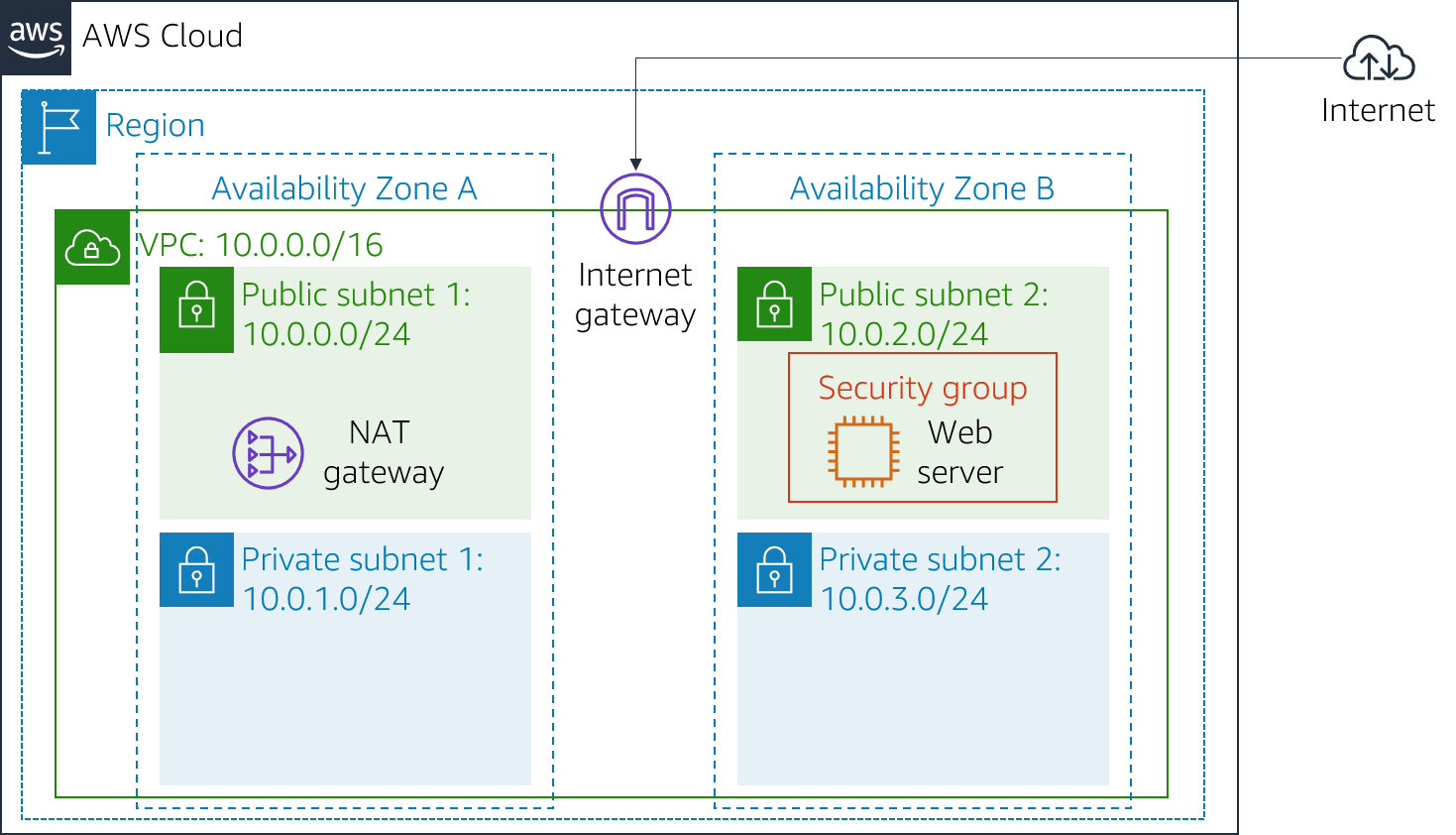
**Roll No:** 20BCE204

**Course:** Cloud Computing

**Practical No:** 9

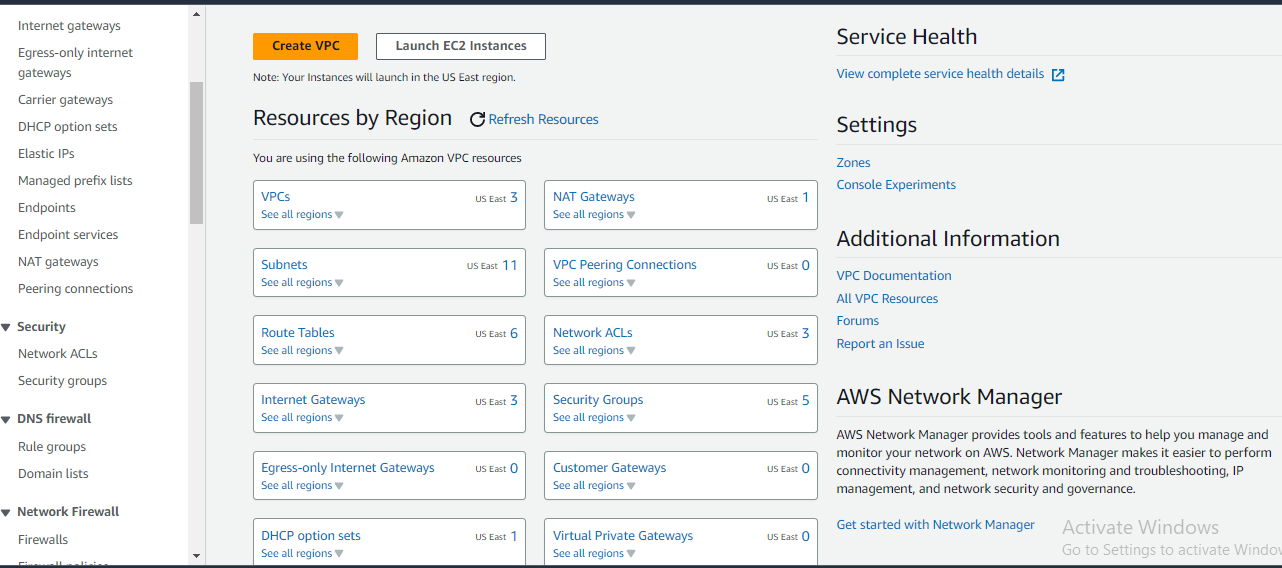
**Aim:** Build Your DB Server and Interact with Your DB Using an App.

