COS20019

Cloud Computing Architechture

Week 5 – ACF Lab 5:
RDS lab (Build a Database Server)

Truong Ngoc Gia Hieu
105565520

Lab 5: Build Your DB Server and Interact With Your DB Using an App

A. Lab Overview and objectives

This lab is designed to reinforce the concept of leveraging an AWS-managed database instance for solving relational database needs.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

By the end of this lab, you will be able to:

- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

Duration

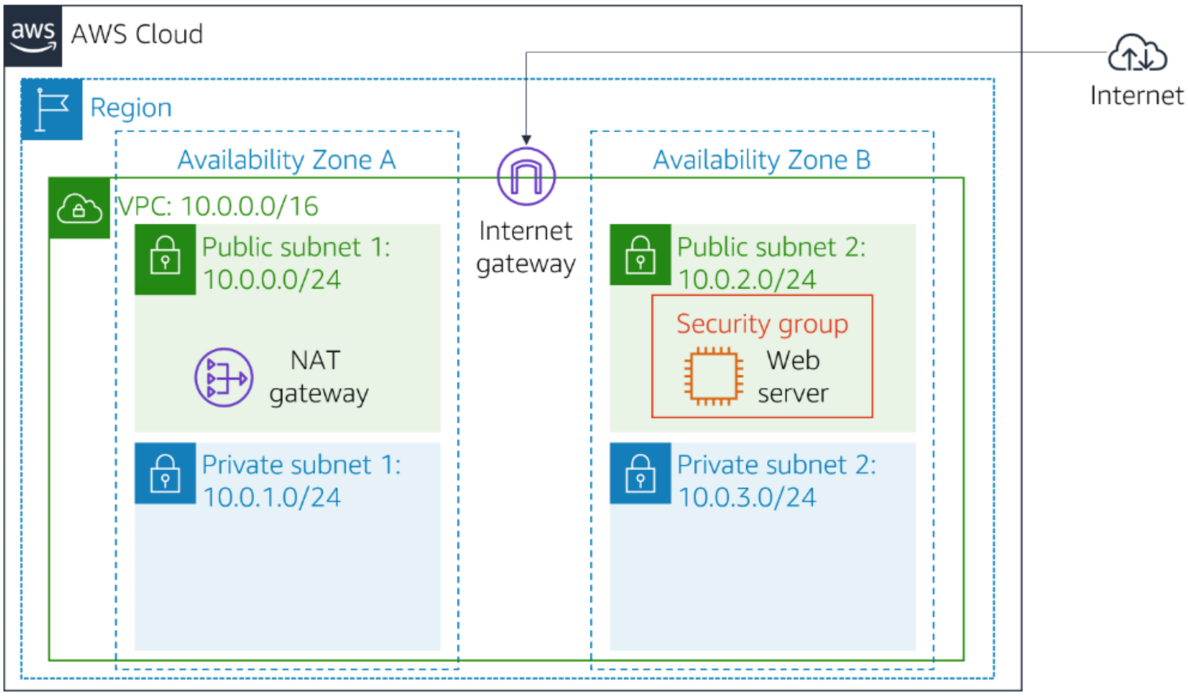
This lab takes approximately **30 minutes.**

AWS service restrictions

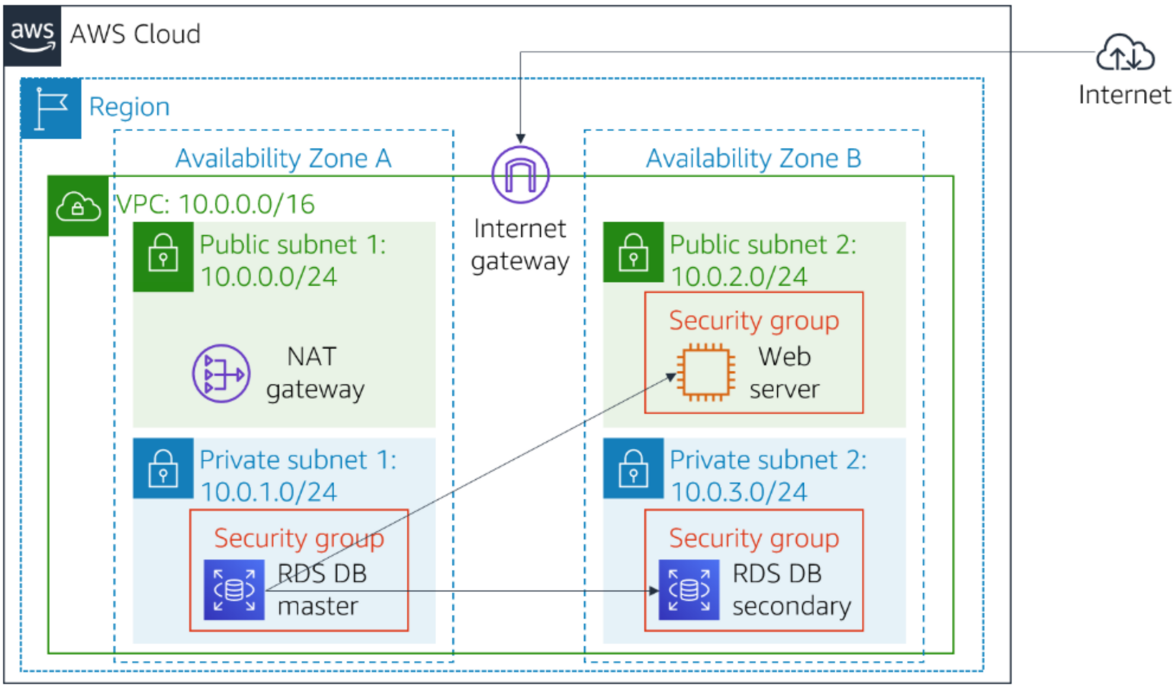
In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Scenario

When you start the lab, the following infrastructure is provided:



By the end of the lab, you will have this infrastructure:



B. Accessing the AWS Management Console

To access the AWS Management Console, clicking the ▶ Start Lab button and wait until the circle next to the AWS changes from yellow to green

Figure 1: AWS Management Console activated

Getting Credit for your work

At the end of this lab you will be instructed to submit the lab to receive a score based on your progress.

- **Tip**: The script that checks you works may only award points if you name resources and set configurations as specified. In particular, values in these instructions that appear in This Format should be entered exactly as documented (case-sensitive).

C. Task 1: Create a Security Group for the RDS DB Instance

In this task, you will create a security group to allow your web server to access your RDS DB instance. The security group will be used when you launch the database instance.

Step 1.1: Open the AWS Management Console Homepage, search and select the VPC

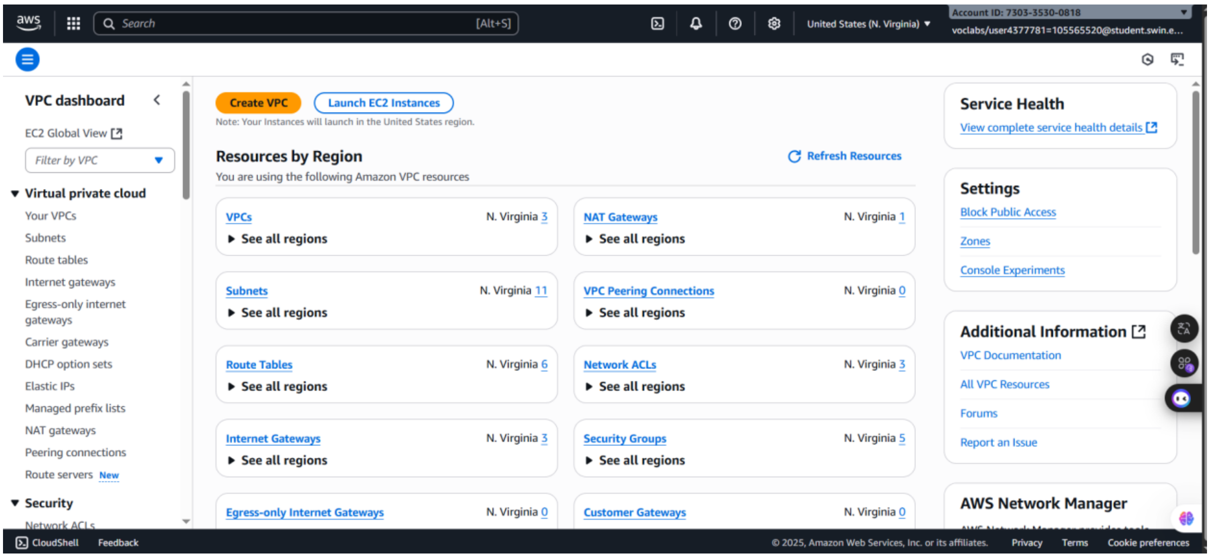
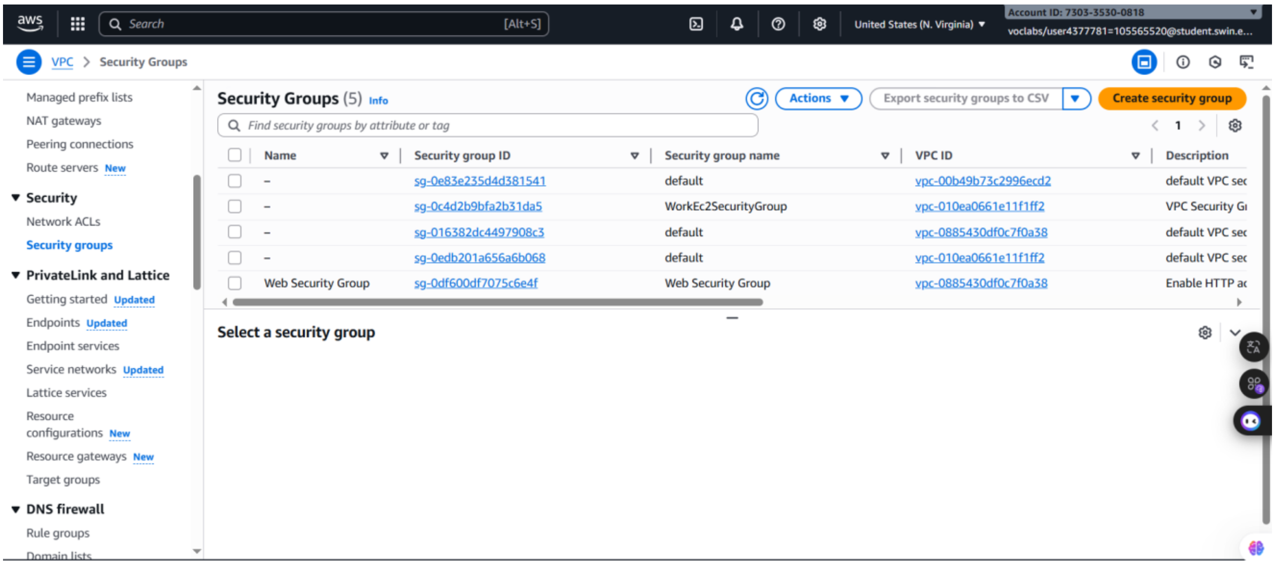


