

## Worksheet 9

**Student Name:** Vishal Saini

**Branch:** CSE

**Semester:** 5th

**Subject Name:** ADBMS

**UID:** 23BCS10163

**Section/Group:** KRG 2-A

**Date of Performance:** 30/10/2025

**Subject Code:** 23CSP-333

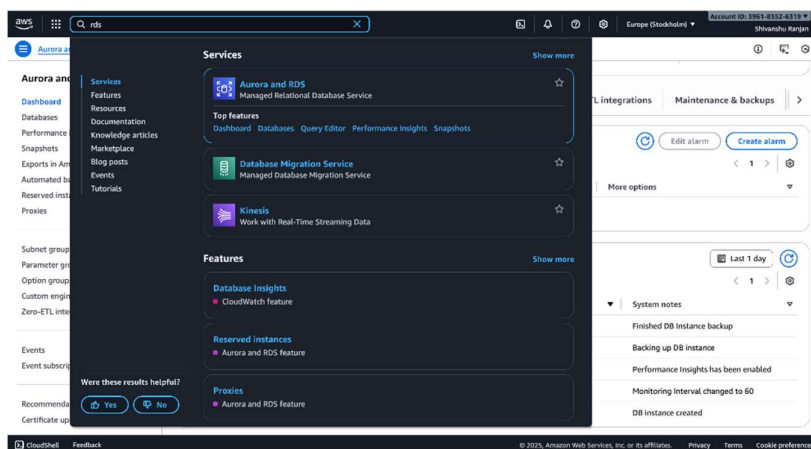
**1. Aim:** To understand and implement the setup of Amazon Relational Database Service (AWS RDS) by creating a database instance, configuring security groups, and establishing a secure connection between the local pgAdmin tool and the RDS instance hosted on the AWS Cloud.

### **2. Objective:**

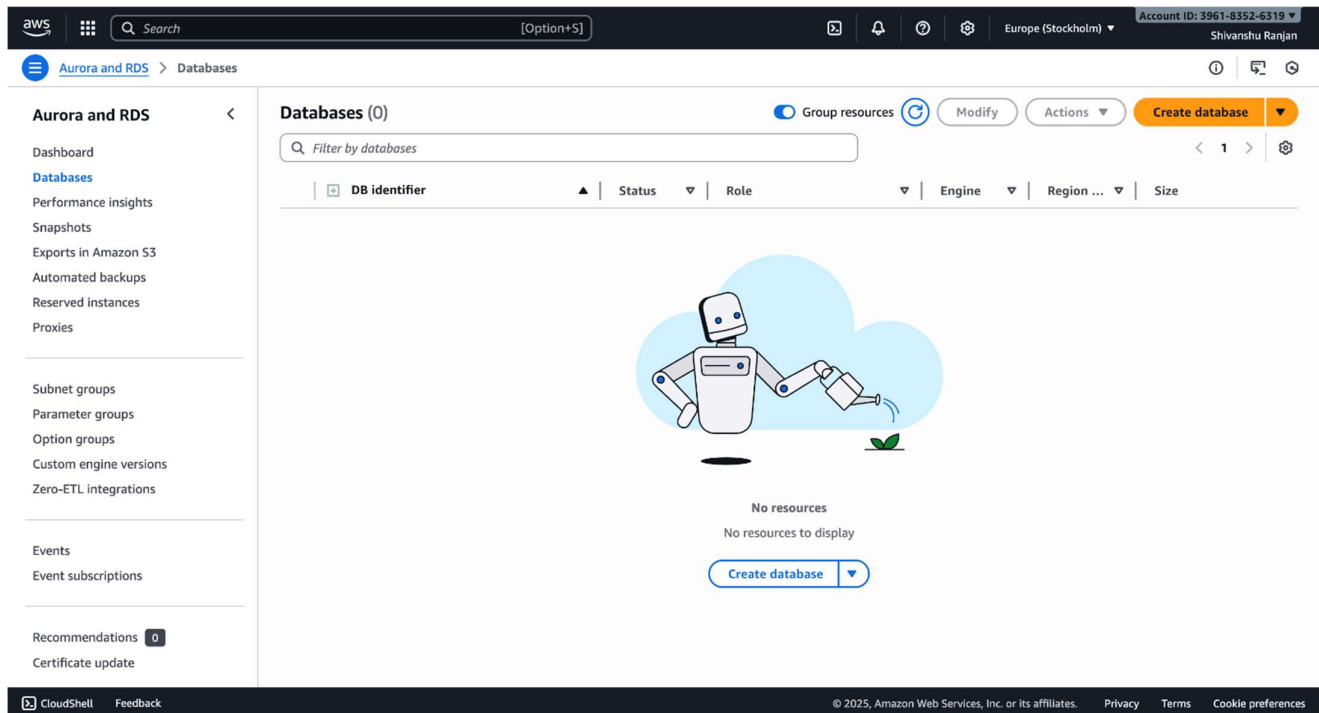
- To learn the basic concepts and features of Amazon Relational Database Service (AWS RDS).
- To create and configure a new RDS database instance on the AWS Management Console.
- To understand the role and configuration of security groups for controlling database access.
- To connect a local pgAdmin client to the AWS RDS instance securely using proper credentials and endpoint details.
- To verify successful database connectivity and perform basic operations through pgAdmin.

