MOKGADI SELEPE

Build Your DB Server and Interact With Your DB Using an App

Here's what happened:

I'm learning about Amazon RDS, a service that helps me set up and manage a relational database in the cloud.

Here's what I'll be doing:

1. Launching a DB instance: I'll create a database instance using Amazon RDS, which will be highly available (meaning it's always accessible).
2. Configuring connections: I'll set up the database to allow connections from my web server.
3. Testing the database: I'll open a web application and interact with my database to see it in action.

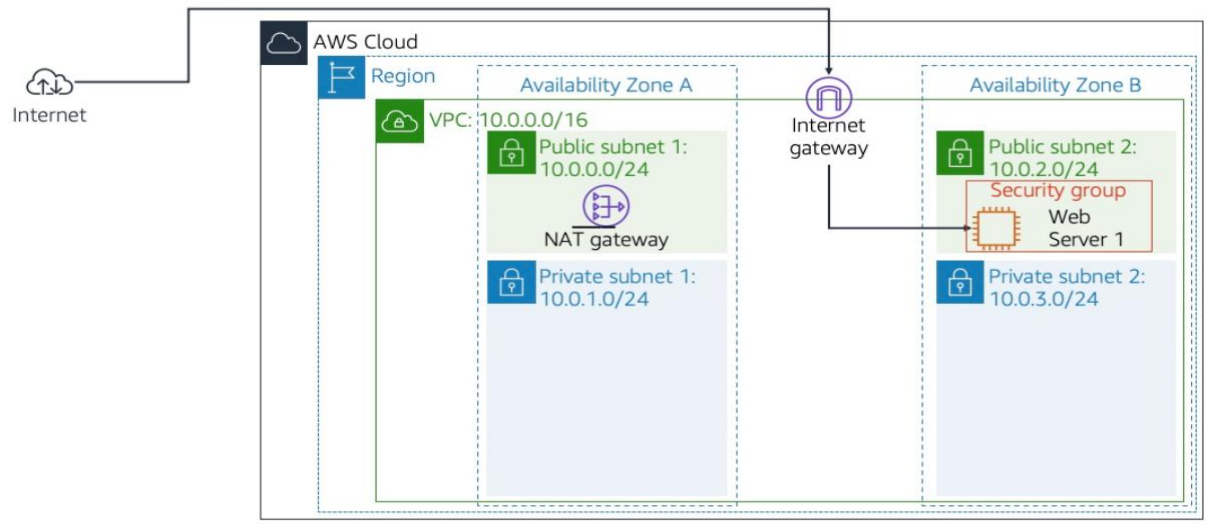
What is Amazon RDS?

Amazon RDS is a service that makes it easy to set up and manage a relational database in the cloud. It takes care of time-consuming tasks, so I can focus on my applications and business.

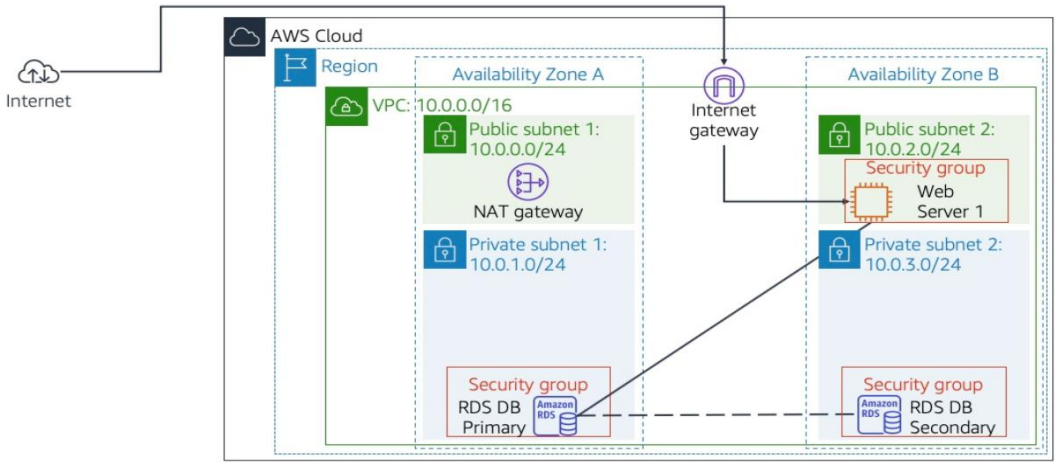
What can I do with Amazon RDS?

I can choose from six popular database engines, including MySQL, PostgreSQL, and Oracle, and scale my database as my needs grow.

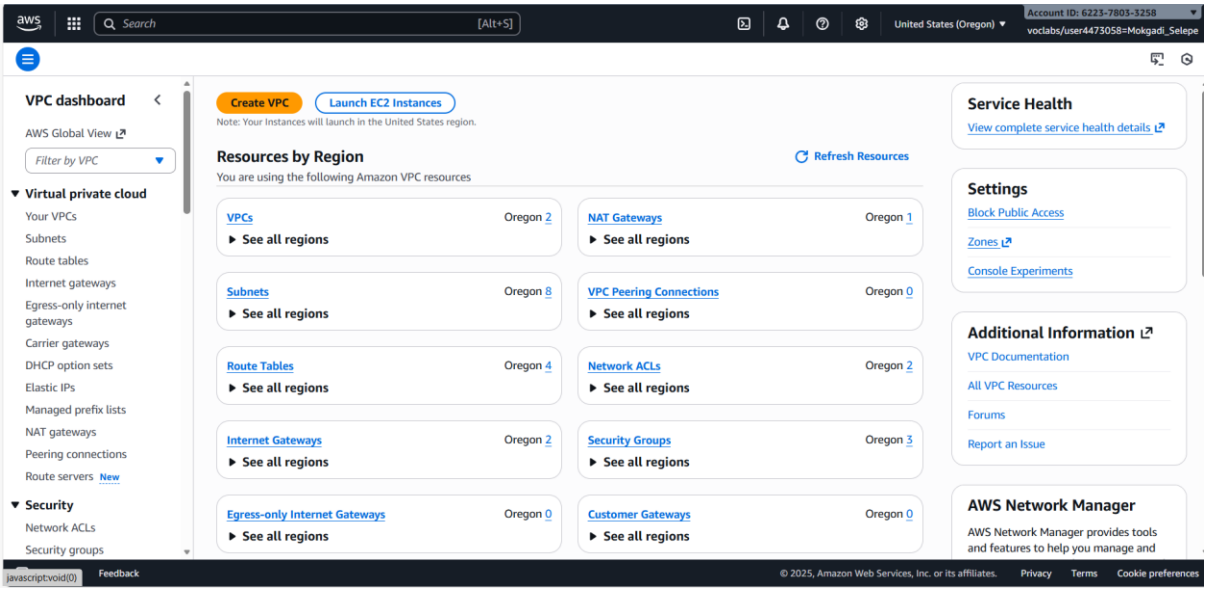
Scenario
You start with the following infrastructure:

