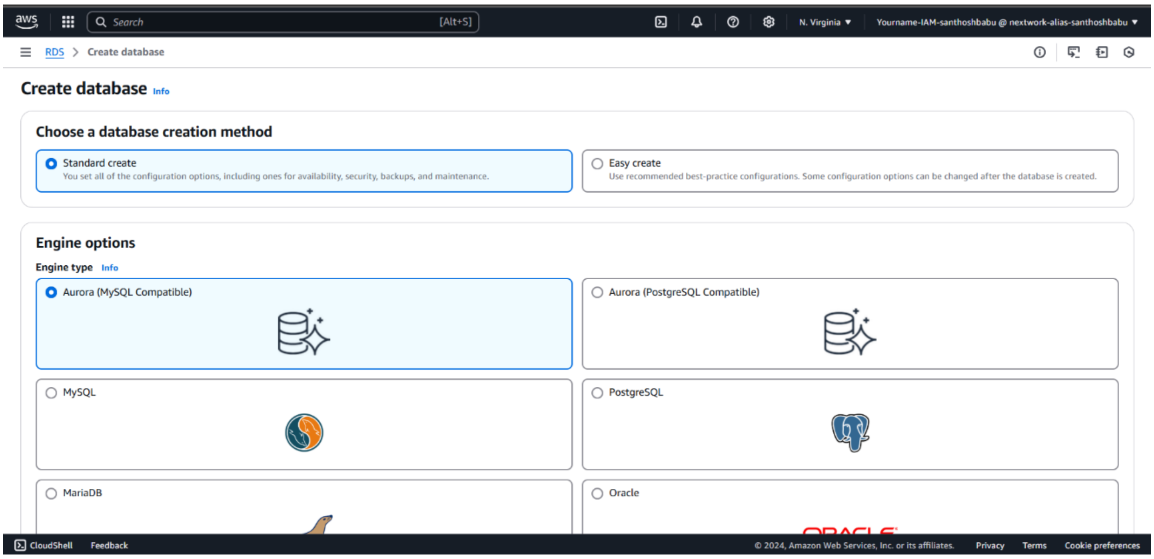
This project took me just half an hour to complete.

In the first part of my project...

Creating an Aurora Cluster

A relational database stores data in tables with rows and columns. Tables are linked using keys to establish relationships between data. SQL is used to manage, query, and update data, ensuring consistency, flexibility, and efficient data handling.

Aurora is a good choice when you need a high-performance, scalable relational database with automatic backups, fault tolerance, and low latency. It supports MySQL and PostgreSQL, offering strong security and seamless integration for large workloads.


aws
Search [Alt+S]
N. Virginia ▼
Yourname-IAM-santhoshbabu @ nextwork-alias-santhoshbabu ▼
RDS > Create database
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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

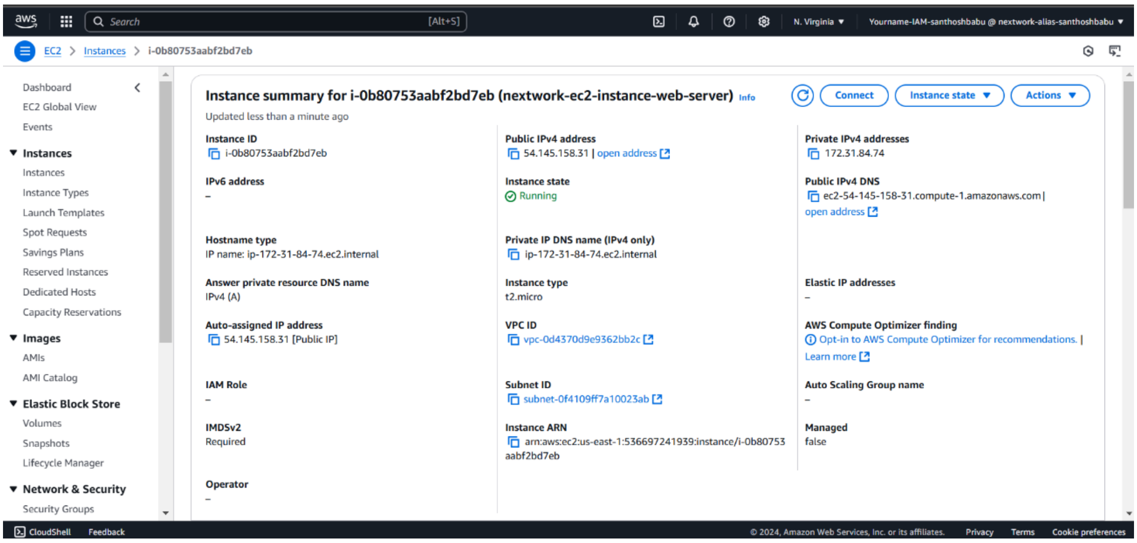
Halfway through I stopped!

