

## 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?

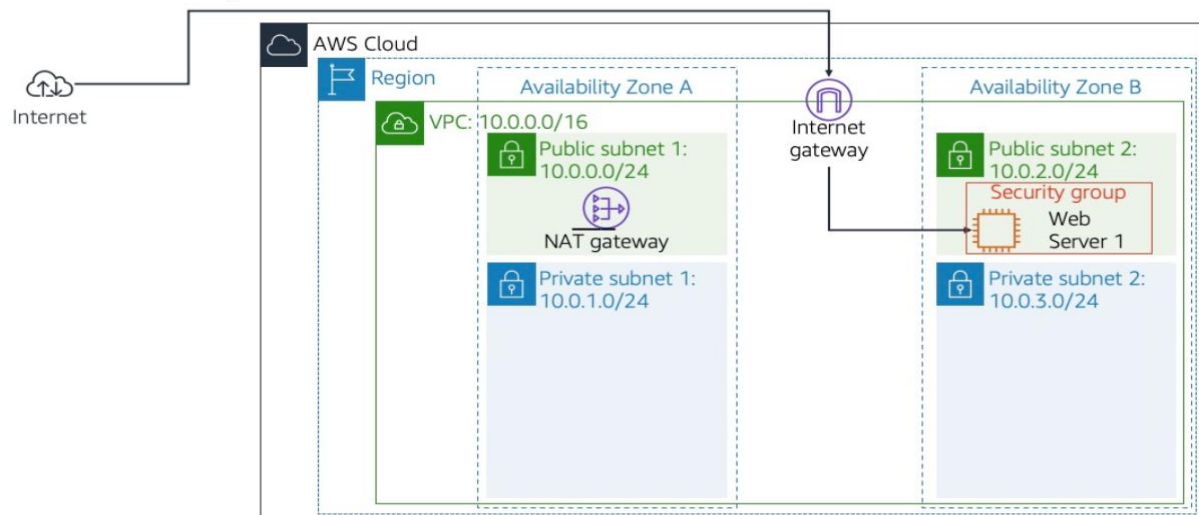
Amazon RDS is a service that makes it easy to set up and manage a relational database in the cloud. It takes care of time-consuming tasks, so I can focus on my applications and business.

What can I do with Amazon RDS?

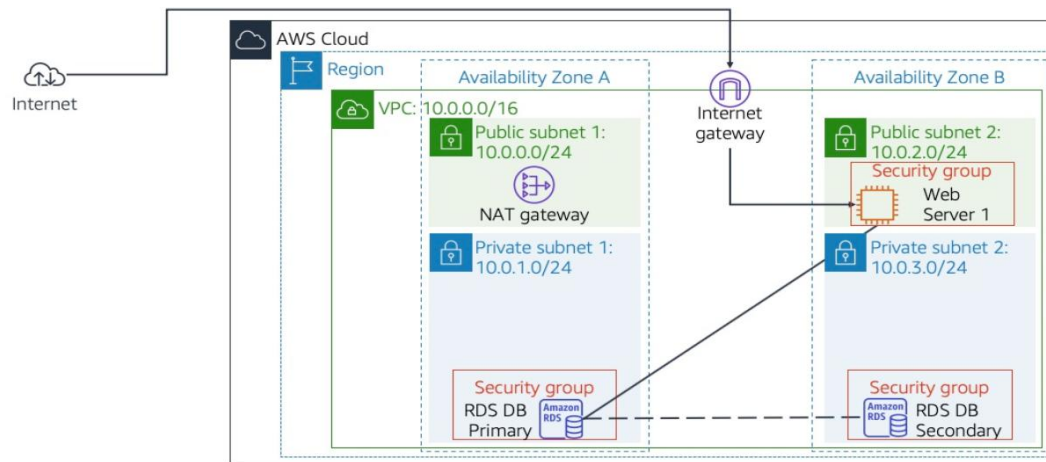
I can choose from six popular database engines, including MySQL, PostgreSQL, and Oracle, and scale my database as my needs grow.

