

## Worksheet 9

**Student Name:** Yash Kotra

**Branch:** CSE

**Semester:** 5th

**Subject Name:** ADBMS

**UID:** 23BCS10599

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

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

**Subject Code:** 23CSP-333

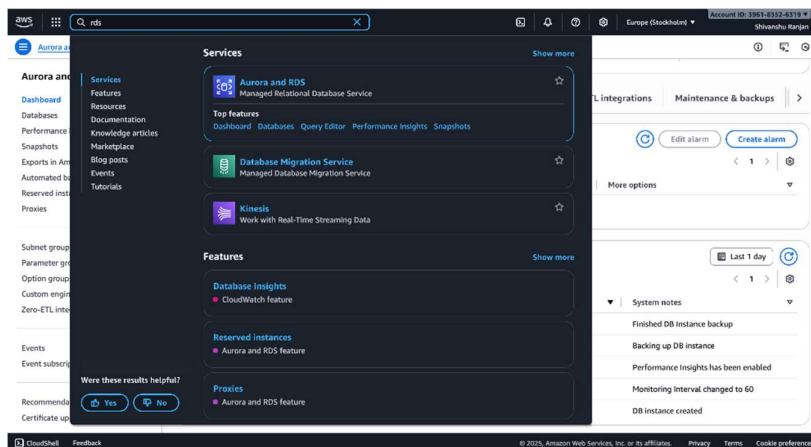
**1. Aim:** To understand and implement the setup of Amazon Relational Database Service (AWS RDS) by creating a database instance, configuring security groups, and establishing a secure connection between the local pgAdmin tool and the RDS instance hosted on the AWS Cloud.

### **2. Objective:**

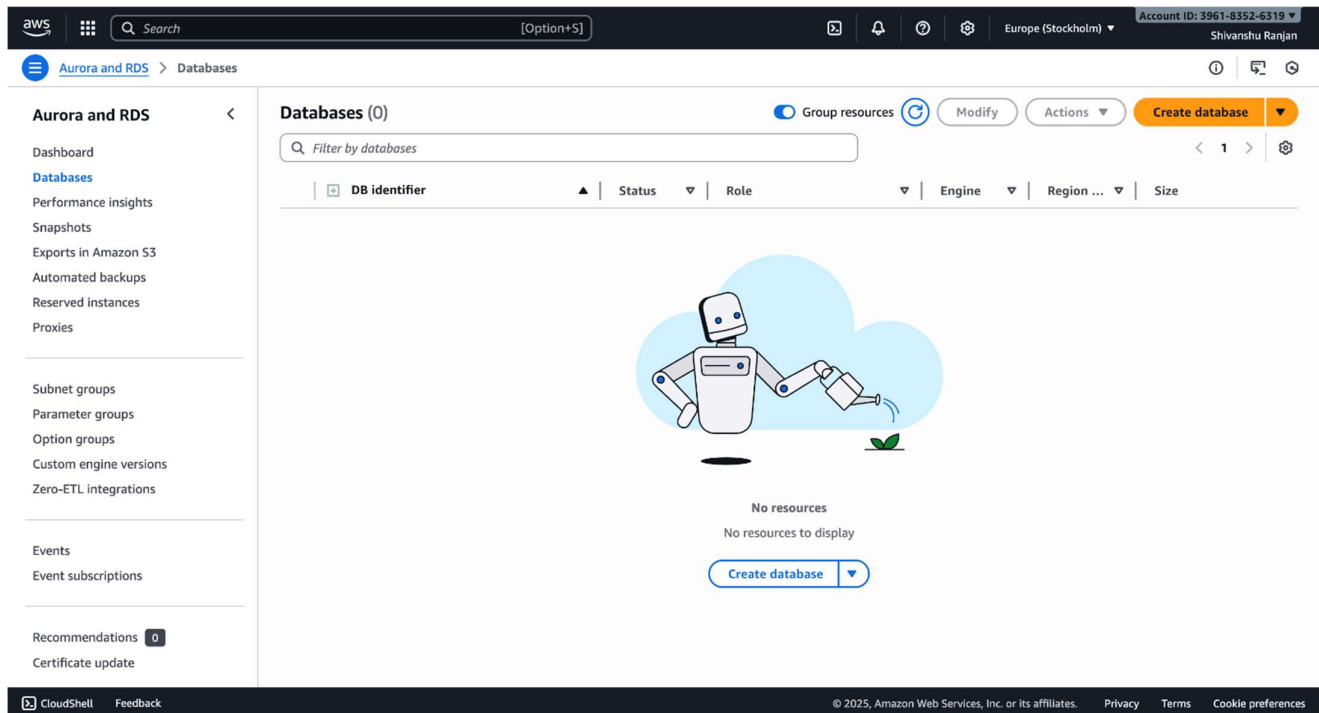
- To learn the basic concepts and features of Amazon Relational Database Service (AWS RDS).
- To create and configure a new RDS database instance on the AWS Management Console.
- To understand the role and configuration of security groups for controlling database access.
- To connect a local pgAdmin 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:**