Figure 2: VPC Homepage

Step 1.2: Choose the **Security Groups** in the left naviagtion pane in the VPC



Security Groups (5) Info

Name	Security group ID	Security group name	VPC ID	Description
–	sg-0e83e235d4d381541	default	vpc-00b49b73c2996ecd2	default VPC sec
–	sg-0c4d2b9bfa2b31da5	WorkEc2SecurityGroup	vpc-010ea0661e11f1ff2	VPC Security Gr
–	sg-016382dc4497908c3	default	vpc-0885430df0c7f0a38	default VPC sec
–	sg-0edb201a656a6b068	default	vpc-010ea0661e11f1ff2	default VPC sec
Web Security Group	sg-0df600df7075c6e4f	Web Security Group	vpc-0885430df0c7f0a38	Enable HTTP ac

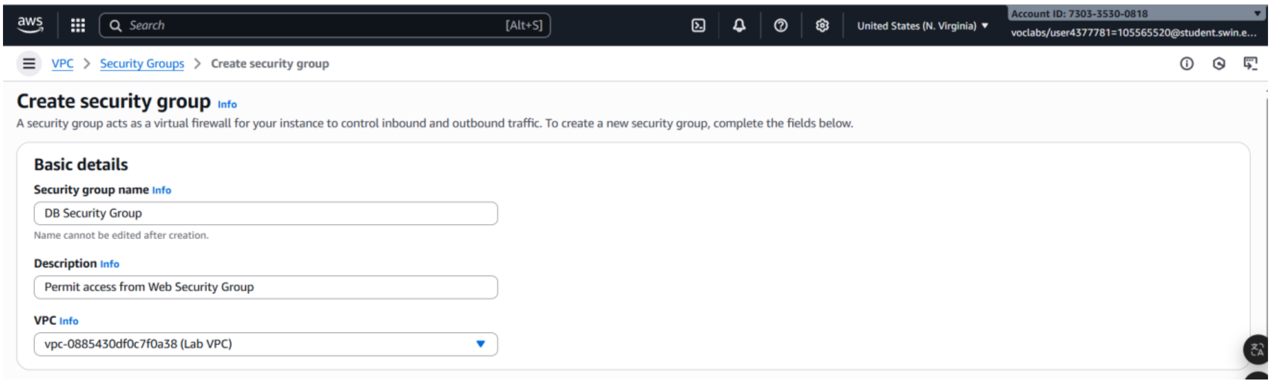
Homepage

Figure 3: VPC Security Groups

Step 1.3: After **Create security group** choosing the , configure the following settings:

- ○ **Security group name:** DB Security Group
- ○ **Description:** Permit access from Web Security Group
- ○ **VPC:** *Lab VPC*

  Tip: Choose the X next to VPC that is already selected, then choose **Lab VPC** from the menu.



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 Info: DB Security Group

Name cannot be edited after creation.

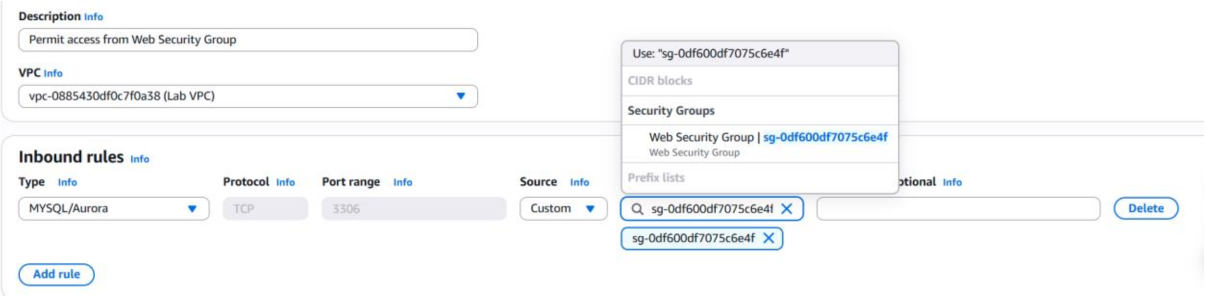
Description Info: Permit access from Web Security Group

VPC Info: vpc-0885430df0c7f0a38 (Lab VPC)

Figure 4: Basic details Configuration

Step 1.4: Then, choosing the in the Inbound rules with following configurations:

- ○ **Type**: MySQL/Aurora (3306)
- ○ **Source**: Place you cursor in the field to the right of Custom, type sg, and then select Web Security Group.



Description Info

Permit access from Web Security Group

VPC Info

vpc-0885430df0c7f0a38 (Lab VPC)

Inbound rules Info

Type	Protocol	Port range	Source		Description - optional	
MYSQL/Aurora	TCP	3306	Custom	sg-0df600df7075c6e4f		Delete

Use: "sg-0df600df7075c6e4f"

CIDR blocks

Security Groups

Web Security Group | sg-0df600df7075c6e4f
Web Security Group

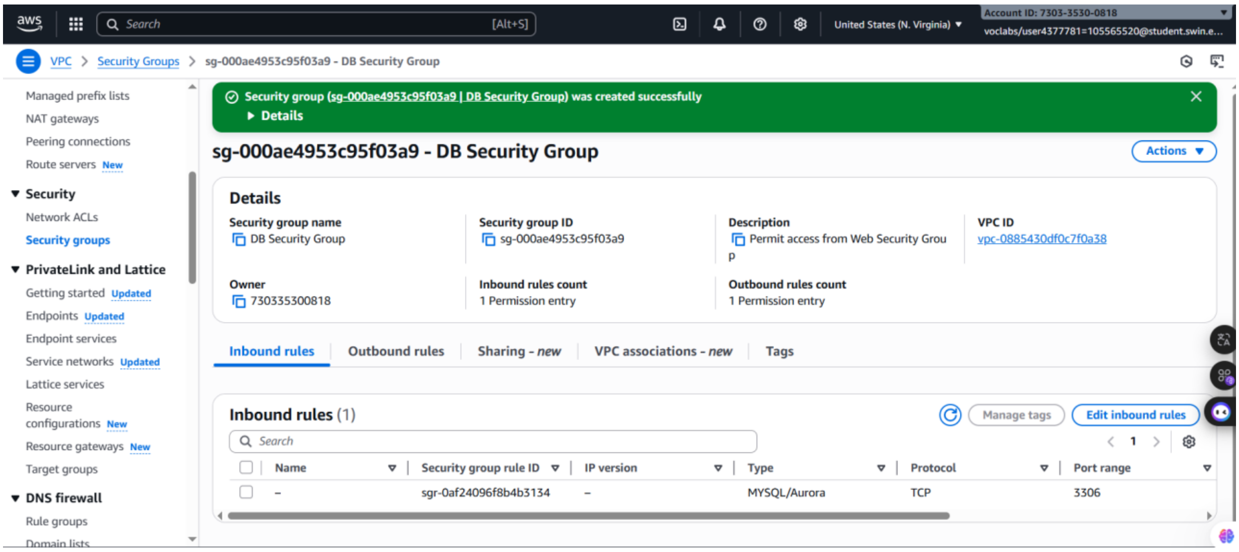
Prefix lists

sg-0df600df7075c6e4f

Add rule

Figure 5: Inbound Configuration

Step 1.5: After configure these settings, clicking the **Create security group** button on the bottom of the page



