

EXPERIMENT NO. 8

Aim :- To study and implement Database as a Service on SQL databases using AWS RDS.

Theory :-

DBaaS :-

Also known as managed database service, is a cloud computing service that lets users access and use a cloud database system without purchasing and setting up their own hardware, installing their own database software, or managing the database themselves (not to mention hiring the high-priced talent required to do so). The service is typically delivered over the internet, and users can access it through a web-based interface or API.

Some DBaaS service providers :-

- 1> Amazon Web Services (AWS)
- 2> Microsoft Azure
- 3> Google Cloud Platform
- 4> IBM Cloud
- 5> Oracle Cloud
- 6> Alibaba Cloud

Amazon RDS is a Relational Database Service.

RDS	Aurora
Amazon RDS supports various relational database engines, like MySQL, PostgreSQL, etc.	Aurora is a MySQL compatible relational database engine developed by AWS.
It makes use of Amazon EBS for storage purposes.	It utilizes a distributed and replicated storage system.
Vertical Scaling by upgrading instance types.	Horizontal scaling with several read replicas.

Endpoint: An endpoint represents the network address or connection point for a database instance.

Snapshot: A snapshot is a point-in-time copy of a database instance's data, capturing entire dataset.

Read Replica: It refers to a duplicate, read-only copy of a source database instance.

Conclusion :-

I understood and was able to successfully implement MySQL database as a service on SQL databases using Amazon RDS.

A.T.
CF
15/3/24

Rishab Mandal
2103110
C23

Cloud Computing Experiment 8

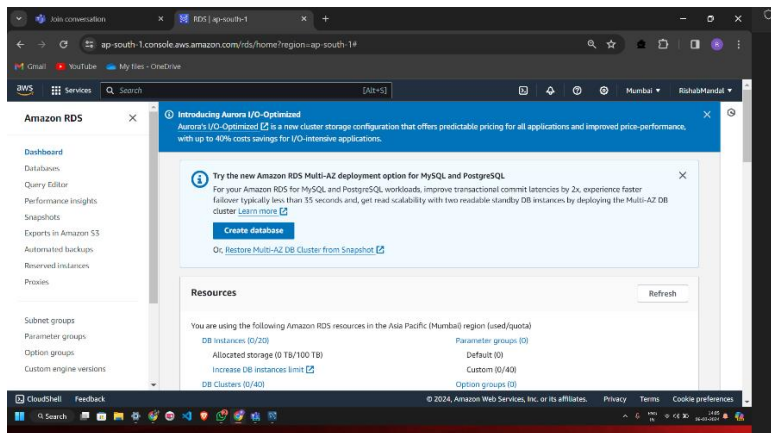
Aim: To study and Implement a Database as a Service on SQL databases Using AWS RDS.

Theory:

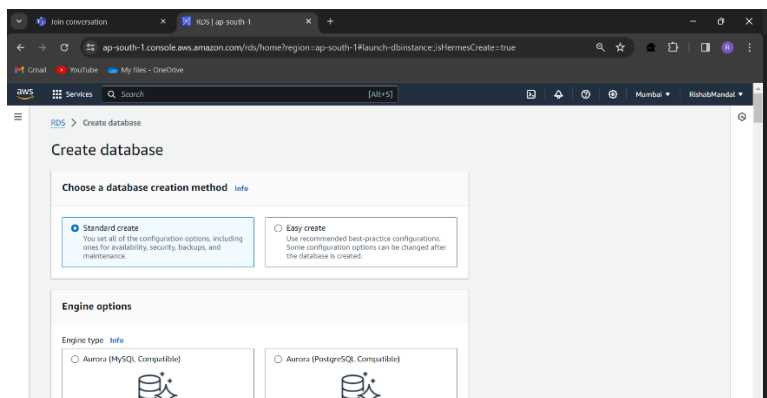
Implementing a Database as a Service (DBaaS) using Amazon Web Services (AWS) Relational Database Service (RDS) involves leveraging AWS's managed database service to deploy and operate SQL databases in the cloud without the need for managing the underlying infrastructure. AWS RDS supports various SQL database engines such as MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB, providing users with flexibility and scalability. By utilizing RDS, users can easily provision, configure, and scale database instances based on their application requirements, while AWS handles tasks like backups, software patching, monitoring, and maintenance. This approach not only reduces the operational overhead associated with managing databases but also ensures high availability, durability, and security of the data. Additionally, AWS RDS offers features like automated backups, multi-AZ deployment for high availability, read replicas for scalability, and encryption at rest and in transit, making it an ideal choice for deploying SQL databases in a cloud-native environment.