### 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

**1: Create a Security Group for the RDS DB Instance**

The screenshot shows the AWS Management Console VPC dashboard and the Security Groups page.

**VPC dashboard:** The dashboard shows resources by region for Oregon 2. Key resources include:

- VPCs:** 1 VPC (Oregon 2)
- Subnets:** 8 Subnets (Oregon 8)
- Route Tables:** 4 Route Tables (Oregon 4)
- Internet Gateways:** 2 Internet Gateways (Oregon 2)
- Egress-only Internet Gateways:** 0 Egress-only Internet Gateways (Oregon 0)
- NAT Gateways:** 1 NAT Gateway (Oregon 1)
- VPC Peering Connections:** 0 VPC Peering Connections (Oregon 0)
- Network ACLs:** 2 Network ACLs (Oregon 2)
- Security Groups:** 3 Security Groups (Oregon 3)
- Customer Gateways:** 0 Customer Gateways (Oregon 0)

**Security Groups (3):** The Security Groups page shows a list of security groups:

Name	Security group ID	Security group name	VPC ID	Description
-	sg-061db649c268060bd	default	ypc-02f6017a8cbd77f0d	default VPC securit
-	sg-0acfd3608b90eb26e	default	ypc-03261aef566b8000a	default VPC securit
Web Security Group	sg-0686e6c93f29387ae	Web Security Group	ypc-03261aef566b8000a	Enable HTTP acces

Below the table, there is a section titled "Select a security group" with a dropdown menu.

Account ID: 6223-7803-3258

voclabs/user4473058=Mokgadi\_Selepe

United States (Oregon)

Search

[Alt+S]

VPC

Security Groups

Create security group

Create security group

info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name

DB Security Group

Name cannot be edited after creation.

Description

Permit access from Web Security Group

VPC

vpc-03261aef566b8000a (Lab VPC)

Inbound rules

info

This security group has no inbound rules.

Add rule

Outbound rules

info

Type

Protocol

Port range

Destination

Description - optional

Account ID: 6223-7803-3258

voclabs/user4473058=Mokgadi\_Selepe

United States (Oregon)

Search

[Alt+S]

VPC

Security Groups

Create security group

Create security group

info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name

DB Security Group

Name cannot be edited after creation.

Description

Permit access from Web Security Group

VPC

vpc-03261aef566b8000a (Lab VPC)

Inbound rules

info

Type

Protocol

Port range

Source

Description - optional

Add rule

Delete

Outbound rules

info

Type

Protocol

Port range

Source

Description - optional

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Search

[Alt+S]

VPC

Security Groups

sg-0b7eeb3f82b9d10e7 - DB Security Group

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

NAT gateways

Peering connections

Route servers

Security

Network ACLs

Security groups

PrivateLink and Lattice

Getting started

Endpoints

Security group (sg-0b7eeb3f82b9d10e7 | DB Security Group) was created successfully

Details

sg-0b7eeb3f82b9d10e7 - DB Security Group

Actions

Details

Security group name

DB Security Group

Security group ID

sg-0b7eeb3f82b9d10e7

Description

Permit access from Web Security Group

VPC ID

vpc-03261aef566b8000a

Owner

622378033258

Inbound rules count

1 Permission entry

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (1)

Manage tags

Edit inbound rules

Search

Name

Security group rule ID

IP version

Type

Protocol

Port range

-

sg-069488a3714e47ffa

-

MySQL/Aurora

TCP

3306

Here's what happened:

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.