VPC > Security Groups > sg-000ae4953c95f03a9 - DB Security Group

Security group (sg-000ae4953c95f03a9 | DB Security Group) was created successfully

▸ Details

sg-000ae4953c95f03a9 - DB Security Group

Details

Security group name	Security group ID	Description	VPC ID
DB Security Group	sg-000ae4953c95f03a9	Permit access from Web Security Group	vpc-0885430df0c7f0a38
Owner	**Inbound rules count**	**Outbound rules count**	
730335300818	1 Permission entry	1 Permission entry	

Inbound rules (1)

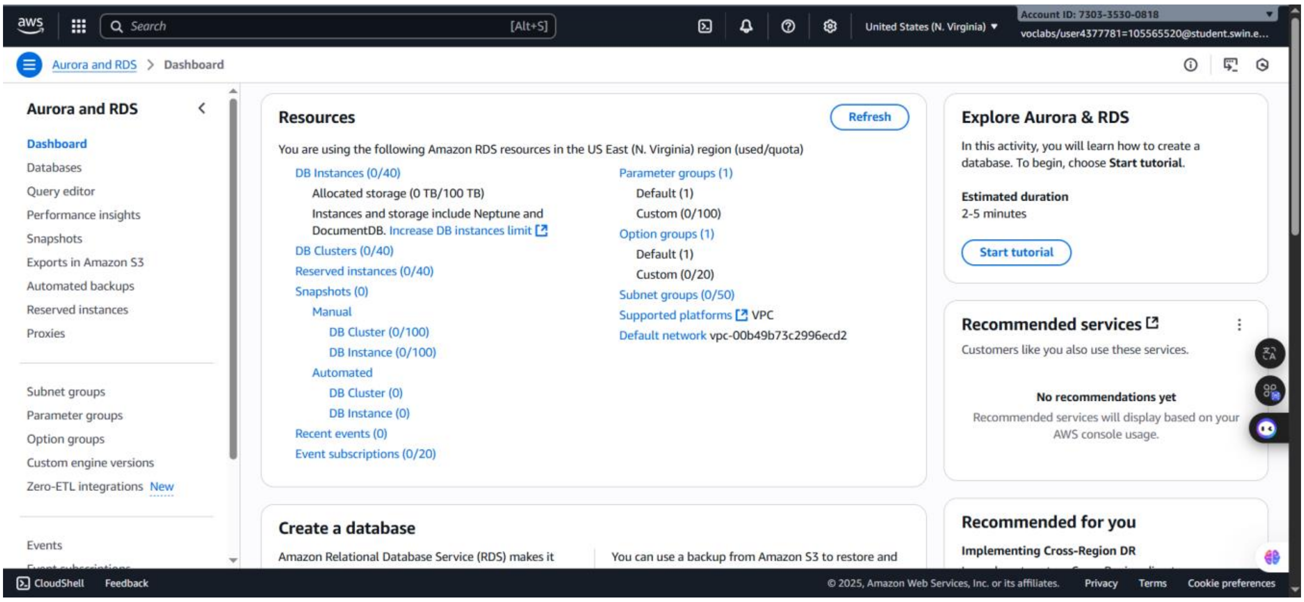
Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-0af24096f8b4b3134	–	MYSQL/Aurora	TCP	3306

Figure 6: Security group created succeassfully

D. Task 2: Create a DB Subnet Group

In this task, you will create a *DB subnet group* that is used to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

Step 2.1: In the search box of VPC page next to the service icon, search and choose the **Aurora and RDS**



Resources

Refresh

You are using the following Amazon RDS resources in the US East (N. Virginia) region (used/quota)

DB Instances (0/40)
Allocated storage (0 TB/100 TB)
Instances and storage include Neptune and DocumentDB. Increase DB instances limit
DB Clusters (0/40)
Reserved instances (0/40)
Snapshots (0)
Manual
DB Cluster (0/100)
DB Instance (0/100)
Automated
DB Cluster (0)
DB Instance (0)
Recent events (0)
Event subscriptions (0/20)

Parameter groups (1)
Default (1)
Custom (0/100)
Option groups (1)
Default (1)
Custom (0/20)
Subnet groups (0/50)
Supported platforms VPC
Default network vpc-00b49b73c2996ecd2

Explore Aurora & RDS

In this activity, you will learn how to create a database. To begin, choose **Start tutorial**.

Estimated duration
2-5 minutes

Start tutorial

Recommended services

Customers like you also use these services.

No recommendations yet

Recommended services will display based on your AWS console usage.

Create a database

Amazon Relational Database Service (RDS) makes it

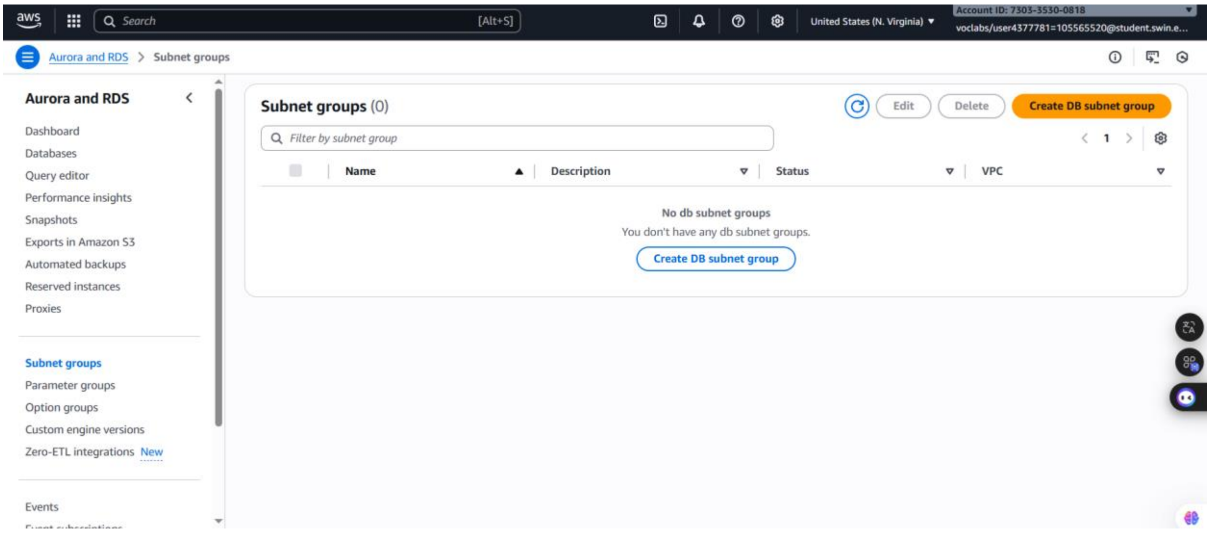
You can use a backup from Amazon S3 to restore and

Recommended for you

Implementing Cross-Region DR

Figure 7: Aurora and RDS Homepage

Step 2.2: Choose the Subnets groups in the left navigation



Subnet groups (0)

Edit | Delete | Create DB subnet group

Filter by subnet group

	Name	Description	Status	VPC

No db subnet groups

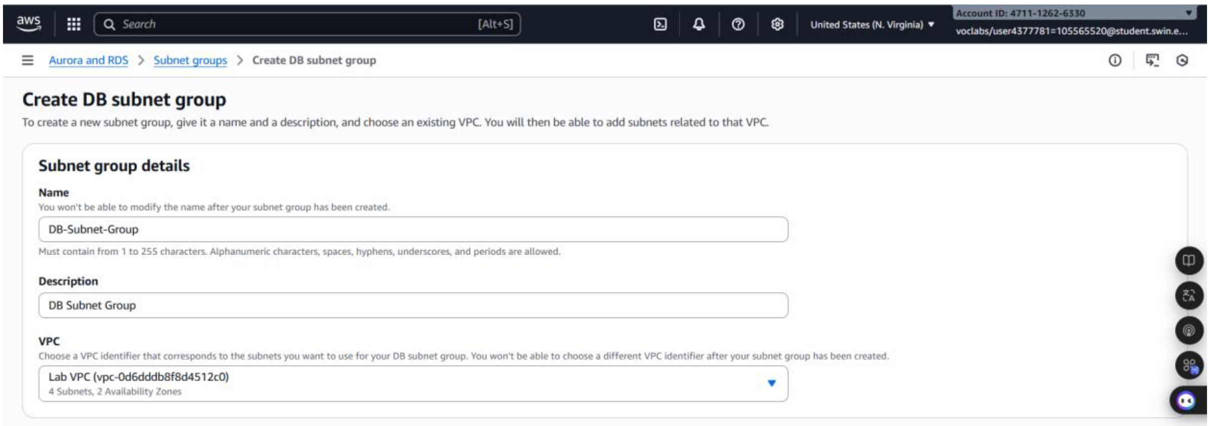
You don't have any db subnet groups.

