Worksheet 9

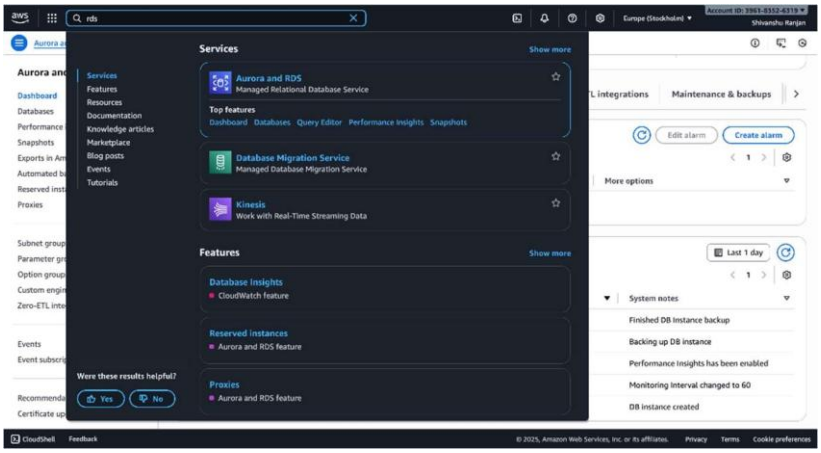
Aim: To understand and implement the setup of Amazon Relational Database Service (AWS RDS) by creating a database instance, configuring security groups, and establishing a secure connection between the local pgAdmin tool and the RDS instance hosted on the AWS Cloud.

1. Objective:

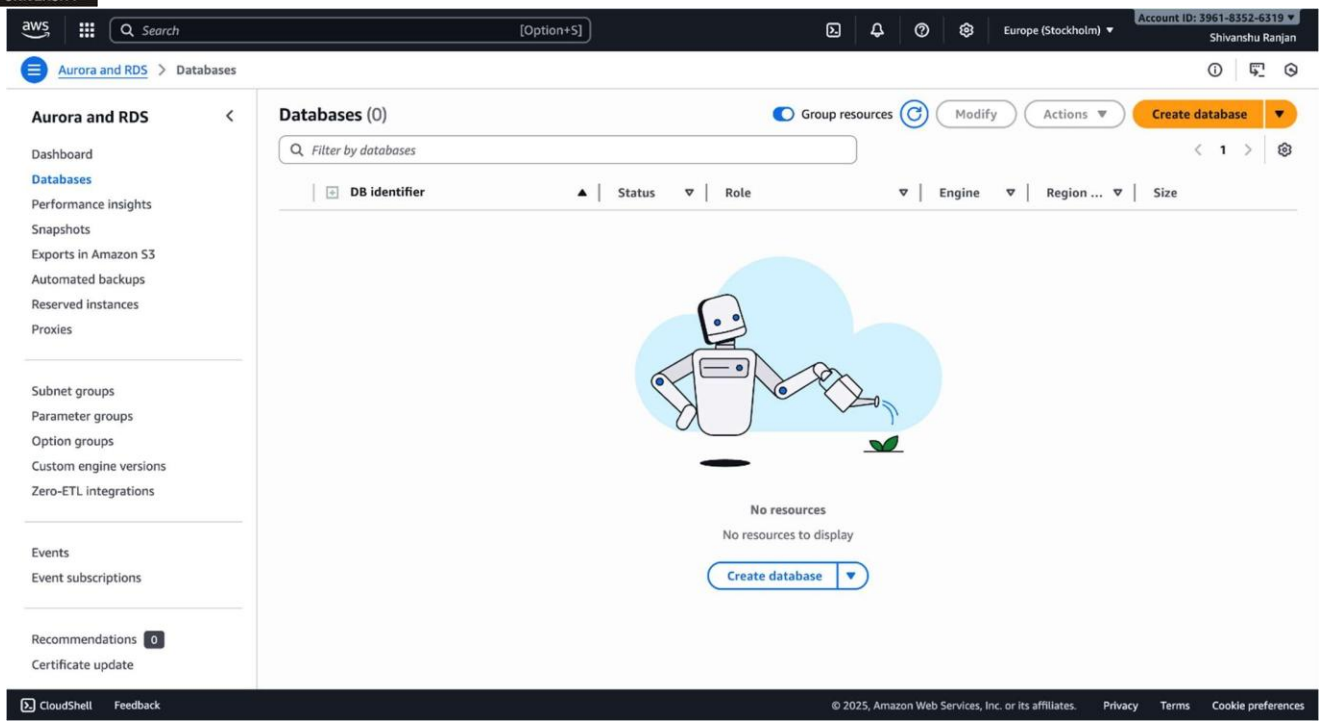
- To learn the basic concepts and features of Amazon Relational Database Service (AWS RDS).
- To create and configure a new RDS database instance on the AWS Management Console.
- To understand the role and configuration of security groups for controlling database access.
- To connect a local pgAdmin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