Steps for creation and usage of database as a service :-

Step1 : Login to aws console and search RDS

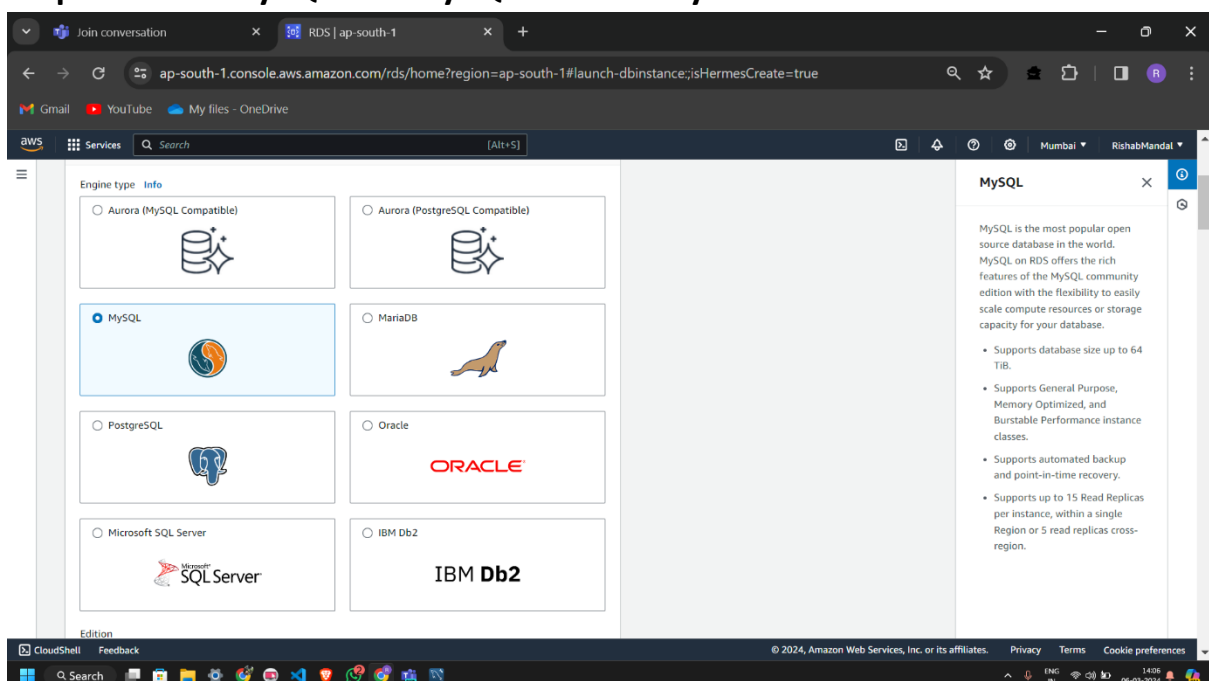


Step2: Click on to RDS and create database

