



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

EXPERIMENT- 09

Student Name: Skand Kumar

Branch: BE-CSE

Semester: 05

Subject Name: ADBMS

UID: 23BCS12703

Section/Group: KRG-1A

Date of Performance: 30/10/25

Subject Code: 23CSP-333

- Aim:** To create and connect a PostgreSQL database instance on **Amazon RDS** (**Relational Database Service**)

2. Objective:

- To understand the steps involved in launching a database instance using Amazon RDS.
- To configure a database for public access and connect it with a local client (pgAdmin).
- To perform basic SQL operations (CREATE, INSERT, SELECT).

3. Tools / Software

- Amazon Web Services (AWS)
- PostgreSQL
- pgAdmin 4
- RDS (Relational Database Service)

4. Program:

Step 1: Create and Configure Database Instance

- Login to AWS Console → RDS → Create database, select Standard create and PostgreSQL under the Free Tier template.
- Set DB identifier: ruchi-db, Username: postgres, choose db.t3.micro, 20 GB gp2 storage,

The screenshot shows the AWS RDS Databases page. On the left, there's a sidebar with options like Aurora and RDS, Databases, Query editor, Performance insights, Snapshots, and Events. The main area is titled 'Databases (1)' and shows a table with one row for 'ruchi-db'. The columns in the table are DB identifier, Status, Role, Engine, Region ..., and Size. The 'ruchi-db' row has 'Config...', 'Instance', 'PostgreSQL', 'eu-north-1a', and 'db.t4g.micro'. There are also 'Group resources', 'Modify', 'Actions', and 'Create database' buttons at the top right of the table area.

and enable Public access.

- Click Create database and wait until the status shows Available in the RDS dashboard.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Step 2: Configure Security Group (Allow Local Access Only)

1. In AWS Console → go to RDS → Databases → click your DB (ruchi-db).
2. Open the Connectivity & Security tab.
3. Under VPC security groups, click the linked group name (it opens EC2 security groups).
4. Click Edit inbound rules → Add rule
 - >Type: PostgreSQL
 - Protocol: TCP
 - Port: 5432
 - Source: My IP
5. Click Save rules.

The screenshot shows the AWS RDS Security Groups console. The top navigation bar has the text "sg-0570f959421927738 - default". Below it, there's a search bar and a table titled "Inbound rules (2)". The table columns are: Name, Security group rule ID, IP version, Type, Protocol, Port range, and Source. The data rows are:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0d39d1bf593210da4	IPv4	PostgreSQL	TCP	5432	106.206.235.43
-	sgr-0ee4f18536cb88772	-	All traffic	All	All	sg-0570f95942

Step 3: Connect Database Using pgAdmin

1. Open pgAdmin 4 on your local system.
2. Right-click Servers → Create → Server.
3. Under the General tab, enter the name: postgres.
4. Under the Connection tab, fill in the following details:
 - Host name/address: ruchidb.xxxxxxx.rds.amazonaws.com
 - Port: 5432
 - Username: postgres
 - Check Save password.
5. Click Save to connect your RDS PostgreSQL database.

The screenshot shows the pgAdmin 4 interface. The left sidebar shows a tree structure under "Default Workspace":

- Servers (2)
 - PostgreSQL 17
 - ruchi-db
 - Databases
 - Login/Group Roles
 - Tablespaces