



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

EXPERIMENT- 09

Student Name: Dushyant Singh Chauhan

UID:23BCS11986

Branch: BE-CSE

Section/Group: KRG 3(B)

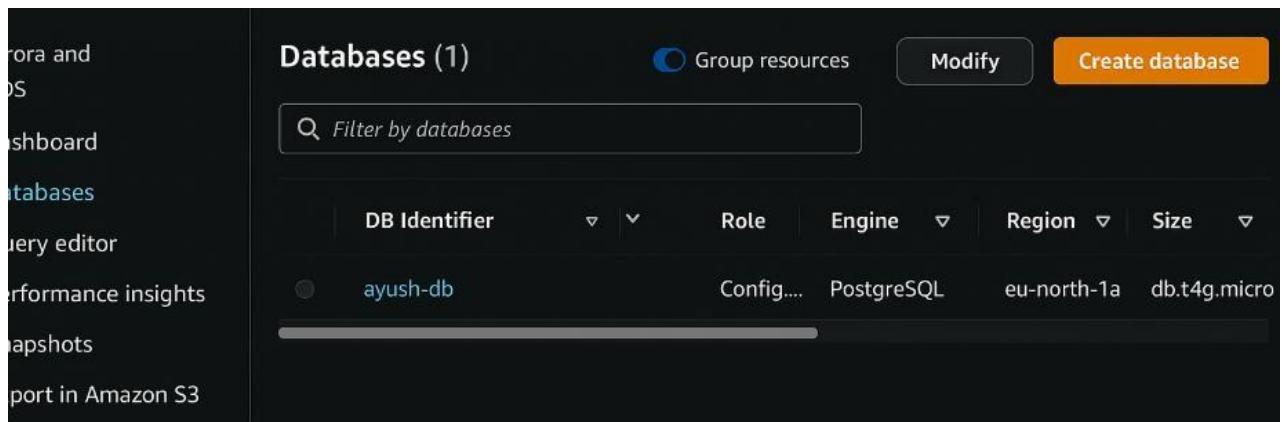
Semester: 05

Date of Performance: 30/10/25

Subject Name: ADBMS

Subject Code: 23CSP-333

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



The screenshot shows the Amazon RDS console interface. On the left, there's a sidebar with navigation links: Home, Databases, Dashboard, Metrics, Query editor, Performance insights, Snapshots, and Export in Amazon S3. The main area is titled 'Databases (1)' and contains a table with one row. The table columns are DB Identifier, Role, Engine, Region, and Size. The data row shows 'ayush-db' as the DB Identifier, 'Config....' as the Role, 'PostgreSQL' as the Engine, 'eu-north-1a' as the Region, and 'db.t4g.micro' as the Size. There are buttons for 'Group resources', 'Modify', and 'Create database' at the top right of the table area. A search bar labeled 'Filter by databases' is also present.

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

1. Login to AWS Console → RDS → Create database, select Standard create and PostgreSQL under the Free Tier template.
2. Set DB identifier: ruchi-db, Username: postgre, choose db.t3.micro, 20 GB gp2 storage, and enable Public access.
3. 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.

Inbound rules (2)						
	Name	Security group rule ID	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0d39d1bf593210da4	IPv4	PostgreSQL	TCP	5432
<input type="checkbox"/>	-	sgr-0ee4f18536cb88772	-	All traffic	All	106.206.235.43 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: postgre.
4. Under the Connection tab, fill in the following details:
 - Host name/address: ruchi- dbxxxxxxxx.rds.amazonaws.com
 - Port: 5432
 - Username: postgre
 - Check Save password.
5. Click Save to connect your RDS PostgreSQL database.

Default Workspace

The screenshot shows a software interface for managing a PostgreSQL database. At the top, there's a toolbar with icons for file operations, search, and export. Below the toolbar is a navigation pane titled "Default Workspace". The pane lists "Servers (2)" and "PostgreSQL 17". Under "PostgreSQL 17", a database named "ayush-db" is selected, indicated by a blue highlight bar. The "ayush-db" node has three children: "Databases", "Login/Group Roles", and "Tablespaces". Each item has a corresponding icon next to it.

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