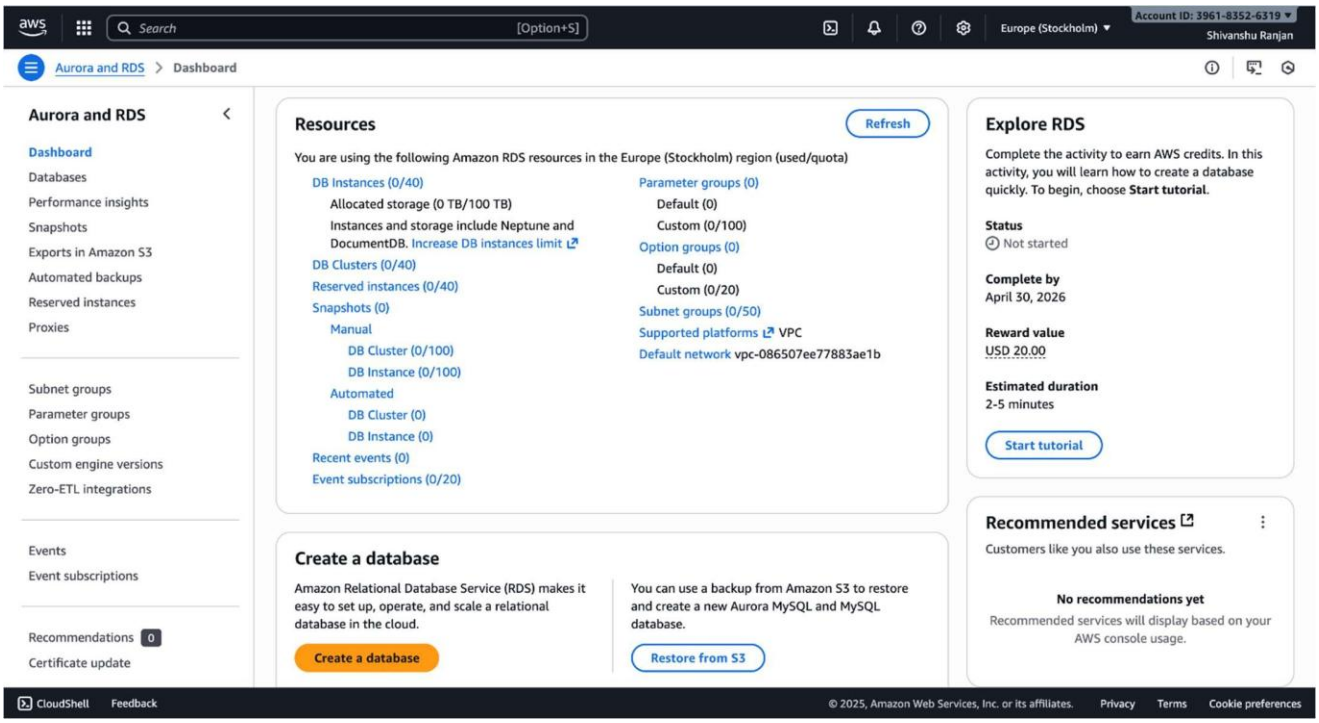
1. Sign-in



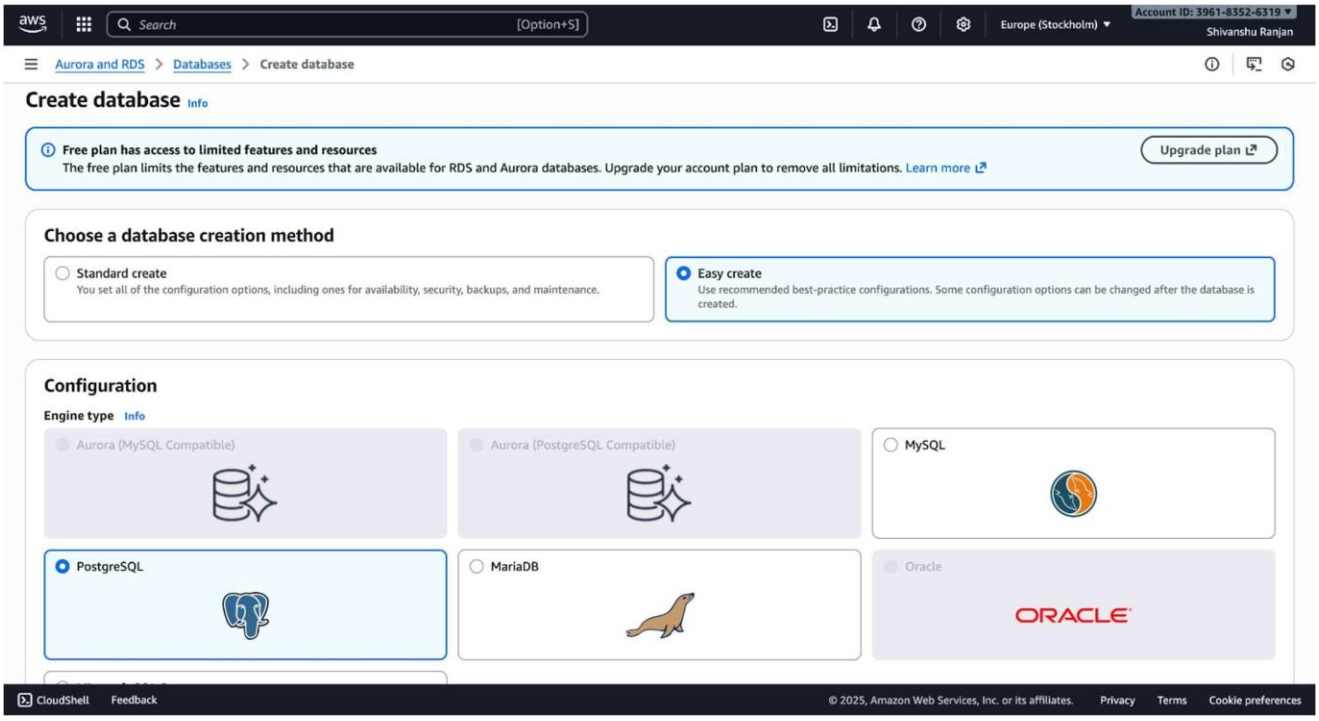
2. Navigating to RDS Service



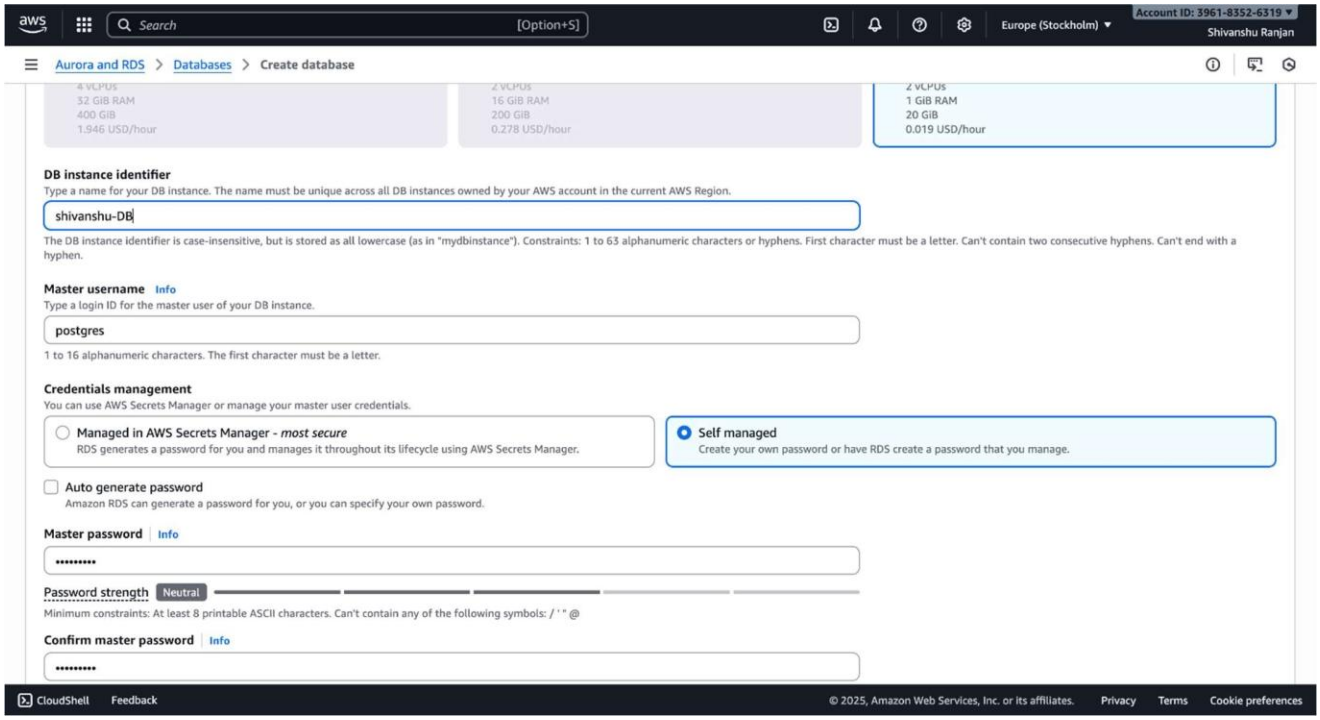
