

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No. 6

A.1 Aim:

To create database using RDS.

A.2 Prerequisite:

A.3 Outcome:

After successful completion of this course, students will be able to, install and appreciate security features and user management.

A.4 Theory:

What is Amazon Relational Database Service (Amazon RDS)?

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Overview of Amazon RDS:

Why do you want a managed relational database service? Because Amazon RDS takes over many of the difficult or tedious management tasks of a relational database:

When you buy a server, you get CPU, memory, storage, and IOPS, all bundled together. With Amazon RDS, these are split apart so that you can scale them independently. If you need more CPU, less IOPS, or more storage, you can easily allocate them.

Amazon RDS manages backups, software patching, automatic failure detection, and recovery.

To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges.

You can have automated backups performed when you need them, or manually create your own backup snapshot. You can use these backups to restore a database. The Amazon RDS restore process works reliably and efficiently.

You can get high availability with a primary instance and a synchronous secondary instance that you can fail over to when problems occur. You can also use MySQL, MariaDB, or PostgreSQL Read Replicas to increase read scaling.

You can use the database products you are already familiar with: MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server.

In addition to the security in your database package, you can help control who can access your RDS databases by using AWS Identity and Access Management (IAM) to define users and permissions. You can also help protect your databases by putting them in a virtual private cloud.

Amazon RDS Interfaces:

AWS Management Console

The AWS Management Console is a simple web-based user interface. You can manage your DB instances from the console with no programming required. To access the Amazon RDS console, sign in to the AWS Management Console and open the Amazon RDS console.

Command Line Interface

You can use the AWS Command Line Interface (AWS CLI) to access the Amazon RDS API interactively. To install the AWS CLI, see [Installing the AWS Command Line Interface](#). To begin using the AWS CLI for RDS, see [AWS Command Line Interface Reference for Amazon RDS](#).

Programming with Amazon RDS

If you are a developer, you can access the Amazon RDS programmatically. For more information, see [Amazon RDS Application Programming Interface \(API\) Reference](#).

For application development, we recommend that you use one of the AWS Software Development Kits (SDKs). The AWS SDKs handle low-level details such as authentication, retry logic, and error handling, so that you can focus on your application logic. AWS SDKs are available for a wide variety of languages. AWS also provides libraries, sample code, tutorials, and other resources to help you get started more easily. For more information, see [Sample Code & Libraries](#).

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

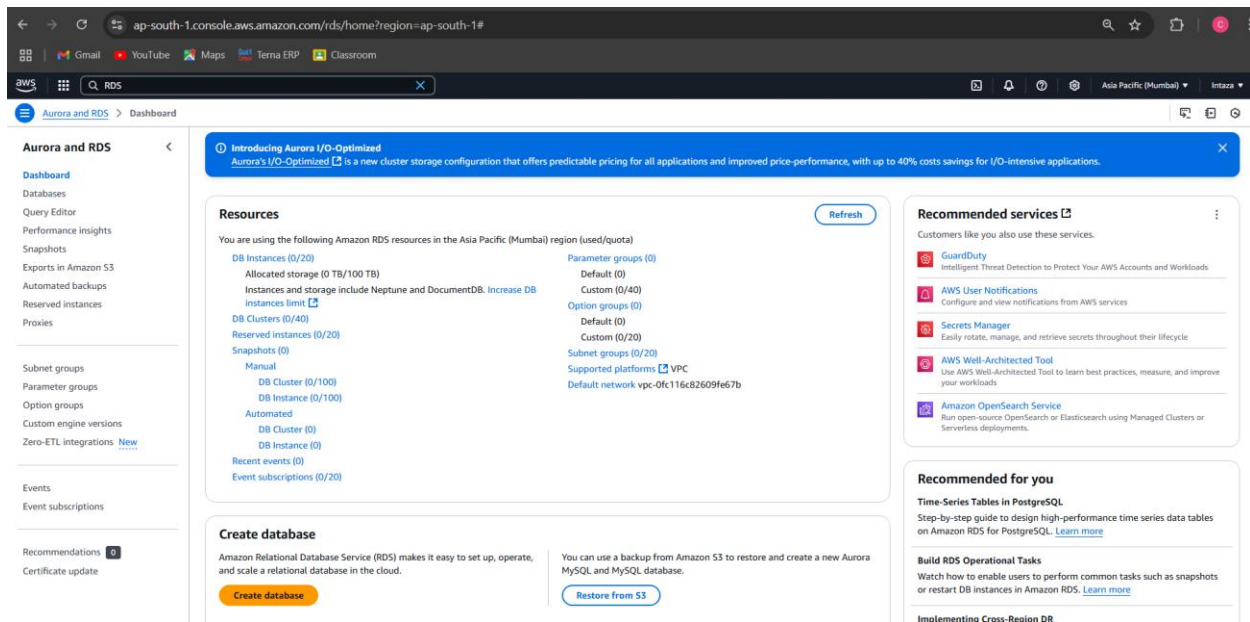
(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No.B48	Name: Aryan Unhale
Class:TE B COMPS	Batch:B3
Date of Experiment:27/3/25	Date of Submission:29/3/25
Grade:	

