



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment - 9

**Student Name:** Suhani Rawat

**UID:** 23BAI70070

**Branch:** BE-AIT-CSE

**Section/Group:** 23AIT\_KRG-G1\_A

**Semester:** 5th

**Date of Performance:** 29 Oct, 2025

**Subject Name:** ADBMS

**Subject Code:** 23CSP-333

### 1. Aim:

To understand and implement the creation and management of a relational database instance using **AWS RDS**, including setting up database connectivity (via pgAdmin or MySQL client), configuring security groups, and comparing RDS with EC2 database setups in terms of scalability, performance, and manageability.

### 2. Objective:

- To study the concept and features of **Amazon Web Services (AWS) Relational Database Service (RDS)**.
- To understand the **advantages of using RDS** over EC2 and on-premise database setups.
- To learn how to **create a database instance** on AWS RDS.
- To configure and manage **security groups** for secure database access.
- To learn how to **connect AWS RDS to local pgAdmin or MySQL client**.
- To explore various **RDS features** such as automated backups, monitoring, and scaling.
- To understand **Multi-AZ deployment, read replicas, and cross-region replication** for high availability.
- To gain hands-on experience in **launching and managing cloud-based databases** using AWS.

### 3. Theory:

Amazon Web Services (AWS) Relational Database Service (RDS) is a **managed cloud database service** that simplifies the setup, operation, and scaling of relational databases. It automates key administrative tasks such as provisioning, patching, backups, and monitoring, allowing developers to focus on application logic rather than infrastructure management.

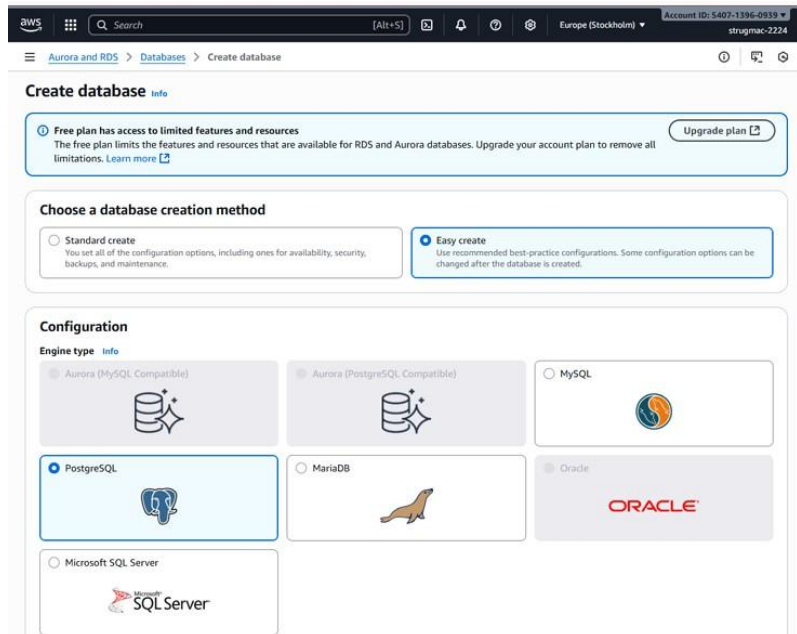
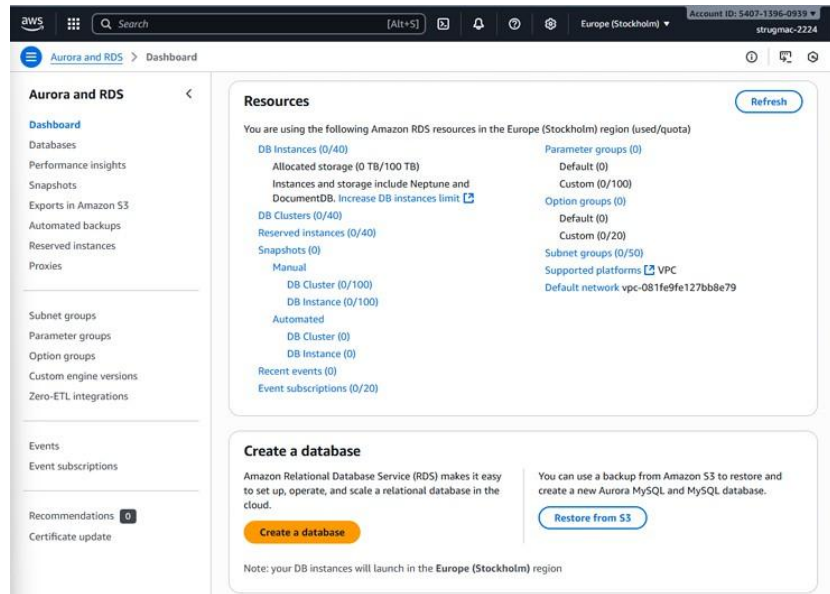
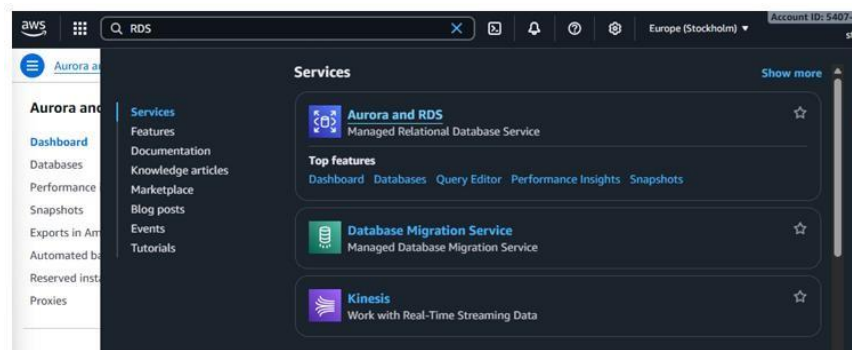
AWS RDS supports multiple database engines, including **MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server**, and provides features like **automated backups, multi-AZ deployment**, and **read replicas** to ensure high availability and reliability. Security is managed through **IAM, KMS encryption**, and **VPC security groups**, which protect databases from unauthorized access.