Create DB subnet group

Figure 8: Aurora and RDS Homepage

Step 2.3: Clicking the and create a new DB group with below settings:

- **Name:** DB-Subnet-Group
- **Description:** DB Subnet Group
- **VPC:** *Lab VPC*



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.

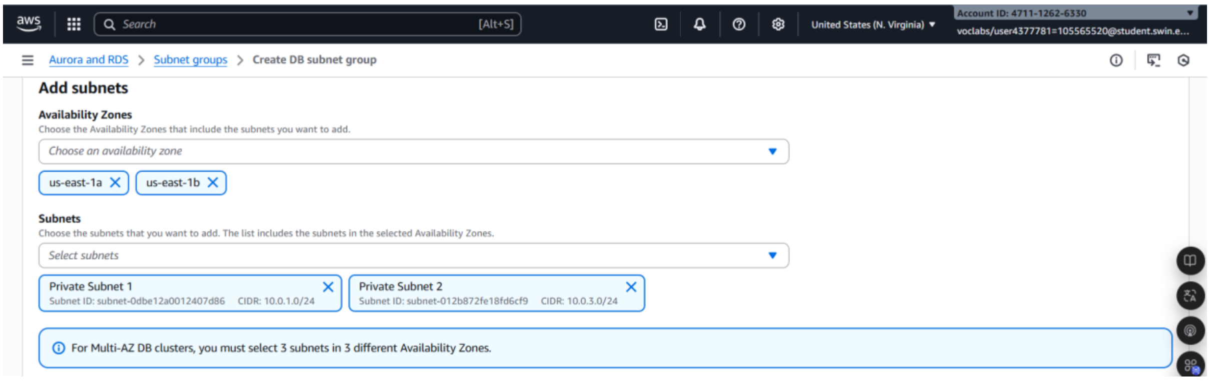
Description
DB Subnet Group

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-0d6dddb8f8d4512c0)
4 Subnets, 2 Availability Zones

Figure 9: Subnet Group Details Configuration

Step 2.4: Then, srolling down to the Add Subnets section and edit it with following configurations:

- **Availability Zones**: **us-east-1a** and **us-east-1b**
- **Subnets: 10.0.1.0/24 and 10.0.3.0/24.**



Aurora and RDS > Subnet groups > Create DB subnet group

Add subnets

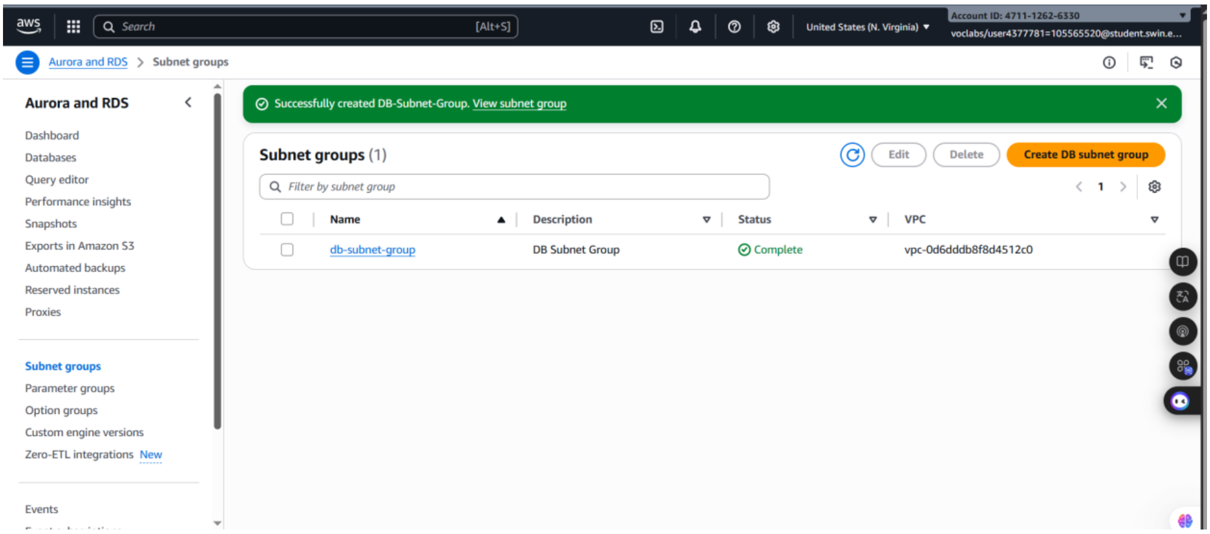
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a ✕ us-east-1b ✕

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
Private Subnet 1 — Subnet ID: subnet-0dbe12a0012407d86 CIDR: 10.0.1.0/24 ✕
Private Subnet 2 — Subnet ID: subnet-012b872fe18fd6cf9 CIDR: 10.0.3.0/24 ✕

ⓘ For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Figure 10: Add subnets Configuration

Step 2.5: After rewatching and confirming the configuration is correct, select **Create**



Aurora and RDS > Subnet groups

✓ Successfully created DB-Subnet-Group. View subnet group

Subnet groups (1)

Name	Description	Status	VPC
db-subnet-group	DB Subnet Group	✓ Complete	vpc-0d6dddb8f8d4512c0

Figure 11: DB subnet group created successfully

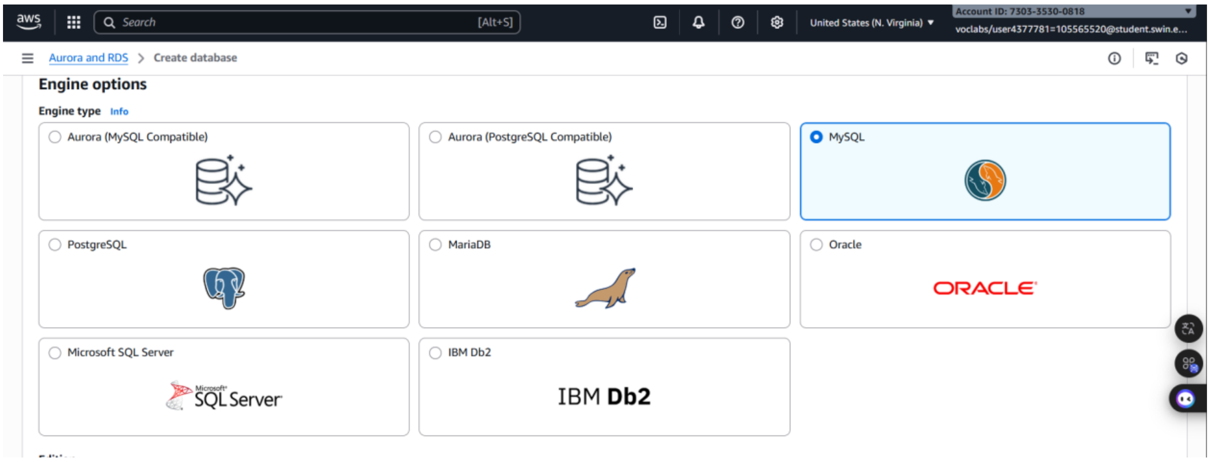
E. Task 3: Create an Amazon RDS DB Instance

In this task, you will configure and launch a Multi-AZ Amazon RDS deployment of a MySQL database instance.

Amazon RDS ***Multi-AZ*** deployments provide enhanced availability and durability for Database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (A:Z).

Step 3.1: Next, on the left navigation pannel, choose **Databases** and then click the button to **Create database** create database. After that, configure these settings:

- **Engine type**: MySQL



aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 7303-3530-0818
voclabs/user4377781=105565520@student.swin.e...
Aurora and RDS > Create database

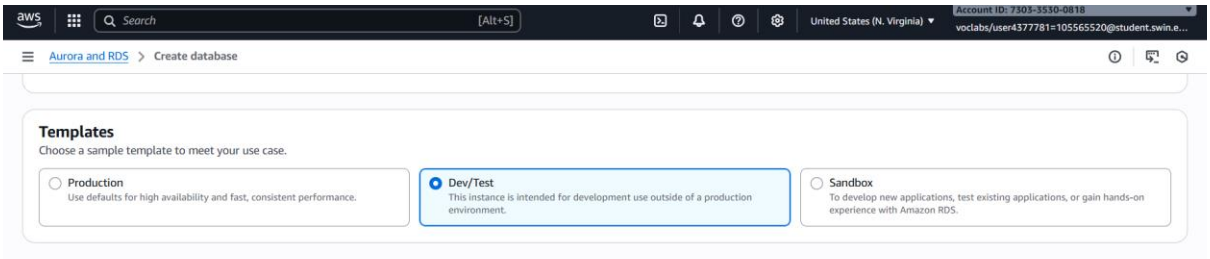
Engine options

Engine type Info

○ Aurora (MySQL Compatible)
○ Aurora (PostgreSQL Compatible)
● MySQL
○ PostgreSQL
○ MariaDB
○ Oracle
○ Microsoft SQL Server
○ IBM Db2

Figure 12: Engine configuration

- **Templates**: Dev/Test



aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 7303-3530-0818
voclabs/user4377781=105565520@student.swin.e...
Aurora and RDS > Create database

Templates

Choose a sample template to meet your use case.