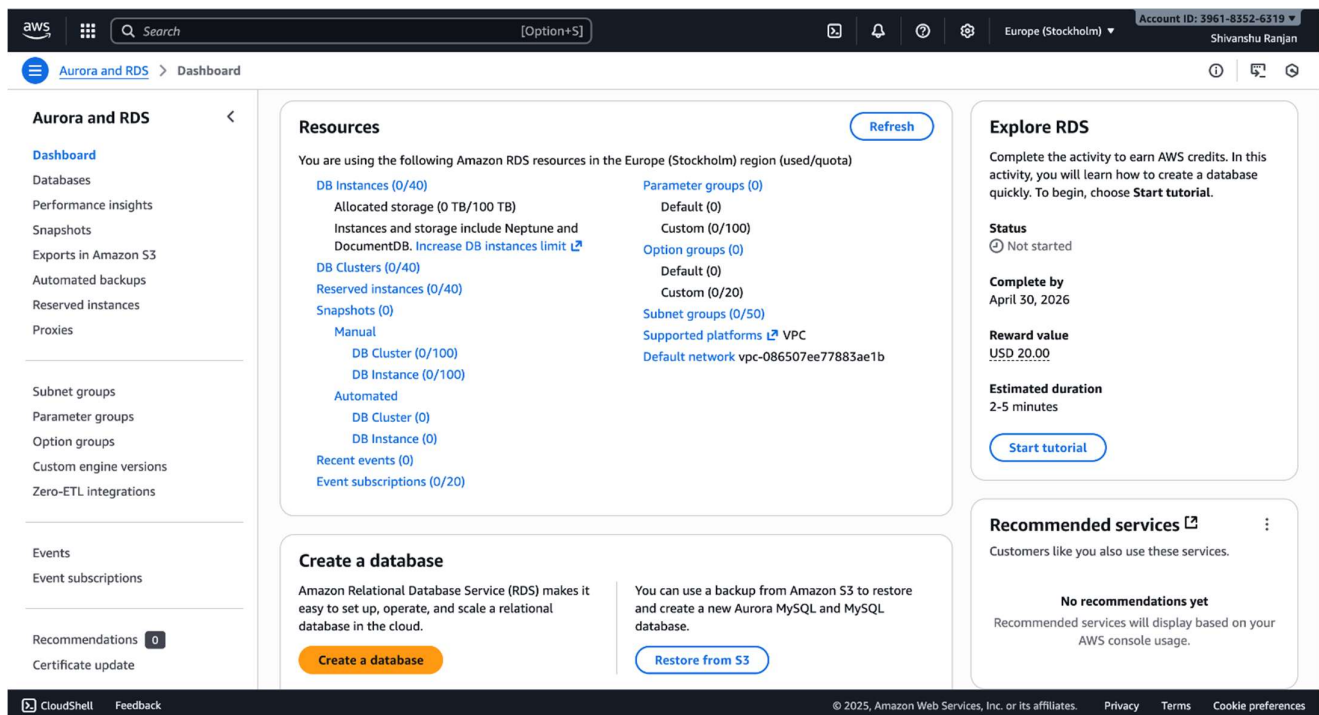
#### **1. Sign-in**



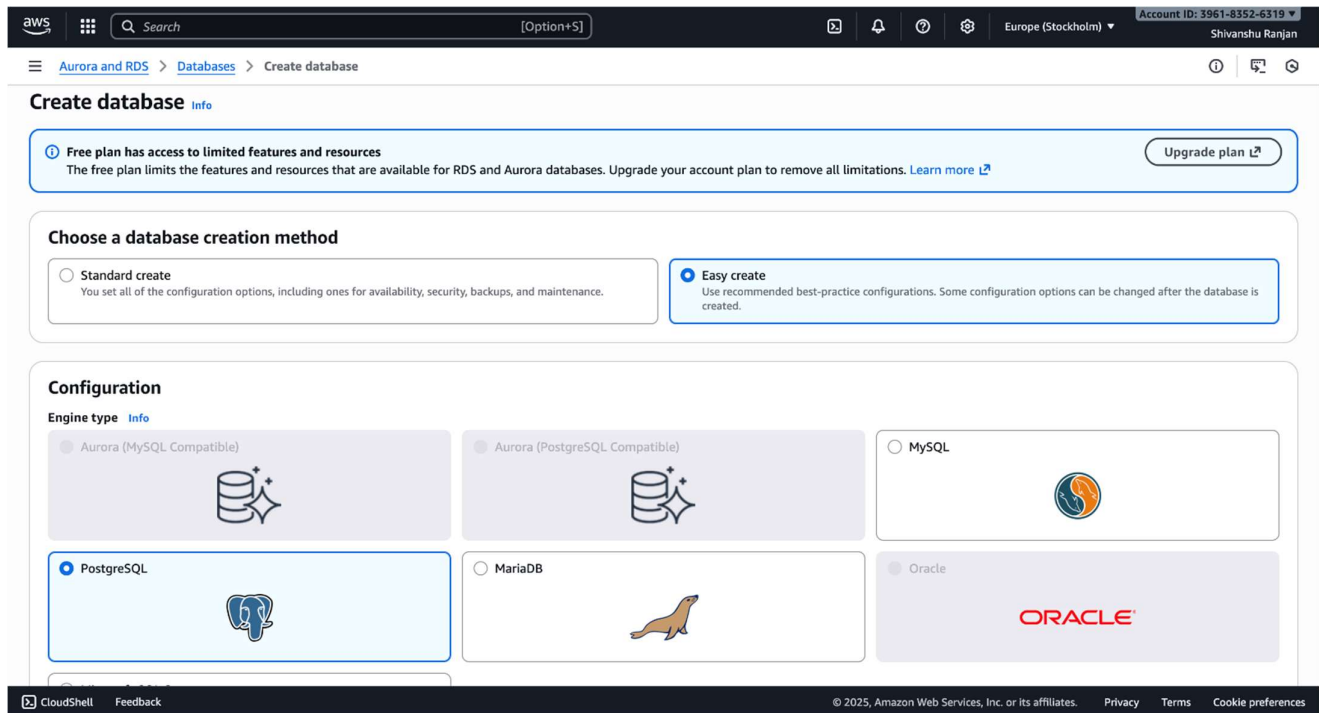
## 2. Navigating to RDS Service



## 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](#)

**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)

☒ PostgreSQL

☐ MariaDB

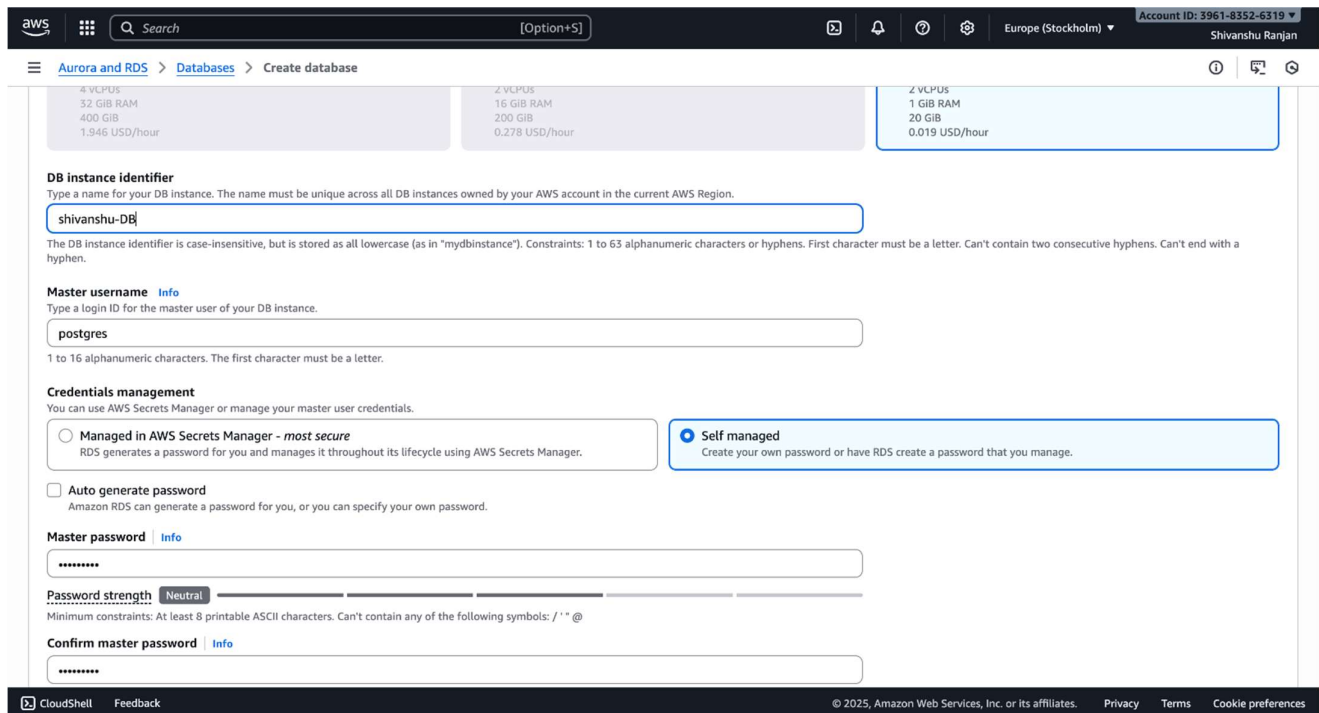
☐ MySQL

☐ Oracle

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 5. Selecting PostgreSQL as Database Engine



**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](#)

**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)

☒ PostgreSQL