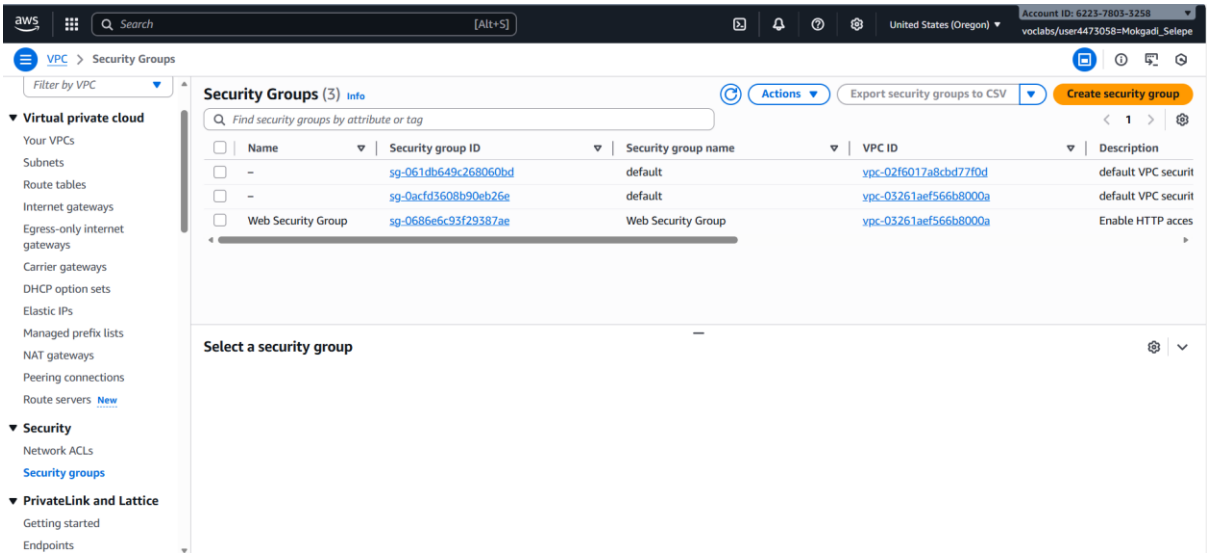


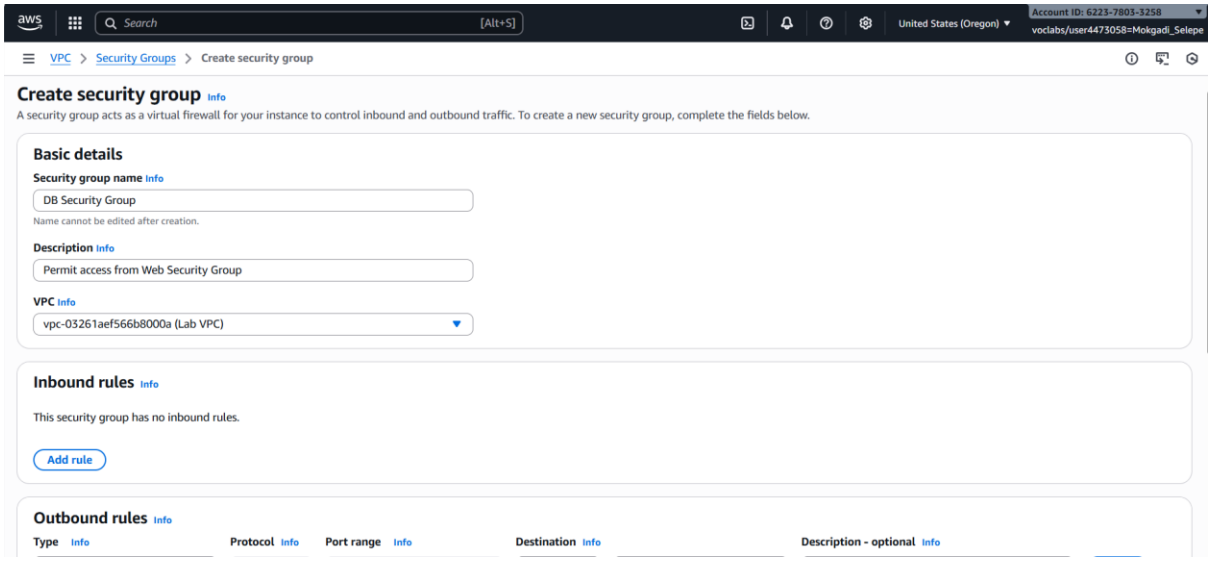
At the end of the lab, this is the infrastructure:

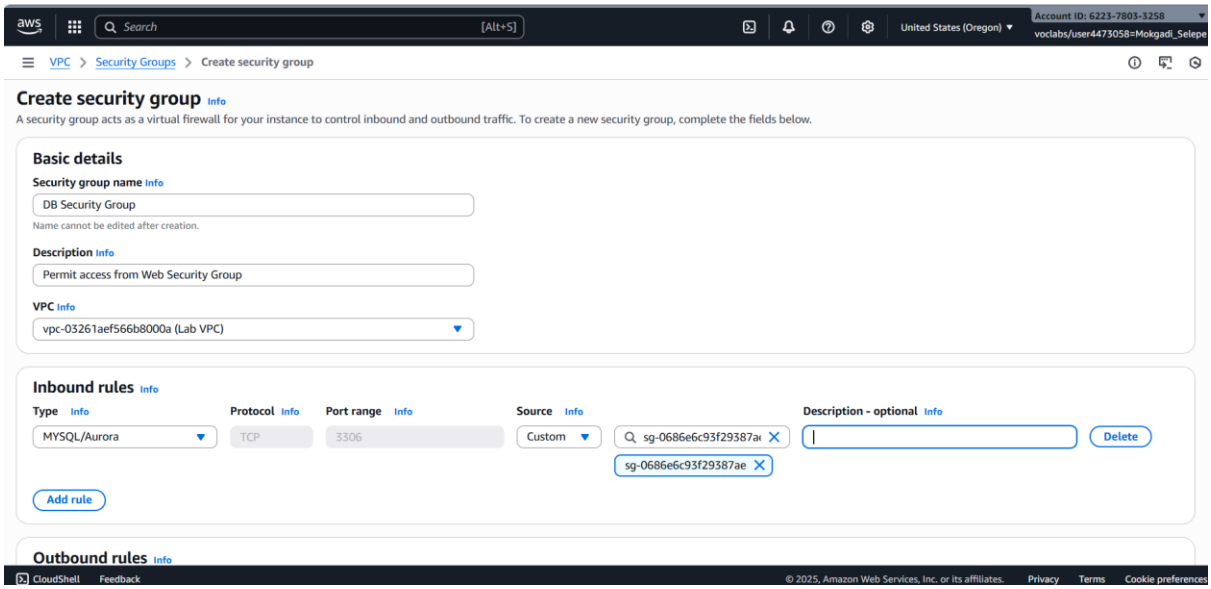


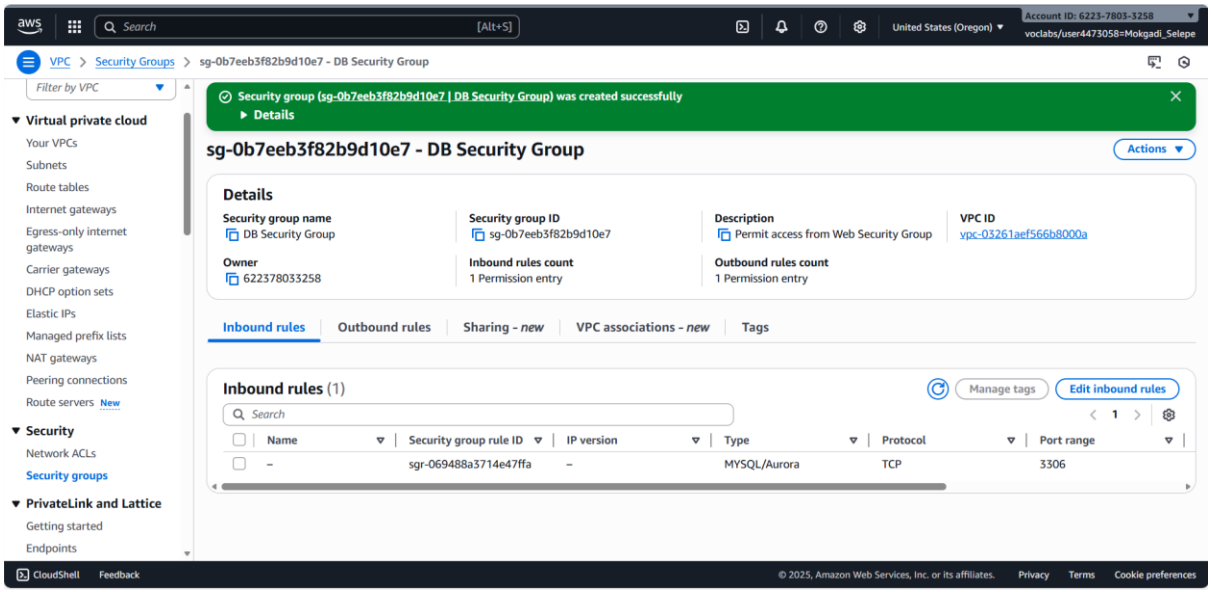
1: Create a Security Group for the RDS DB Instance











Here's what happened:

MOKGADI SELEPE

I created a security group to allow my web server to access my database.

Here's what I did:

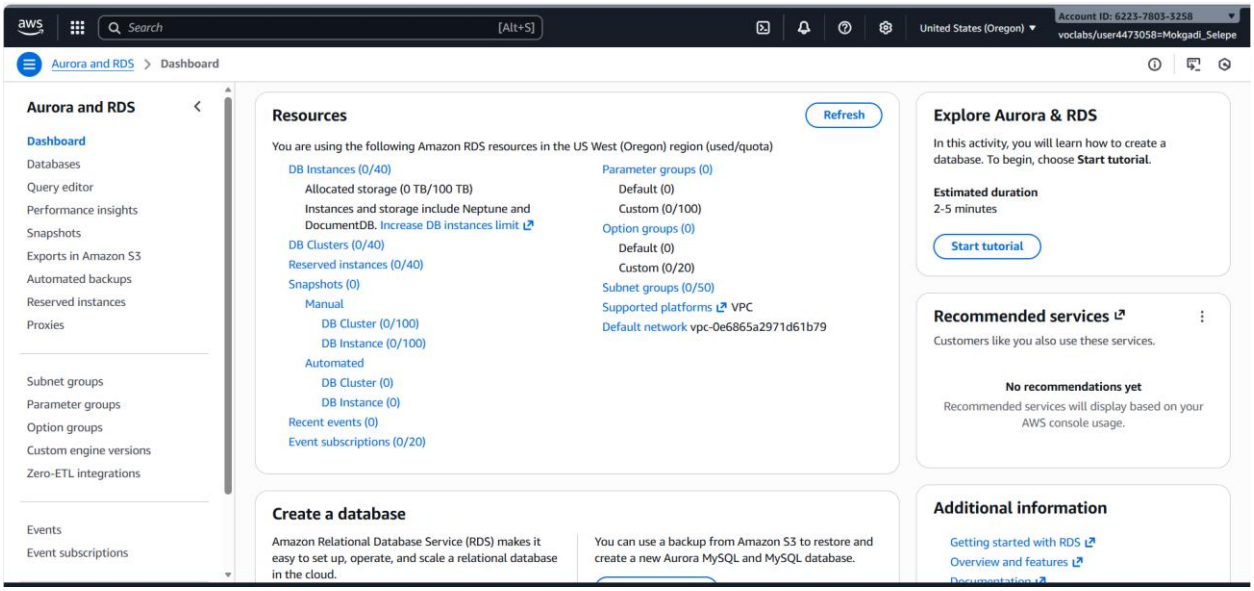
1. Created a security group: I made a new security group called "DB Security Group" to control access to my database.
2. Added a rule: I added a rule to allow incoming database requests from my web server's security group (Web Security Group).
3. Configured the rule: I set the rule to allow MySQL/Aurora traffic on port 3306 from the Web Security Group.