3. Amazon RDS Dashboard Overview



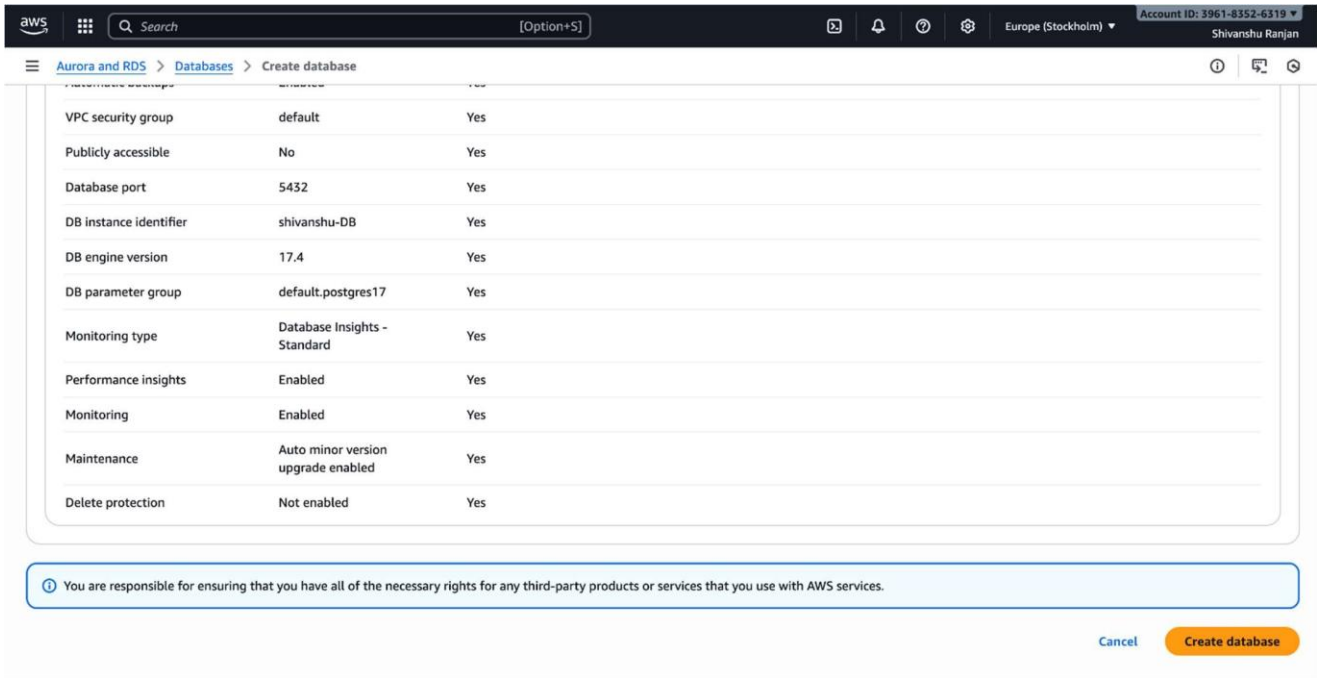
4. Creating a New Database Instance



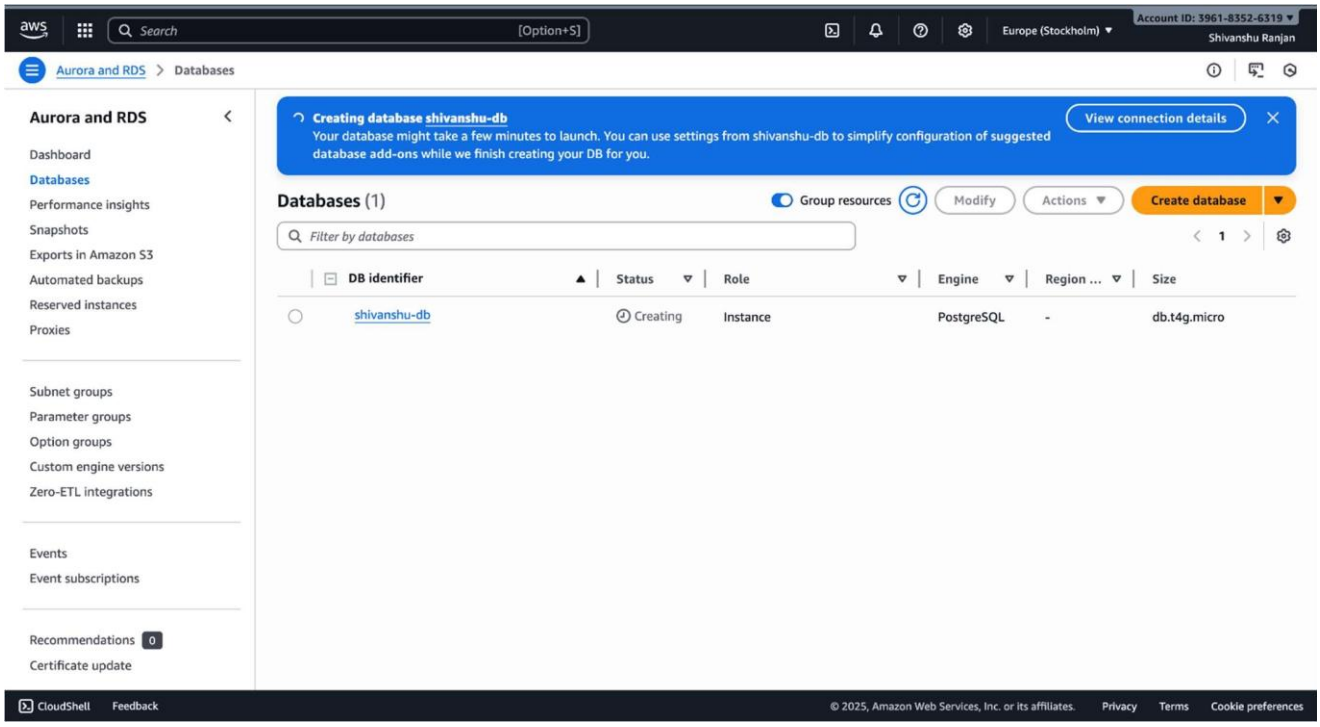
5. Selecting PostgreSQL as Database Engine