B.1 Software Code written by student:

B.2 Input and Output:

Step 1: Open the Amazon RDS Service(search for **RDS** in the search bar and Click on **Aurora and RDS** to open the service.)



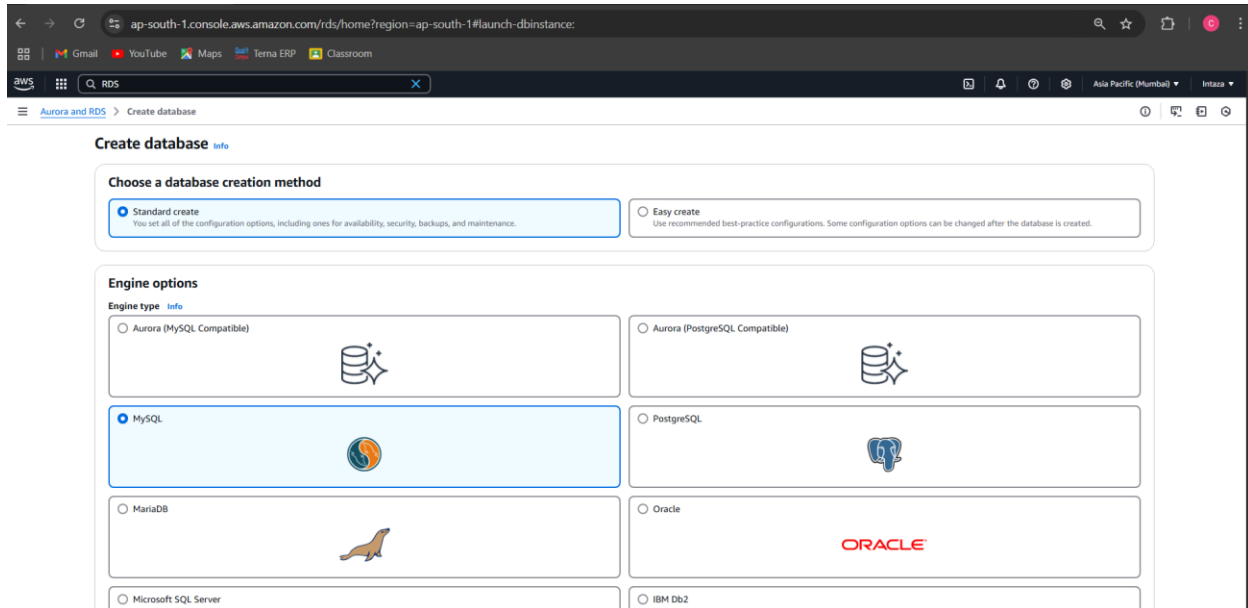
The screenshot displays the Amazon RDS console dashboard. The top navigation bar shows the region as Asia Pacific (Mumbai). The left sidebar contains a menu with options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main content area is titled 'Resources' and shows a summary of RDS resources in the Asia Pacific (Mumbai) region. It includes a 'Refresh' button and a list of resources such as DB instances, clusters, snapshots, and parameter groups. Below the resource summary, there is a 'Create database' section with a 'Create database' button and a 'Restore from S3' button. On the right side, there are sections for 'Recommended services' and 'Recommended for you', which include links to various AWS services and documentation.

Step 2: Scroll down and Click on **Create database**.

- Choose **Standard create** for full configuration options.

