



# CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

## EXPERIMENT - 9

**Student Name:** Tanay Manish Nesari

**UID:** 23BCS13761

**Branch:** BE-CSE

**Section/Group:** KRG 1-B

**Semester:** 5th

**Date of Performance:** 21/10/2025

**Subject Name:** ADBMS

**Subject Code:** 23CSP-333

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

**Objective:** To understand the steps involved in launching a database instance using Amazon RDS, and configure a database for public access and connect it with a local client (pgAdmin).

### **Code:**

1. GO TO AWS HOMEPAGE -> CLICK ON SIGN IN-> ENTER USER NAME WITH EMAIL ADDRESS
2. AFTER SIGN-IN -> GO TO SEARCH BAR -> SEARCH FOR RDS -> HIT ENTER

A screenshot of the AWS Management Console homepage. The search bar at the top contains the text 'rds'. Below the search bar, there are two sections: 'Services' and 'Features'. The 'Services' section is expanded, showing three items: 'Aurora and RDS' (Managed Relational Database Service), 'Database Migration Service' (Managed Database Migration Service), and 'Kinesis' (Work with Real-Time Streaming Data). The 'Aurora and RDS' item is highlighted with a blue border. The 'Features' section is collapsed. On the right side of the screen, there is a sidebar with options like 'Reset to default layout', 'Add widgets', 'Info', 'Create application', and a search bar for 'Find applications'. Below the sidebar, there is a message: 'No applications. Get started by creating an application.' with a 'Create application' button.

### 3. To create database go to RDS Dashboard

The screenshot shows the AWS RDS Dashboard. On the left sidebar, there are several navigation options: Aurora and RDS, Dashboard, Databases, Query editor, Performance insights, Snapshots, Exports to Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations (0), and Certificate updates. The main content area is titled 'Resources' and displays various RDS metrics: DB Instances (0/400), DB Clusters (0/40), Reserved Instances (0/40), and Snapshots (0). It also lists Parameter groups (1), Subnet groups (1/50), and Supported platforms (2 - VPC). A 'Create a database' section is present, along with a 'Service health' status bar indicating the service is operating normally.

### 4. CLICK ON CREATE DATABASE.

The screenshot shows the 'Create database' page. At the top, there are two creation methods: 'Standard create' (selected) and 'Easy create'. Below this, the 'Engine options' section allows selecting an engine type: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible) (selected), MySQL, PostgreSQL, MariaDB, Oracle, Microsoft SQL Server, and IBM Db2. Under 'Engine version', it says 'View the engine versions that support the following database features.' and includes filters for 'Show only versions that support the Babelfish for PostgreSQL feature' and 'Show only versions that support Aurora Limitless Database'. The bottom of the page includes standard footer links: CloudShell, Feedback, © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

IN THE STANDALONE CREATE, WE CAN SET EVERYTHING FOR OUR DATABASE, THE INCOMING TRAFFIC, IP ADDRESSES TO BE USED, BACKUP ETC.

5. Select PostgreSQL and add configurations.

The screenshot shows the 'Create database' step in the AWS RDS console. The 'DB instance size' section is expanded, showing three options: 'Production' (db.r7.large), 'Dev/Test' (db.r7.large), and 'Free tier' (db.t1g.micro). The 'Free tier' option is selected and highlighted with a blue border. The 'DB instance identifier' field contains 'database-1'. The 'Master username' field is set to 'postgres'. Under 'Credentials management', the 'Self managed' option is selected, which is also highlighted with a blue border. A note below it says 'Create your own password or have RDS create a password that you manage.' The 'Master password' field is empty and has a placeholder '(Required)' with a note: 'The Master password field is required. Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ ^ @'. The top right corner shows the account ID '4429-4254-2666' and the user 'Anshpreet Singh'.

6. GET CONNECTIONS URL AND CONNECT LOCAL PG ADMIN WITH CLOUD DB USING URL AND PASSWORD

The screenshot shows the 'Register - Server' dialog in DBeaver. The 'Connection' tab is selected. The 'Host name/address' field contains 'strugmac-postgresql.czqk2qqwqtc0.eu-north-1.rds.amazonaws.com'. The 'Port' field is set to '5432'. The 'Maintenance database' field is 'postgres'. The 'Username' field is 'postgres'. The 'Kerberos authentication?' checkbox is unchecked. The 'Password' field is masked with '\*\*\*\*\*'. A note at the bottom states: 'In edit mode the password field is enabled only if Save Password is set to true.' The top right corner shows the 'Save' and 'Cancel' buttons.