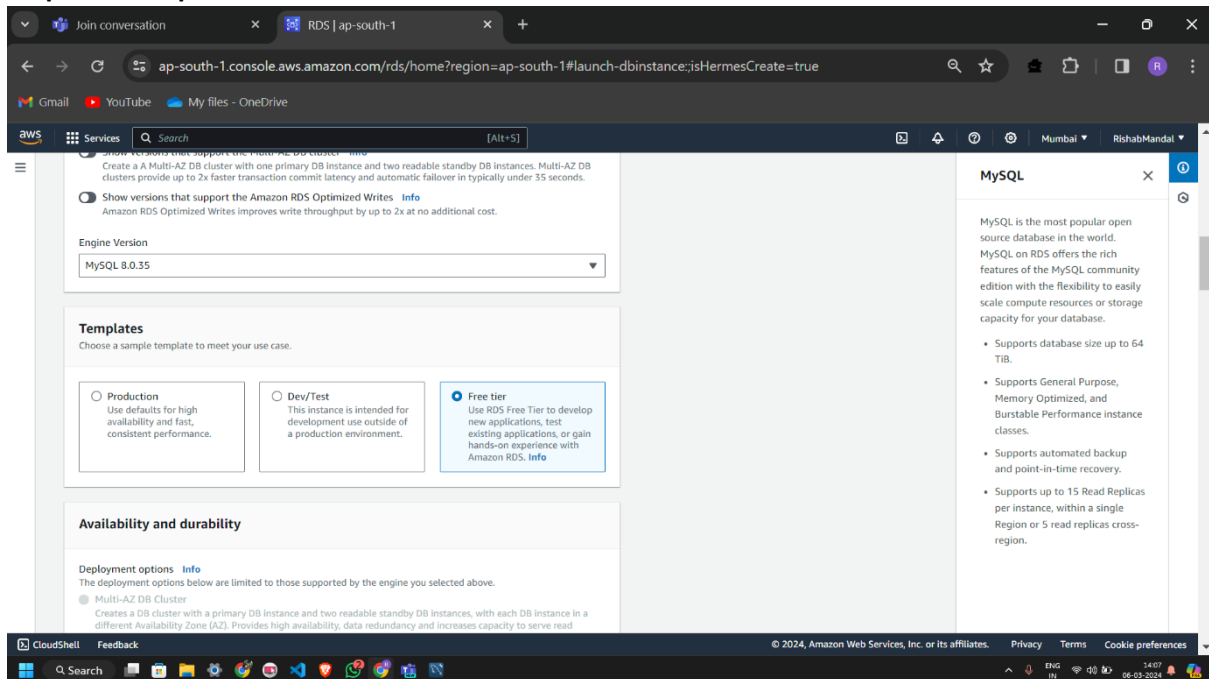


Step 3: Select standard database

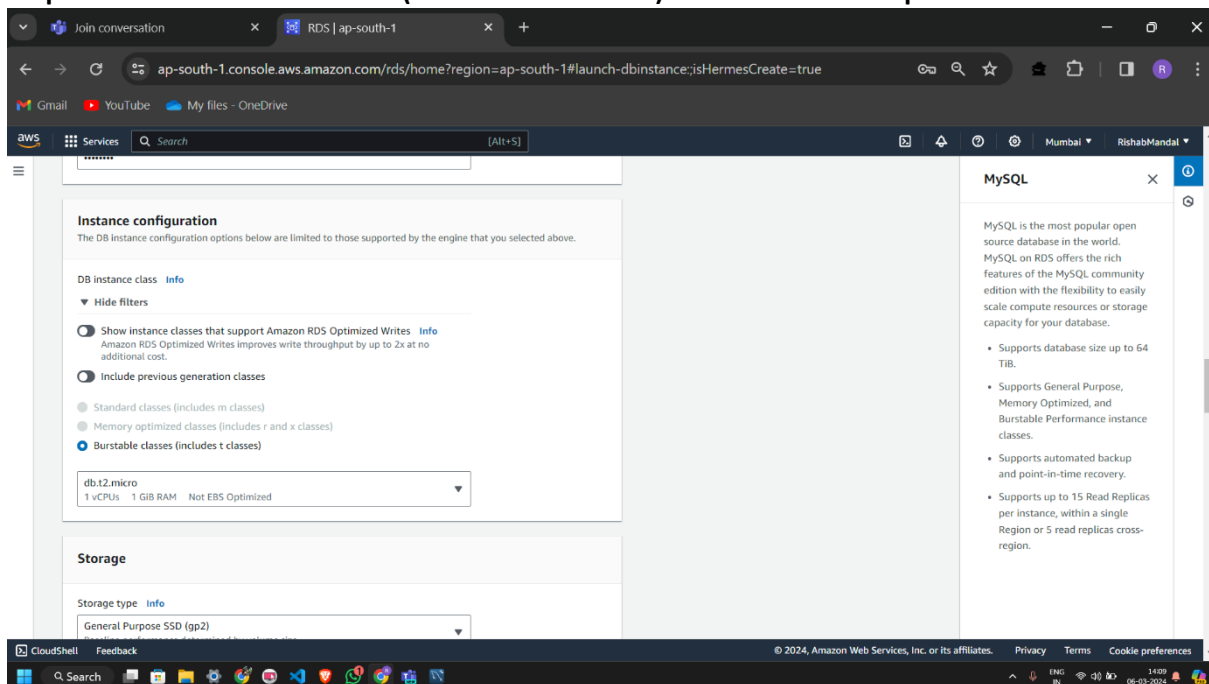
Step 4: Select MySQL and MySQL Community edition