*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.

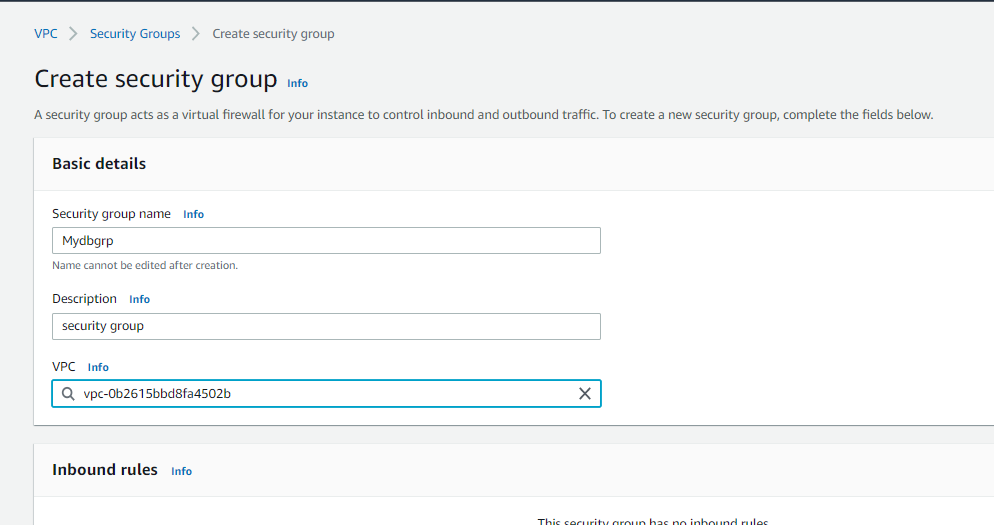
****

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

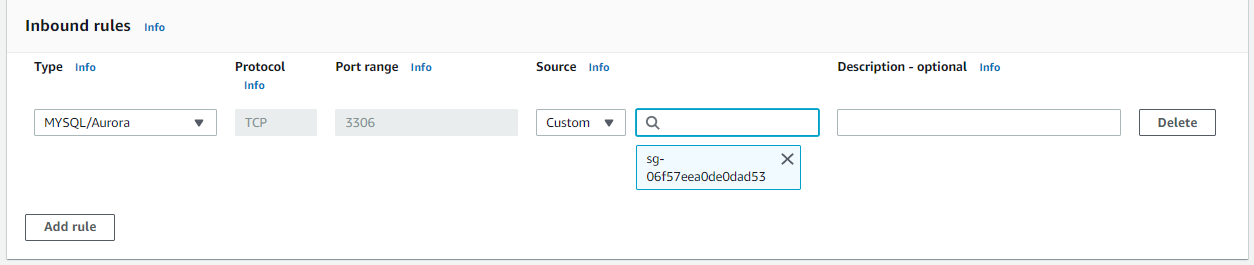
1. In the AWS Management Console, on the Services menu, choose VPC.
2. In the left navigation pane, choose Security Groups.

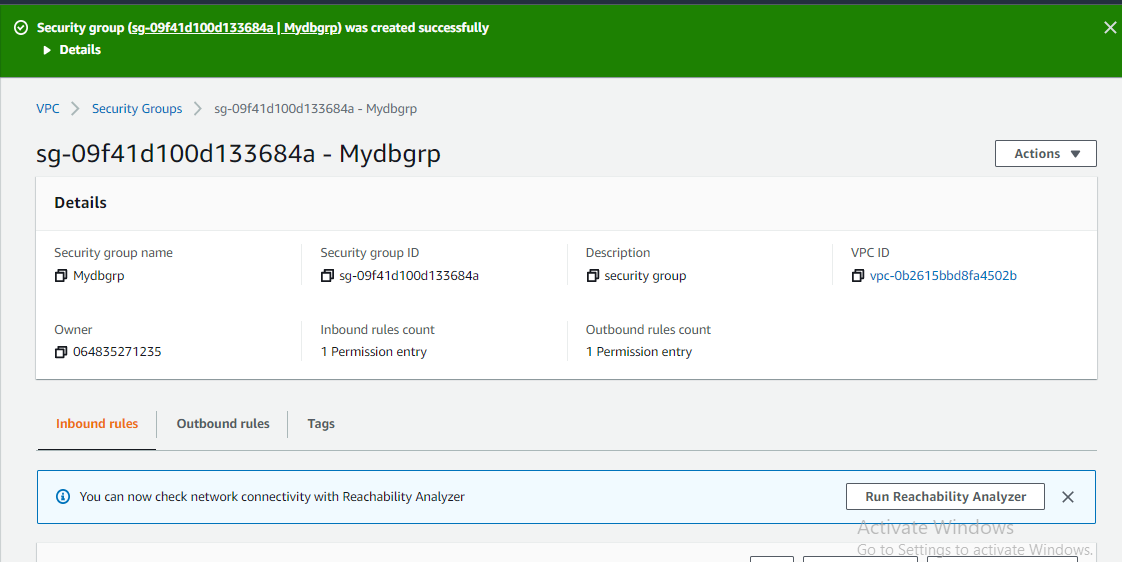
****

1. Choose Create security group and then configure:
   * 1. Security group name: Mydbgrp
     2. Description: security group
     3. VPC: *Lab VPC*