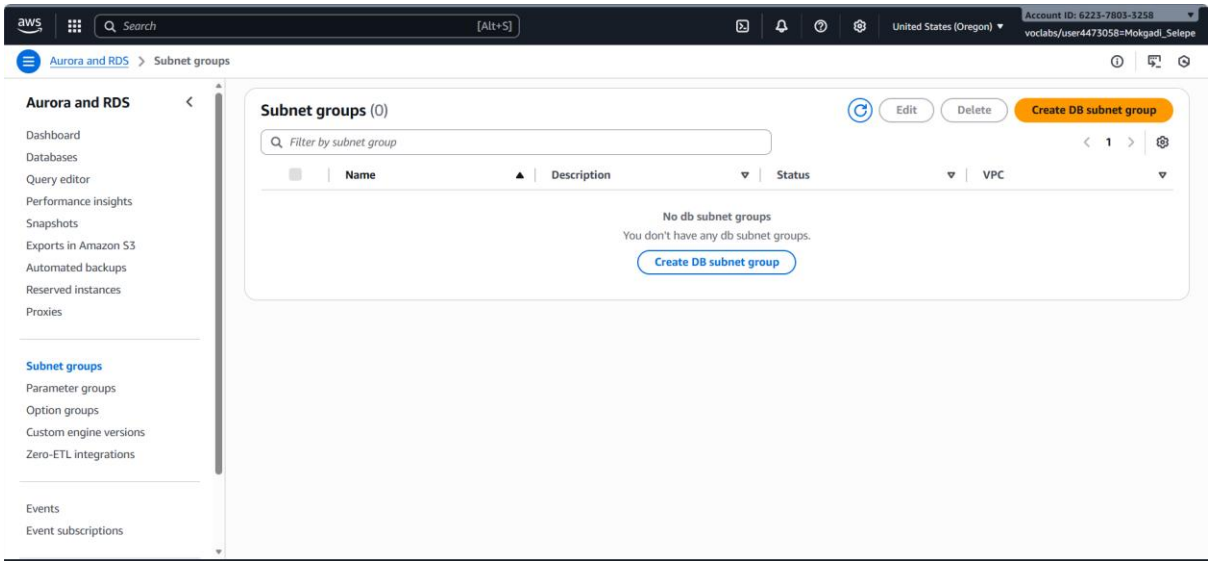
What does this mean?

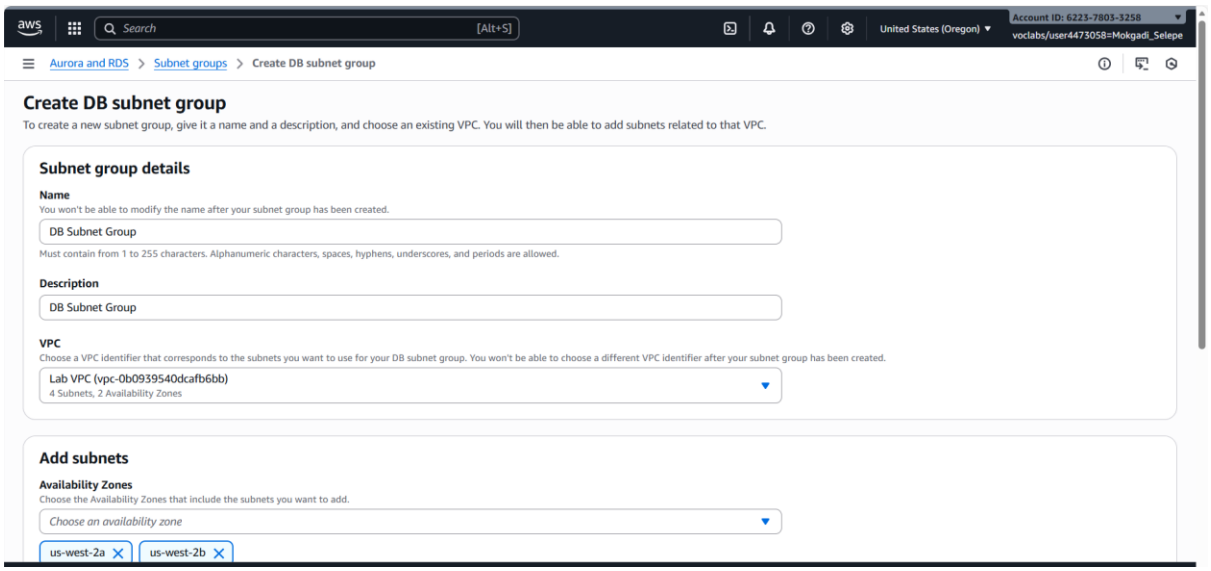
This security group will allow my web server to connect to my database, but block connections from other sources. It's like setting up a firewall to protect my database.

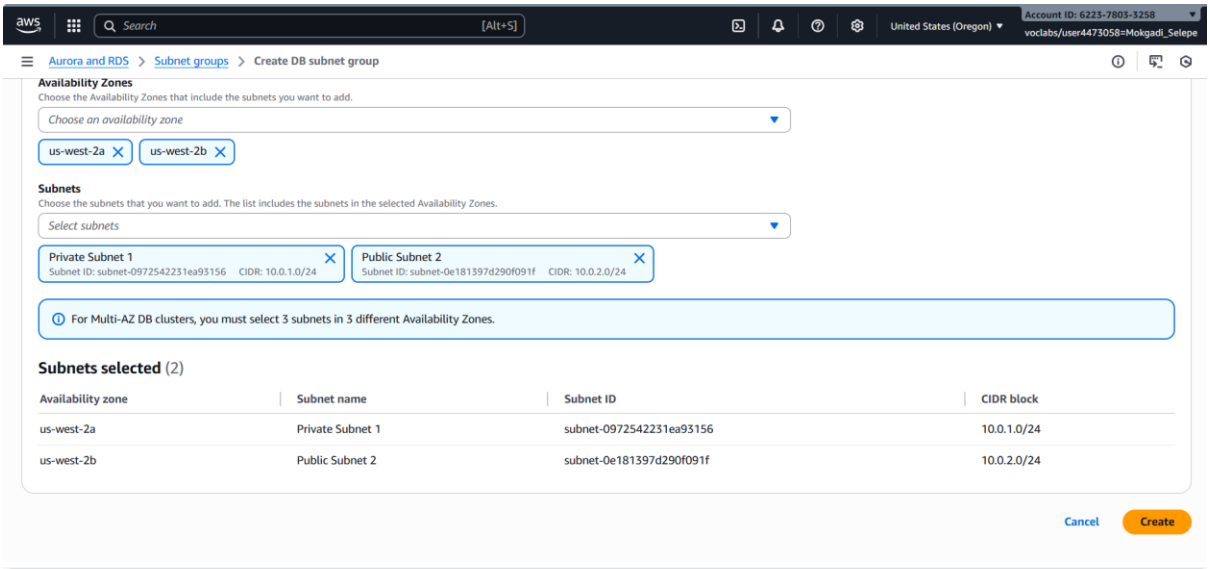
I'll use this security group when I create my database instance.