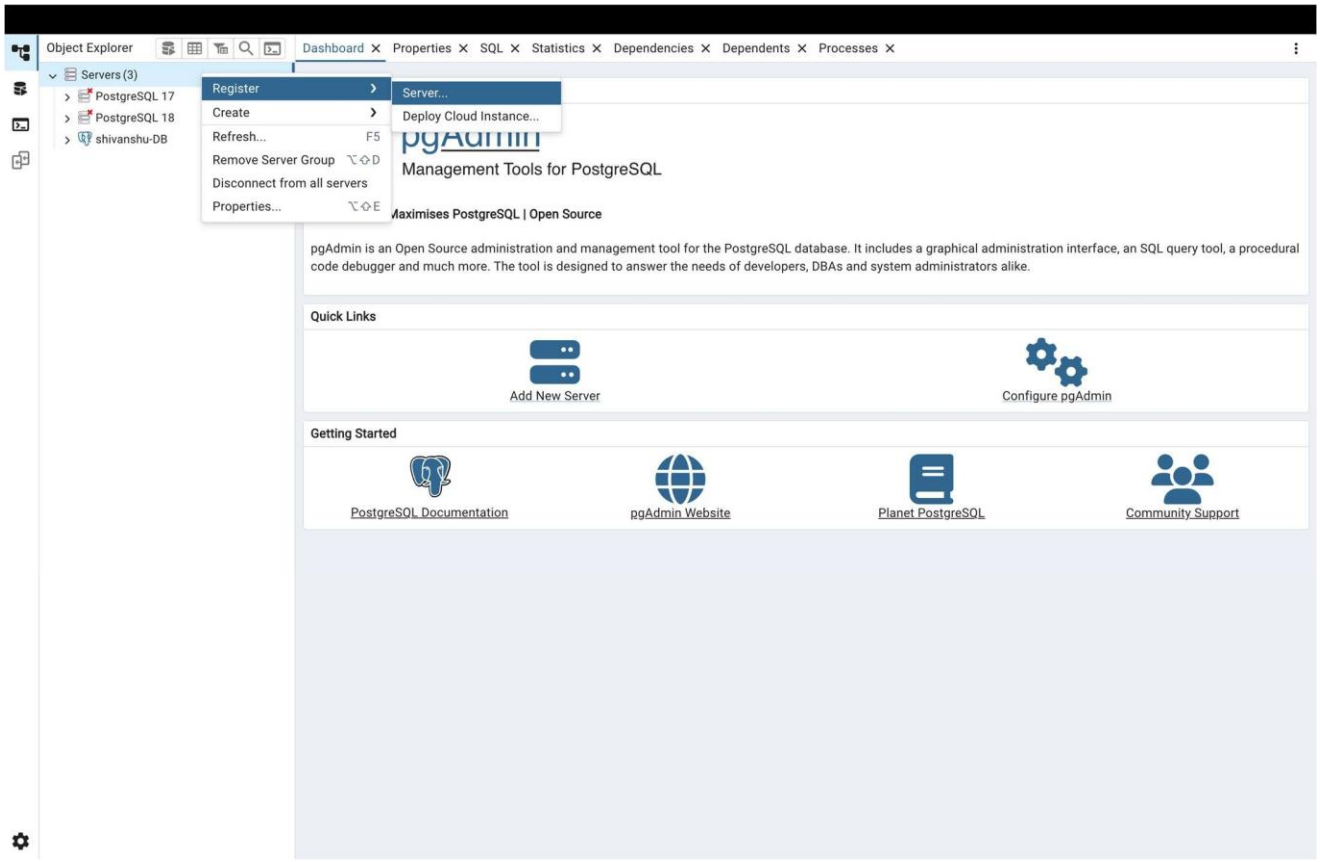
6. Choosing Deployment Option and Template



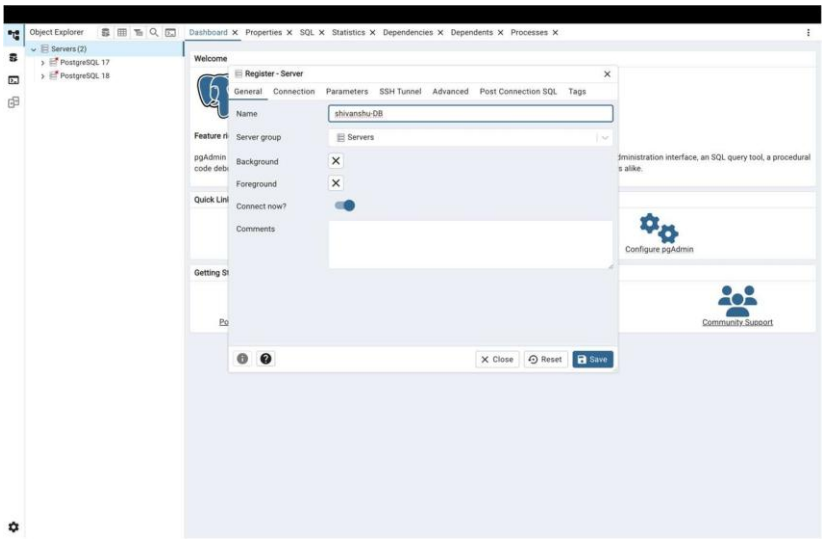
7. Configuring Database Settings (Name, Username, Password)