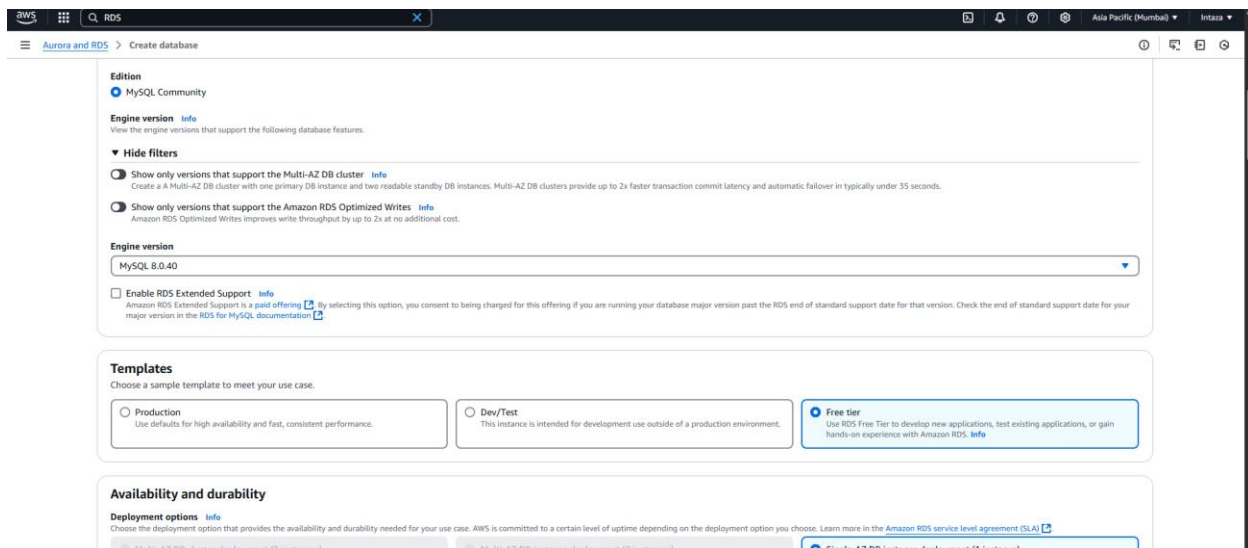
Step 3: Select Database Engine

- Under **Engine options**, select a relational database (e.g., **MySQL**).
- Choose the **Engine Version** (default recommended).



Step 4: Configure Database Settings

- **Templates:** Select **Free Tier** (if eligible) to avoid costs.
- **DB instance identifier:** Enter a unique name (e.g., myddb-instance).
- **Master username:** Enter admin (or a custom username).
- **Master password:** Set a secure password.



Instance Specifications (default for Free Tier).

Step 5: Configure Connectivity

- **Virtual Private Cloud (VPC):** Use the default VPC.
- **Public access:** Select **Yes** (for lab purposes only).

The screenshot shows the 'Create database' page in the AWS RDS console. The 'Connectivity' section is active, showing options for connecting to an EC2 instance or not. The 'Virtual private cloud (VPC)' section shows the 'Default VPC' selected. The 'DB subnet group' is set to 'default'. The 'Public access' section has 'Yes' selected. The 'VPC security group (firewall)' section has 'Choose existing' selected, with 'default' chosen from the dropdown.

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.

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

Default VPC (vpc-0fc116c82609e67b)
3 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default

Public access info
☒ Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

☐ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the 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

default

Step 6: Set Initial Database Name

- Under **Additional configuration**, enter a database name (e.g., mydatabase).

The screenshot shows the 'Additional configuration' section of the 'Create database' page. It includes options for 'RDS Proxy', 'Certificate authority', 'Database port', 'Tags', and 'Database authentication'. The 'Database port' is set to '3306'. The 'Database authentication' section has 'Password authentication' selected.

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

☐ Create an RDS Proxy info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

Certificate authority - optional info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
(Expiry: May 20, 2021)

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

Database port info
TCP/IP port that the database will use for application connections.

3306

Tags - optional
A tag consists of a case-sensitive key-value pair.

No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tags.

Database authentication

Database authentication options info

☒ Password authentication
Authenticates using database passwords.

☐ Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

aws RDS

Aurora and RDS > Create database

Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)

intaza_mysql

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.

Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MySQL, refer to details [here](#).

Backup retention period [Info](#)

The number of days (1-35) for which automatic backups are kept.

1 day

Backup window [Info](#)

The daily time range (in UTC) during which RDS takes automated backups.

☐ Choose a window

☒ No preference

☒ Copy tags to snapshots

- **Step 7 : Create the Database**(Wait 5–10 minutes for the instance status to turn to **Available**.)

aws RDS

Aurora and RDS > Create database

Maintenance

Auto minor version upgrade [Info](#)

☒ **Enable auto minor version upgrade**

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)

Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

☐ Choose a window

☒ No preference

Deletion protection

☐ **Enable deletion protection**

Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).