****

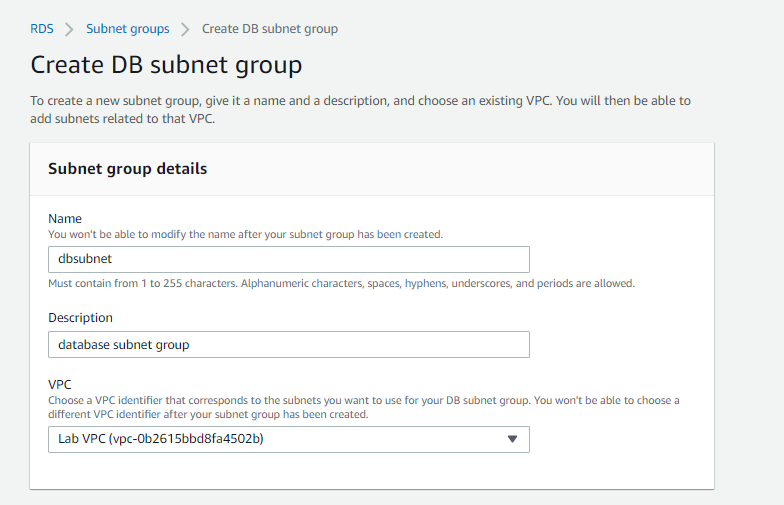
1. In the Inbound rules pane, choose Add rule
2. Configure the following settings:
3. Type: *MySQL/Aurora (3306)*
4. CIDR, IP, Security Group or Prefix List: Type sg and then select *Web Security Group*.

****

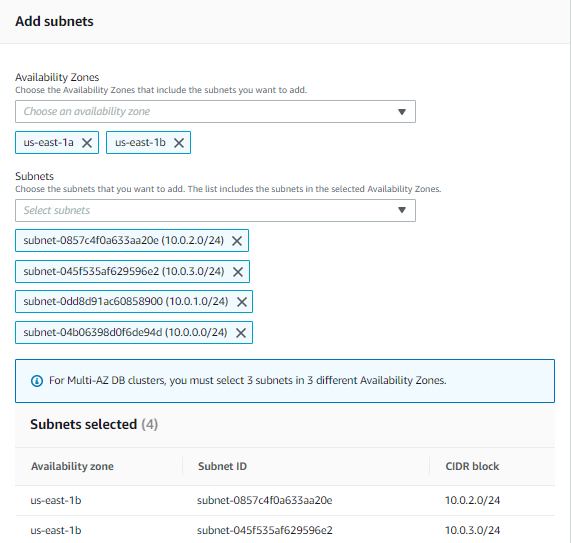
****

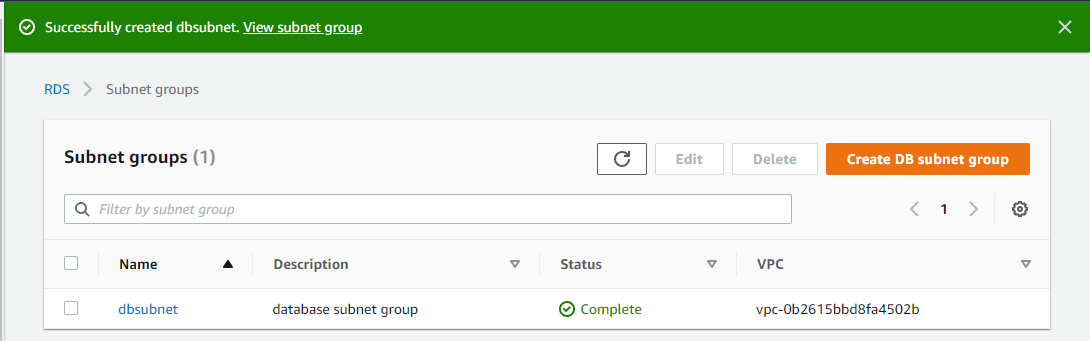
**Task 2: Create a DB Subnet Group**

1. On the Services menu, choose RDS.
2. In the left navigation pane, choose Subnet groups.
3. Choose Create DB Subnet Group then configure:
   * 1. Name: dbsubnet
     2. Description: database subnet group
     3. VPC: *Lab VPC*

****

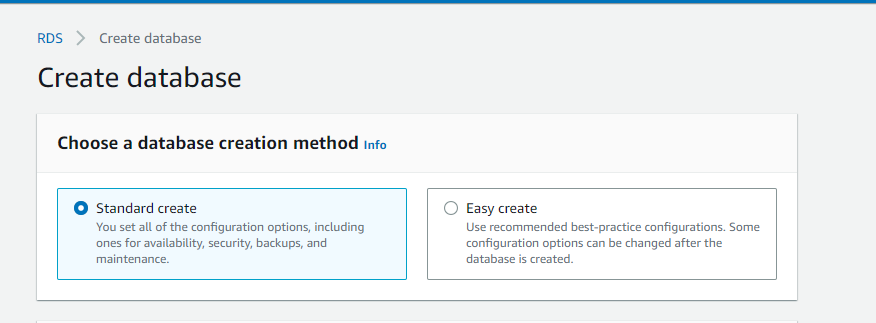
1. Scroll down to the Add Subnets section.
2. Expand the list of values under Availability Zones and select the first two zones: us-east-1a and us-east-1b.
3. Expand the list of values under Subnets and select the subnets associated with the CIDR ranges 10.0.1.0/24 and 10.0.3.0/24.
4. Choose Create