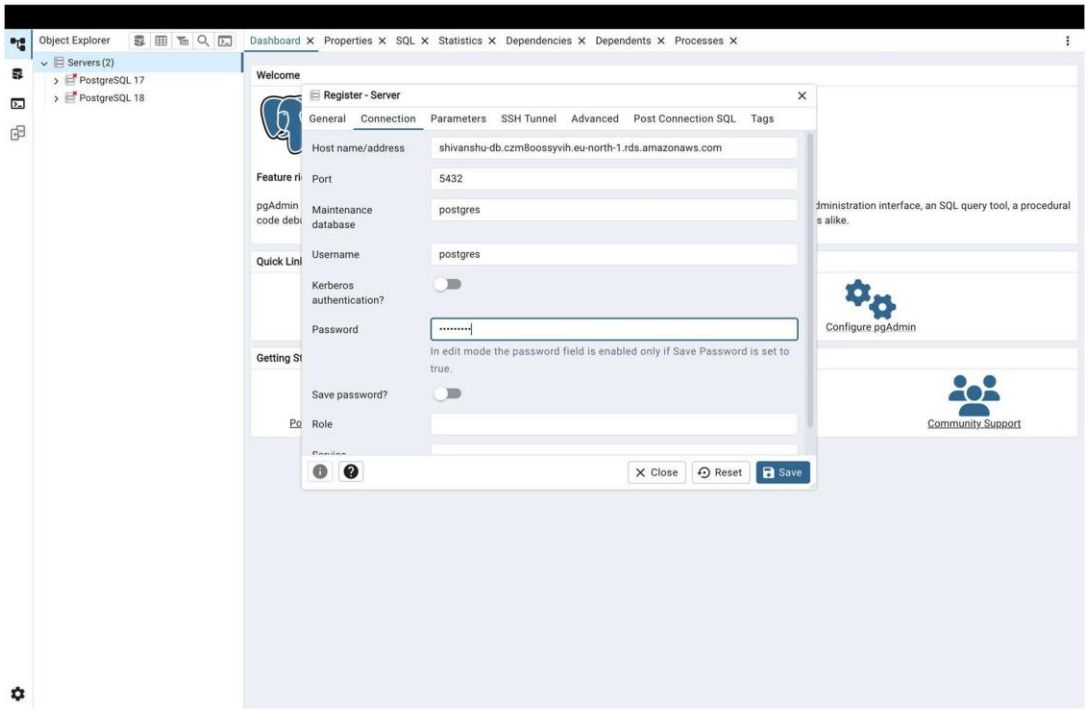
8. Setting Up Instance Size and Storage



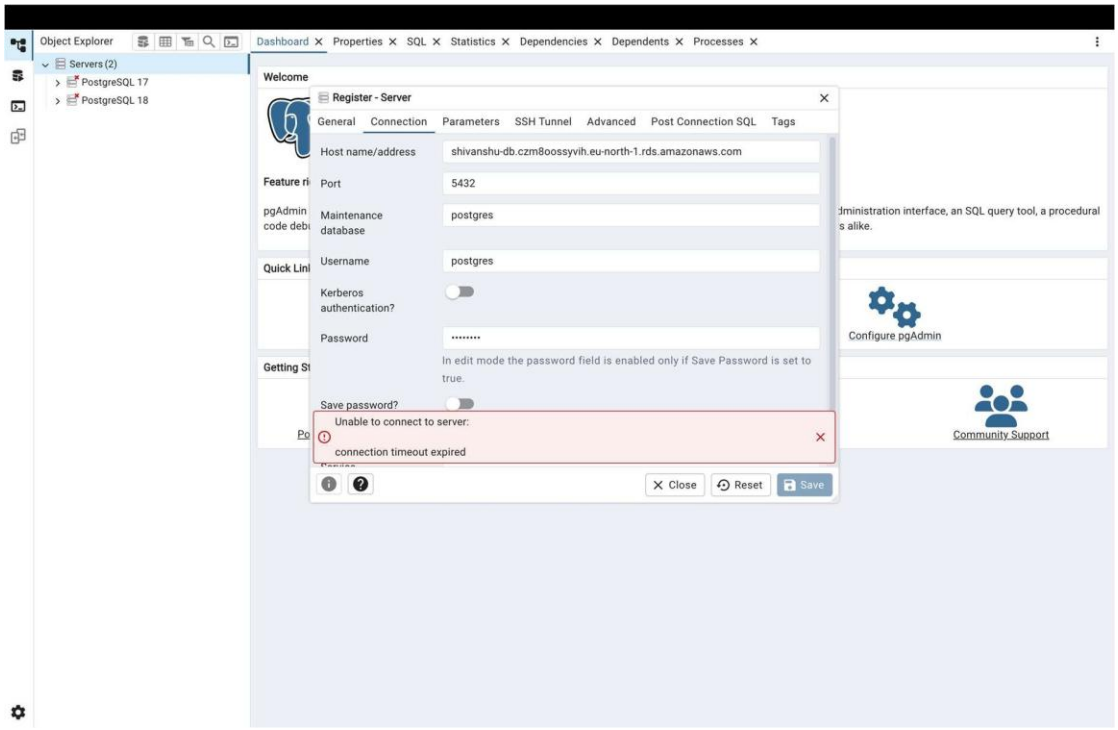
9. Configuring Connectivity and VPC Settings