### **3. Code & Output:**

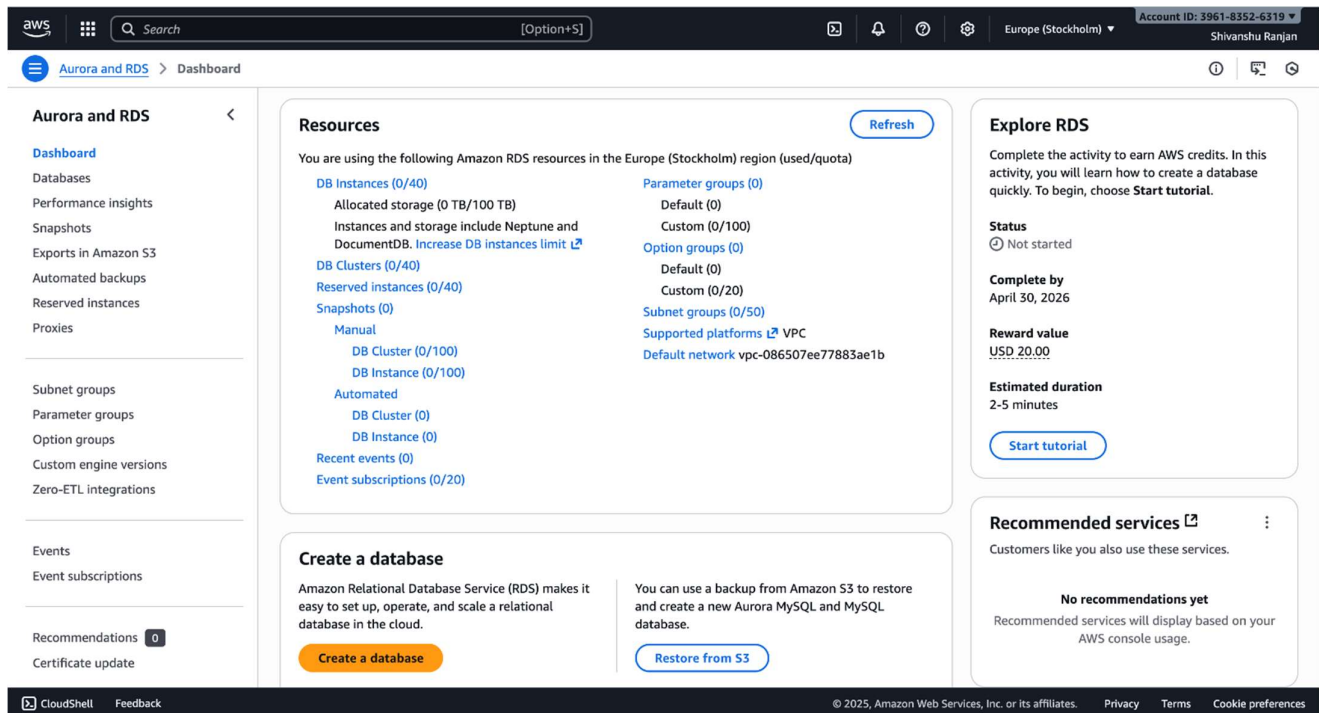
#### **1. Sign-in**



## 2. Navigating to RDS Service



## 3. Amazon RDS Dashboard Overview





## 4. Creating a New Database Instance

**Create database** Info

Free plan has access to limited features and resources  
The free plan limits the features and resources that are available for RDS and Aurora databases. Upgrade your account plan to remove all limitations. [Learn more](#)

Upgrade plan

**Choose a database creation method**

☐ Standard create  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

☒ Easy create  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Configuration**

Engine type Info

☐ Aurora (MySQL Compatible)

☐ Aurora (PostgreSQL Compatible)

☐ MySQL

☒ PostgreSQL

☐ MariaDB

☐ Oracle

## 5. Selecting PostgreSQL as Database Engine

**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.

shivanshu-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.