****

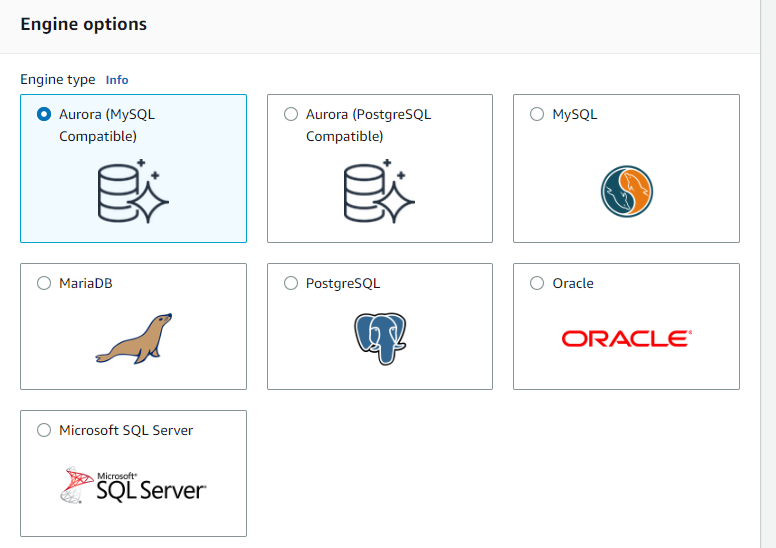
****

**Task 3: Create an Amazon RDS DB Instance**

1. In the left navigation pane, choose Databases.
2. Choose Create database

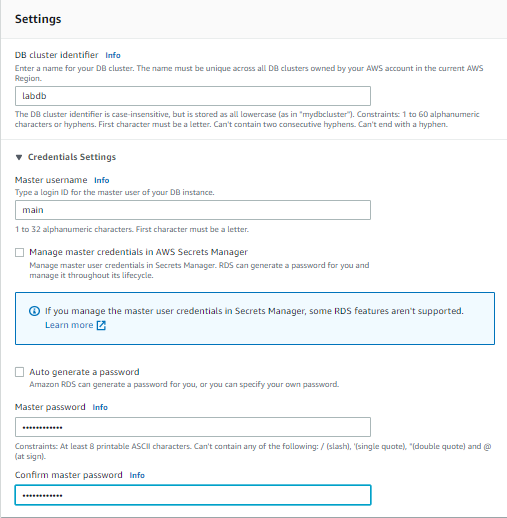


1. Select MySQL under Engine Options.
2. Under Templates choose Dev/Test.
3. Under Availability and durability choose Multi-AZ DB instance.

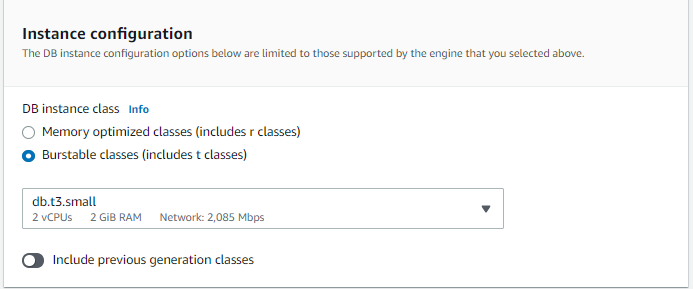
****

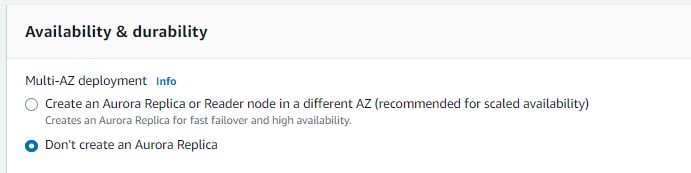
****

1. Under Settings, configure:
   * 1. DB instance identifier: labdb
     2. Master username: main
     3. Master password: lab-password
     4. Confirm password: lab-password

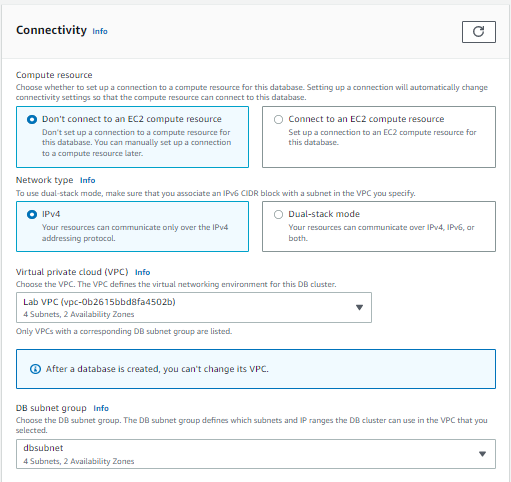


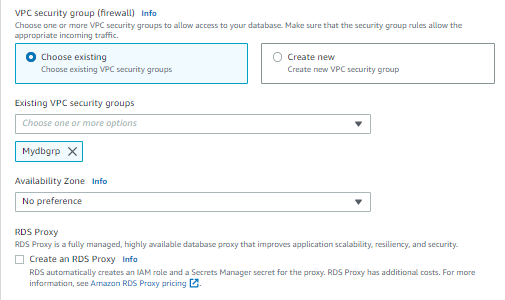
1. Under DB instance class, configure:
   * 1. Select Burstable classes (includes t classes).
     2. Select *db.t3.micro*

****

****

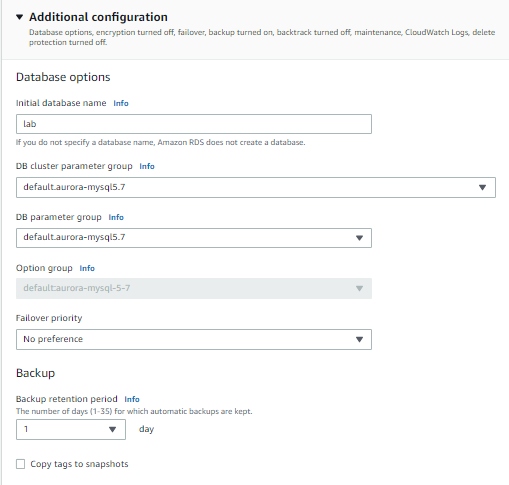
1. Under Connectivity, configure:
   * 1. Virtual Private Cloud (VPC): *Lab VPC*
2. Under Existing VPC security groups, from the dropdown list:
   * 1. Choose *DB Security Group*.
     2. Deselect *default*.