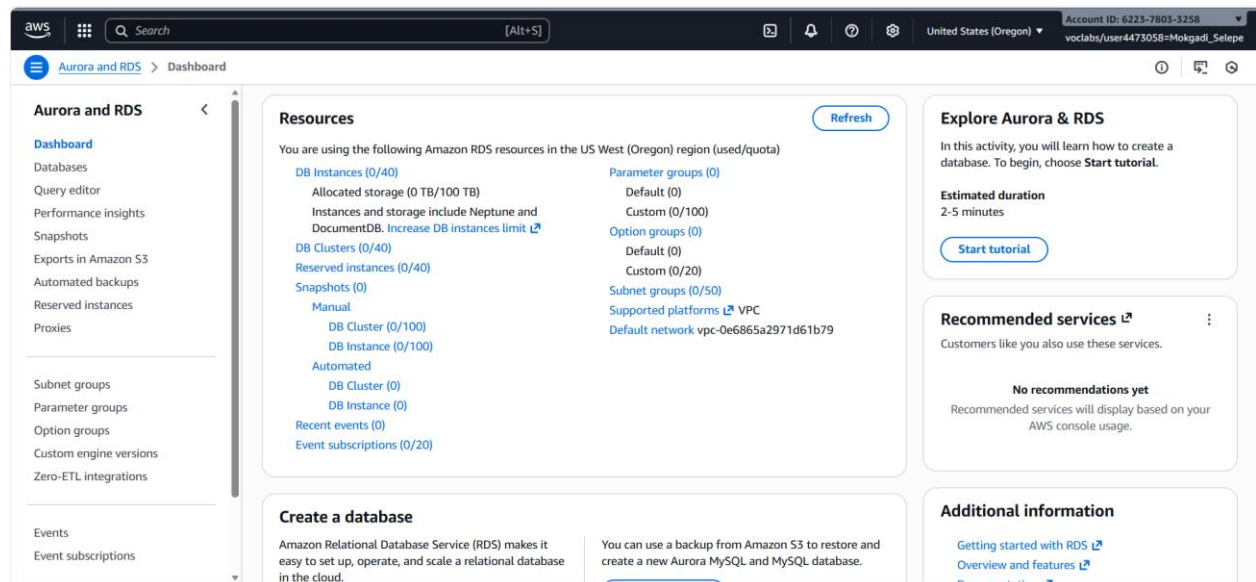
What does this mean?

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



The image shows two screenshots of the AWS Management Console. The top screenshot displays the 'Subnet groups' page under the 'Aurora and RDS' section. It shows a table with 0 subnet groups and a 'Create DB subnet group' button. The bottom screenshot shows the 'Create DB subnet group' wizard. It includes fields for 'Name' (DB Subnet Group), 'Description' (DB Subnet Group), and 'VPC' (Lab VPC). The 'Add subnets' section shows 'Availability Zones' with 'us-west-2a' and 'us-west-2b' selected.

**AWS Management Console - Subnet groups**

Account ID: 6223-7803-3258  
voclabs/user4473058=Mokgadi\_Selepe

**Subnet groups (0)**

Filter by subnet group

Name	Description	Status	VPC
No db subnet groups You don't have any db subnet groups. <a href="#">Create DB subnet group</a>			

**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-0b0939540dcdfb6bb)  
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

The first screenshot shows the 'Create DB subnet group' wizard in the AWS Management Console. Under 'Availability Zones', two zones are selected: 'us-west-2a' and 'us-west-2b'. Under 'Subnets', two subnets are selected: 'Private Subnet 1' (Subnet ID: subnet-0972542231ea93156, CIDR: 10.0.1.0/24) and 'Public Subnet 2' (Subnet ID: subnet-0e181397d290f091f, CIDR: 10.0.2.0/24). A note states: 'For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.' Below, a table lists the selected subnets:

Availability zone	Subnet name	Subnet ID	CIDR block
us-west-2a	Private Subnet 1	subnet-0972542231ea93156	10.0.1.0/24
us-west-2b	Public Subnet 2	subnet-0e181397d290f091f	10.0.2.0/24

The second screenshot shows the 'Subnet groups' page after successful creation. A green banner at the top says 'Successfully created DB Subnet Group. View subnet group'. The 'Subnet groups (1)' table shows one group:

Name	Description	Status	VPC
db subnet group	DB Subnet Group	Complete	vpc-0b0939540dcfb6bb

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

**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
- ☐ Microsoft SQL Server
- ☐ IBM Db2

**Suggested add-ons for lab-db**

Simplify the configuration of the following suggested add-ons by using settings from your new database.