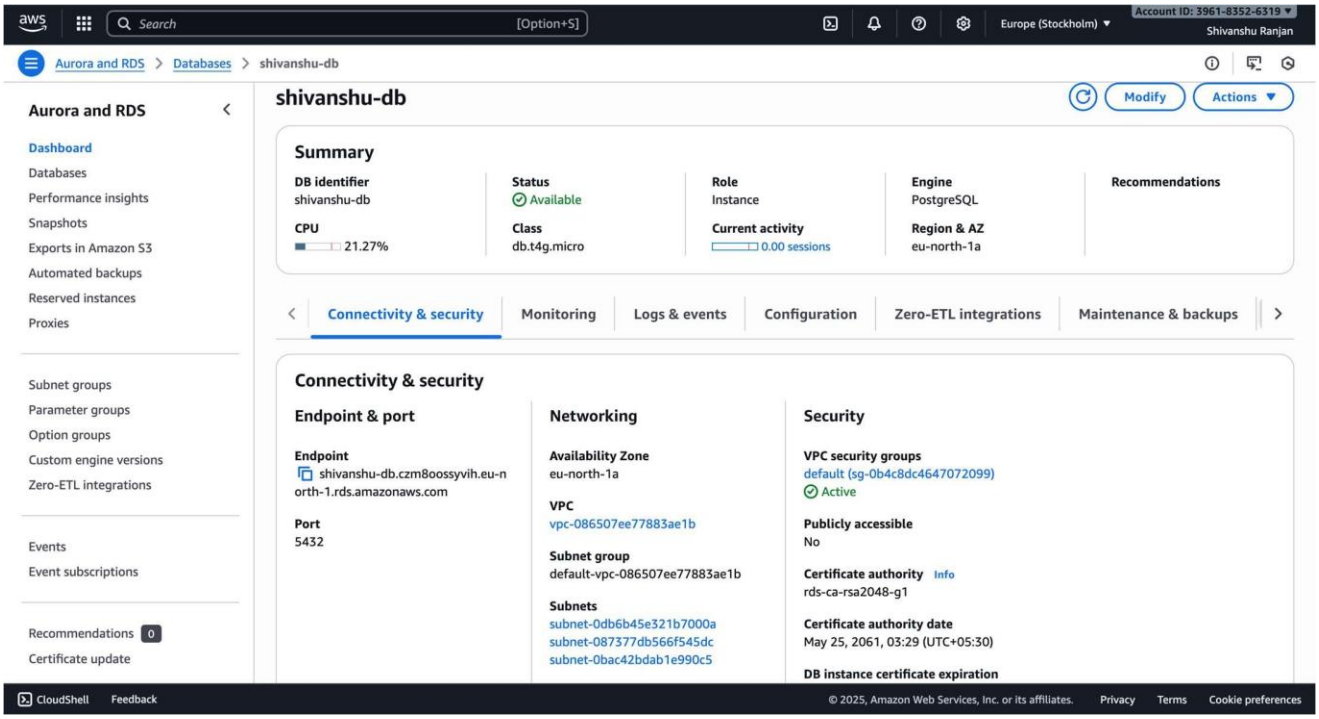
10. Gr Setting Up Security Groups for RDS Access



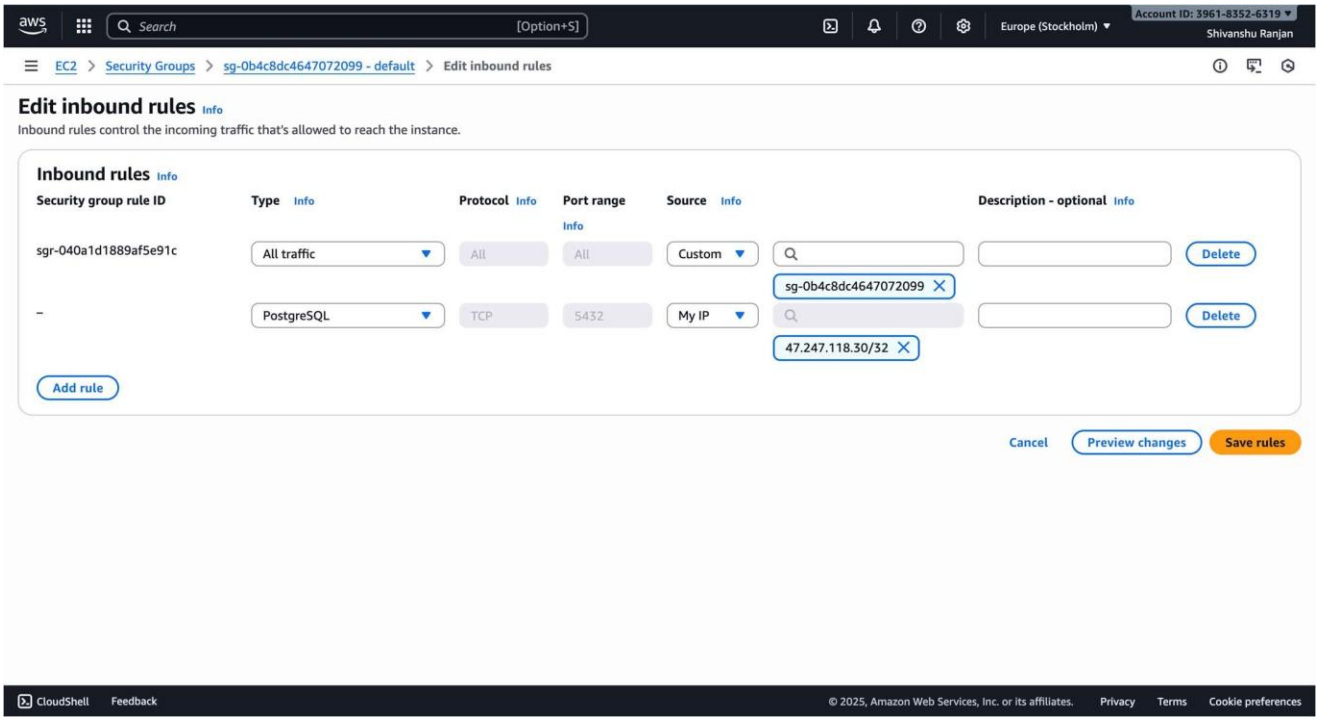
11. Additional Database Configuration Options



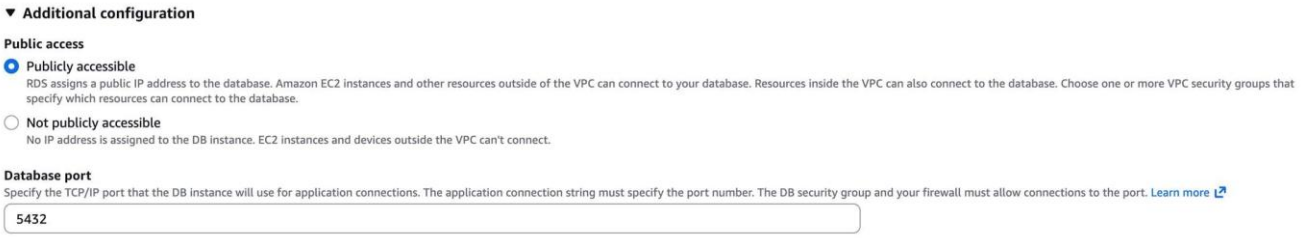
12. Reviewing and Creating the Database Instance