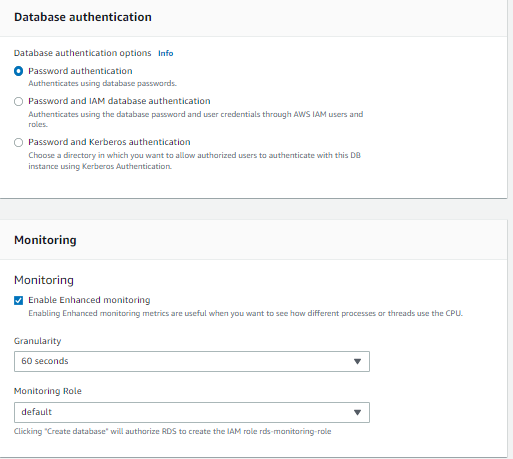


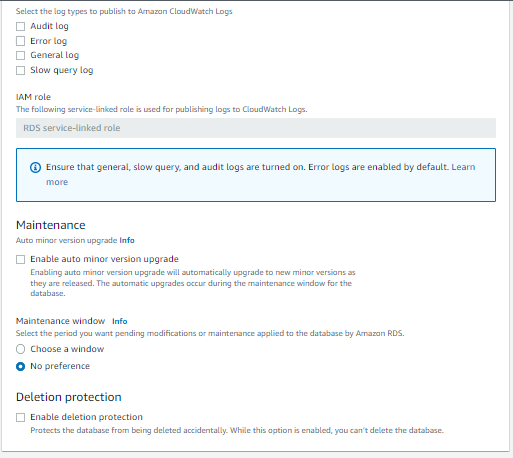


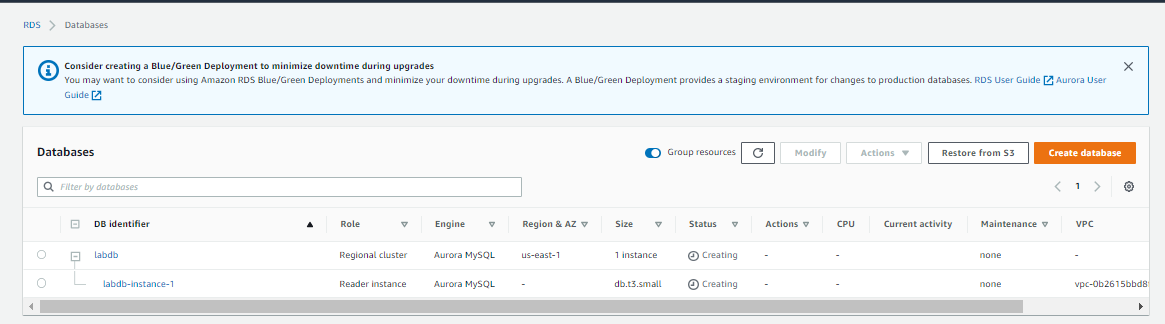
1. Expand Additional configuration, then configure:

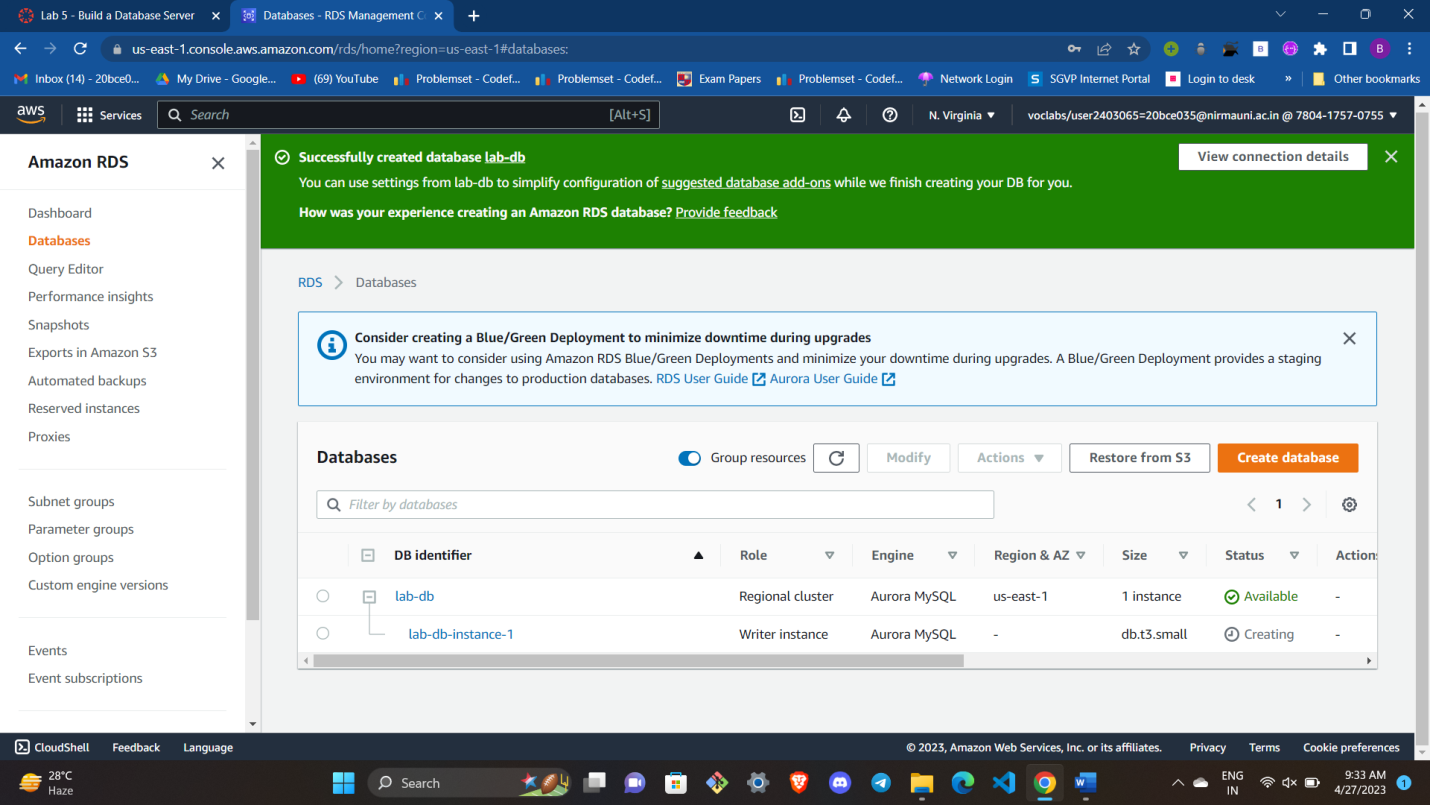
* Initial database name: lab
* Uncheck Enable automatic backups.
* Uncheck Enable encryption
* Uncheck Enable Enhanced monitoring.