☐ MariaDB

☐ MySQL

☐ Oracle

**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.

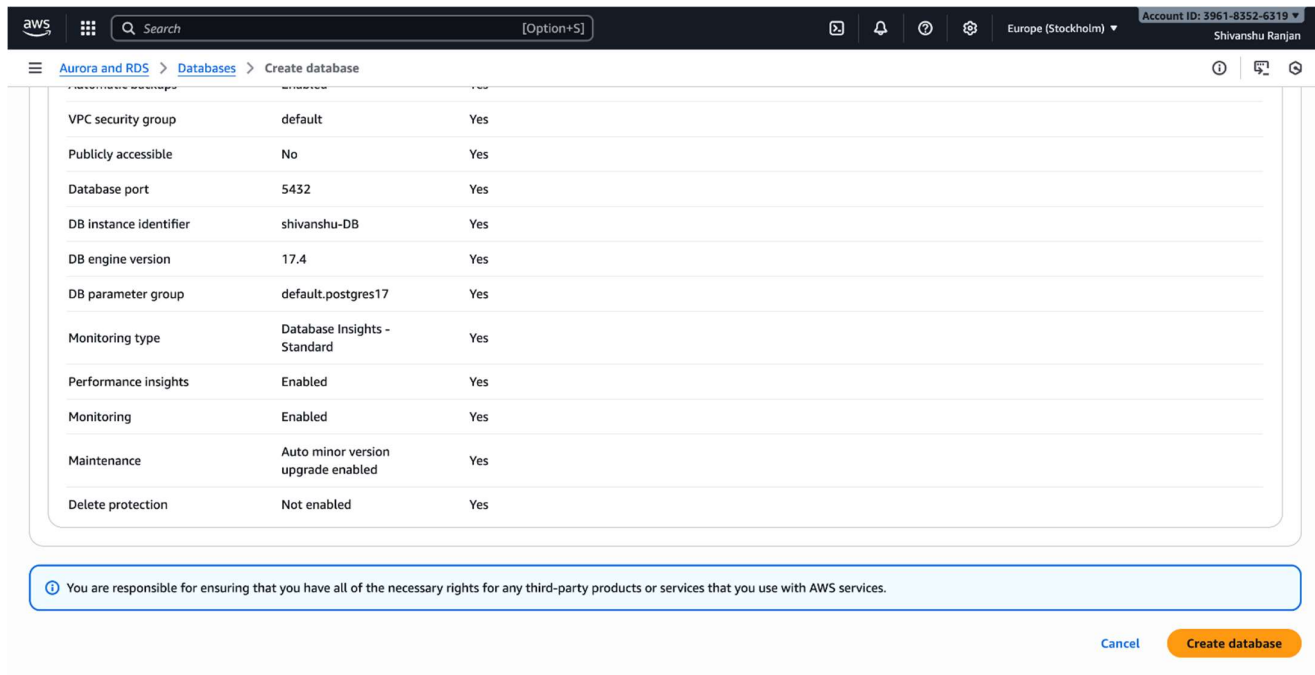
**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.  
☐ Managed in AWS Secrets Manager - most secure  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.  
☒ Self managed  
Create your own password or have RDS create a password that you manage.

**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

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 6. Choosing Deployment Option and Template