Additionally, AWS RDS integrates with **CloudWatch** for performance monitoring and offers **storage auto-scaling** to handle growing data needs efficiently. Compared to running databases on EC2 instances, RDS provides greater scalability, reduced administrative overhead, and enhanced performance, making it a cost-effective and reliable choice for cloud-based database management.

### 4. Procedure:

- Log in to the AWS Management Console using your credentials.
- Search for and open the **RDS** service from the AWS dashboard.
- Click on **Create Database** to start a new RDS instance setup.
- Choose the **Standard Create** option for manual configuration.
- Select the required **database engine** (MySQL or PostgreSQL).
- Enter the **DB instance name**, **master username**, and **password**.
- Choose the **instance class** and configure **storage settings**.
- Enable **storage auto-scaling** if needed.
- Configure **VPC** and **security groups** for database connectivity.
- Set the database to be **publicly accessible** (if connecting locally).
- Enable **automated backups** and optional **Multi-AZ deployment**.
- Review all settings and click **Create Database**.
- Wait for the instance status to become **Available** in the RDS dashboard.
- Copy the **endpoint** (host name) of the created database.
- Open **pgAdmin** or **MySQL Workbench** on your local machine.
- Create a new connection using the endpoint, username, and password.
- If connection fails, modify **inbound rules** in the security group to allow your local IP.
- Test the connection and perform basic SQL operations to verify setup.

## 5. Output:



DB instance size

Production

db.r7g.xlarge

4 vCPUs

32 GiB RAM

400 GiB

1.946 USD/hour

Dev/Test

db.r7g.large

2 vCPUs

16 GiB RAM

200 GiB

0.278 USD/hour

Free tier

db.t4g.micro

2 vCPUs

1 GiB RAM

20 GiB

0.019 USD/hour

DB instance identifier

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.

strugmac-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.

Master username

Info

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

admin

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

Very strong

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

Confirm master password

Info

\*\*\*\*\*

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use Standard create.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-081fe9fe127bb8e79)	No
Multi-AZ	No	Yes
Option group	default:mysql-8-0	Yes
Subnet group	Create new DB Subnet Group	Yes
Automatic backups	Enabled	Yes
VPC security group	default	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	strugmac-DB	Yes
DB engine version	8.0.42	Yes
DB parameter group	default.mysql8.0	Yes
Monitoring type	Database Insights - Standard	Yes
Performance insights	Not enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes

Aurora and RDS

Databases

Dashboard

Databases

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Creating database strugmac-db

Your database might take a few minutes to launch. You can use settings from strugmac-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

Filter by databases

DB identifier

Status

Role

Engine

strugmac-db

Creating

Instance

MySQL Co...