****

****

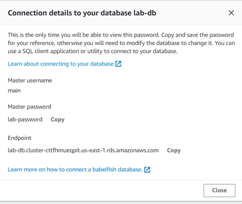
****

****



1. Scroll down to the Connectivity & security section and copy the Endpoint field.

It will look similar to: *lab-db.cggq8lhnxvnv.us-west-2.rds.amazonaws.com*

****

Name :- main

Password :- lab-password

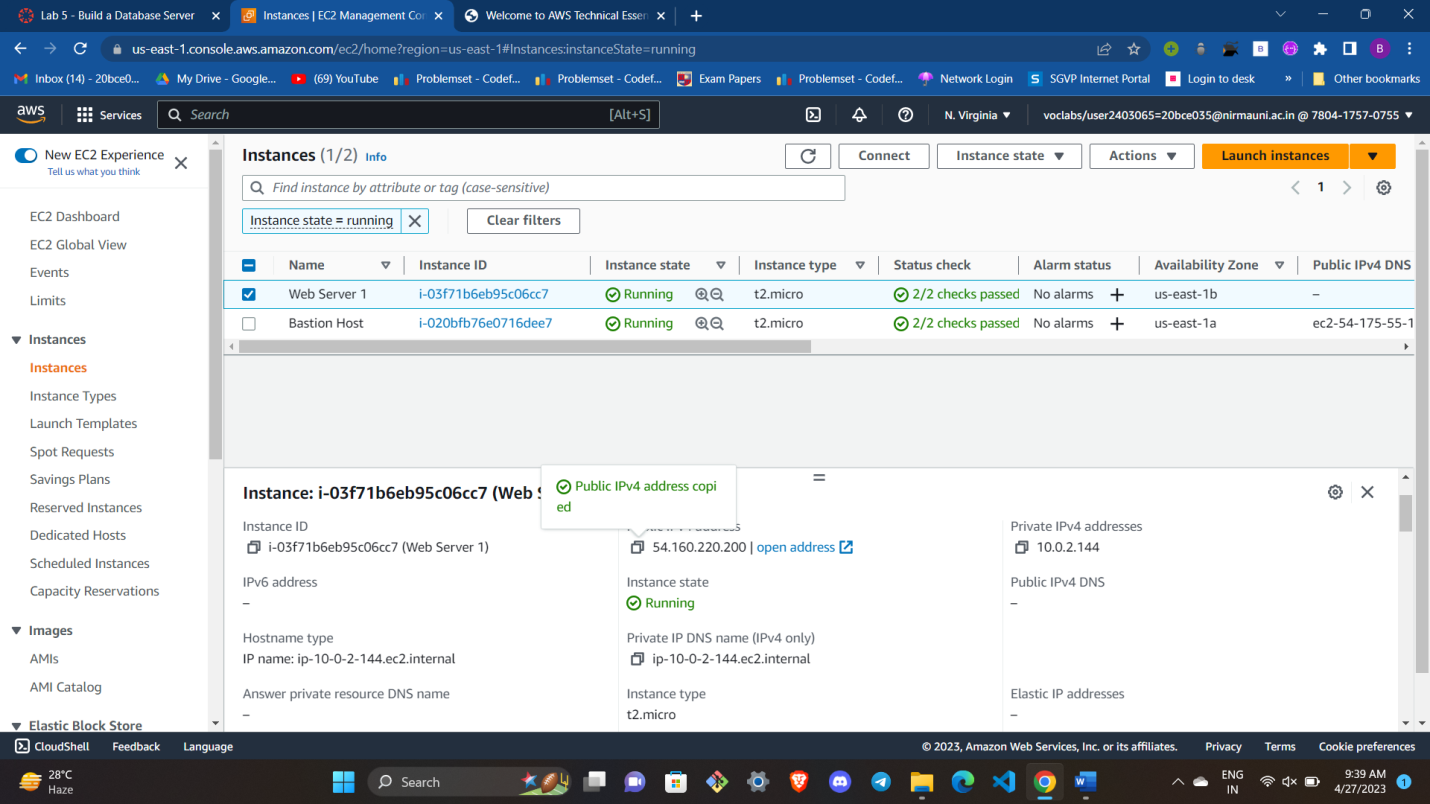
Endpoint :- lab-db.cluster-cttfhmueqpit.us-east-1.rds.amazonaws.com