postgres

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

\*\*\*\*\*

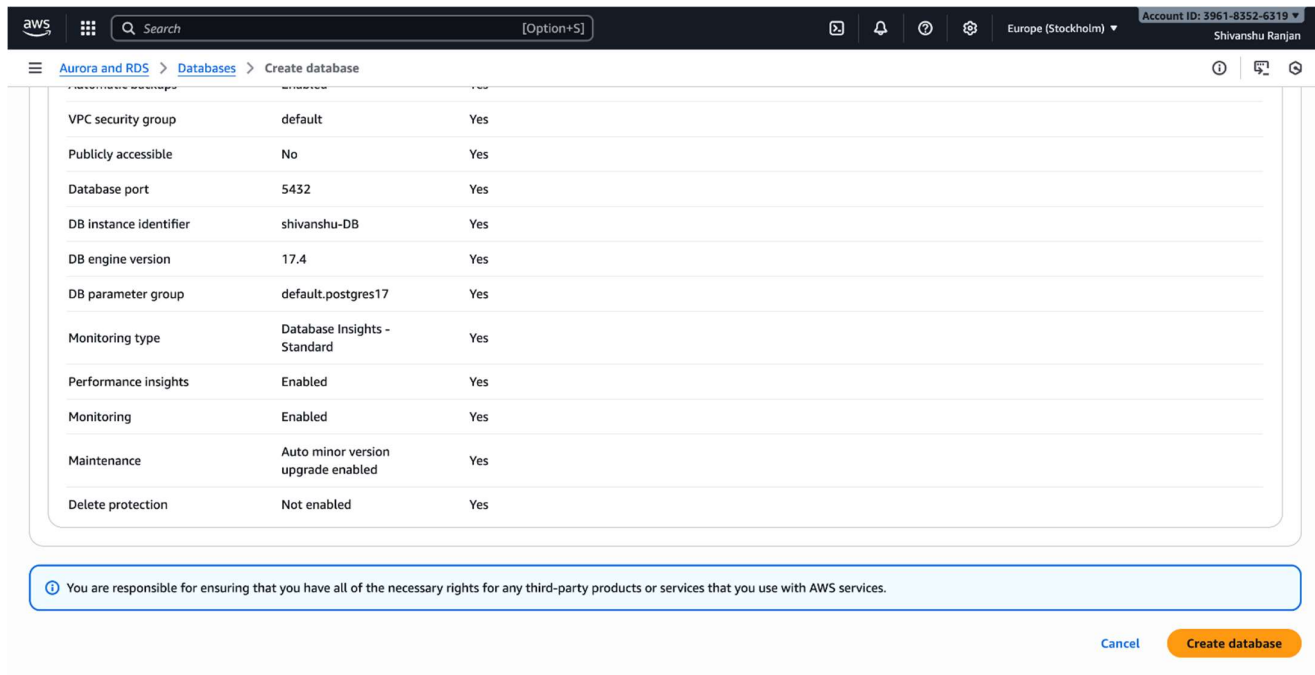
**Password strength** Neutral

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

**Confirm master password** Info

\*\*\*\*\*

## 6. Choosing Deployment Option and Template

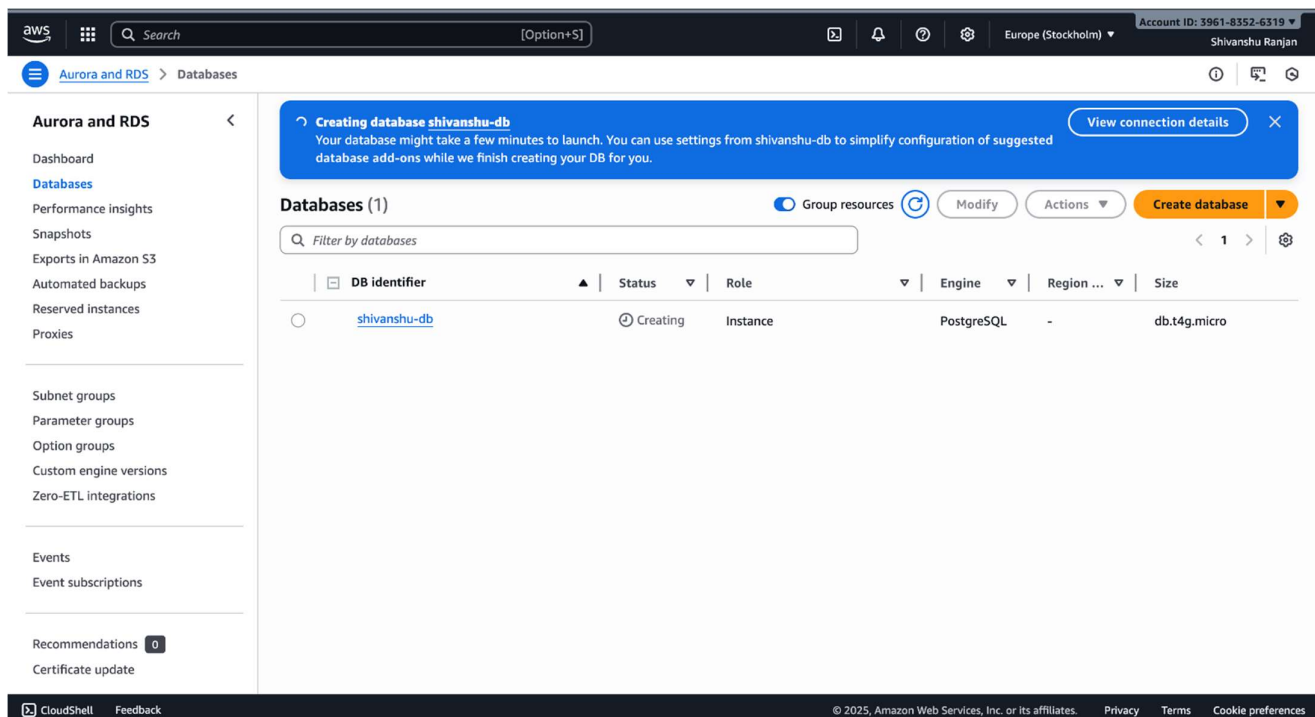


The screenshot shows the 'Create database' configuration page in the AWS Management Console. The configuration table is as follows:

Configuration Option	Value	Required
VPC security group	default	Yes
Publicly accessible	No	Yes
Database port	5432	Yes
DB instance identifier	shivanshu-DB	Yes
DB engine version	17.4	Yes
DB parameter group	default.postgres17	Yes
Monitoring type	Database Insights - Standard	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