Console Home

makeplications

All services

All services

Services by category

Compute

EC2

Internal

Livefulg

Builds

Eneral Benestalk

APY Slght Application Repository

ARB Durienos

BDV Chdel Deliter

APJ T,quaimization

ARA Sabtcaroe Measer

Vinural Jrinüqating Service

Amkational View

Containers

Machine Learning

Amazon Supplistion AI

Amazon Dssigization AI

Amazon Confqcon

Amazon Services View

Amazon DevelopDiver

Amazon Mzanizeh

Amazon Cnual Detector

Amazon Fletillöy

Amazon Foncervillies

Amazon Engly

Amazon Transigation

Amazon Transide

Amazon Transpiree

Amazon Translors

EC2

Dashboard

AMB adunut View

Events

Instances

Instance

Instance Types

Launch Templates

Sych Resourcee

Savings State

Black laft Instances

Dreftlanine lxxardo,

Creeting Reservations

Images

Benefits and features

EC2 offers ultimate scalability and control

Fully available comminceo pfectly to support virtually any workload. This service in bever ait the eltt

- Highest level of control of the entire technology stack, allowing full information in-Jeae:dithe:nsornes
- Vvutual with nuntth of confrmations.
- Vidiate coicolutent of up:rating systems to choose from including Unoer, Dovine ...:i ard marcto
- Global availability

Find out more about EC2

Launch a virtual server

Launch Instance

View das:hbord

dew stearch ruinrisutation

Get started ratorial

Additional actions

View execting stairgtings

Migrate a server

EC2

Dashboard

AMB adunut View

Events

Instances rate

Connect

Instance state

Actions

Launch instance

Thet travsuare by attribute of mgl feartir sensioding

All defaus

Name of	Instance ID	Instance state	Instance type	Status check
No Instances				

## Application and OS Images (Amazon Machine Image)

An AMI contains operating systems, application servers, and applications that run on your EC2 instance. If you're still unsure, visit [AWS documentation](#) for more information.

Search our full catalog including 1000s of applications and OS images



## Key pair (login) [info](#)

You can use a key pair to securely connect to your instance. If you don't have access to the selected key pair you launch the instance.

### Key pair recommended

Proceed without a key pair (Not recommended)

Default value ▶

## ▼ Network settings [info](#)

Edit

### Network [info](#)

vpc-65f01/eh72teb7h9

### Subnet [info](#)

No preference: (Default subnet in availability zone)

### Auto-assign public IP [info](#)

Enable

### Firewall (security groups) [info](#)

Amazon recommends that you create a security group that controls traffic to and from the instances. You can also use an existing security group.

☐ Create security group

☒ Select existing security group

### Common security groups [info](#)

Select security groups ▼

default - sg-0671ettb482425 ✕  
default

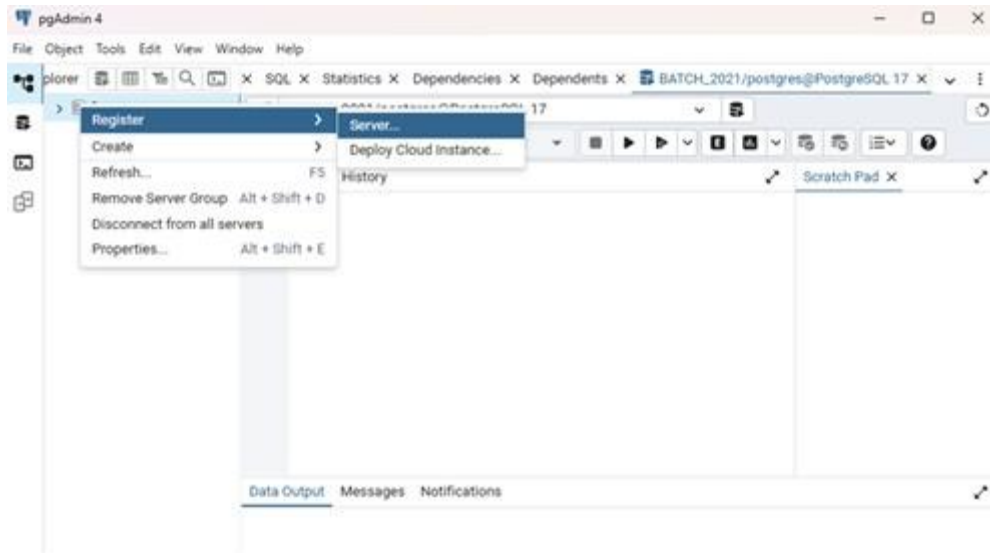
Security groups that you add or remove from will be added or removed from all your network interfaces.