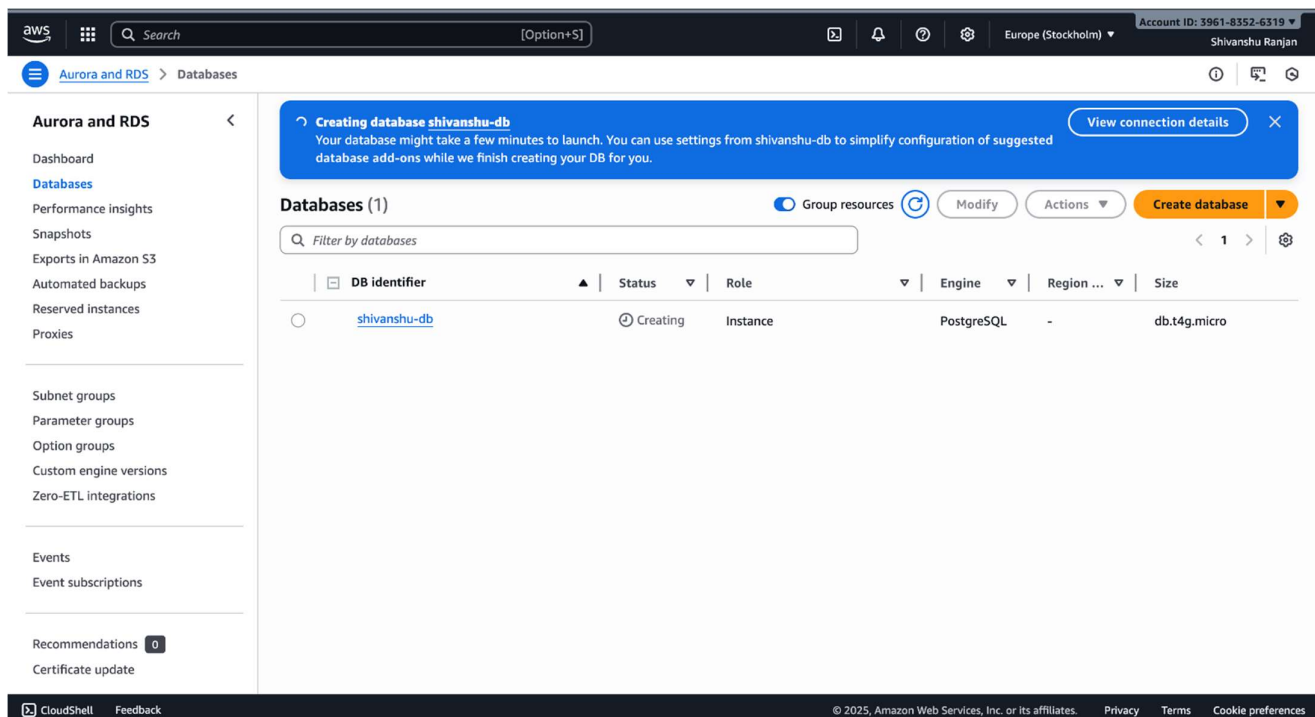


The screenshot shows the 'Create database' configuration page in the AWS Management Console. The configuration is as follows:

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 warning message: "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, there are two buttons: "Cancel" and "Create database".

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



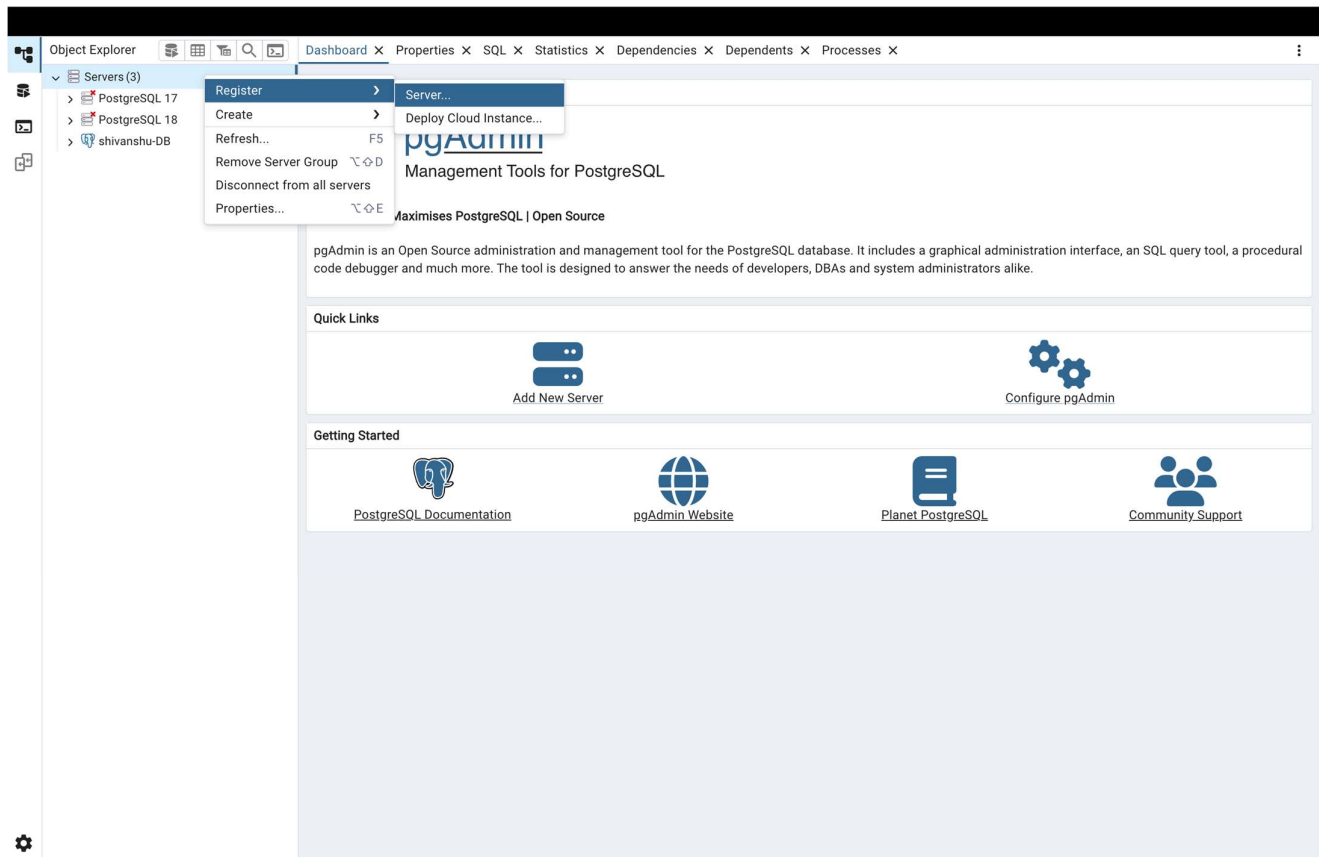
The screenshot shows the 'Databases' page in the AWS Management Console. A blue banner at the top indicates: "Creating database shivanshu-db. Your database might take a few minutes to launch. You can use settings from shivanshu-db to simplify configuration of suggested database add-ons while we finish creating your DB for you." There is a "View connection details" link.

Below the banner, the 'Databases (1)' section shows a table with the following data:

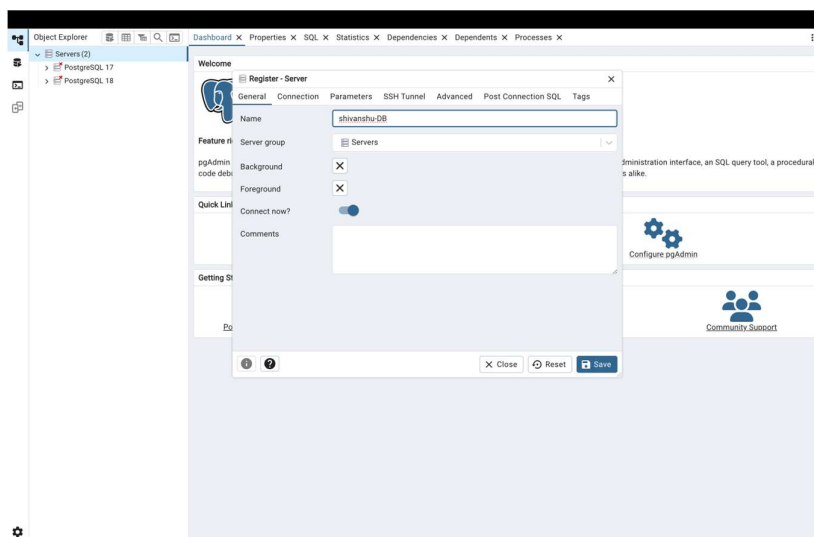
DB identifier	Status	Role	Engine	Region	Size
<a href="#">shivanshu-db</a>	Creating	Instance	PostgreSQL	-	db.t4g.micro

The left sidebar shows the 'Aurora and RDS' navigation 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, and Certificate update.