You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

[Cancel](#)
[Create database](#)

Click on close:

Suggested add-ons for intaza-rds-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 - new

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](#)

i 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](#)

- instance status to turn to **Available**.

AWS

Aurora and RDS

Databases

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

Recommendations

Certificate update

Creating database intaza-rds-db

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

View credential details

Notifications

Group resources

Modify

Actions

Restore from S3

Create database

Databases (1)

Filter by databases

DB identifier

Status

Role

Engine

Region

Size

Recommendations

CPU

Current activity

Mal

intaza-rds-db	Creating	Instance	MySQL Co...	-	db.t4g.micro	-	non
---------------	----------	----------	-------------	---	--------------	---	-----

ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#databases:

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 [New](#)

Events
Event subscriptions

Recommendations [6](#)
Certificate update

Successfully created database intaza-rds-db

You can use settings from intaza-rds-db to simplify configuration of suggested database add-ons while we finish creating your DB for you.

[View connection details](#)

Databases (1)

Filter by databases

DB identifier	Status	Role	Engine	Region	Size	Recommendations	CPU	Current activity	Mal
intaza-rds-db	Available	Instance	MySQL Co...	ap-south-1a	db.t4g.micro				non

Step 8 : Copy Endpoint :

intaza-rds-db

[Modify](#) [Actions](#)

Summary

DB identifier intaza-rds-db	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 4.74%	Class db.t4g.micro	Current activity 0 Connections	Region & AZ ap-south-1a	

[Connectivity & security](#) [Monitoring](#) [Logs & events](#) [Configuration](#) [Zero-ETL integrations](#) [Maintenance & backups](#) [Data migrations - new](#) [Tags](#) [Recommendations](#)

Connectivity & security

Endpoint & port Endpoint intaza-rds-db.crkc0c8aggl4.ap-south-1.rds.amazonaws.com Port 3306	Networking Availability Zone ap-south-1a VPC vpc-0fc116c82609fe67b Subnet group default-vpc-0fc116c82609fe67b Subnets subnet-01166486613ebf5bd subnet-0c540017df34a6ba subnet-01c01a90627fa8ff Network type IPv4	Security VPC security groups default (sg-02b8c65ff9495b93) Publicly accessible Yes Certificate authority rds-ca-rsa2048-g1 Certificate authority date May 20, 2061, 00:10 (UTC+05:30) DB instance certificate expiration date March 29, 2026, 20:37 (UTC+05:30)
---	---	--

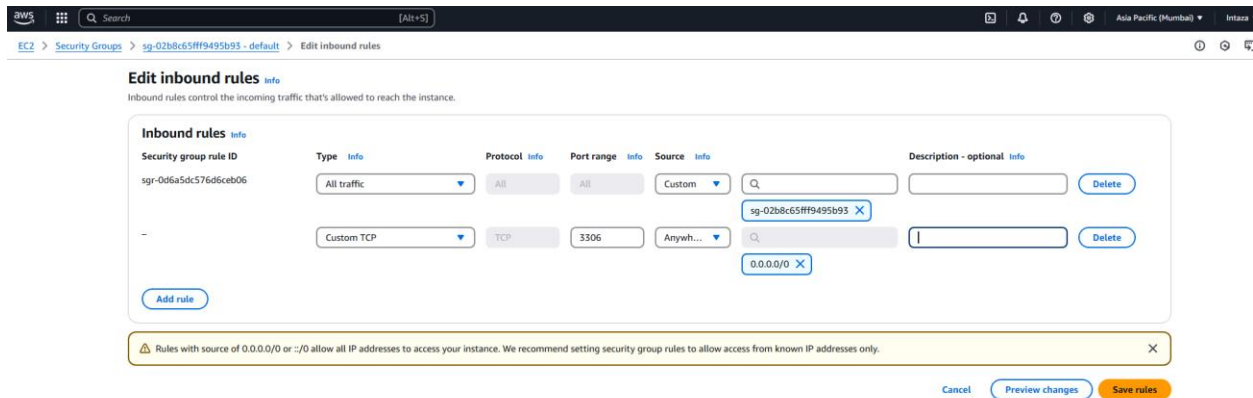
Step 9: Scroll Down and Click on EC2 security group-inbound from Security group rule.