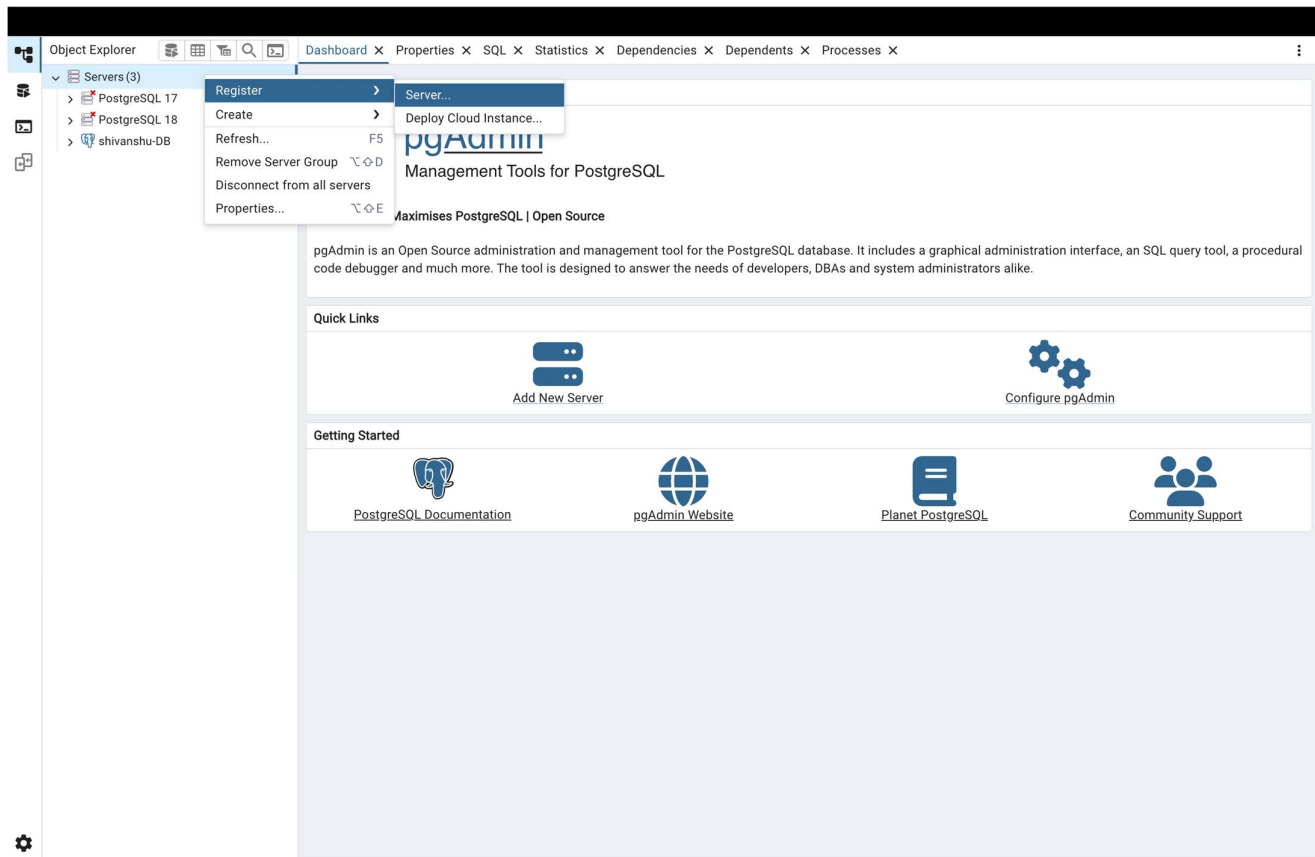
At the bottom, there is a blue box with a warning icon and text: "You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services." Below this box are two buttons: "Cancel" and "Create database".