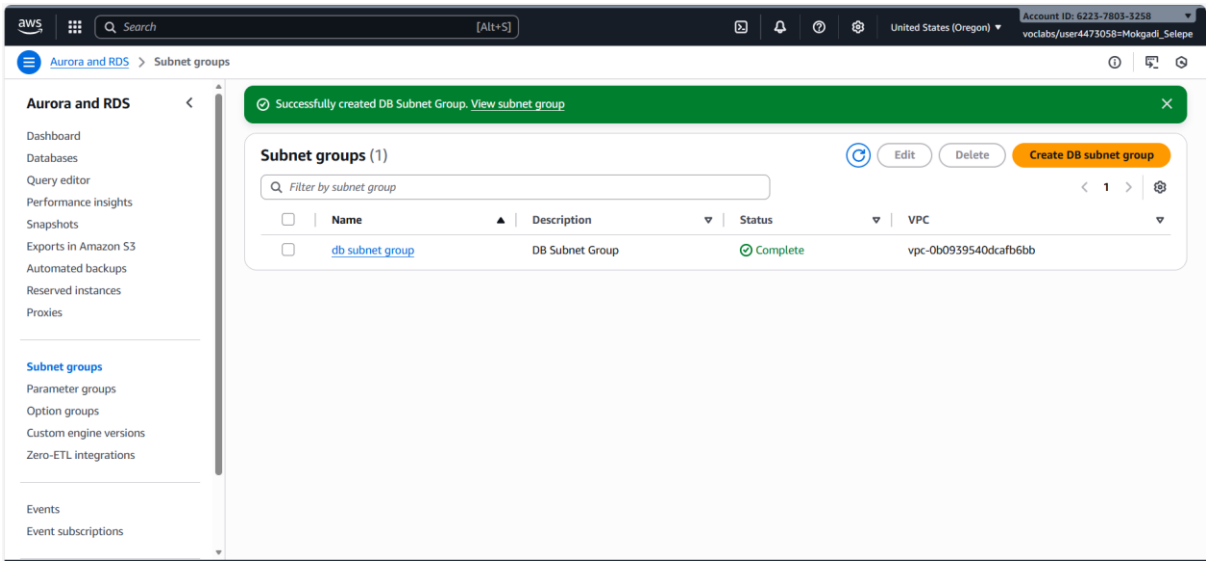
2: Create a DB Subnet Group




Search
[Alt+S]
United States (Oregon)
Account ID: 6223-7803-3258
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Subnet groups
Aurora and RDS
Dashboard
Databases
Query editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Event subscriptions
Subnet groups (0)
Edit
Delete
Create DB subnet group
Filter by subnet group
Name
Description
Status
VPC
No db subnet groups
You don't have any db subnet groups.
Create DB subnet group


Search
[Alt+S]
United States (Oregon)
Account ID: 6223-7803-3258
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Subnet groups > Create DB subnet group
Create DB subnet group
To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.
Subnet group details
Name
You won't be able to modify the name after your subnet group has been created.
DB Subnet Group
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.
Description
DB Subnet Group
VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
Lab VPC (vpc-0b0939540dcafb6bb)
4 Subnets, 2 Availability Zones
Add subnets
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-west-2a
us-west-2b





I've created a special group called a "DB Subnet Group" for my database.

This group tells Amazon's database service (RDS) which parts of my network (called "subnets") it can use to store my database.

Here's what I did:

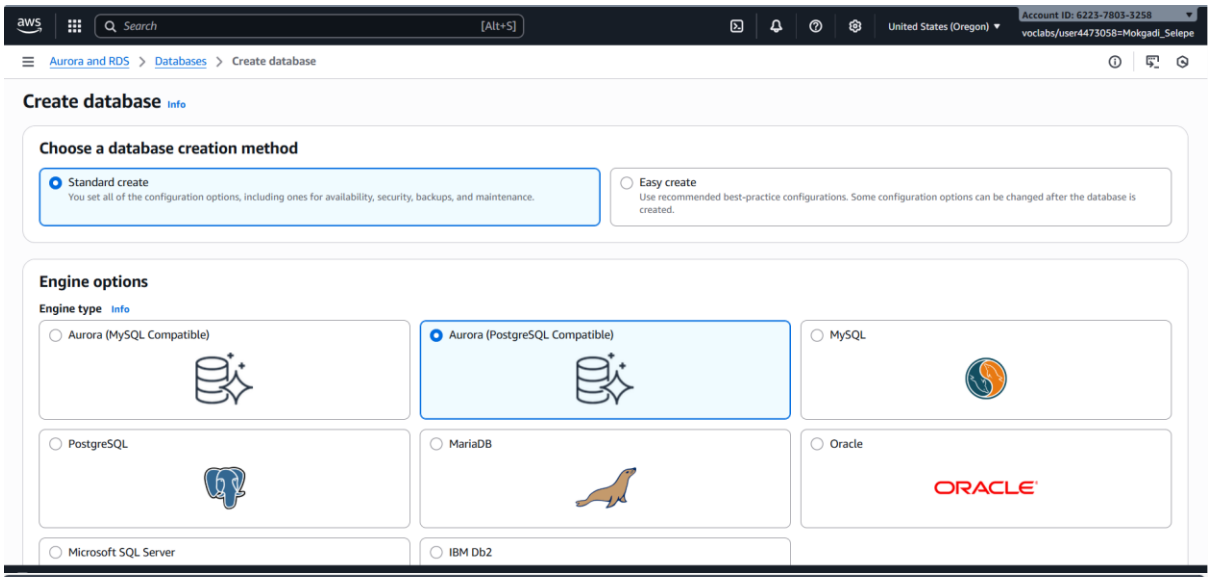
1. I chose two private areas of my network (called "subnets") in different locations (called "Availability Zones").
2. I put these subnets into a group called "DB Subnet Group".
3. I told RDS that it's okay to use these subnets for my database.