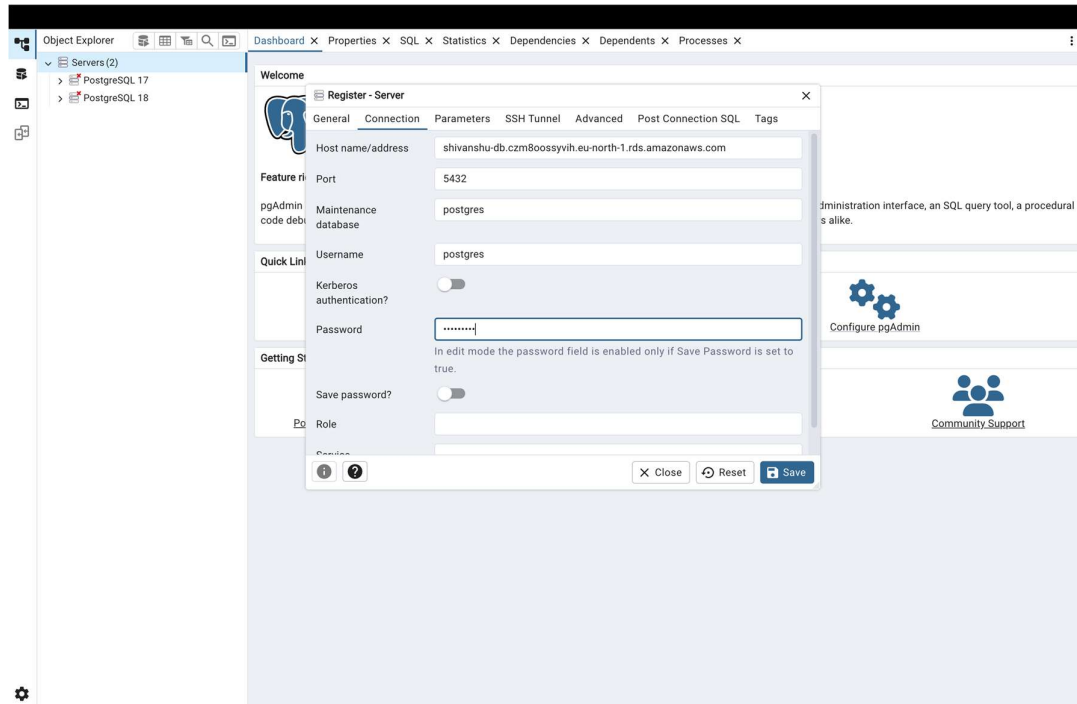
## 8. Setting Up Instance Size and Storage



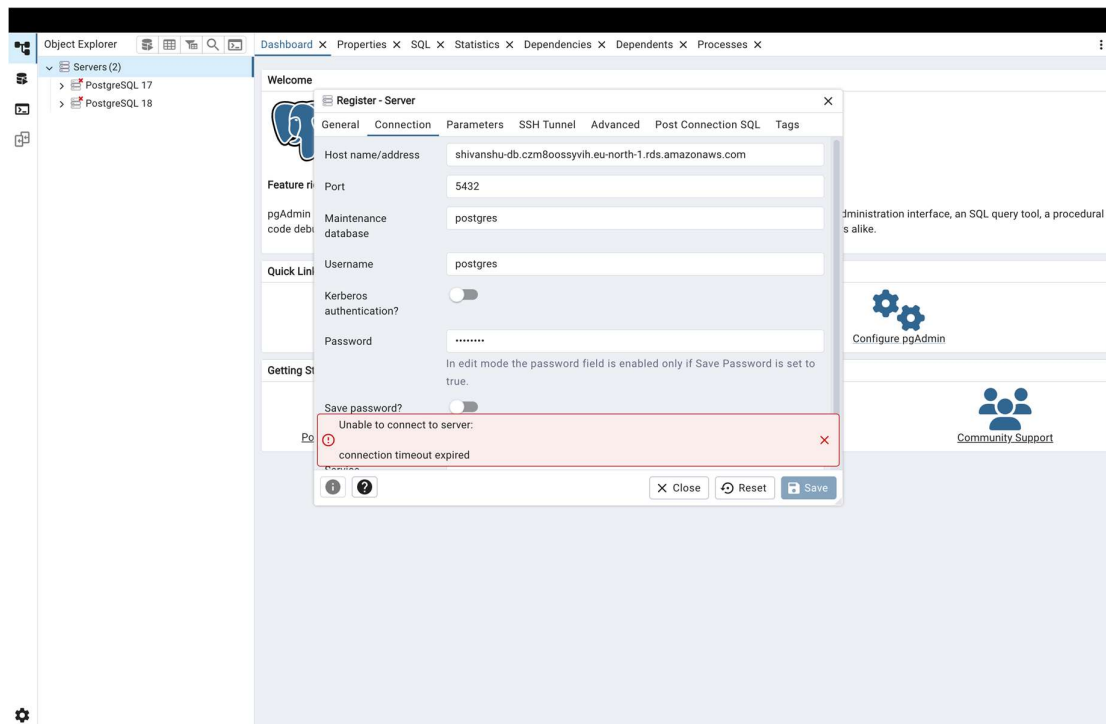
## 9. Configuring Connectivity and VPC Settings