The screenshot shows the AWS Aurora and RDS console. The left sidebar contains navigation links for Aurora and RDS, 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, Recommendations, and Certificate update. The main content area is titled 'Proxies (0)' and shows a table with columns for Proxy identifier, Status, and Engine family. Below this, the 'Security group rules (2)' section is expanded, showing a table with columns for Security group, Type, and Rule. The first rule is 'default (sg-02b8c65ff9495b93)' with type 'EC2 Security Group - Inbound' and rule 'sg-02b8c65ff9495b93'. The second rule is 'default (sg-02b8c65ff9495b93)' with type 'CIDR/IP - Outbound' and rule '0.0.0.0/0'. Below this, the 'Replication (1)' section is expanded, showing a table with columns for DB identifier, Role, Region & AZ, Replication source, Replication state, and Lag. The first entry is 'intaza-rds-db' with role 'Instance', region 'ap-south-1a', replication source '-', replication state '-', and lag '-'. A 'Create proxy' button is visible in the Proxies section.

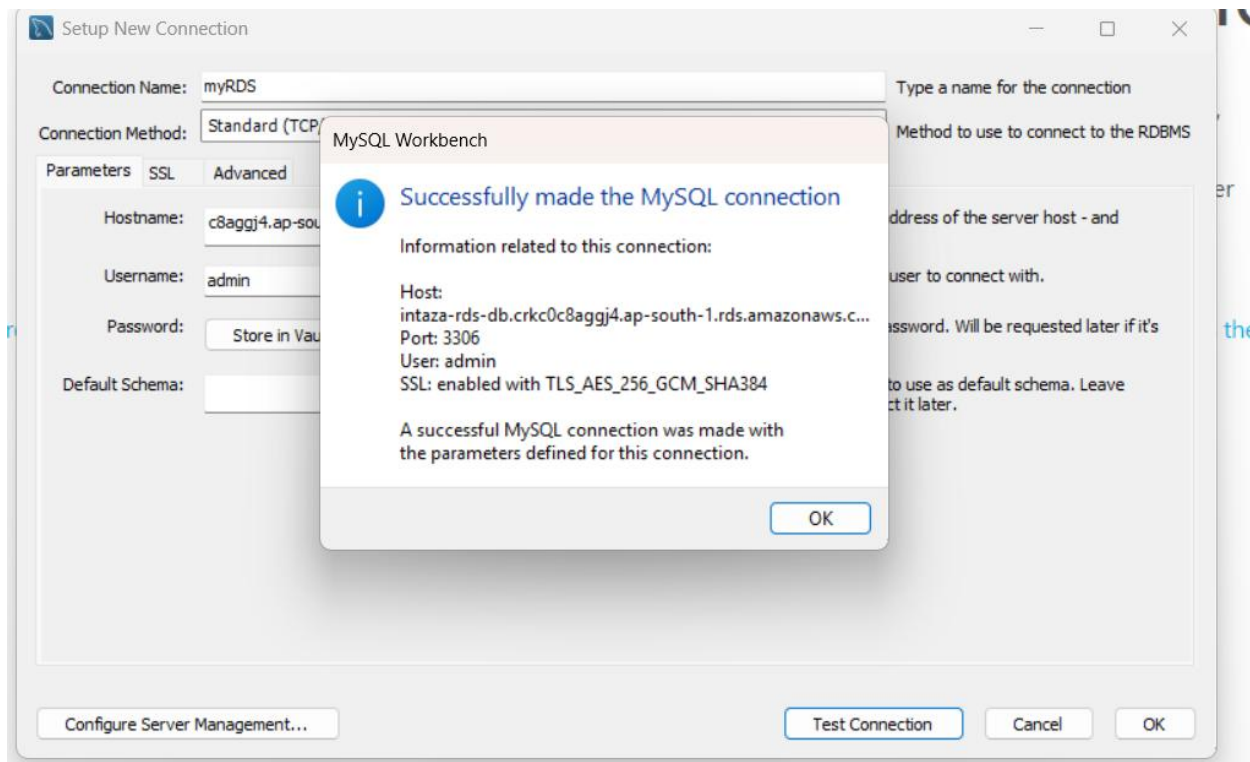
Step 10: Tick Check box and action>Edit inbound rule