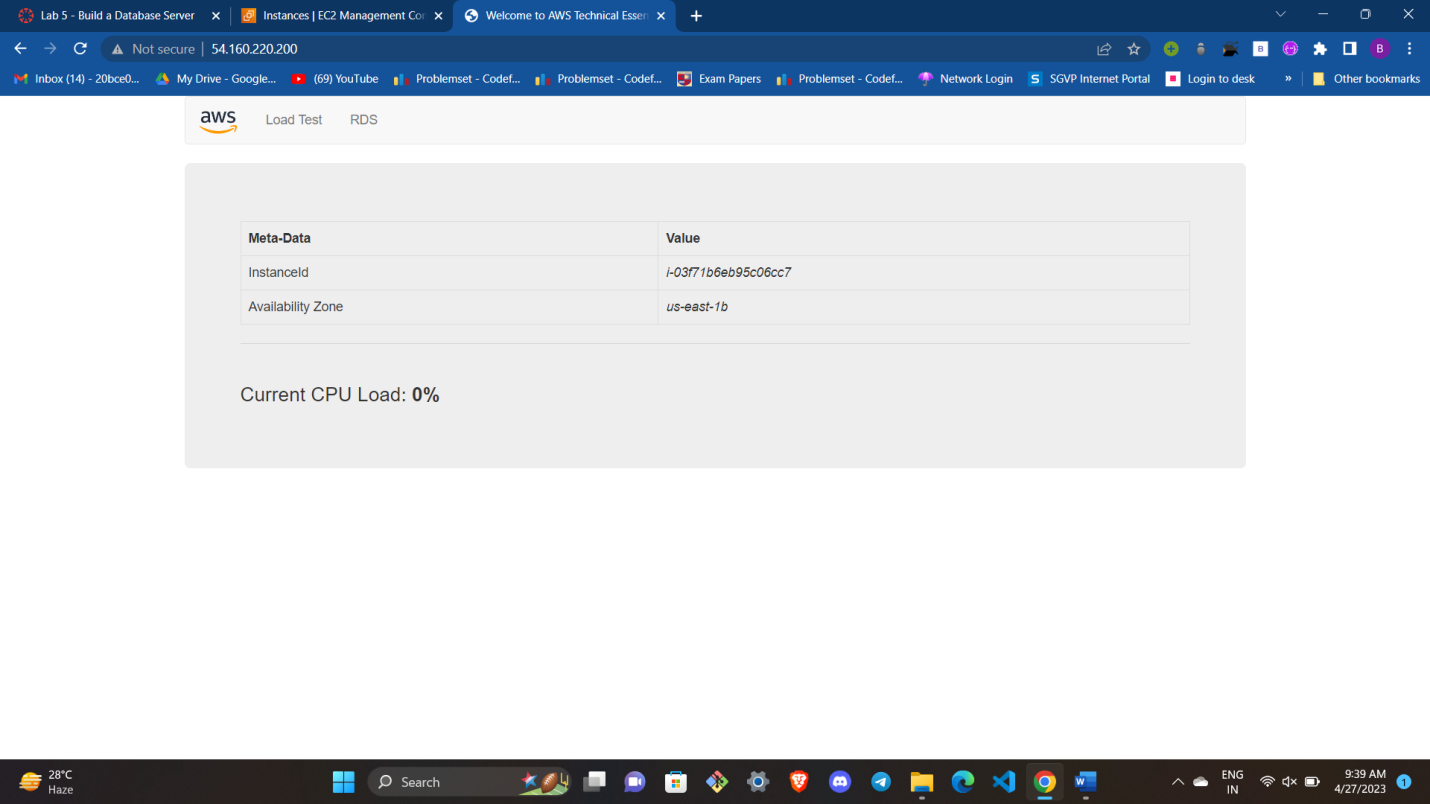
**Task 4: Interact with Your Database**

1. 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.

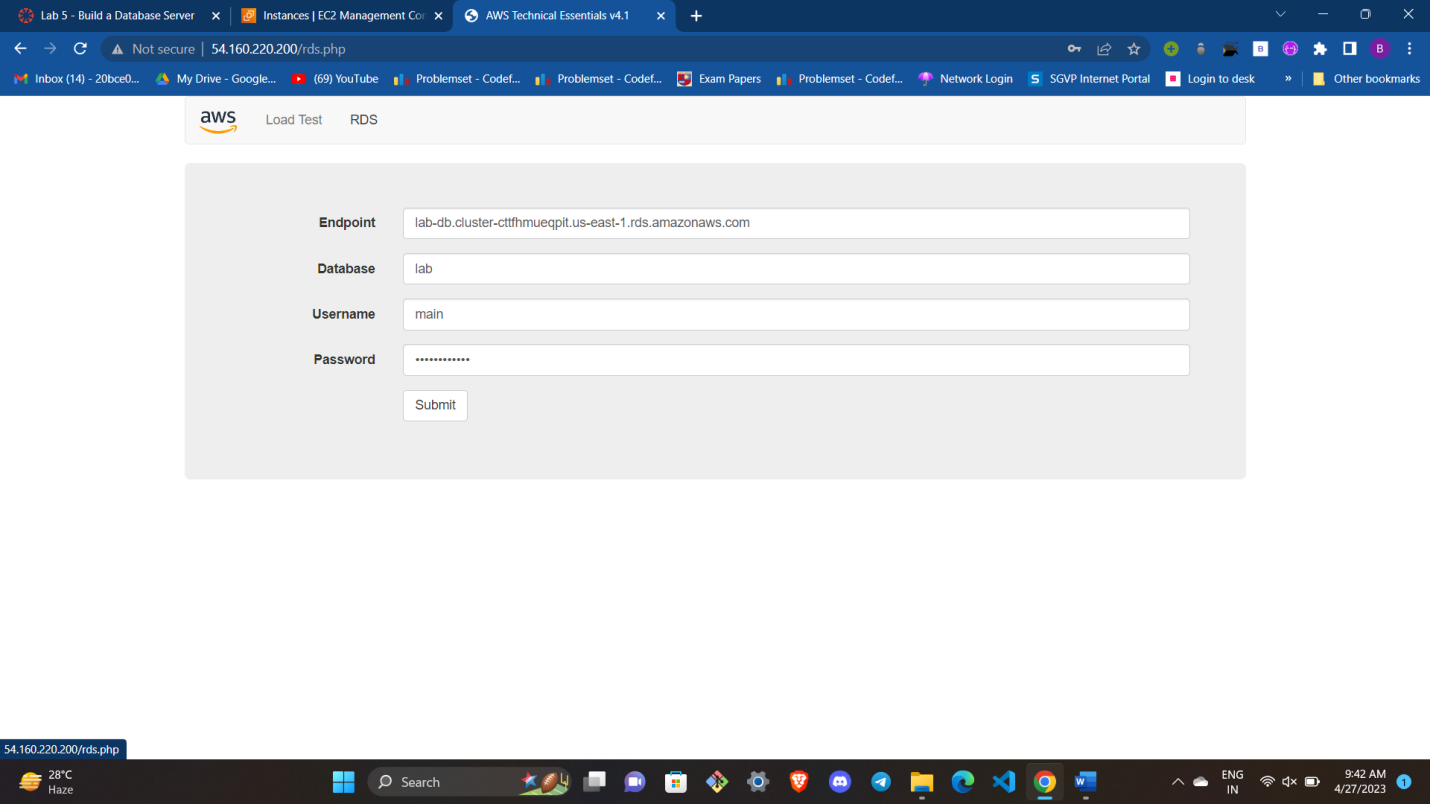
1. Choose the RDS link at the top of the page.