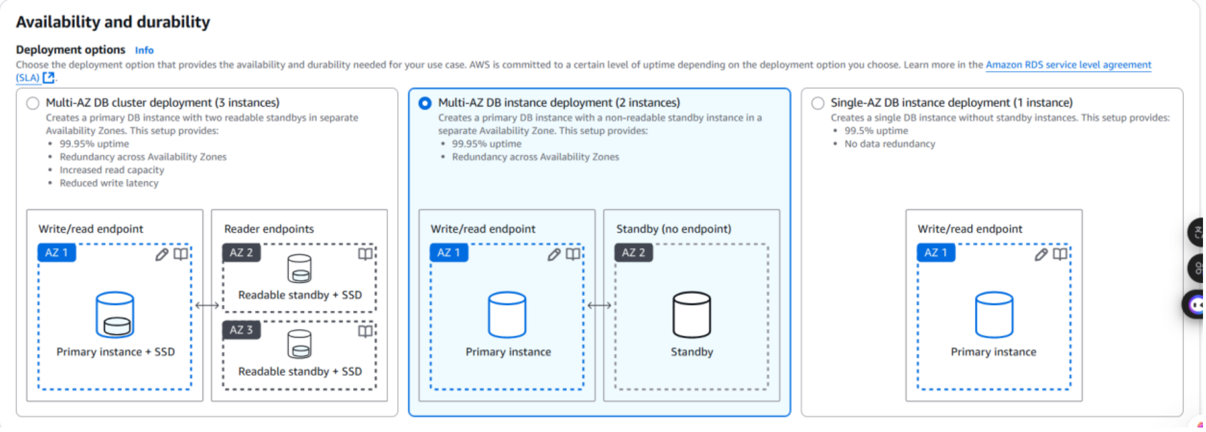
○ Production
Use defaults for high availability and fast, consistent performance.

● Dev/Test
This instance is intended for development use outside of a production environment.

○ Sandbox
To develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

Figure 13: Templates configuration

- **Availability and durability**: Multi-AZ DB instance.



Availability and durability

Deployment options Info

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the Amazon RDS service level agreement (SLA).

○ Multi-AZ DB cluster deployment (3 instances)
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

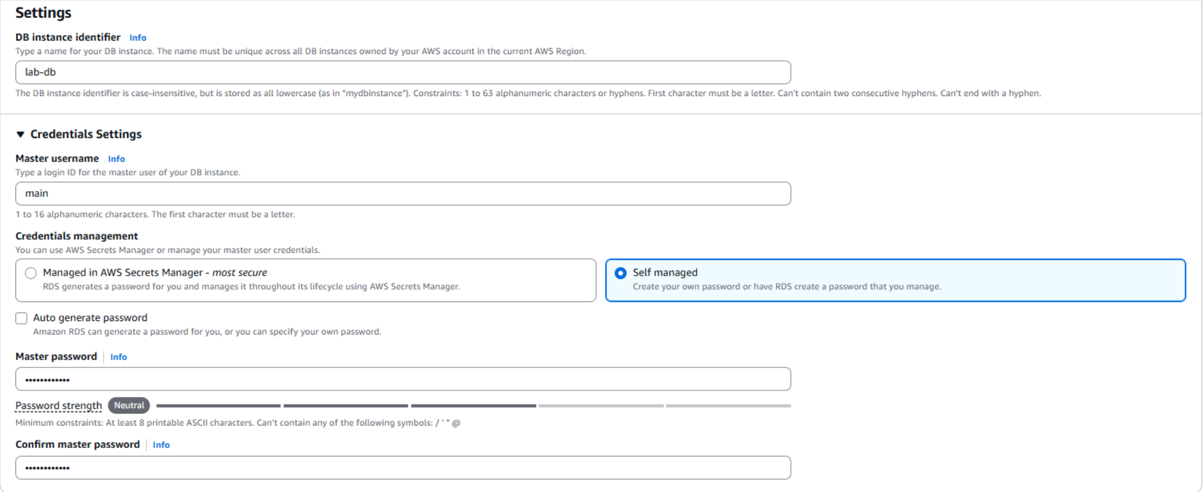
● Multi-AZ DB instance deployment (2 instances)
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones

○ Single-AZ DB instance deployment (1 instance)
Creates a single DB instance without standby instances. This setup provides:
- 99.5% uptime
- No data redundancy

Figure 13: Availability and durability configuration

- **Under Settings**, configure:
  - **DB instance identifier:** lab-db
  - **Master username:** main

  ****Note***: For creating password, choose Self Managed
  - **Master password:** lab-password
  - **Confirm password:** lab-password



Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

lab-db

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username Info

Type a login ID for the master user of your DB instance.

main

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

○ Managed in AWS Secrets Manager - *most secure*
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

● Self managed
Create your own password or have RDS create a password that you manage.

□ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password | Info

••••••••••••

Password strength Neutral

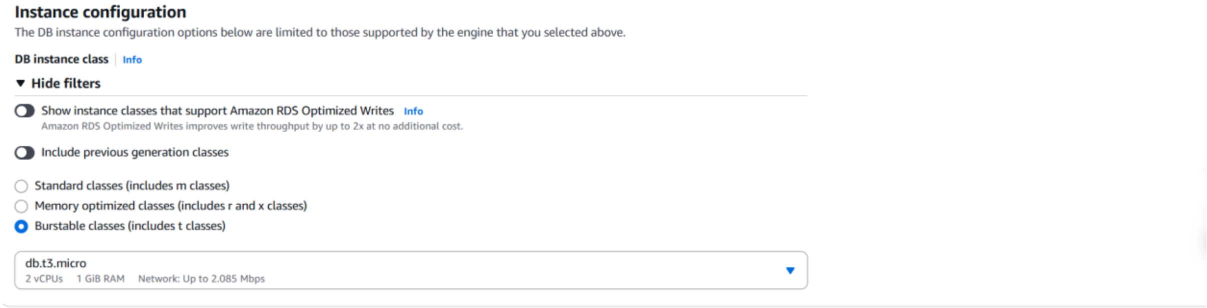
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password | Info

••••••••••••

Figure 14: Settings configuration

- Under **DB instance class**
  - Select DB instance class: Burstable classes (includes t classes).
  - Select *db.t3.micro*



Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class Info

▼ Hide filters

Show instance classes that support Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

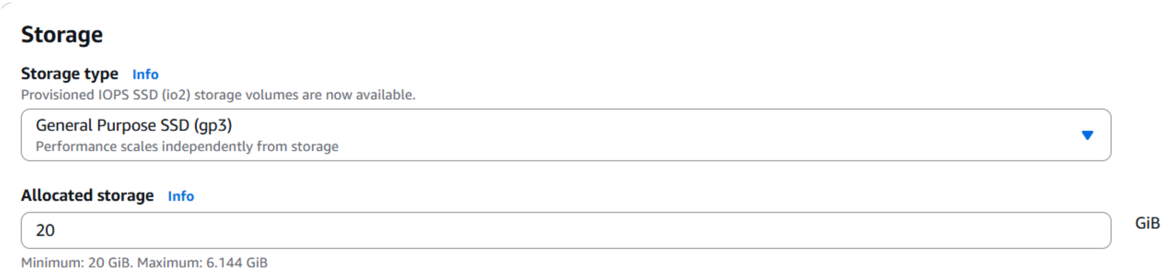
Include previous generation classes

○ Standard classes (includes m classes)
○ Memory optimized classes (includes r and x classes)
● Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM Network: Up to 2.085 Mbps

Figure 15: DB instance class configuration

- Under **Storage**, configure:
  - **Storage type**: General Purpose (SSD)
  - **Allocated storage**: 20



Storage

Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp3)
Performance scales independently from storage

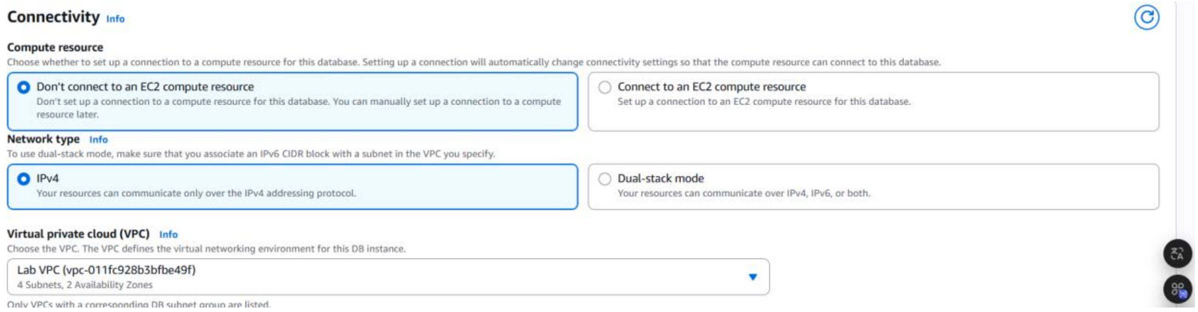
Allocated storage Info

20 GiB

Minimum: 20 GiB. Maximum: 6.144 GiB

Figure 16: Storage configuration

- Under Connectivity, configure: **Virtual Private Cloud (VPC) - Lab VPC**



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.