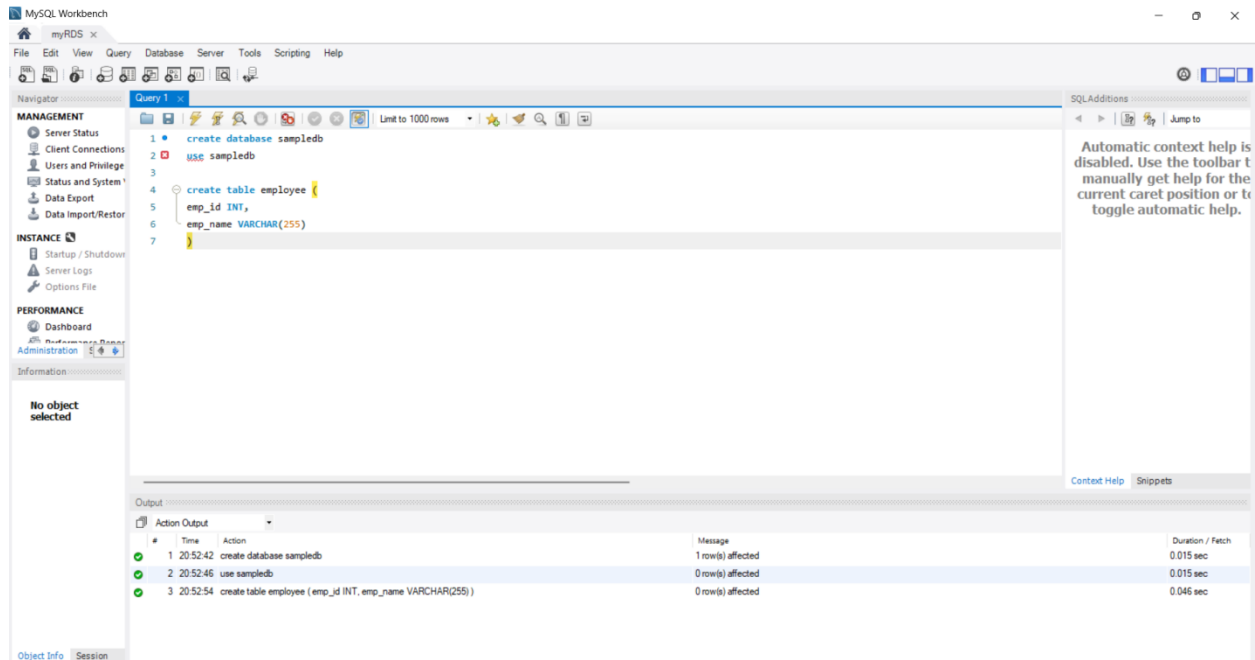
The screenshot shows the AWS EC2 console. The left sidebar contains navigation links for EC2, Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, and Placement Groups. The main content area is titled 'Security Groups (1/1)' and shows a table with columns for Name, Security group ID, Security group name, VPC ID, and Owner. The first entry is 'sg-02b8c65ff9495b93' with security group ID 'sg-02b8c65ff9495b93', security group name 'default', VPC ID 'vpc-0fc116c82609fe67b', and owner '699475959073'. The 'Actions' dropdown menu is open, showing options: View details, Edit inbound rules, Edit outbound rules, Manage tags, Manage stale rules, Copy to new security group, Share security group, and Delete security groups. The 'Edit inbound rules' option is selected. Below the table, the 'Details' tab for the security group is visible, showing the security group name 'default', security group ID 'sg-02b8c65ff9495b93', description 'default VPC security group', VPC ID 'vpc-0fc116c82609fe67b', owner '699475959073', and inbound/outbound rules count of 1.