I stopped creating my Aurora database because we haven't even created an EC2 instance yet. Without the proper infrastructure in place, such as an EC2 instance to connect to the database, it wouldn't be possible to set up or effectively use Aurora.

Features of my EC2 instance

I created a new key pair for my EC2 instance because it's required for securely accessing the instance via SSH. The key pair ensures that only authorized users can connect to the instance, providing an extra layer of security.

When I created my EC2 instance, I took note of the number of instances, OS type, virtual server type, firewall settings, and storage. These factors impact performance, security, compatibility, and data management for my applications.



Instance summary for i-0b80753aabf2bd7eb (nextwork-ec2-instance-web-server) Info

Updated less than a minute ago

Instance ID
i-0b80753aabf2bd7eb

IPv6 address
–

Hostname type
IP name: ip-172-31-84-74.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
54.145.158.31 [Public IP]

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
54.145.158.31 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-84-74.ec2.internal

Instance type
t2.micro

VPC ID
vpc-0d4370d9e9362bb2c

Subnet ID
subnet-0f4109ff7a10023ab

Instance ARN
arn:aws:ec2:us-east-1:536697241939:instance/i-0b80753aabf2bd7eb

Private IPv4 addresses
172.31.84.74

Public IPv4 DNS
ec2-54-145-158-31.compute-1.amazonaws.com | open address

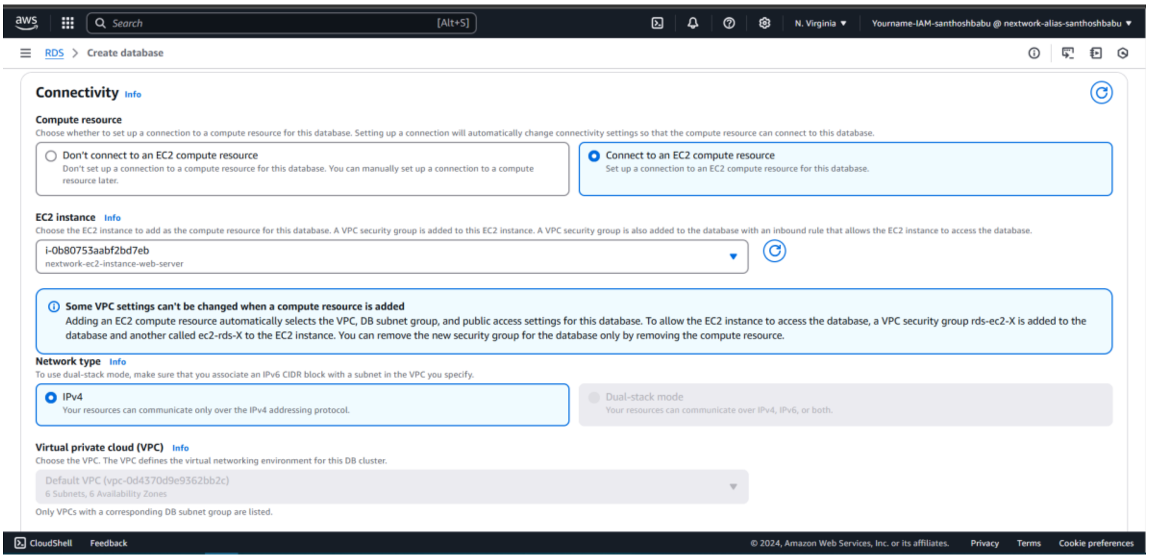
Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
–

Managed
false

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-0b80753aabf2bd7eb
nextwork-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
Your resources can communicate only over the IPv4 addressing protocol.

○ Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

Default VPC (vpc-0d4370d9e9362bb2c)
6 Subnets, 6 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

Aurora Database uses clusters because they offer high availability, fault tolerance, and automatic scaling. Clusters distribute data across multiple nodes, ensuring continuous performance, quick recovery, and seamless load balancing for reliability.

Today you've learnt how to:

- 🧱 **Create an Aurora MySQL Database**: You created an Aurora MySQL database instance from scratch in AWS, configured it with the appropriate settings, and connected it to an EC2 instance to prepare for hosting a web application.

- 💻 **Launch and Configure an EC2 Instance**: You successfully launched an EC2 instance using the Amazon Linux 2023 AMI, configured the instance to connect to your Aurora database, and set up the necessary security groups to allow traffic.