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.

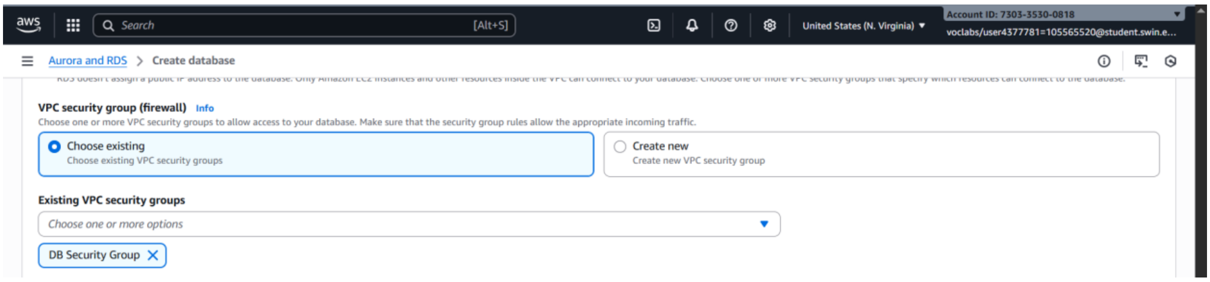
Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Lab VPC (vpc-011fc928b3bfbe49f)
4 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

Figure 17: Connectivity configuration

- Under **Existing VPC security groups**, from the dropdown list:
  - o Choose DB Security Group.
  - o Deselect default.



aws
Search [Alt+S]
United States (N. Virginia) ▾
Account ID: 7303-3530-0818
voclabs/user4377781=105565520@student.swin.e...
Aurora and RDS > Create database

VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

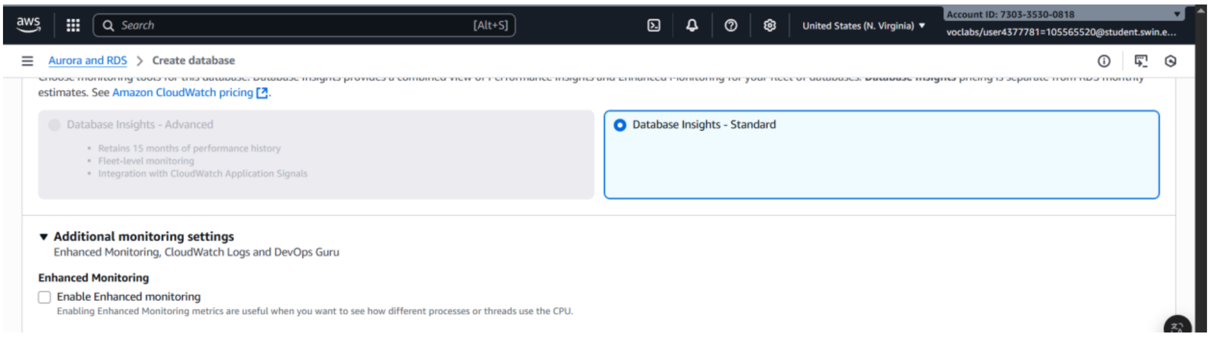
Existing VPC security groups

Choose one or more options

DB Security Group ✕

Figure 17: VPC configuration

- Next, click the **Aditional configuration** and deselect **Enable Enhanced monitoring**.


aws
Search [Alt+S]
United States (N. Virginia) ▾
Account ID: 7303-3530-0818
voclabs/user4377781=105565520@student.swin.e...
Aurora and RDS > Create database

Choose monitoring tools for this database. Database Insights provides a combined view of Performance Insights and Enhanced Monitoring for your fleet of databases. Database Insights pricing is separate from RDS monthly estimates. See Amazon CloudWatch pricing.

○ Database Insights - Advanced
- Retains 15 months of performance history
- Fleet-level monitoring
- Integration with CloudWatch Application Signals

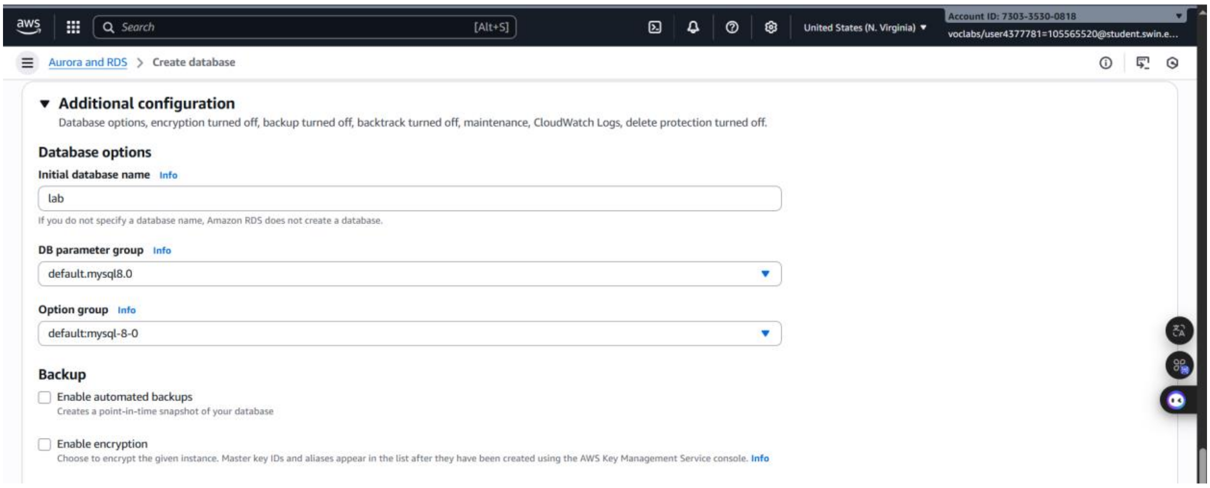
● Database Insights - Standard

▼ Additional monitoring settings
Enhanced Monitoring, CloudWatch Logs and DevOps Guru

Enhanced Monitoring
□ Enable Enhanced monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Figure 18: Uncheck the **Enable Enhanced monitoring**

- Next, expand the Additional configuration:
  - **Initial database name**: lab
  - Uncheck **Enable automatic backups.**
  - Uncheck **Enable encryption**


aws
Search [Alt+S]
United States (N. Virginia) ▾
Account ID: 7303-3530-0818
voclabs/user4377781=105565520@student.swin.e...
Aurora and RDS > Create database

▼ Additional configuration
Database options, encryption turned off, backup turned off, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options
Initial database name Info
lab
If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group Info
default.mysql8.0

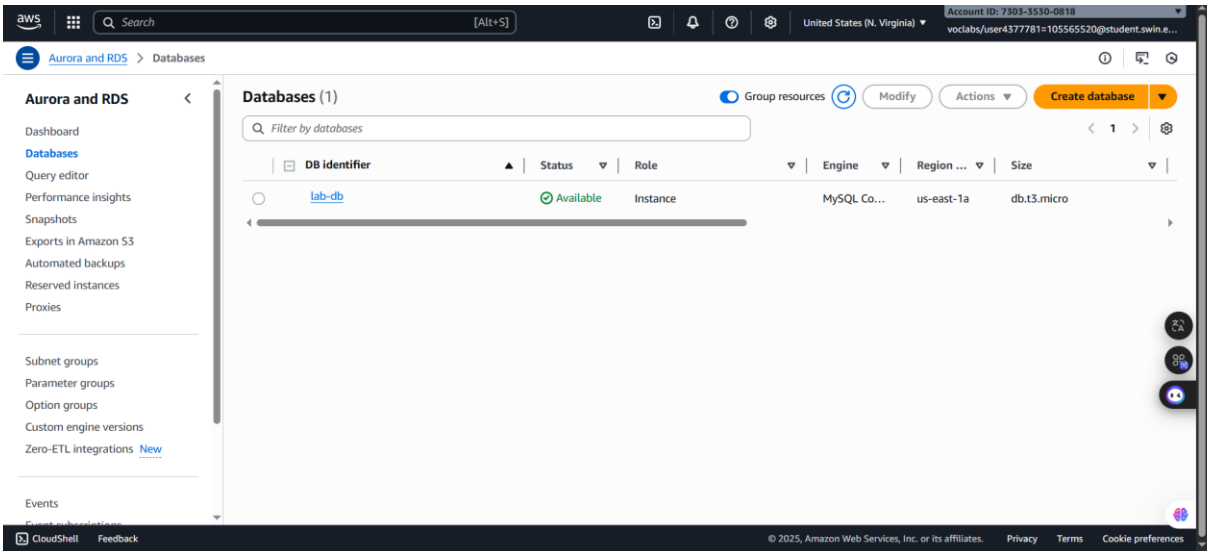
Option group Info
default:mysql-8-0

Backup
□ Enable automated backups
Creates a point-in-time snapshot of your database

□ Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. Info

Figure 19: Additional Configuration

- After clicking the Create database button , wait approximately 4 minutes to launch the database available