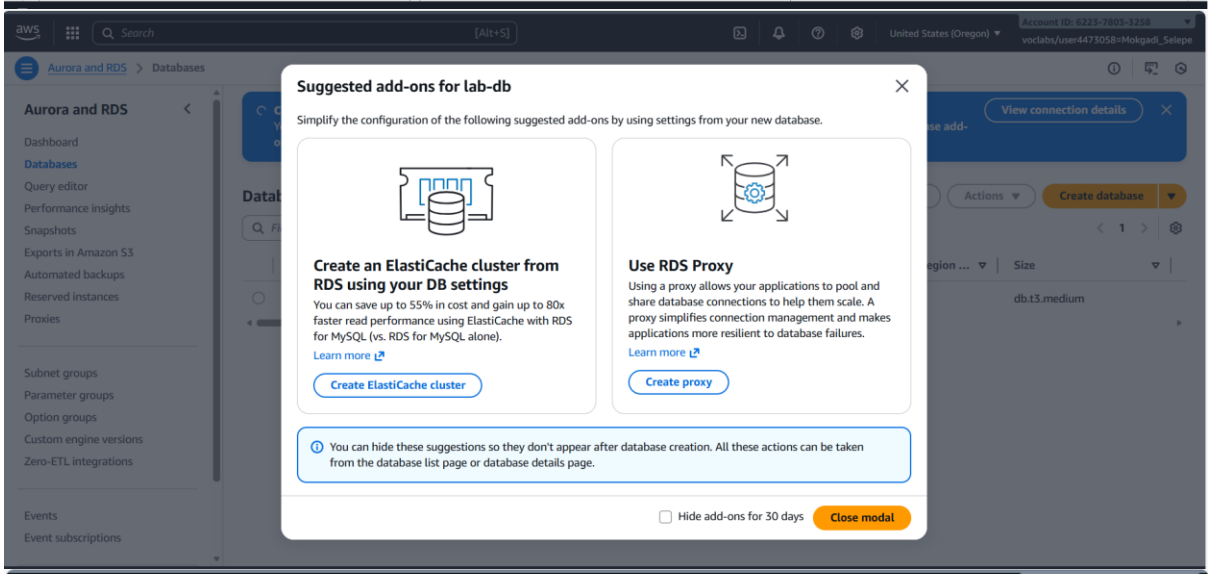
Why is this important?

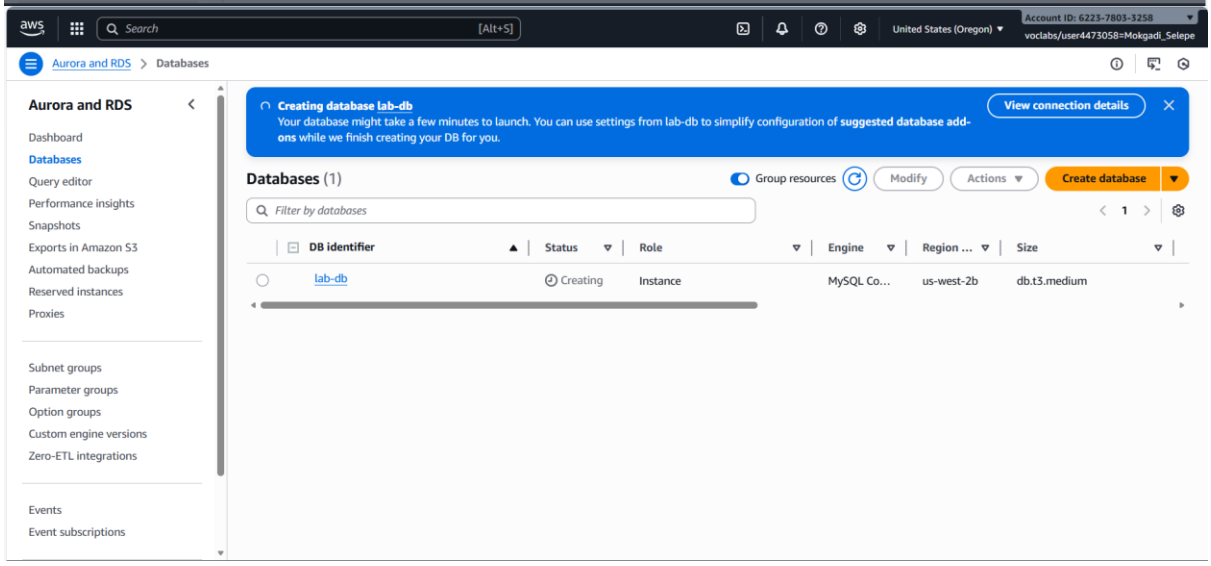
- It helps keep my database safe by putting it in a secure part of my network.
- It makes sure my database is available in multiple locations, so if one location has a problem, my database is still available.

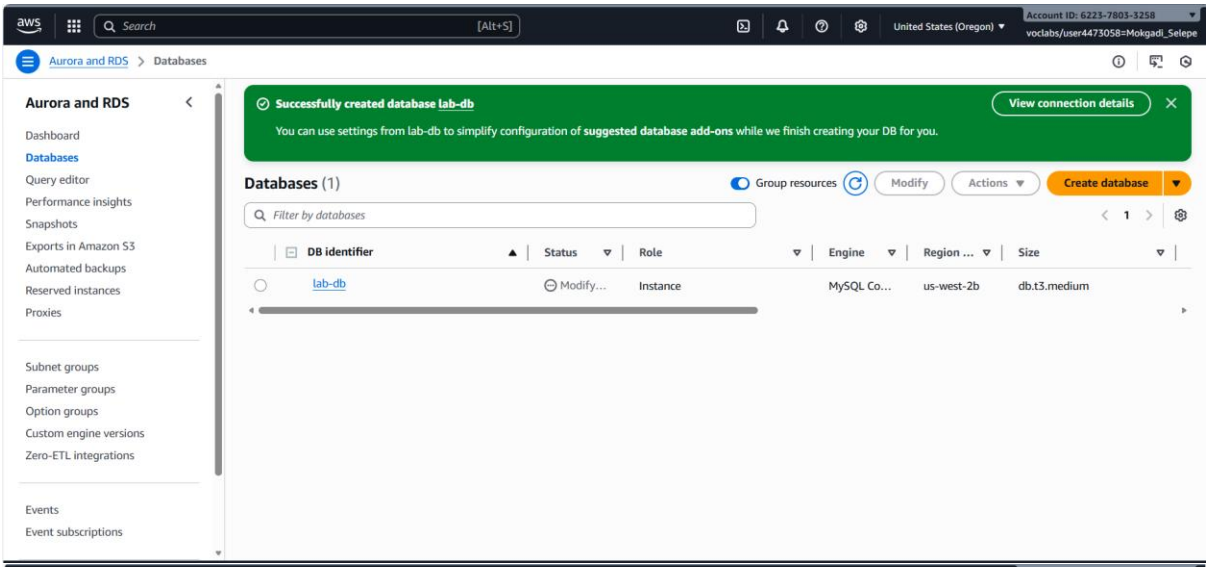
Think of it like choosing two safe and separate locations for my valuable data!