Step 5: In Templates select Free tier



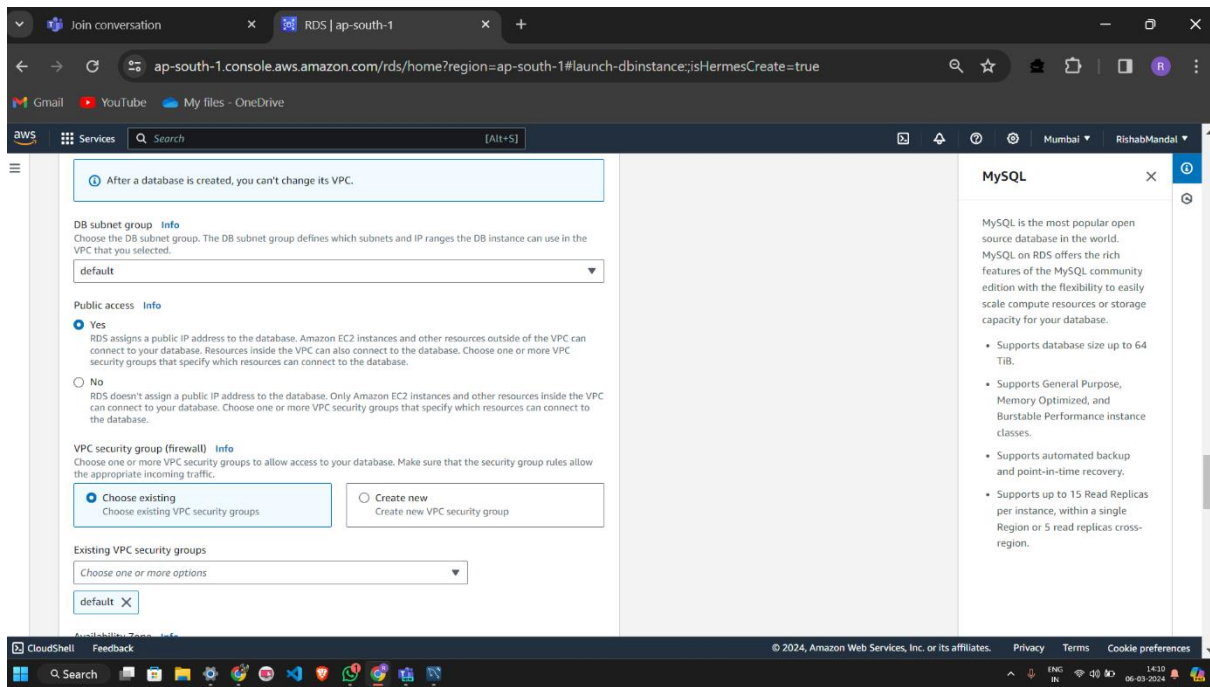
Step 6: Mention database name (default is database1) and username and password