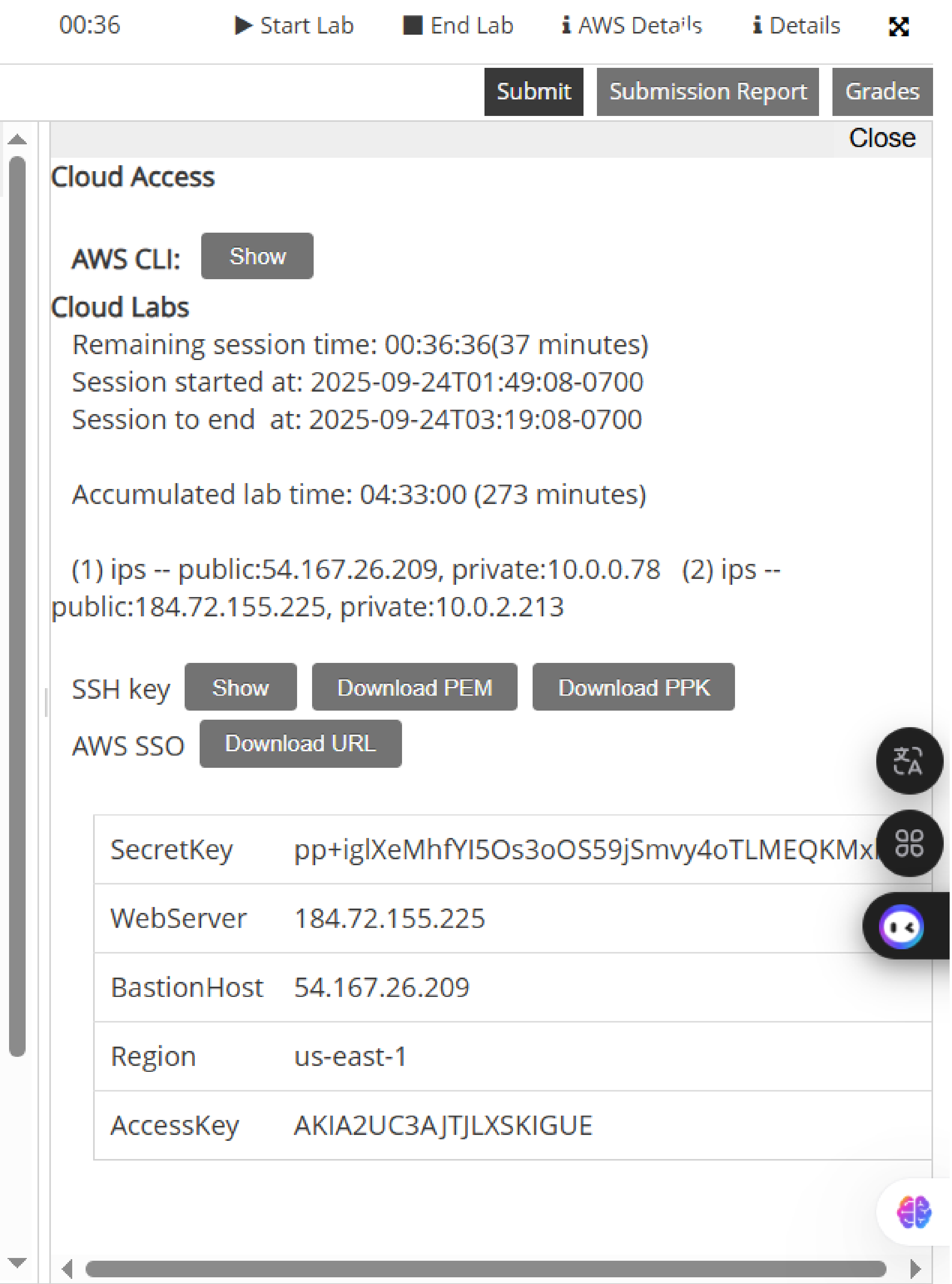
	DB identifier	Status	Role	Engine	Region ...	Size
○	lab-db	✓ Available	Instance	MySQL Co...	us-east-1a	db.t3.micro

Figure 20: Database available

Step 3.2: Click the **lab-db** itself, scroll down to the **Connectivity & security** section and copy the Endpoint. Then, copy the endpoint and paste it into the text editor

F. Task 4: Interact with Your Database

Step 4.1: ℹ AWS Details Choose the on the top right corner



Cloud Access

AWS CLI:

Cloud Labs

Remaining session time: 00:36:36(37 minutes)
Session started at: 2025-09-24T01:49:08-0700
Session to end at: 2025-09-24T03:19:08-0700

Accumulated lab time: 04:33:00 (273 minutes)

(1) ips -- public:54.167.26.209, private:10.0.0.78 (2) ips -- public:184.72.155.225, private:10.0.2.213

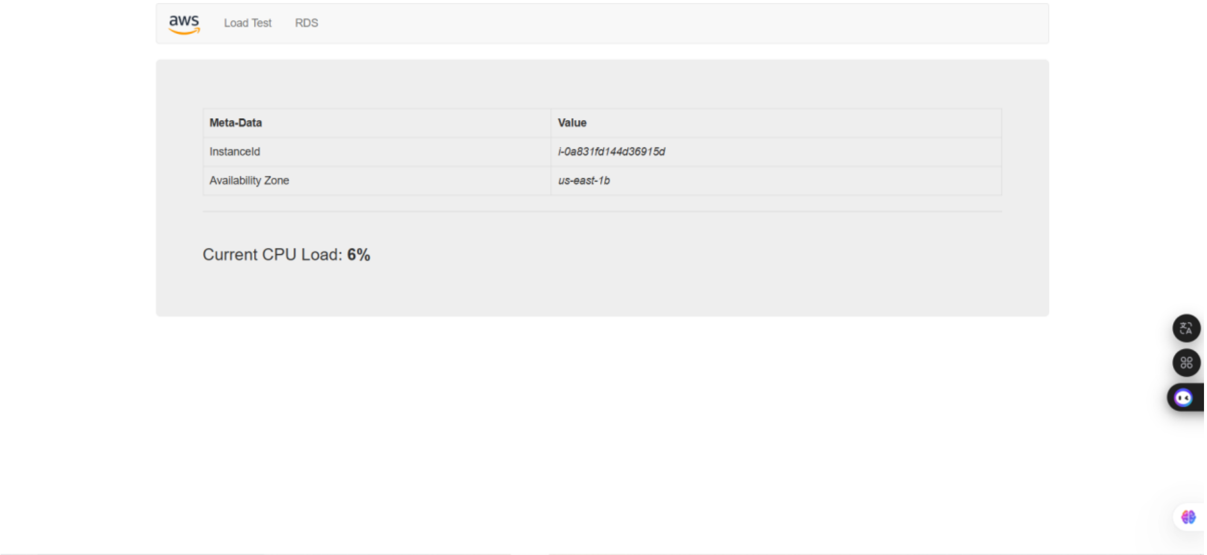
SSH key

AWS SSO

SecretKey	pp+iglXeMhfYI5Os3oOS59jSmvy4oTLMEQKMx
WebServer	184.72.155.225
BastionHost	54.167.26.209
Region	us-east-1
AccessKey	AKIA2UC3AJTJLXSKIGUE

Figure 21: AWS Details

Step 4.2: Pate the Webserver link into a new browser tab



Meta-Data	Value
InstanceId	*i-0a831fd144d36915d*
Availability Zone	*us-east-1b*

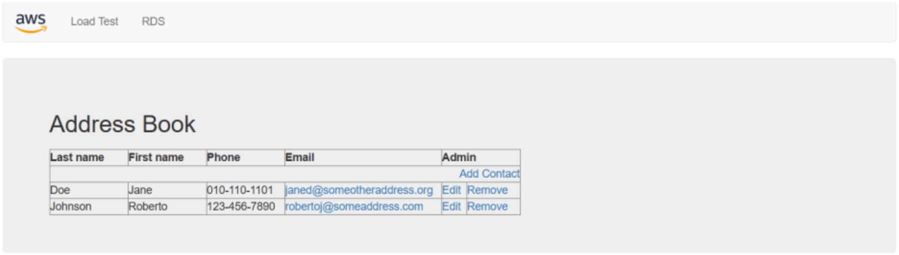
Current CPU Load: **6%**

Figure 22: Webserver page

Step 4.3: Choose the **RDS** and then configure following settings:

- **Endpoint:** lab-db.c16agicay2t3.us-east-1.rds.amazonaws.com
- **Database:** lab
- **Username:** main
- **Password:** lab-password

And finally click **Submit**

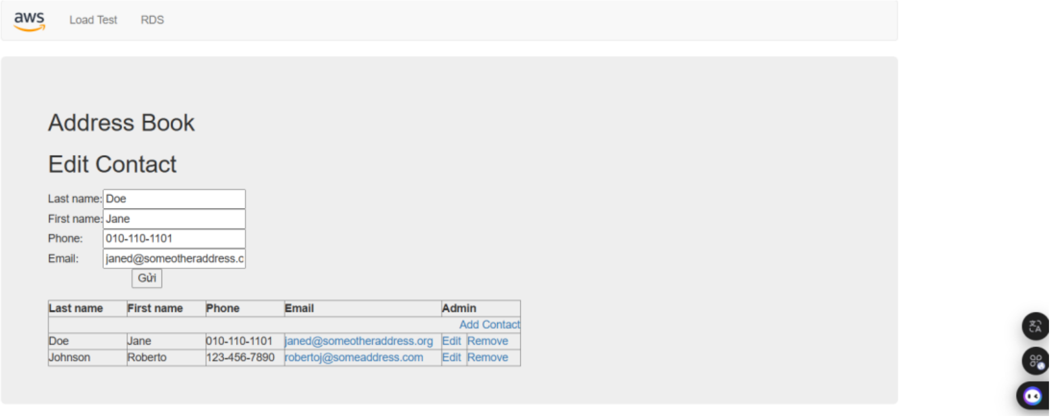


Address Book

Last name	First name	Phone	Email	Admin
				Add Contact
Doe	Jane	010-110-1101	janed@someotheraddress.org	Edit Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

Figure 23: Address Book page

Step 4.4: Try to adding, editing and removing contactson the web application.