## 7. Configuring Database Settings (Name, Username, Password)

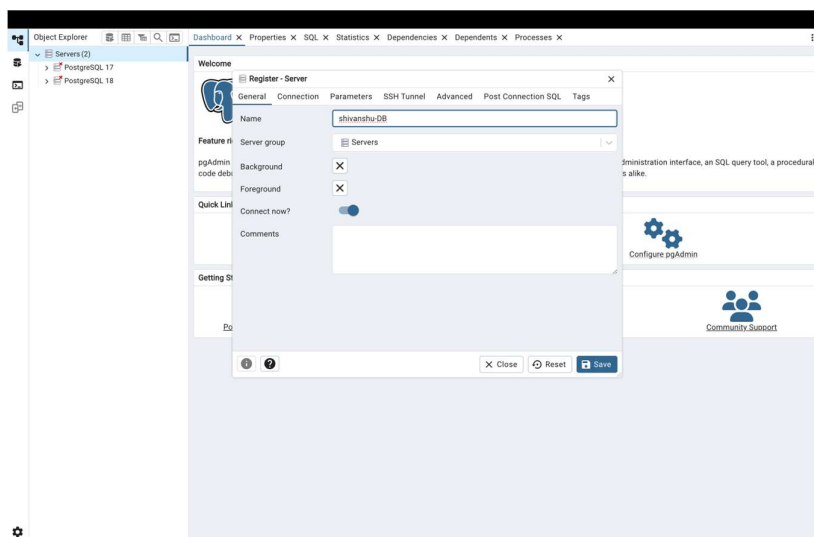


The screenshot shows the 'Databases' page in the AWS Management Console. On the left is a sidebar with navigation links for 'Aurora and RDS', 'Dashboards', 'Databases', 'Performance insights', 'Snapshots', 'Exports in Amazon S3', 'Automated backups', 'Reserved instances', 'Proxies', 'Subnet groups', 'Parameter groups', 'Option groups', 'Custom engine versions', 'Zero-ETL integrations', 'Events', 'Event subscriptions', 'Recommendations', and 'Certificate update'. The main content area shows a blue banner at the top with the text: "Creating database shivanshu-db. Your database might take a few minutes to launch. You can use settings from shivanshu-db to simplify configuration of suggested database add-ons while we finish creating your DB for you." Below the banner, there is a 'Databases (1)' section with a search bar and a table of databases. The table has columns for 'DB identifier', 'Status', 'Role', 'Engine', 'Region', and 'Size'. The table contains one entry: 'shivanshu-db' with status 'Creating', role 'Instance', engine 'PostgreSQL', region '-', and size 'db.t4g.micro'. At the bottom of the page, there is a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

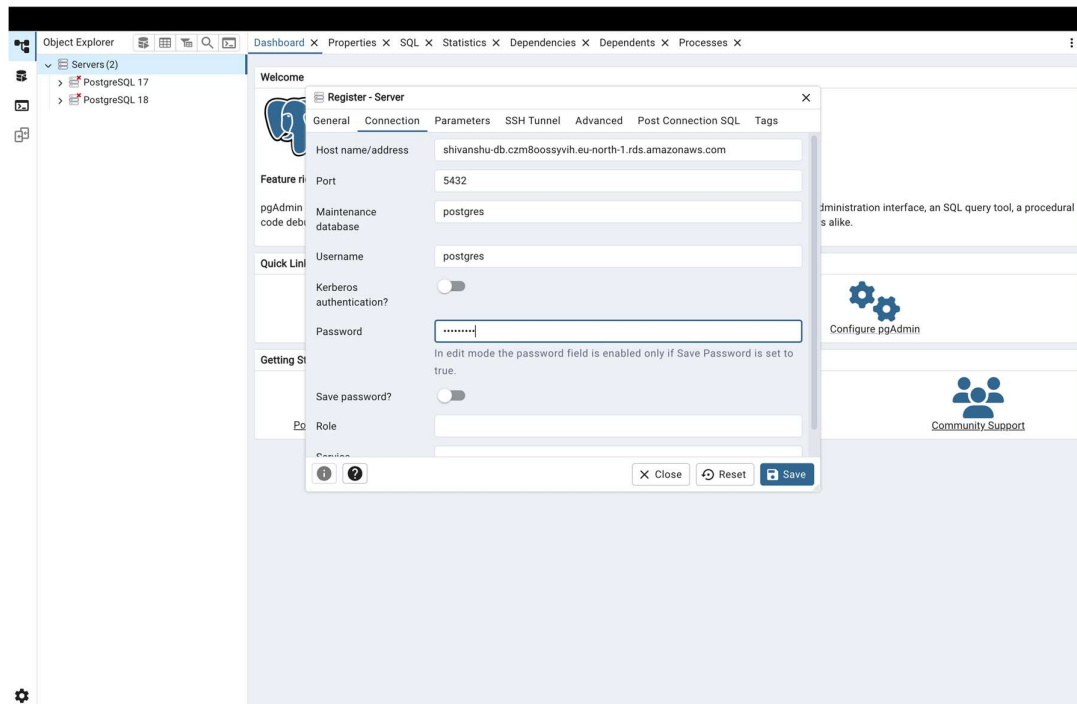
## 8. Setting Up Instance Size and Storage



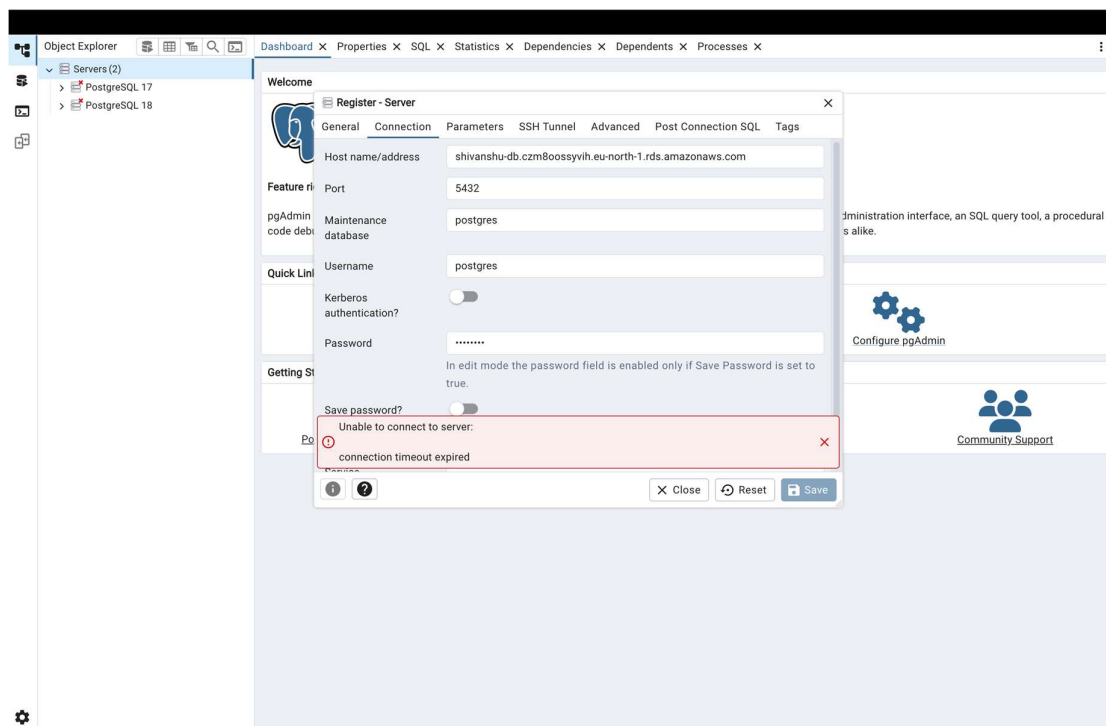
## 9. Configuring Connectivity and VPC Settings