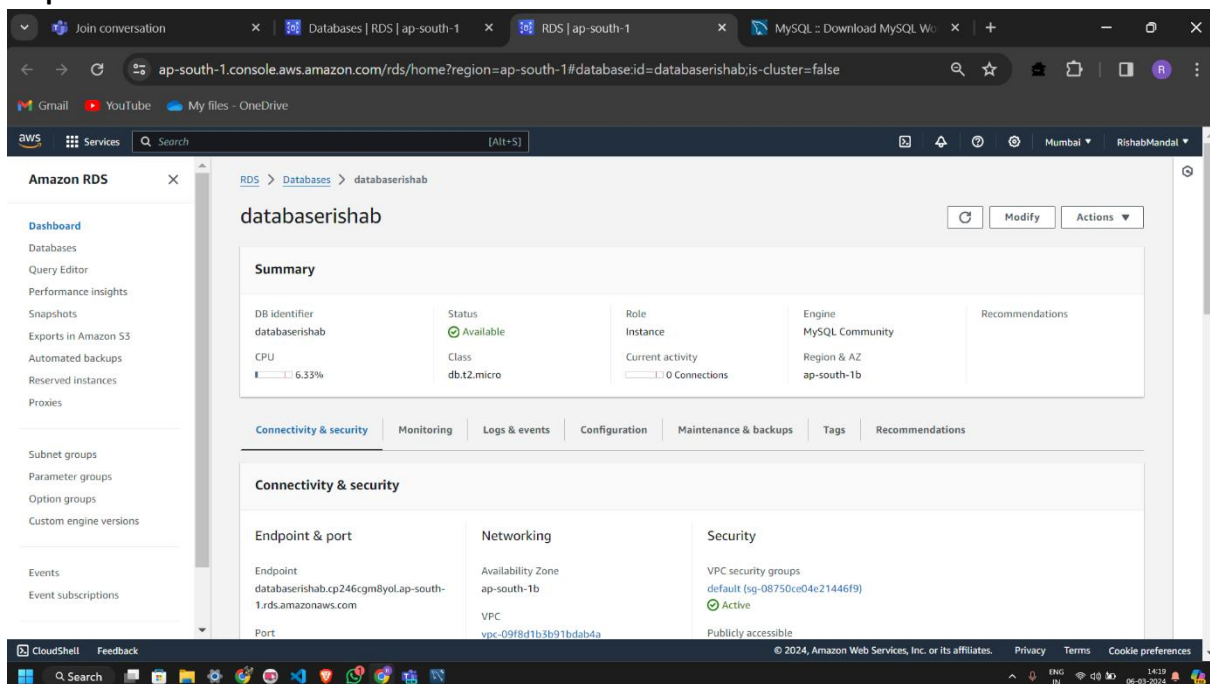
Step 7: Instance is t2.micro

Rest of things keep default

Step 8: Select Public Access -Yes



Step 9: Click on to create Database



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-08...

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sg-0b7241691956aa716	All traffic	All	All	Custom		Delete
-	All traffic	All	All	Anywh...		Delete

Rules with source of 0.0.0.0/0 or ::0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Share this window Cancel Preview changes Save rules

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SecurityGroups:

EC2 Dashboard
EC2 Global View
Events
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager

Inbound security group rules successfully modified on security group (sg-08750ce04e21446f9 | default)

Security Groups (2)

Name	Security group ID	Security group name	VPC ID	Description
-	sg-02d39f81849fa8224	launch-wizard-1	vpc-09f8d1b3b91bdab4a	launch-wizard-1 created 2024-02-
-	sg-08750ce04e21446f9	default	vpc-09f8d1b3b91bdab4a	default VPC security group