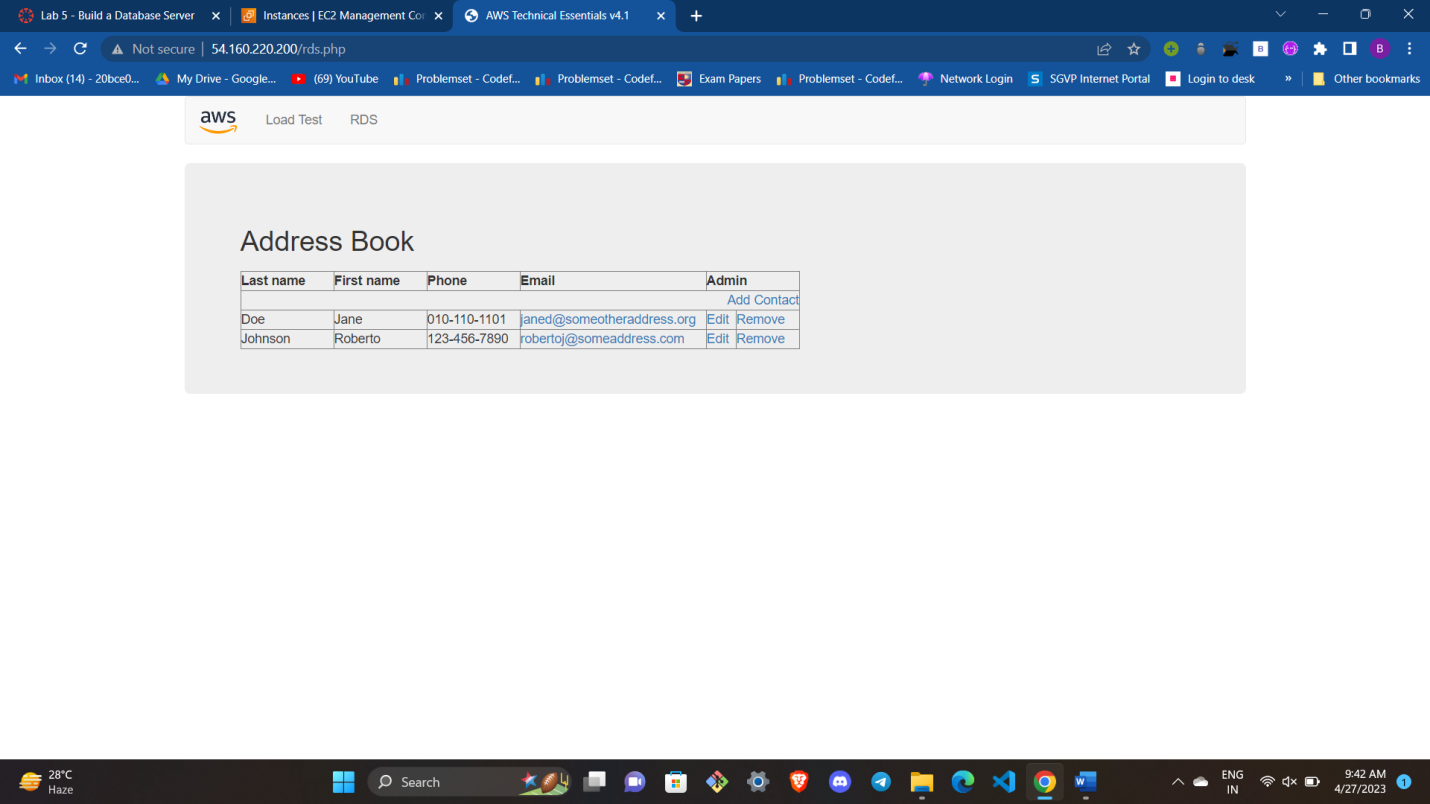
****

****

1. Configure the following settings:

* Endpoint: Paste the Endpoint you copied to a text editor earlier
* Database: lab
* Username: main
* Password: lab-password
* Choose Submit

****

****

**CONCLUSION :-**

By successfully implementing a dedicated database server and an application to interact with it, you can streamline data management processes, enhance user experience, and improve overall productivity. Additionally, with a well-designed architecture, you can scale your database server as your data grows and handle increasing user demands.

Overall, building a dedicated database server and developing an application to interact with your database is a powerful combination that enables efficient data management, accessibility, and scalability. It empowers you to harness the full potential of your data and leverage it effectively for your business or project needs.