## 10. Gr Setting Up Security Groups for RDS Access



## 11. Additional Database Configuration Options





## 12. Reviewing and Creating the Database Instance

The screenshot shows the AWS Management Console for an Amazon RDS instance named 'shivanshu-db'. The console is in the 'Europe (Stockholm)' region, and the user is 'Shivanshu Ranjan' with account ID '3961-8352-6319'. The instance is in the 'Summary' tab, showing a status of 'Available'. The DB identifier is 'shivanshu-db', the engine is 'PostgreSQL', and the region is 'eu-north-1a'. The CPU usage is 21.27%, and the class is 'db.t4g.micro'. The current activity shows 0.00 sessions. The 'Connectivity & security' tab is selected, showing the endpoint 'shivanshu-db.czm8oosyvi.h.eu-north-1.rds.amazonaws.com' on port 5432. The availability zone is 'eu-north-1a', and the VPC is 'vpc-086507ee77883ae1b'. The security group is 'sg-0b4c8dc4647072099', which is active and publicly accessible. The certificate authority is 'rds-ca-rsa2048-g1', and the certificate authority date is 'May 25, 2061, 03:29 (UTC+05:30)'. The DB instance certificate expiration is also shown.

## 13. RDS Instance Creation in Progress

The screenshot shows the 'Edit inbound rules' page for a security group in the AWS Management Console. The page is titled 'Edit inbound rules' and includes a description: 'Inbound rules control the incoming traffic that's allowed to reach the instance.' The security group rule ID is 'sgr-040a1d1889af5e91c'. The type is 'All traffic', the protocol is 'All', the port range is 'All', and the source is 'Custom'. The description is 'sg-0b4c8dc4647072099'. The 'Add rule' button is visible. The 'Preview changes' and 'Save rules' buttons are also present.



## 14. Viewing Database Instance Details

### ▼ Additional configuration

#### Public access

##### ☒ Publicly accessible

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

##### ☐ Not publicly accessible

No IP address is assigned to the DB instance. EC2 instances and devices outside the VPC can't connect.

#### Database port

Specify the TCP/IP port that the DB instance will use for application connections. The application connection string must specify the port number. The DB security group and your firewall must allow connections to the port. [Learn more](#)

5432

## 15. Copying the RDS Endpoint for Connection

### Connectivity & security

#### Endpoint & port

##### Endpoint

[shivanshu-db.czm8oosyvih.eu-north-1.rds.amazonaws.com](#)

##### Port

5432

#### Networking

##### Availability Zone

eu-north-1a

##### VPC

[vpc-086507ee77883ae1b](#)

##### Subnet group

default-vpc-086507ee77883ae1b

##### Subnets

[subnet-0db6b45e321b7000a](#)

[subnet-087377db566f545dc](#)

[subnet-0bac42bdab1e990c5](#)

##### Network type

IPv4

#### Security

##### VPC security groups

[default \(sg-0b4c8dc4647072099\)](#)

✓ Active

##### Publicly accessible

Yes

##### Certificate authority [Info](#)

rds-ca-rsa2048-g1

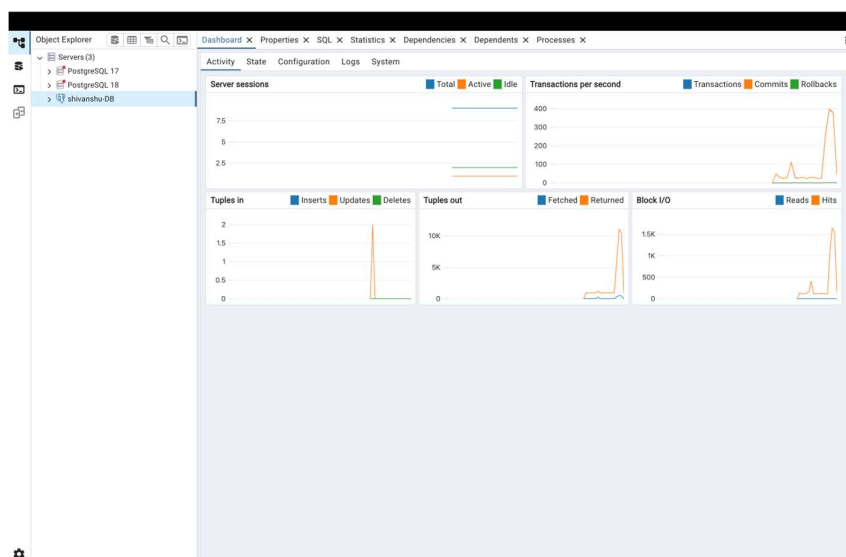
##### Certificate authority date

May 25, 2061, 03:29 (UTC+05:30)