## ▼ Summary

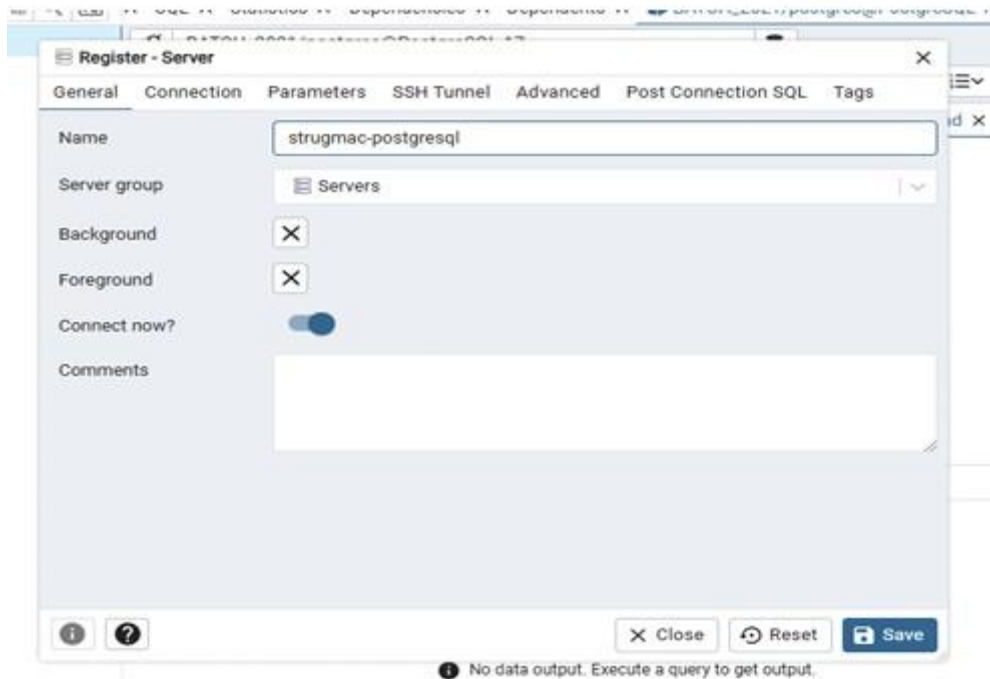
Number of instances: 1

Cancel

Launch instance



Copy the API Endpoints from the dashboard of AWS RDS Database instance.





Register - Server

General

Connection

Parameters

SSH Tunnel

Advanced

Post Connection

SQL

Tags

Host name/address

strugmac-postgresql.czqk2qqwqtc0.eu-north-1.rds.am

Port

5432

Maintenance database

postgres

Username

postgres

Kerberos authentication?

☐

Password

.....

In edit mode the password field is enabled only if Save Password is set to

Save password?

☐

Role

!

?

X Close

↺ Reset

💾 Save

!

No data output. Execute a query to get output.

Register - Server

Host name/address

strugmac-postgresql.czqk2qqwqtc0.eu-north-1.rds

Port

5432

Maintenance database

postgres

!

?

X Close

↺ Reset

↻ Save

OK

!

No data output. Execute a query to get output.

Connectivity & security

Endpoint & port

Endpoint

struginacan-postgresql  
Csa62qgwetcq.eu-north-1.rds.amazonaws.com

Port

5432

Networking

Availability Zone

eu-north-1c

VPC

vpc-081fefe127Dbb879

Subnet group

default-vpc-081fefeg127bb9e78

Subnets

subnet  
00b10747db8495492  
subnet  
0ao1f0688o7d8cecc  
subnet  
0f9ea32b6ab9688f28

Security

VPC security groups

default-tsg  
Ce987b74baff34225)

Publicly accessible

No

Certificate authority

Info  
caii-rs-rsa2048-g1

Certificate authority date

May 25, 2061, 03:20  
(UTC-03:30)

Edit inbound rules

Inbound rule Tab

Range

Port-range

Source - optional

Description - optional

ssg-08FF501010783ae0

All traffic

All

All

PodgesQL-v0

3462

53

Raw

236.51.100.173457

Delete

Add rule

Cancel

Preview changes

See table



## 6. Learning Outcomes:

- Understood the concept and functionality of **Amazon RDS (Relational Database Service)**.
- Learned how to **create and configure a database instance** on AWS RDS.
- Gained practical knowledge of **connecting AWS RDS with local pgAdmin/MySQL Workbench**.
- Understood how to **set up and modify VPC security groups** to manage inbound and outbound traffic.
- Learned to **edit inbound rules** to allow database access from specific IP addresses securely.
- Understood the importance of **network configuration and security** in cloud database management.
- Learned to **troubleshoot common connectivity errors** like “connection timeout expired.”
- Gained experience in **launching and configuring EC2 instances** for database connectivity.
- Understood **differences between databases on EC2 and AWS RDS** in terms of scalability and maintenance.
- Developed the ability to **deploy, manage, and secure cloud-based relational databases** effectively.