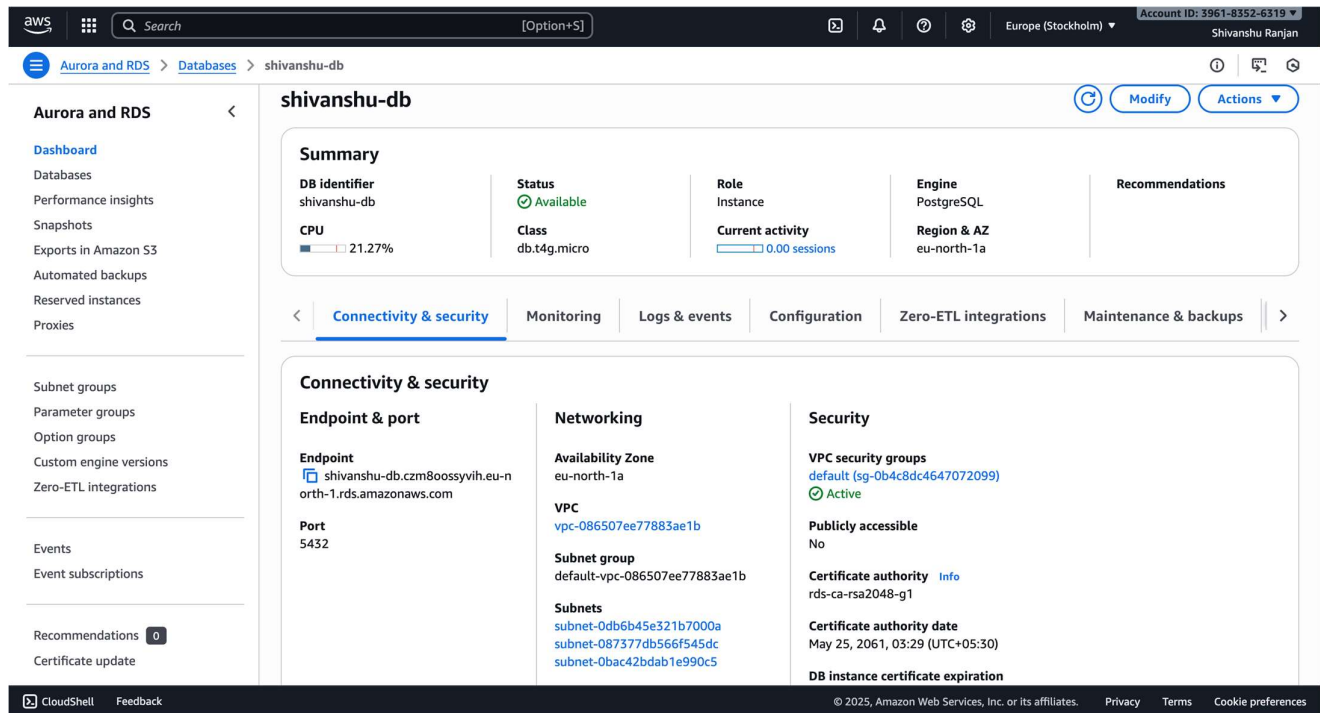
## 10. Gr Setting Up Security Groups for RDS Access



## 11. Additional Database Configuration Options



## 12. Reviewing and Creating the Database Instance

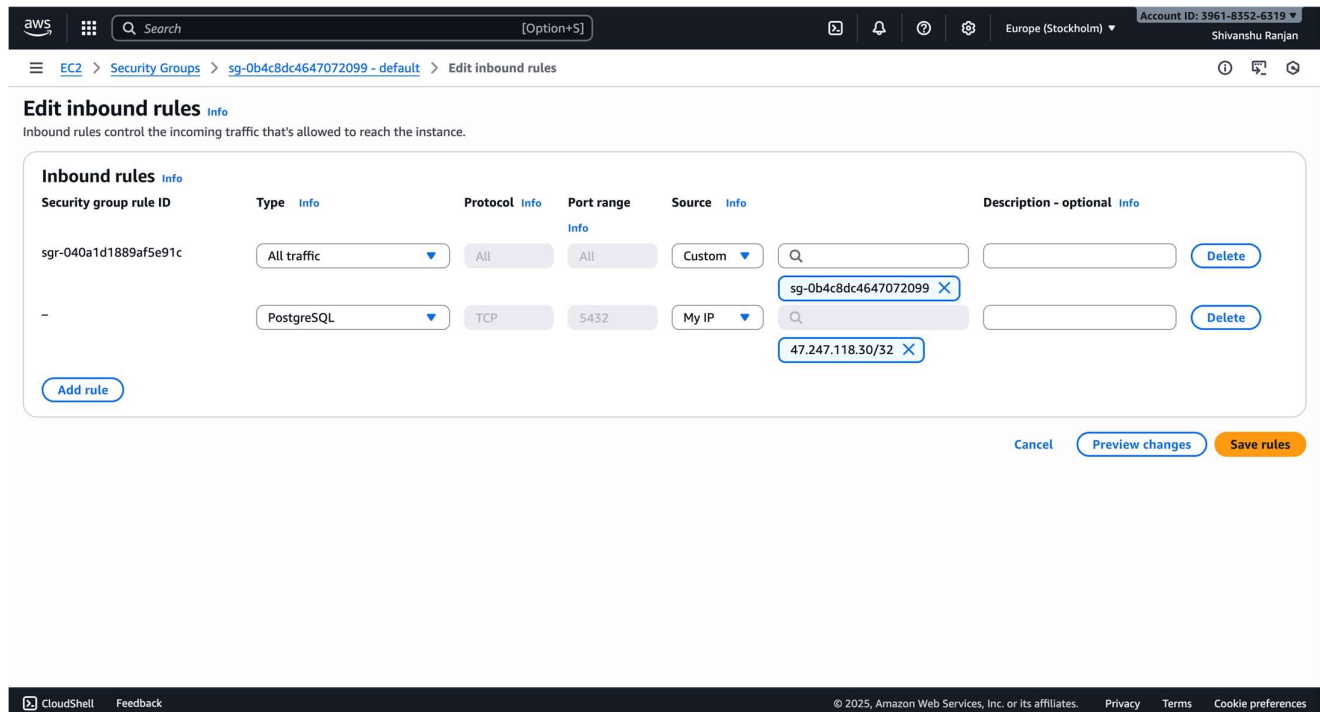


The screenshot shows the AWS Management Console interface for an Amazon RDS instance named 'shivanshu-db'. The left sidebar contains navigation links for Aurora and RDS, Databases, and various database management tasks. The main content area displays the instance details under the 'Summary' tab, including the DB identifier, status (Available), role (Instance), engine (PostgreSQL), and region (eu-north-1a). The 'Connectivity & security' tab is selected, showing the endpoint, port, availability zone, VPC, and security groups. The 'Security' section shows the VPC security groups and publicly accessible status.