##### DB instance certificate expiration date

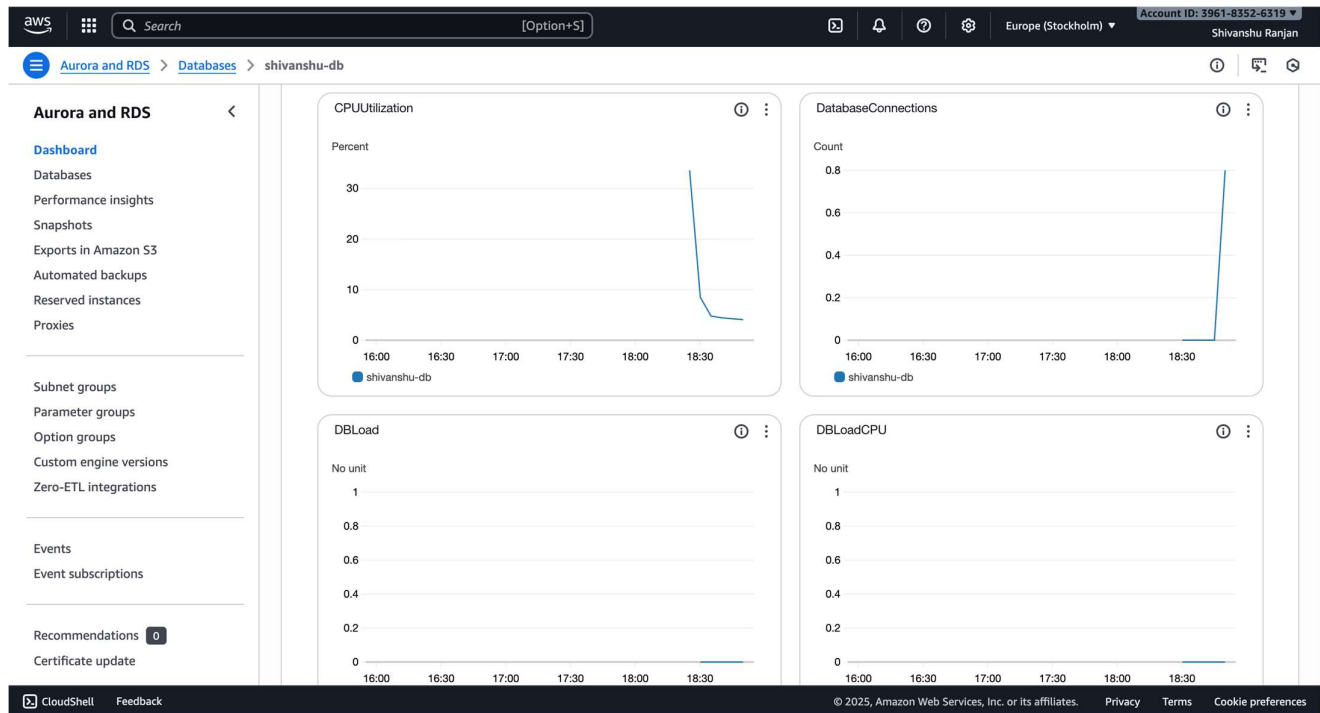
October 30, 2026, 23:59 (UTC+05:30)