Step 11: Type = Custom TCP and source= Anywhere IPv4 , save rule



Step 12: Connect from Local Machine(Mysql Workbench) and run some queries



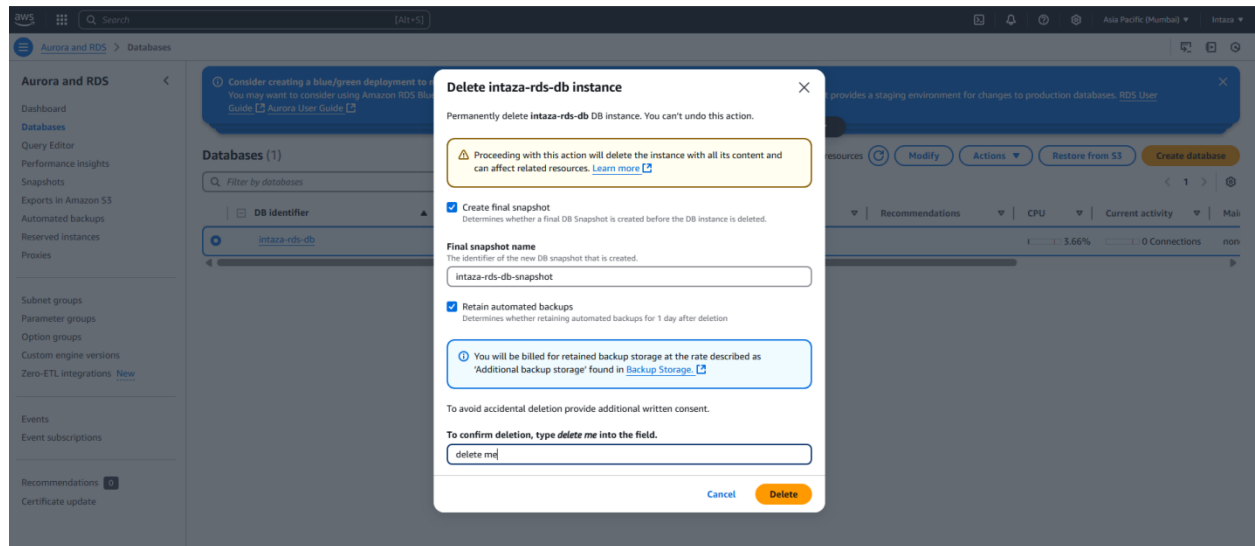


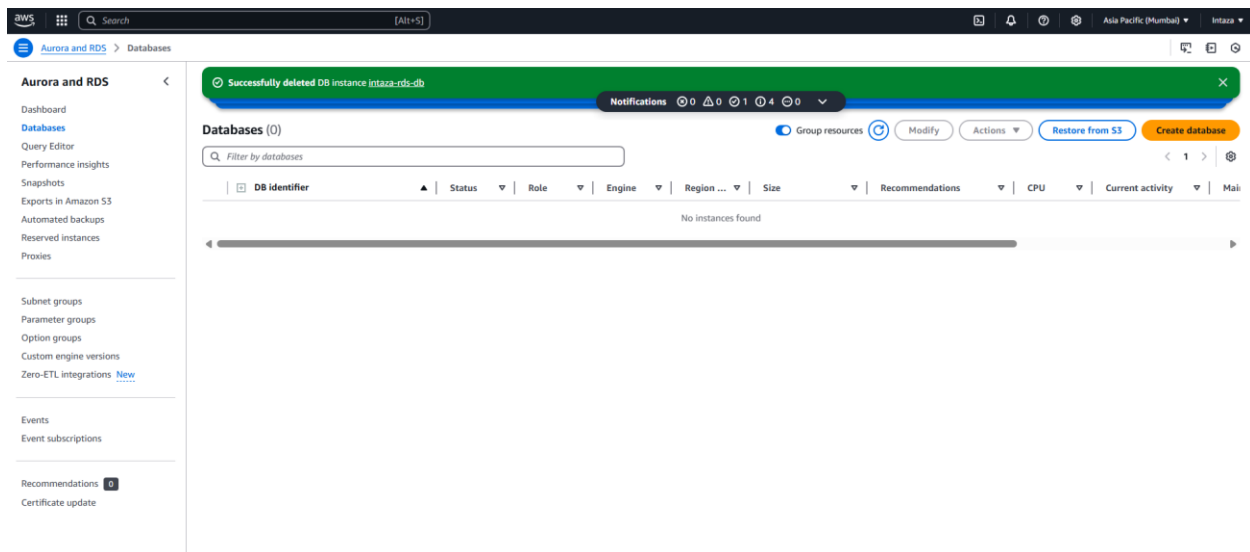
Step 13: Delete the RDS Database Instance

Go to AWS Console → RDS → Databases

Select your database instance (ma231t.ap-south-1.rds.amazonaws.com)

Click "Actions" → "Delete"