Summary				
<b>DB identifier</b> shivanshu-db	<b>Status</b> Available	<b>Role</b> Instance	<b>Engine</b> PostgreSQL	<b>Recommendations</b>
<b>CPU</b> 21.27%	<b>Class</b> db.t4g.micro	<b>Current activity</b> 0.00 sessions	<b>Region &amp; AZ</b> eu-north-1a	

Connectivity & security		
<b>Endpoint &amp; port</b>	<b>Networking</b>	<b>Security</b>
<b>Endpoint</b> shivanshu-db.czm8oossvih.eu-north-1.rds.amazonaws.com	<b>Availability Zone</b> eu-north-1a	<b>VPC security groups</b> default (sg-0b4c8dc4647072099) Active
<b>Port</b> 5432	<b>VPC</b> vpc-086507ee77883ae1b	<b>Publicly accessible</b> No
	<b>Subnet group</b> default-vpc-086507ee77883ae1b	<b>Certificate authority</b> rds-ca-rsa2048-g1
	<b>Subnets</b> subnet-0db6b45e321b7000a subnet-087377db566f545dc subnet-0bac42bdab1e990c5	<b>Certificate authority date</b> May 25, 2061, 03:29 (UTC+05:30)
		<b>DB instance certificate expiration</b>

## 13. RDS Instance Creation in Progress



The screenshot shows the AWS Management Console interface for editing inbound rules for a security group. The left sidebar contains navigation links for EC2, Security Groups, and Edit inbound rules. The main content area displays the 'Edit inbound rules' page, which includes a table of existing rules and an 'Add rule' button. The table has columns for Security group rule ID, Type, Protocol, Port range, Source, and Description. The 'Add rule' button is located at the bottom left of the table.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-040a1d1889af5e91c	All traffic	All	All	Custom	
-	PostgreSQL	TCP	5432	My IP	

Buttons: Add rule, Cancel, Preview changes, Save rules



## 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

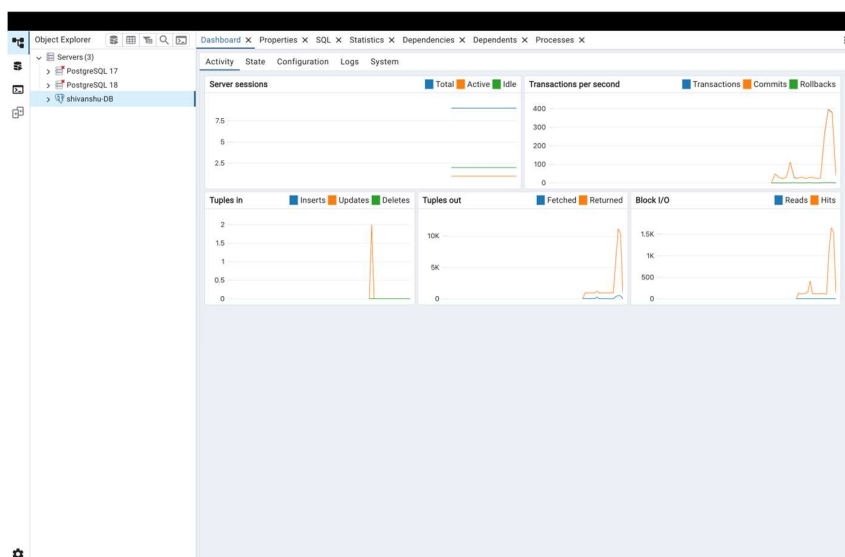
##### Certificate authority date

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

##### DB instance certificate expiration date

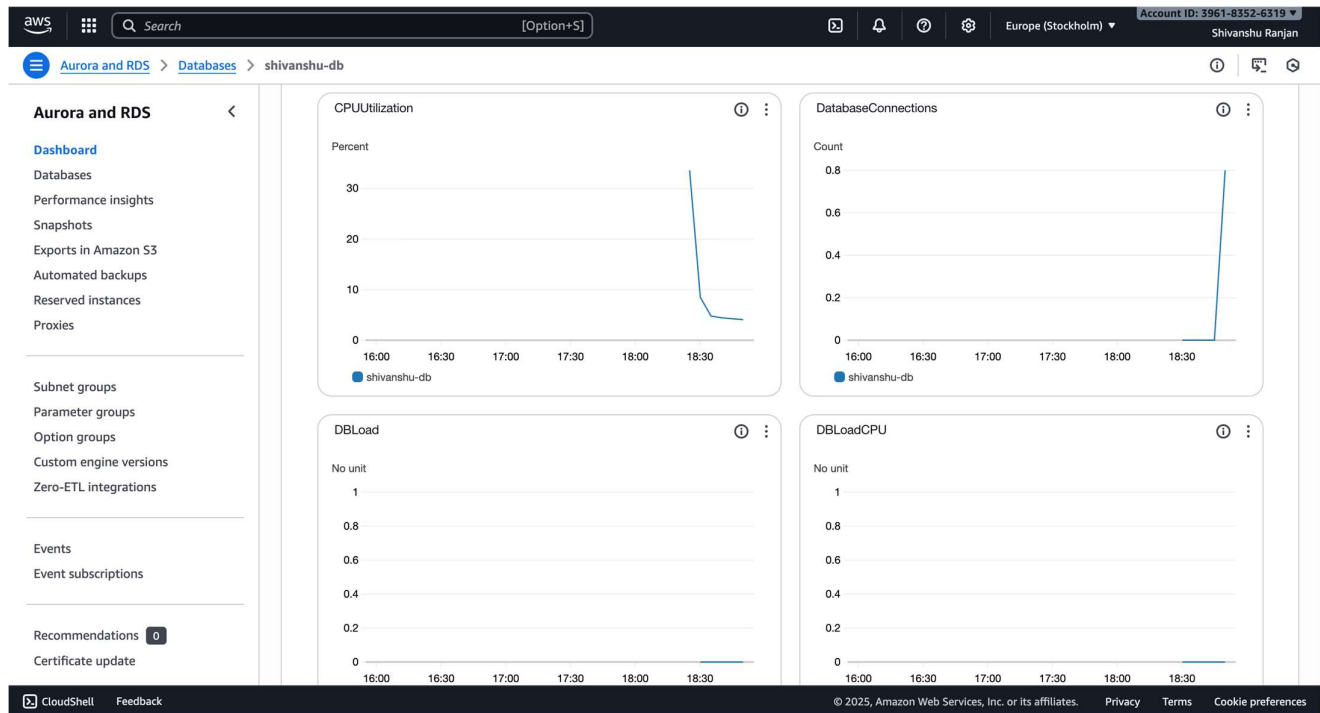
October 30, 2026, 23:59 (UTC+05:30)