**Create an ElastiCache cluster from RDS using your DB settings**

You can save up to 55% in cost and gain up to 80x faster read performance using ElastiCache with RDS for MySQL (vs. RDS for MySQL alone).

[Learn more](#)

[Create ElastiCache cluster](#)

**Use RDS Proxy**

Using a proxy allows your applications to pool and share database connections to help them scale. A proxy simplifies connection management and makes applications more resilient to database failures.

[Learn more](#)

[Create proxy](#)

**You can hide these suggestions so they don't appear after database creation. All these actions can be taken from the database list page or database details page.**

☐ Hide add-ons for 30 days [Close modal](#)

**Creating database lab-db**

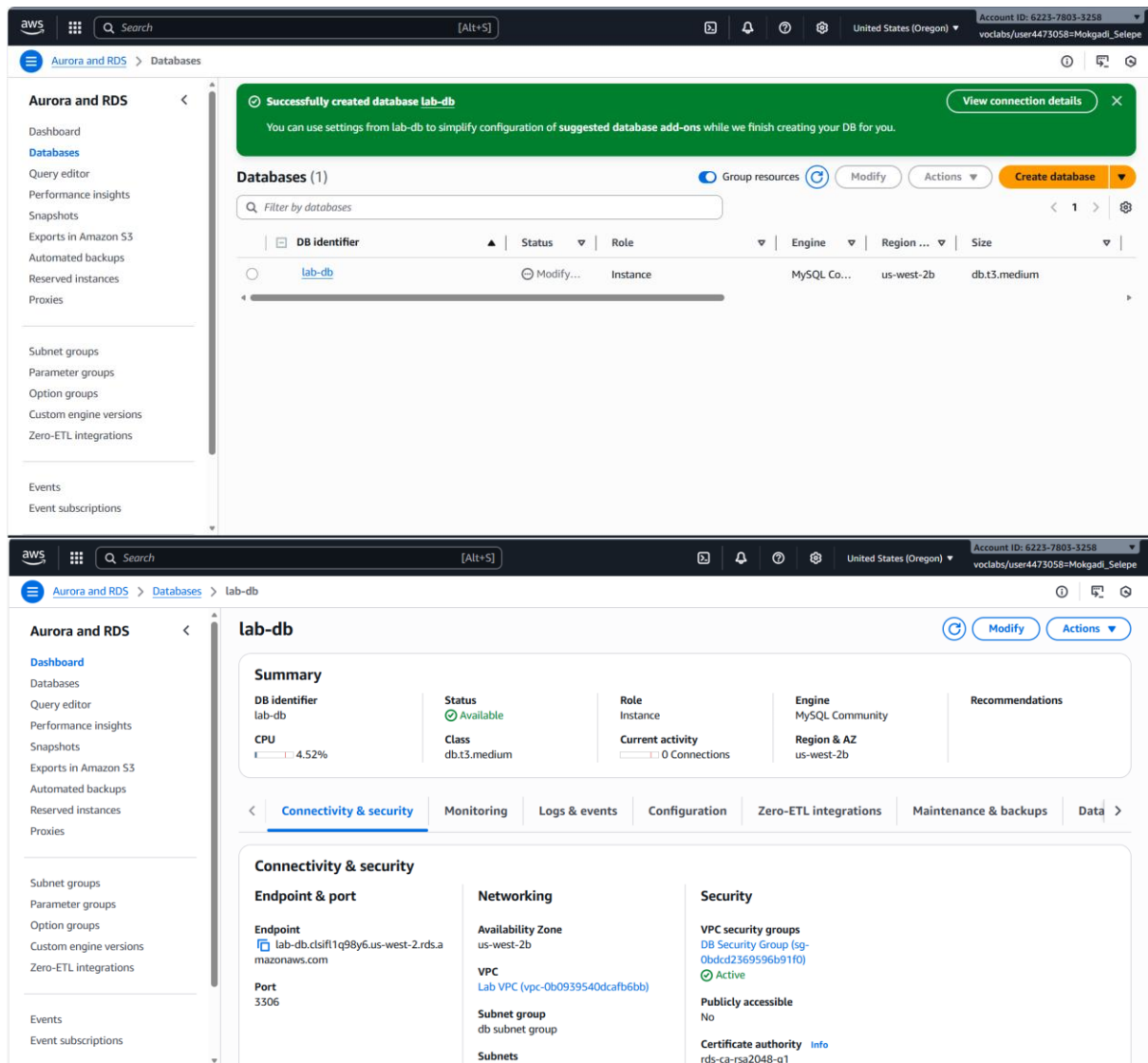
Your database might take a few minutes to launch. You can use settings from lab-db to simplify configuration of suggested database add-ons while we finish creating your DB for you.