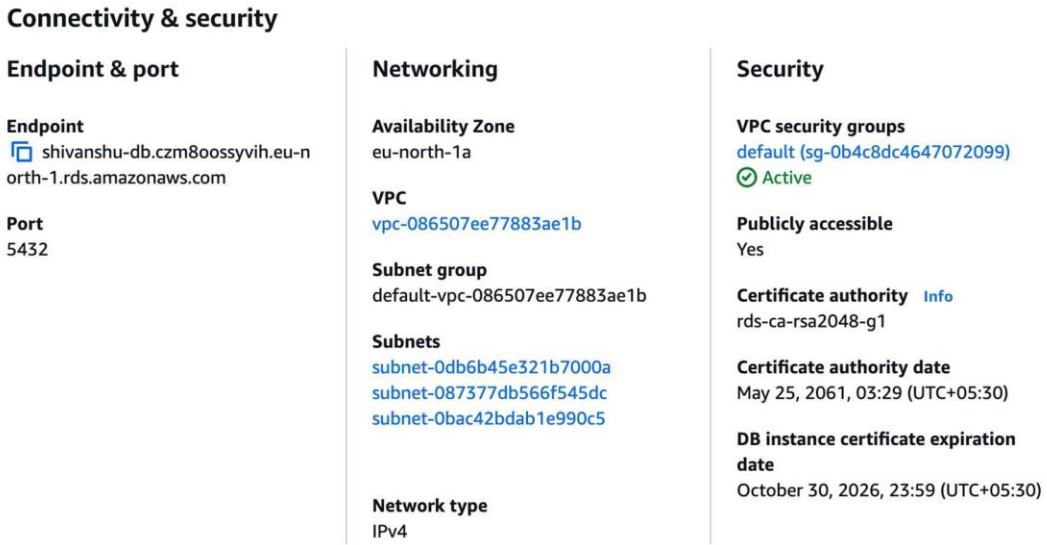
13. RDS Instance Creation in Progress



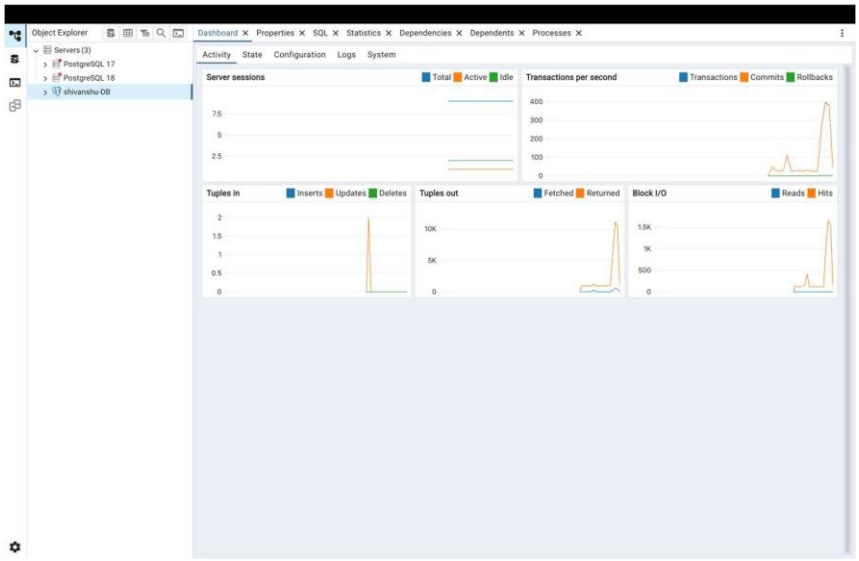
14. Viewing Database Instance Details



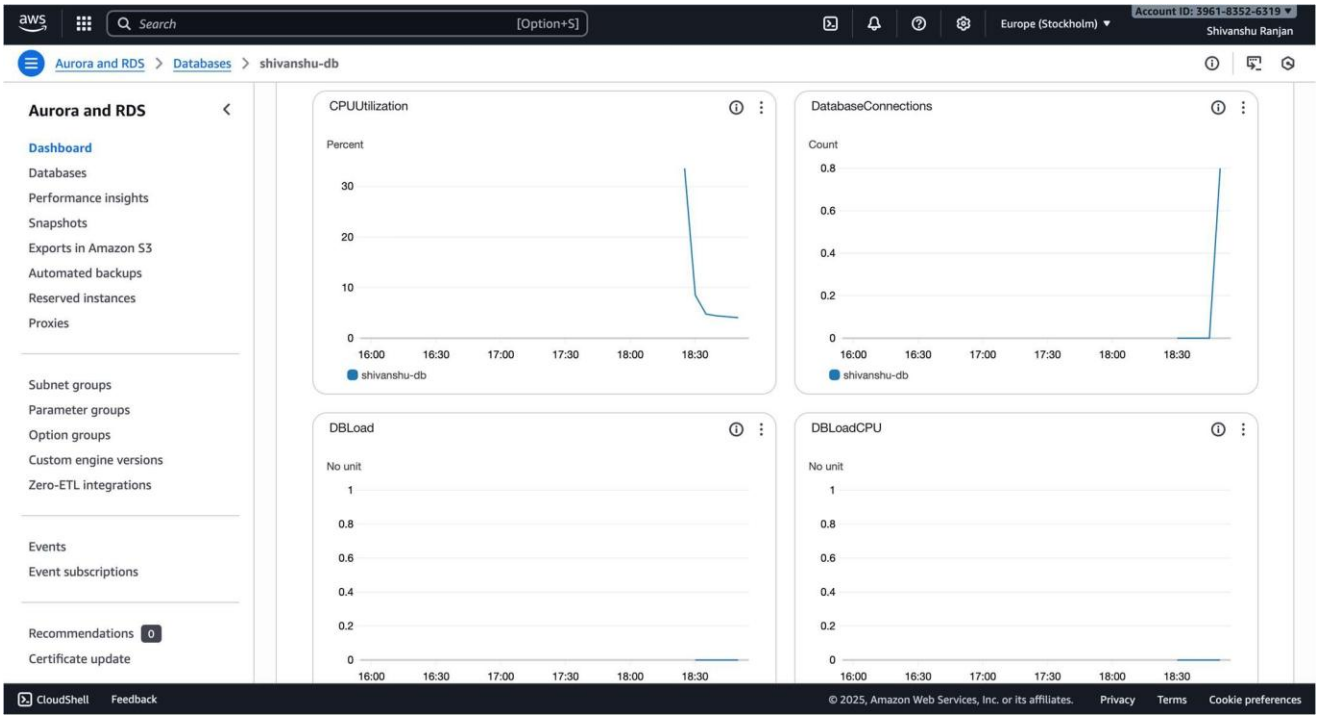
15. Copying the RDS Endpoint for Connection