## 16. Launching pgAdmin on Local Machine

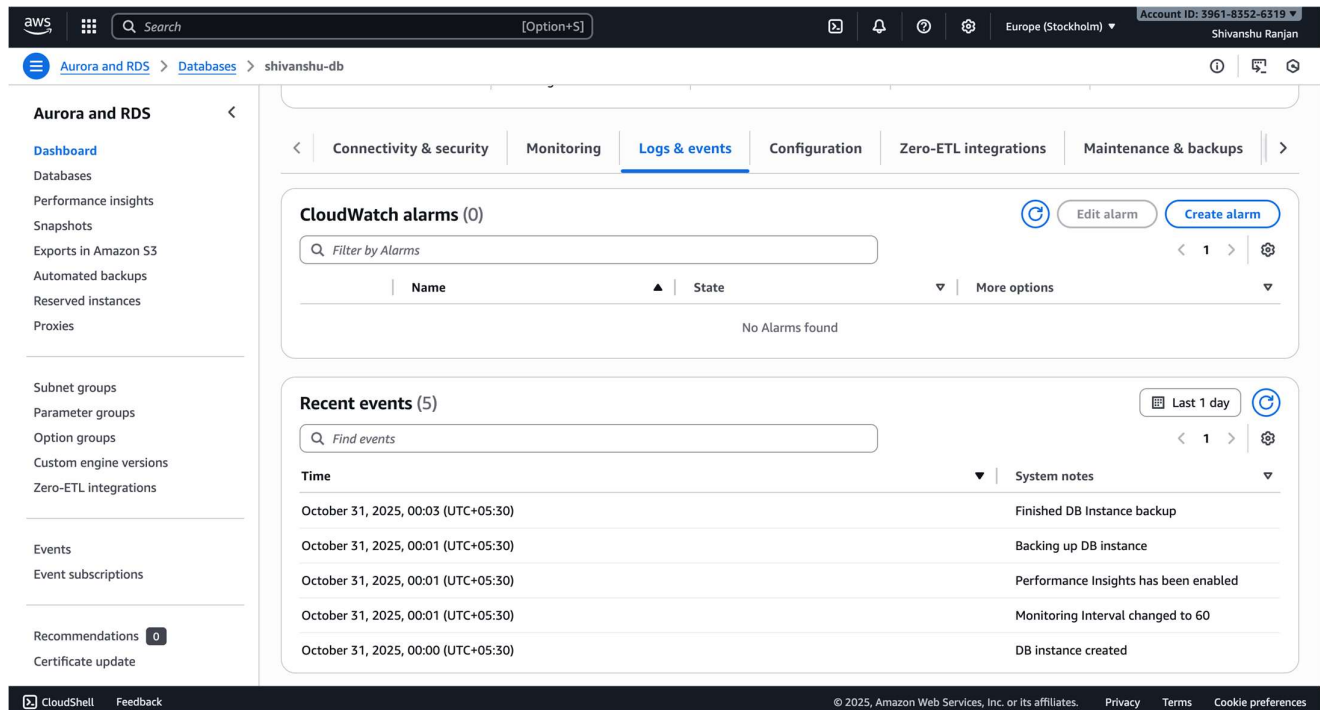




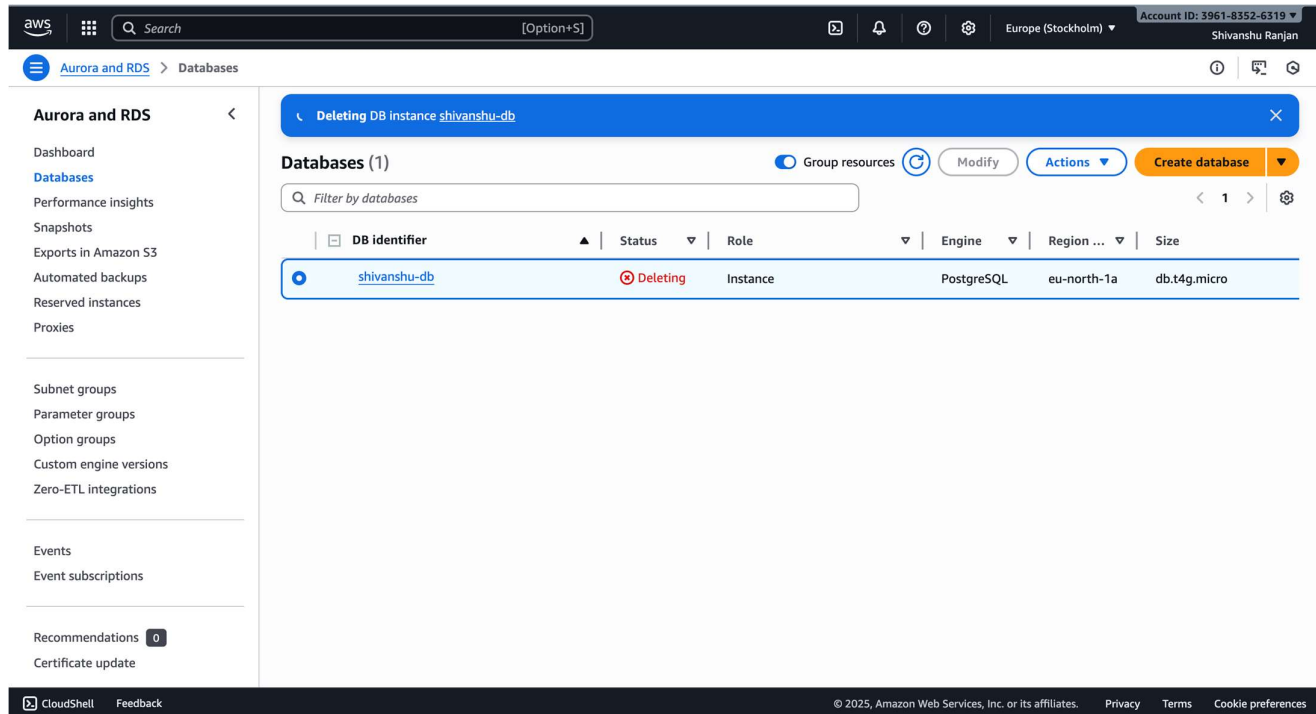
## 17. Adding a New Server in pgAdmin



## 18. Entering Connection Details (Endpoint, Username, Password)



## 19. Successful Connection to AWS RDS Database via pgAdmin



The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and account information (Europe (Stockholm), Account ID: 3961-8352-6319, Shivanshu Ranjan). The left sidebar shows the 'Aurora and RDS' menu with options like Dashboard, Databases, Performance insights, Snapshots, Exports in 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 update. The main content area is titled 'Deleting DB instance shivanshu-db'. It shows a table with one database instance, 'shivanshu-db', which is in the 'Deleting' status. The table columns are DB identifier, Status, Role, Engine, Region, and Size. The instance is a PostgreSQL database in the eu-north-1a region, using the db.t4g.micro instance type.

## 4. Learning Outcomes:

- Understand the fundamental concepts and benefits of using Amazon RDS for relational database management in the cloud.
- Gain practical knowledge of creating and configuring an RDS database instance on AWS.
- Learn how to manage and secure database access using AWS security groups.
- Develop skills to connect a local pgAdmin client to a cloud-hosted RDS instance.
- Be able to monitor, manage, and test database connectivity and performance in a cloud environment.