[View connection details](#)

**Databases (1)**

☒ Group resources [Modify](#) [Actions](#) [Create database](#)

<input type="checkbox"/>	DB identifier	Status	Role	Engine	Region ...	Size
<input type="radio"/>	lab-db	Creating	Instance	MySQL Co...	us-west-2b	db.t3.medium



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

Used \$0.3 of \$10

02:57

Start Lab

End Lab

AWS Details

Details

Reset

Submit

Submission Report

Grades

While you are waiting, you might want to review the [Amazon RDS FAQs](#) or grab a cup of coffee.

33. Wait until the **Status** changes to **Modifying** or **Available**.

34. Scroll down to the **Connectivity & Security** section and copy the **Endpoint** field.

It will look similar to: `lab-db.cgqg8lhnvmv.us-west-2.rds.amazonaws.com`

35. Paste the Endpoint value into a text editor. You will use it later in the lab.

### Task 4: Interact with Your Database

In this task, you will open a web application running on your web server and configure it to use the database.

36. Copy the **WebServer IP address** by selecting **i AWS Details** above these instructions you are currently reading.

37. Open a new web browser tab, paste the **WebServer IP address** and press Enter.

The web application will be displayed, showing information about the EC2 instance.

38. At the top of the web application page, click the **RDS** link.

aws

Load Test

RDS

Meta-Data	Value
InstanceId	i-07064a6fa36f93e61
Availability Zone	us-west-2b

Current CPU Load: 0%

aws

Load Test

RDS

Under High CPU Load! (auto refresh in 5 seconds)

Current CPU Load: 75%

Cloud Access

AWS CLI: Show

Cloud Labs

Remaining session time: 02:56:19(177 minutes)

Session started at: 2025-11-09T04:54:35-0800

Session to end at: 2025-11-09T10:18:01-0800

Accumulated lab time: 05:27:00 (327 minutes)