Address Book

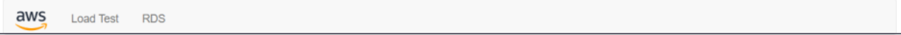
Edit Contact

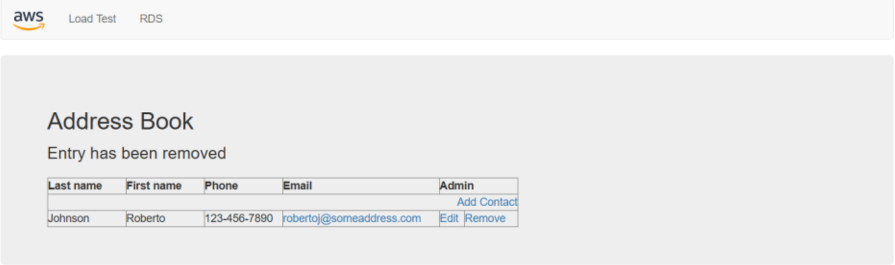
Last name: Doe
First name: Jane
Phone: 010-110-1101
Email: janed@someotheraddress.c

Last name	First name	Phone	Email	Admin
				Add Contact
Doe	Jane	010-110-1101	janed@someotheraddress.org	Edit Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

Figure 24: Edit Data processing

Figure 25: Edit data successfully



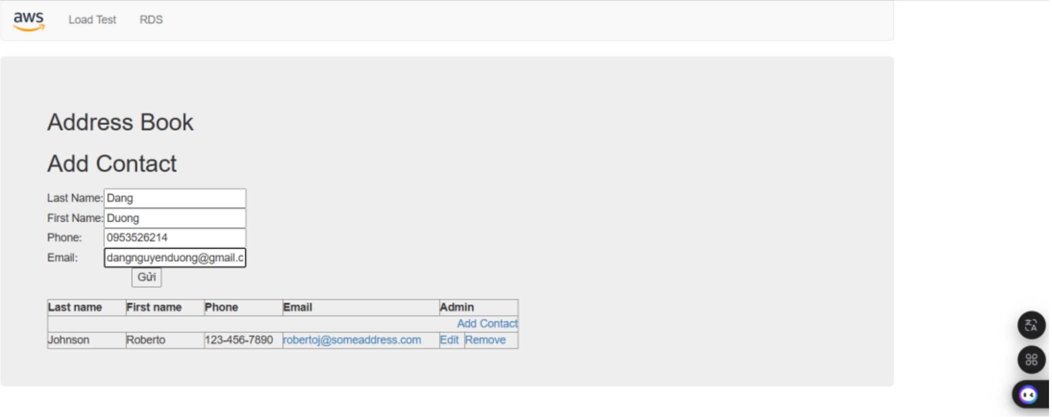


Address Book

Entry has been removed

Last name	First name	Phone	Email	Admin
				Add Contact
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

Figure 26: Remove data successfully



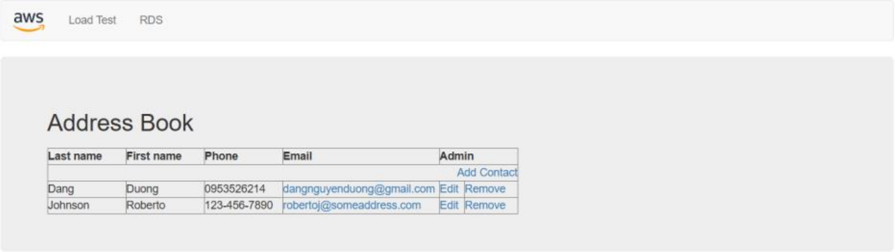
Address Book

Add Contact

Last Name: Dang
First Name: Duong
Phone: 0953526214
Email: dangnguyenduong@gmail.c

Last name	First name	Phone	Email	Admin
				Add Contact
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

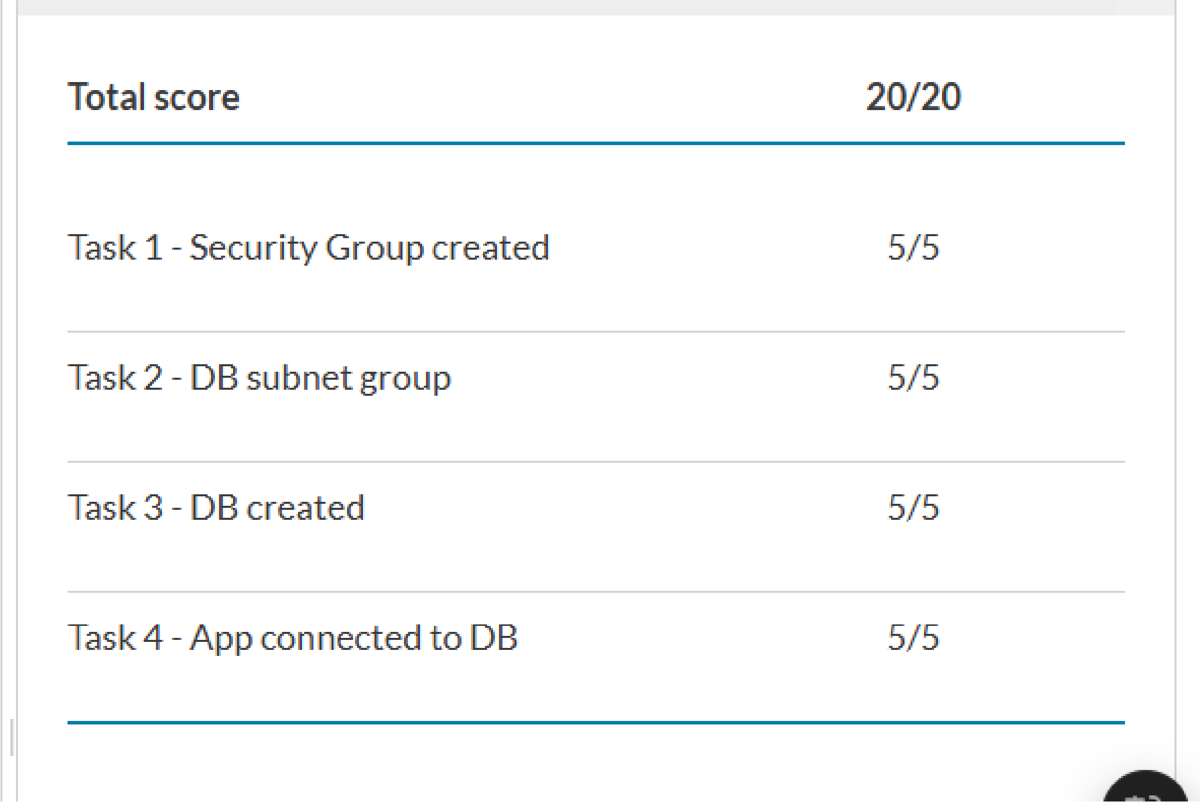
Figure 27: Add data process



Address Book

Last name	First name	Phone	Email	Admin	
				Add Contact	
Dang	Duong	0953526214	dangnguyenduong@gmail.com	Edit	Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit	Remove

Figure 28: Add data successfully



Total score	20/20
Task 1 - Security Group created	5/5
Task 2 - DB subnet group	5/5
Task 3 - DB created	5/5
Task 4 - App connected to DB	5/5

Figure 29: Completely lab

G. Conclusion

- In this lab, I learned how to set up and work with Amazon RDS by creating a relational database instance with high availability. I started by configuring security groups and subnet groups, then launched a Multi-AZ RDS MySQL instance. After that, I connected the database to a web application and successfully added, edited, and removed data through the app**.**

- Through this process, I gained hands-on experience with managed database services in AWS. I understood how Amazon RDS simplifies database administration tasks such as scaling, availability, and security, while still allowing me to interact with the database like a traditional system. This lab helped me build confidence in deploying and managing relational databases in the cloud, which is an important skill for real-world applications.