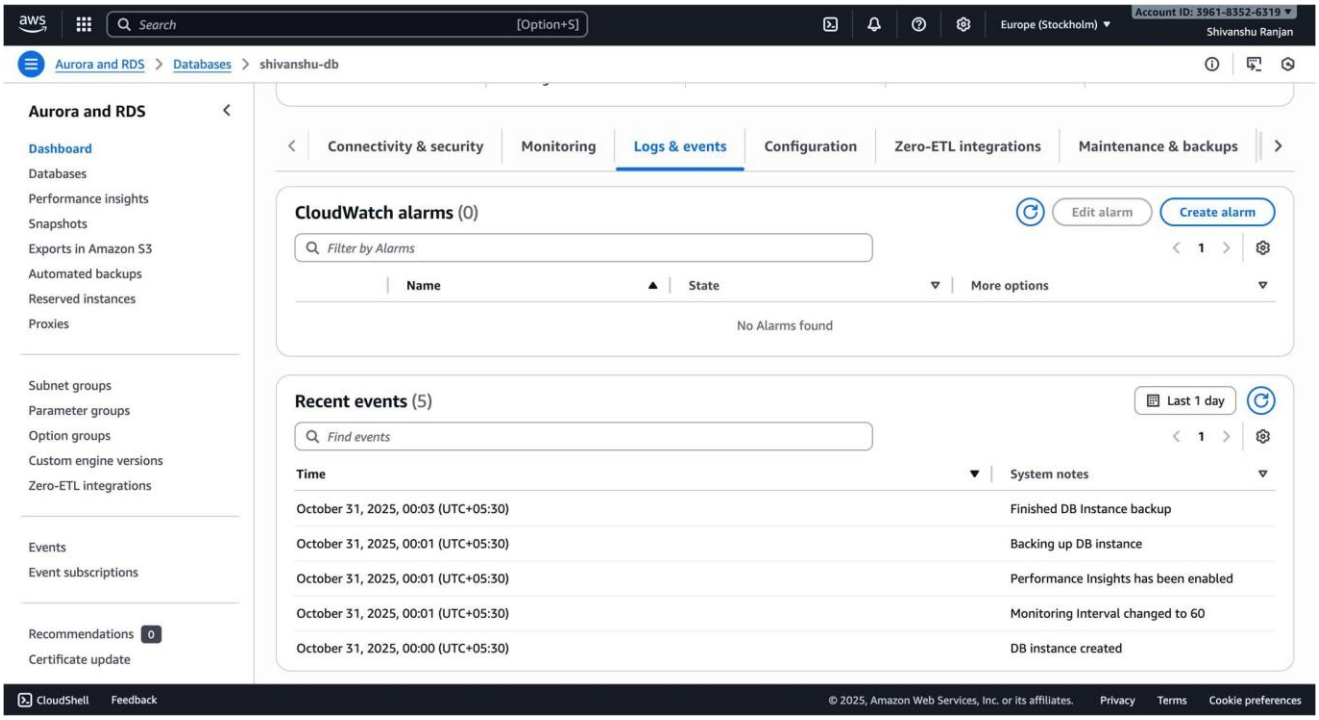
16. Launching pgAdmin on Local Machine



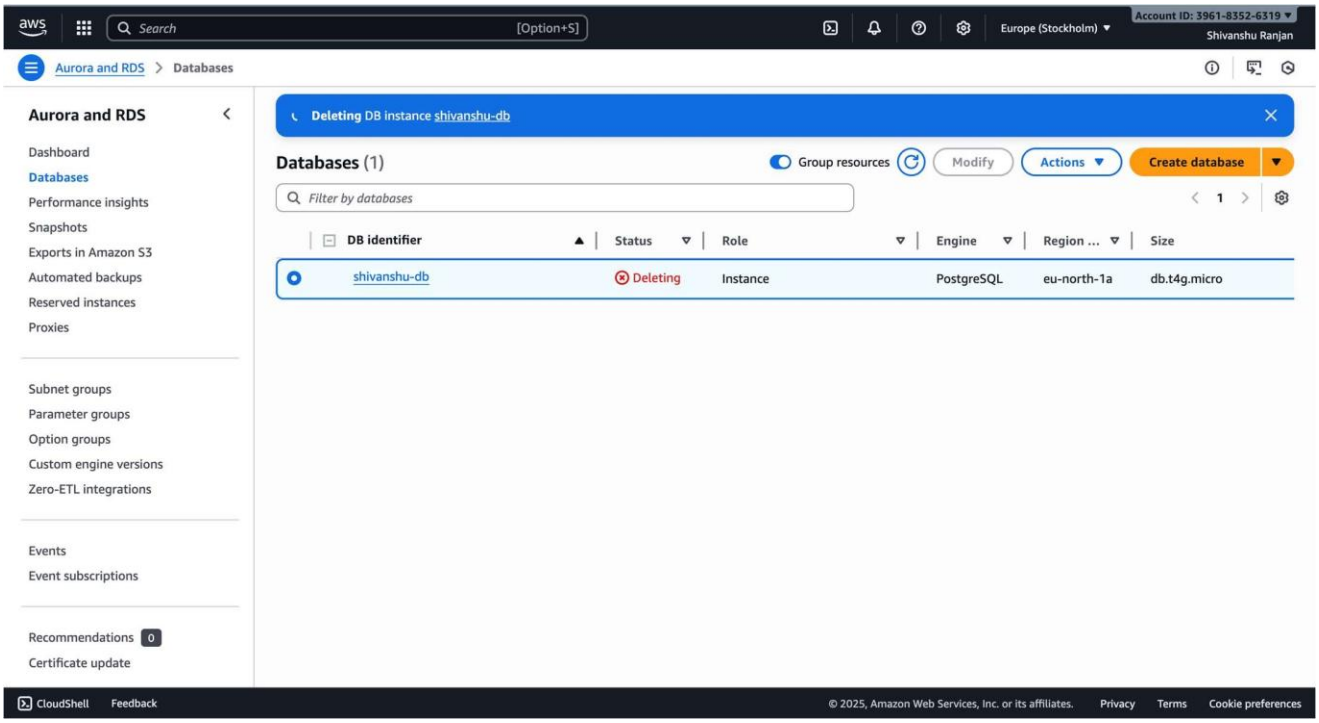
17. Adding a New Server in pgAdmin



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



19. Successful Connection to AWS RDS Database via pgAdmin



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.