ips -- public:35.93.203.246, private:10.0.2.61

SSH key: Show Download PEM Download PPK

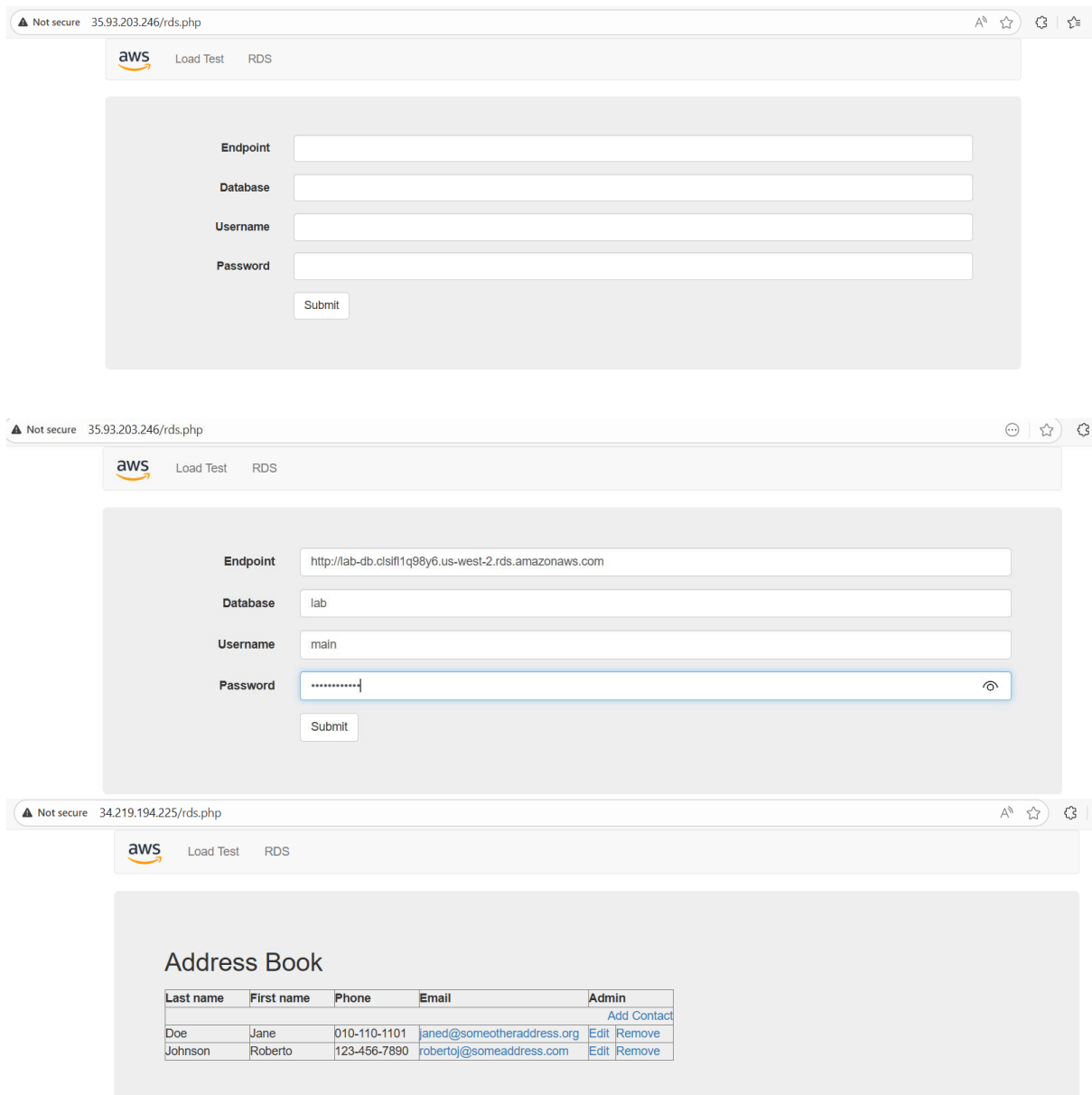
AWS SSO: Download URL

awsstudentAccessSecretKey Access Key: AKIAZB2E7EBVNPWWY

Access Key: 0fyXWC/TurjgqOrjwzksZG+haploIV

WebServer 35.93.203.246

9



Not secure 35.93.203.246/rds.php

aws Load Test RDS

Endpoint

Database

Username

Password

Submit

Not secure 35.93.203.246/rds.php

aws Load Test RDS

Endpoint

Database

Username

Password

Submit

Not secure 34.219.194.225/rds.php

aws Load Test RDS

### Address Book

Last name	First name	Phone	Email	Admin
Doe	Jane	010-110-1101	<a href="#">janed@someotheraddress.org</a>	<a href="#">Add Contact</a>
Johnson	Roberto	123-456-7890	<a href="#">roberto@someaddress.com</a>	<a href="#">Edit</a> <a href="#">Remove</a>

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!

---