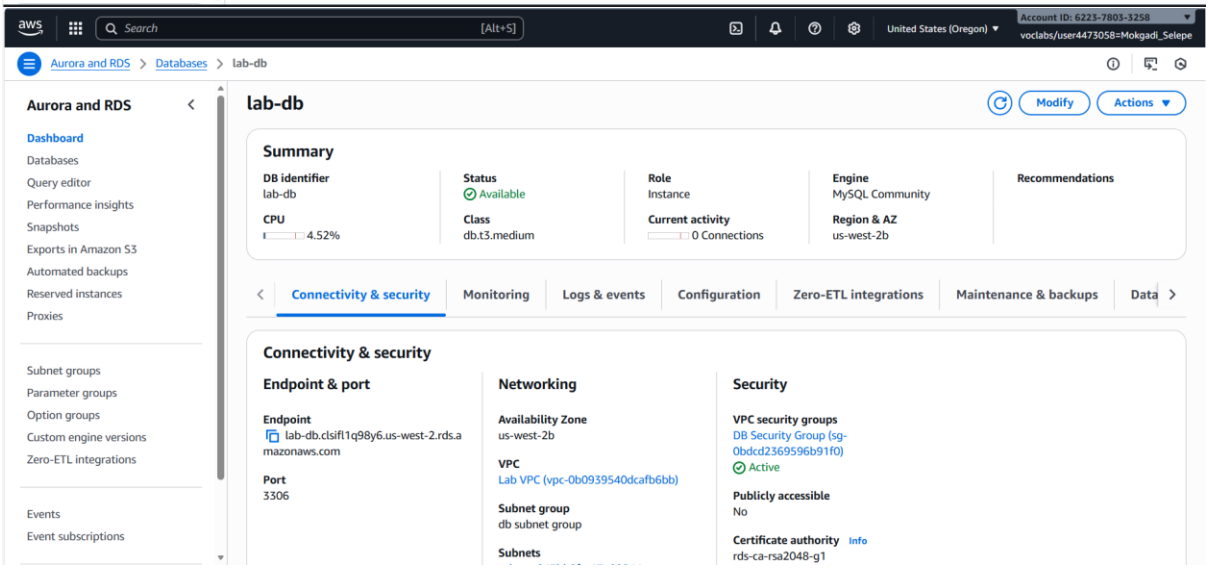
3: Create an Amazon RDS DB Instance











I've created a MySQL database instance on Amazon RDS that's designed for high availability and durability. Here's what happened:

I configured a Multi-AZ deployment, which means Amazon RDS creates a primary database instance and replicates it to a standby instance in a different Availability Zone (AZ).

I chose the latest MySQL version and a Dev/Test template, and set up my database instance with a specific identifier, username, and password. I selected a db.t3.medium instance class and General Purpose SSD storage, and configured my Virtual Private Cloud (VPC) and security group settings.

I created the database, and it's currently deploying in two different Availability Zones (AZs). Once it's available, I'll get an endpoint URL that I can use to connect to my database.

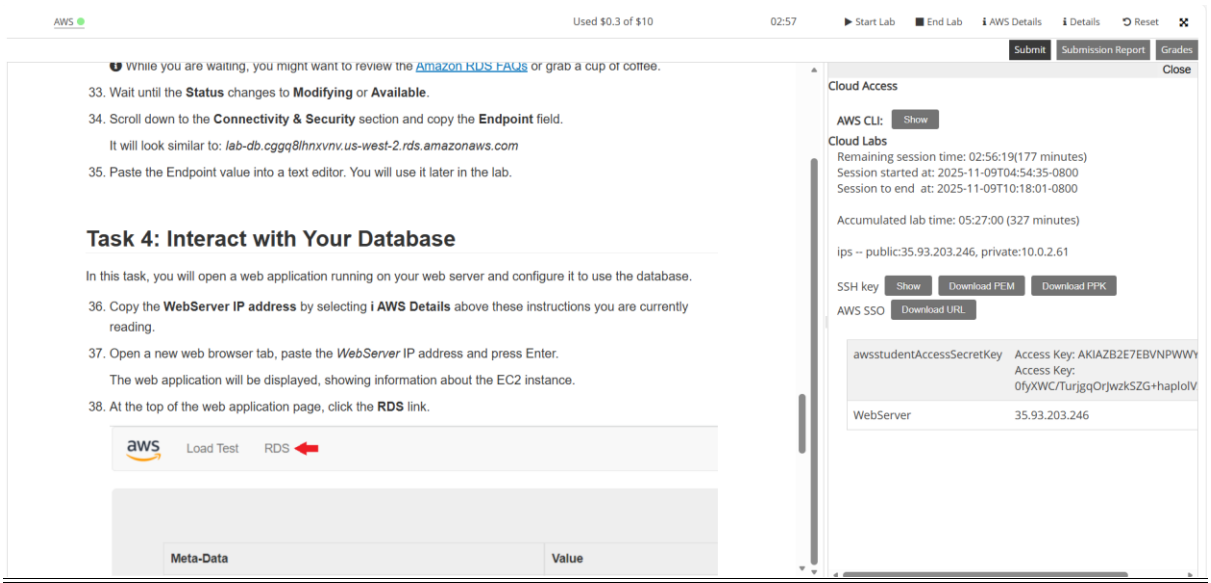
Here's what I did in simple terms:

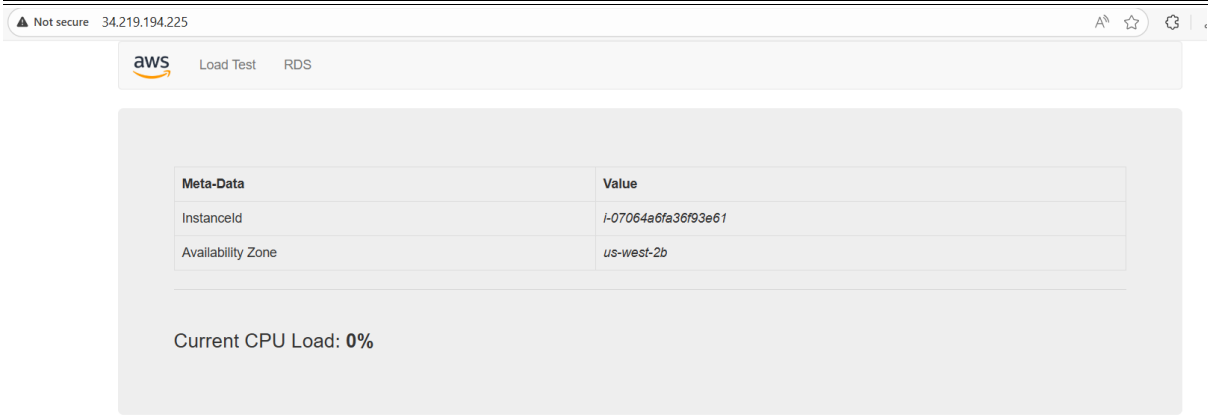
1. I created a database instance with a primary and standby instance in different AZs.

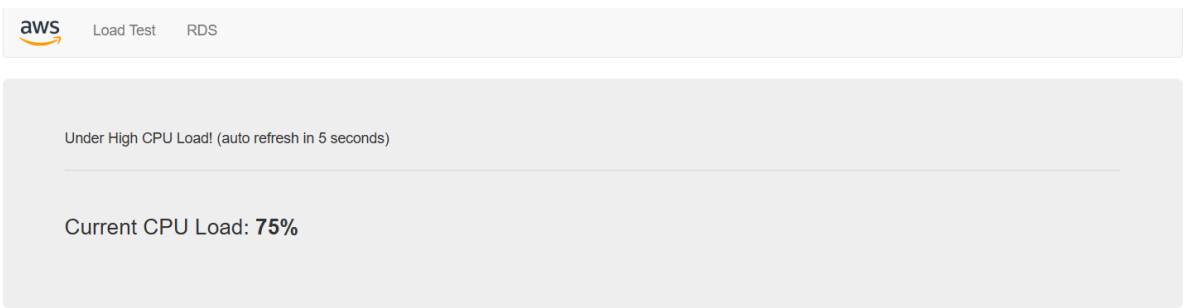
2. I configured the database settings, like instance type and storage.
3. I created the database, and it's deploying now.
4. I'll get an endpoint URL to connect to my database once it's available.

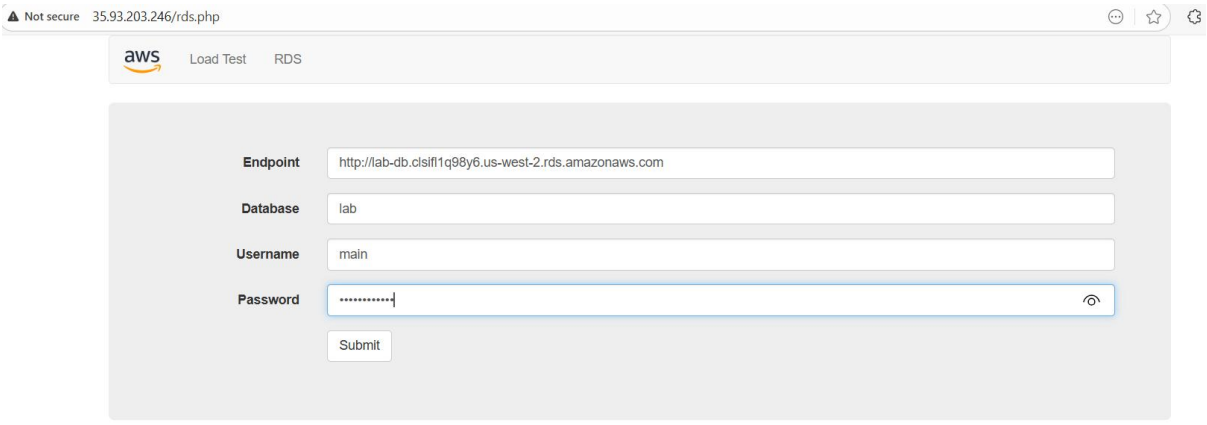
Think of it like setting up a highly available database that's safe and secure!

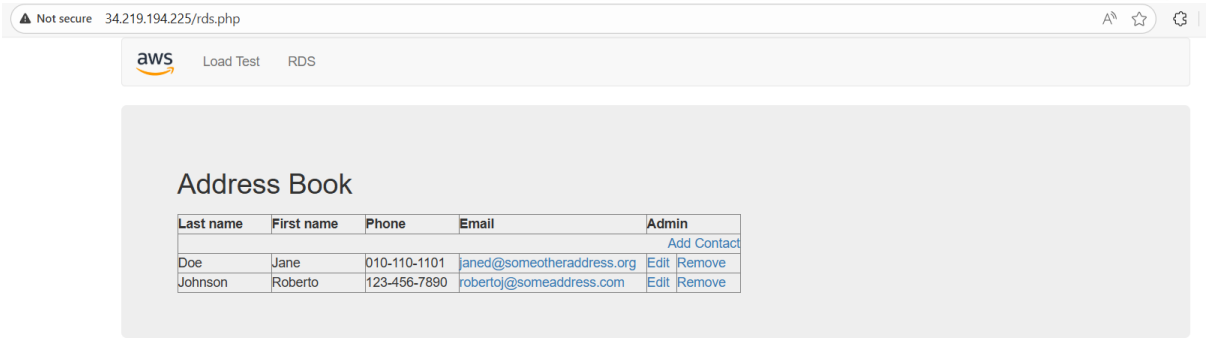
4: Interact with Your Database











I've interacted with my database by opening a web application on my web server and configuring it to use the database.

Here's what I did:

- I copied the WebServer IP address and opened the web application in a new browser tab.
- I clicked the RDS link and configured the application with my database's endpoint, username, and password.
- The application connected to the database and started storing data in it.
- I tested the application by adding, editing, and removing contacts, and the data was saved to the database.
- The database is replicating the data to a second Availability Zone, ensuring it's safe and available.

In simple terms, I've set up a web application that uses my RDS database to store and retrieve data, and it's working as expected!