Step 14: Reset the Default Security Group

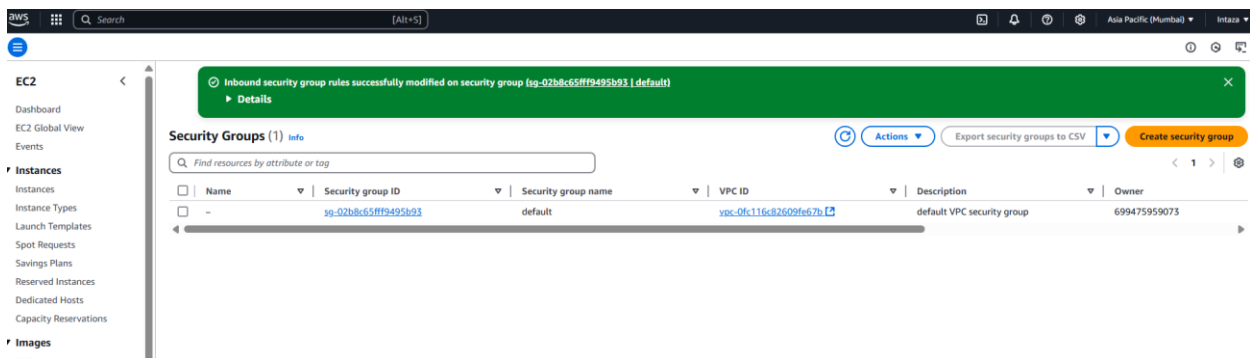
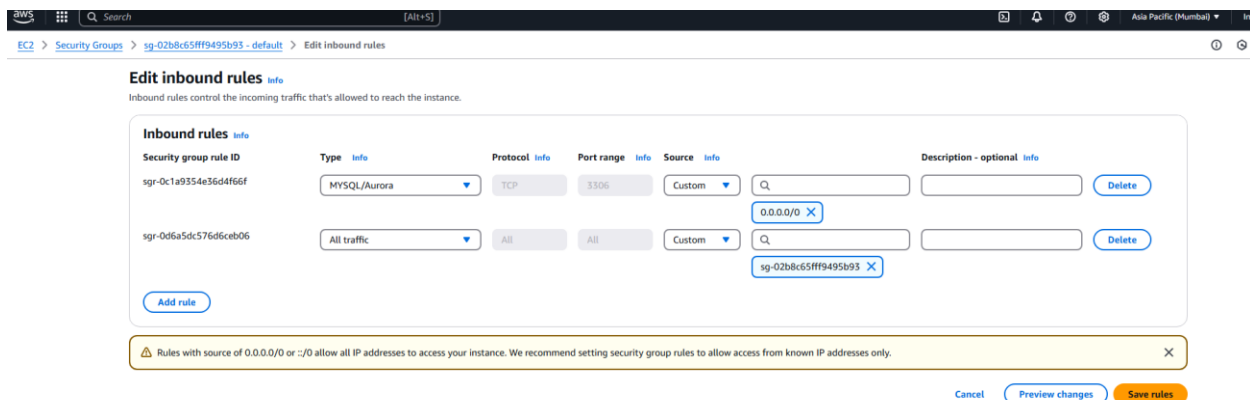
Go to AWS Console → EC2 → Security Groups

Find and select `sg-067fae1521f348144` (or any other related security groups)

Click "Actions" → "Click "Edit inbound rules"

Remove any custom inbound rules (if you mistakenly allowed external access).

Click "Save rules"



B.3 Observations and learning:

The experiment involved setting up a database using Amazon RDS, exploring its features, and understanding its management interfaces. We were able to configure and manage a relational database instance using AWS Management Console, AWS CLI, and API. Security features such as access control through IAM and database backups were observed as essential components of RDS. The ability to scale resources independently (CPU, memory, storage, and IOPS) was demonstrated. Automated database management, including backups, software patching, and failure recovery, was tested. Various database engines like MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server were available for deployment.

B.4 Conclusion:

The experiment successfully demonstrated the ease of deploying and managing a relational database using Amazon RDS. The automated features of RDS reduce the administrative overhead while providing scalability, security, and reliability. By utilizing AWS Management Console, CLI, and SDKs, students gained practical insights into database operations in a cloud environment, reinforcing the importance of cloud-based database solutions in modern applications.

B.5 Question of Curiosity

Q1: What types of RDS databases are currently available?

Ans:

Different types of database engines are currently available in RDS:

1. Amazon Aurora
2. MySQL
3. MariaDB
4. Microsoft SQL Server
5. Oracle
6. PostgreSQL

Q2: Where are RDS backups stored?

Ans:

Backups are stored in Amazon S3. For more information about backup storage costs, see Amazon RDS pricing. If you chose to retain automated backups when you delete a DB instance, the automated backups are saved for the full retention period.

Q3: Is RDS A PaaS?

Amazon RDS is SaaS, not PaaS. It's a fully managed service; it just turns out that users are often software developers. It's not a platform with an integrated stack where one can develop and/or deploy applications, so not PaaS. Google photos is SaaS for people who take photos. You can access Google Photos via an API, so having an end-user facing app is not the deterministic criterion IMO.