Share this window

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

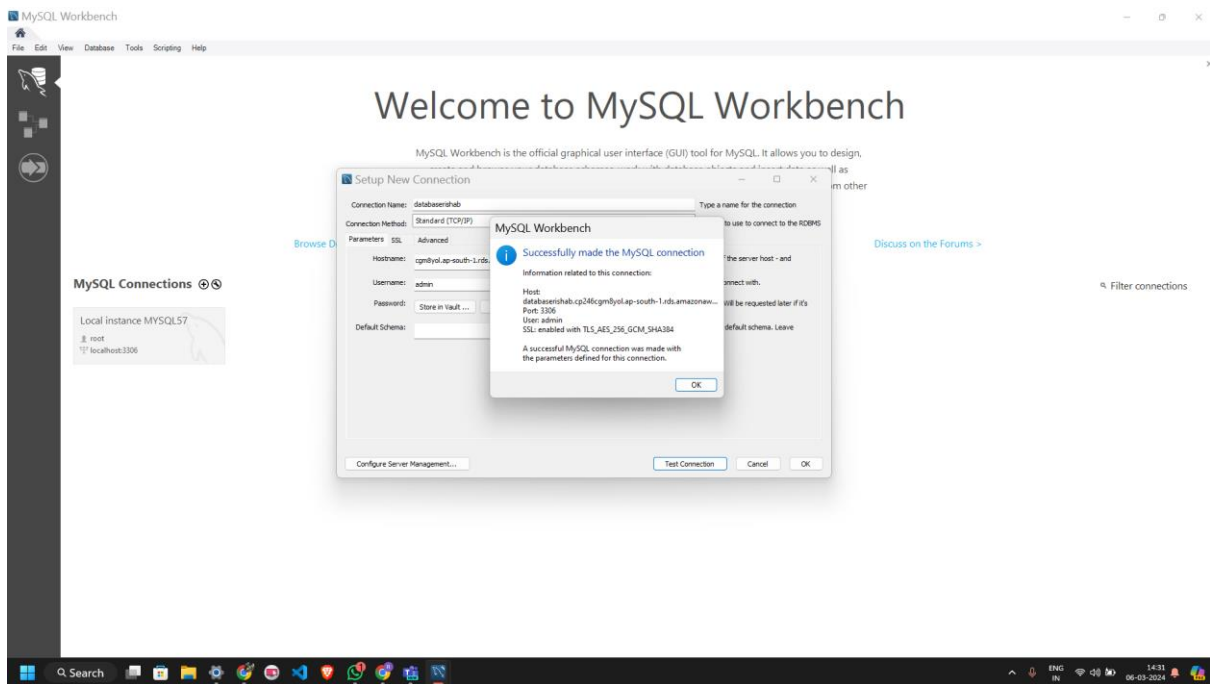
Click on to MySQL connection

Paste copied endpoint in Hostname

Connection Name: databaseRishab

Username: admin

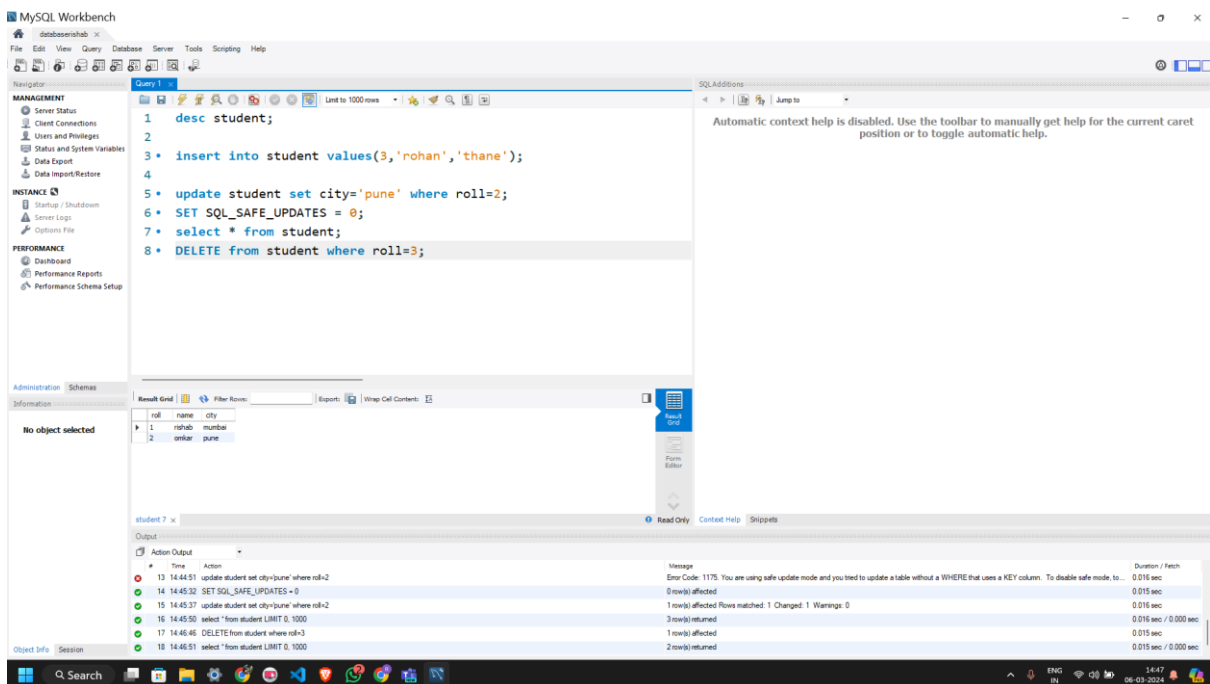
Click on to Test Connection



Click on Ok button

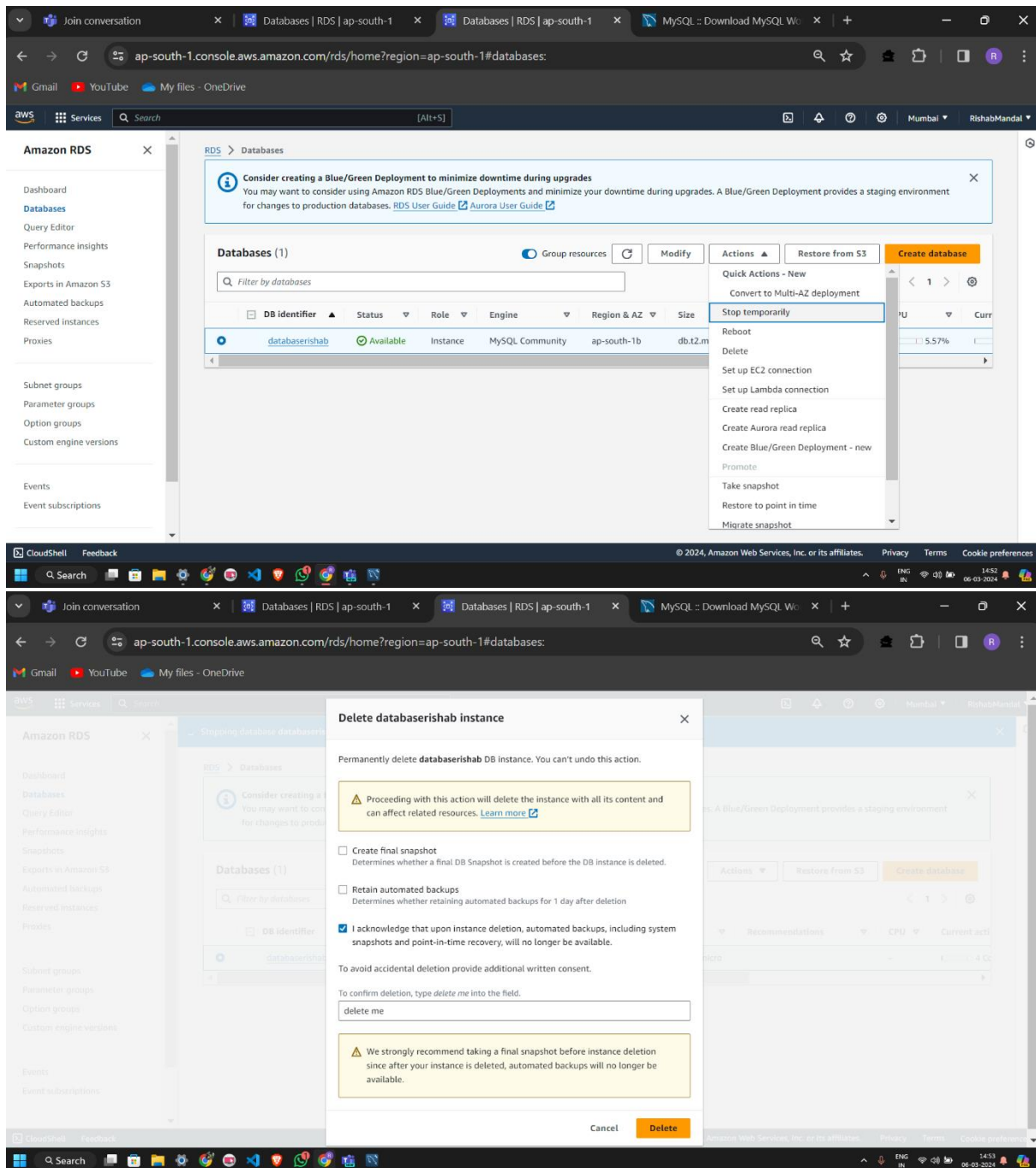
Go to workbench double click on connection,

Write query and execute



Now delete the instance (once you have done with it)

Select instance go to action stop instance and then delete instance



Conclusion:

In conclusion, leveraging AWS RDS for Database as a Service offers a streamlined approach to deploying and managing SQL databases in the cloud, providing scalability, reliability, and security for applications relying on SQL data storage.