## 16. Launching pgAdmin on Local Machine

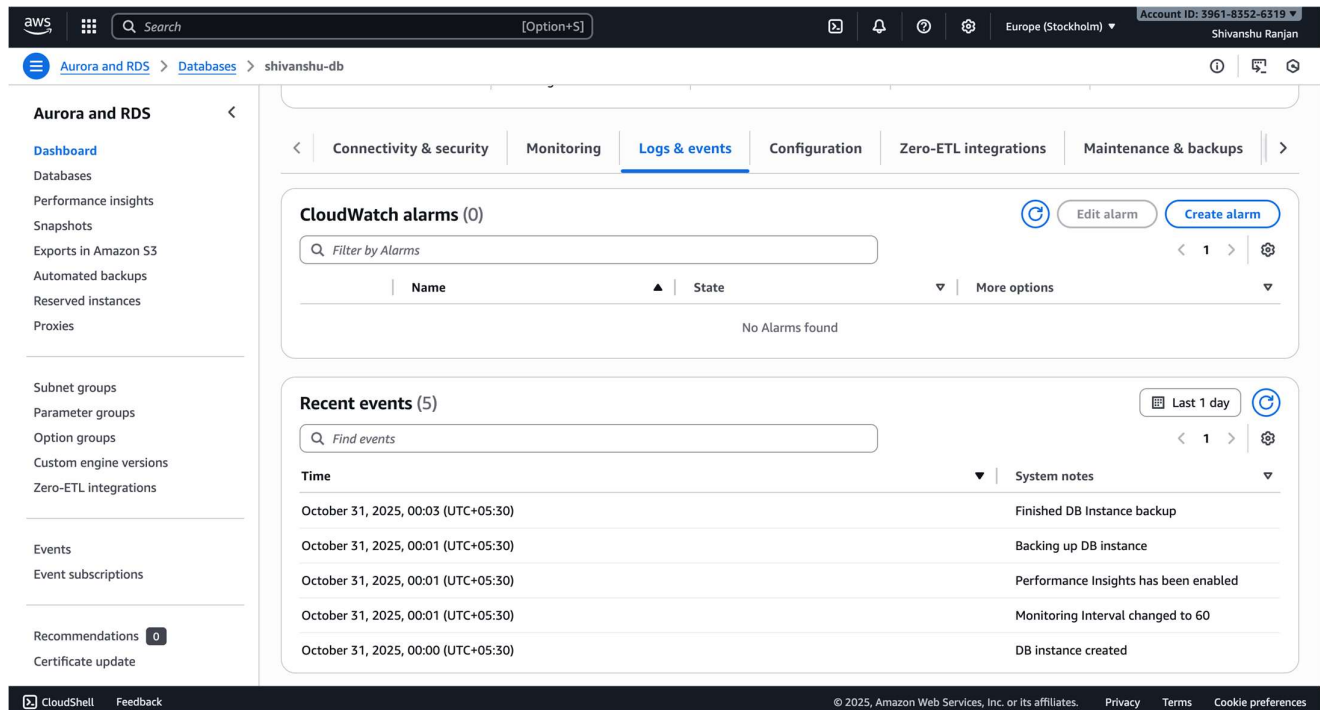




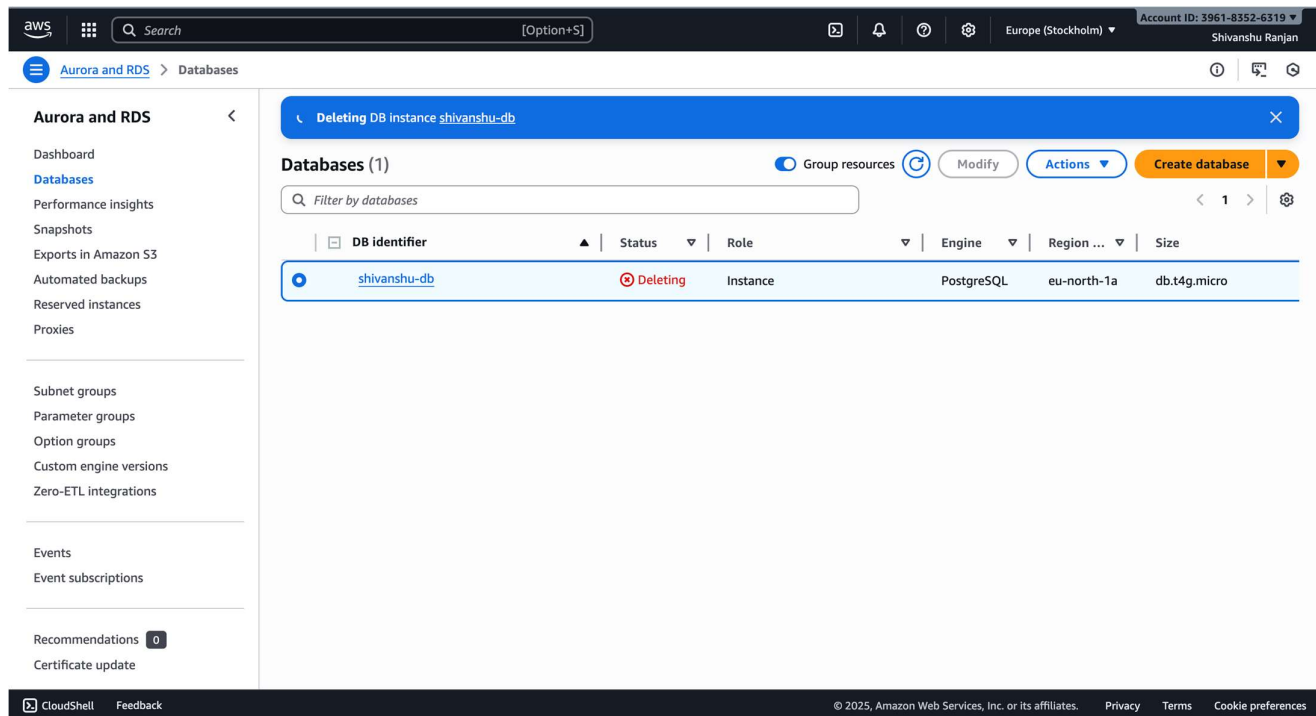
## 17. Adding a New Server in pgAdmin



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



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



The screenshot displays the AWS Management Console interface. At the top, the navigation bar shows the AWS logo, a search bar, and the account ID: 3961-8352-6319. The main content area is titled 'Aurora and RDS' and 'Databases'. A blue banner at the top of the main area indicates 'Deleting DB instance shivanshu-db'. Below this, the 'Databases (1)' section shows a table with one entry: 'shivanshu-db'. The table columns are 'DB identifier', 'Status', 'Role', 'Engine', 'Region ...', and 'Size'. The 'Status' column for 'shivanshu-db' shows a red 'Deleting' status. The 'Engine' is 'PostgreSQL', the 'Region' is 'eu-north-1a', and the 'Size' is 'db.t4g.micro'. The left sidebar contains a navigation 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', and 'Certificate update'.

## 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.