



Worksheet 9

Student Name: Avreet Kaur

Branch: CSE

Semester: 5th

Subject Name: ADBMS

UID: 23BCS12246

Section/Group: KRG 3-A

Date of Performance: 30/10/2025

Subject Code: 23CSP-333

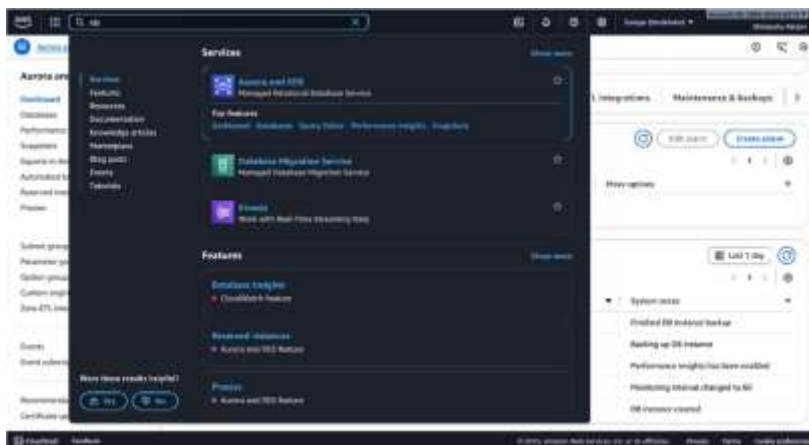
1. Aim: To understand and implement the setup of Amazon Relational Database Service (AWS RDS) by creating a database instance, configuring security groups, and establishing a secure connection between the local pgAdmin tool and the RDS instance hosted on the AWS Cloud.

2. Objective:

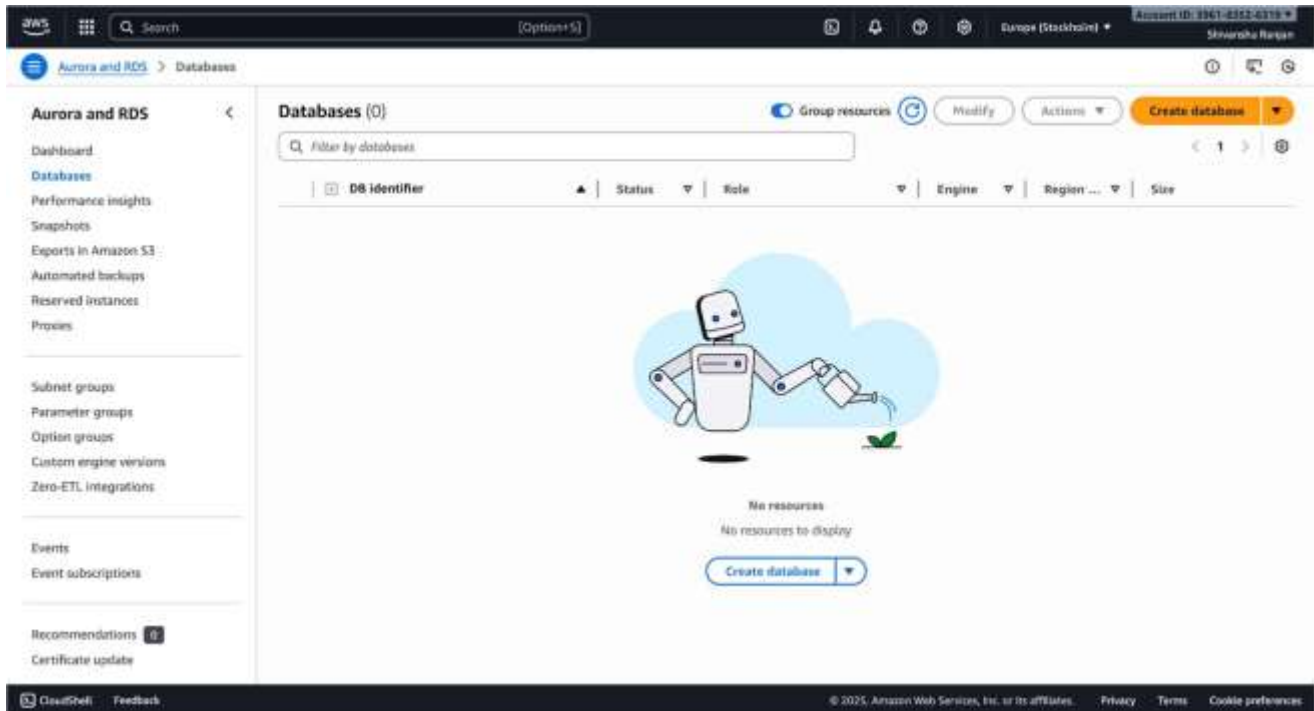
- To learn the basic concepts and features of Amazon Relational Database Service (AWS RDS).
- To create and configure a new RDS database instance on the AWS Management Console.
- To understand the role and configuration of security groups for controlling database access.
- To connect a local pgAdmin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

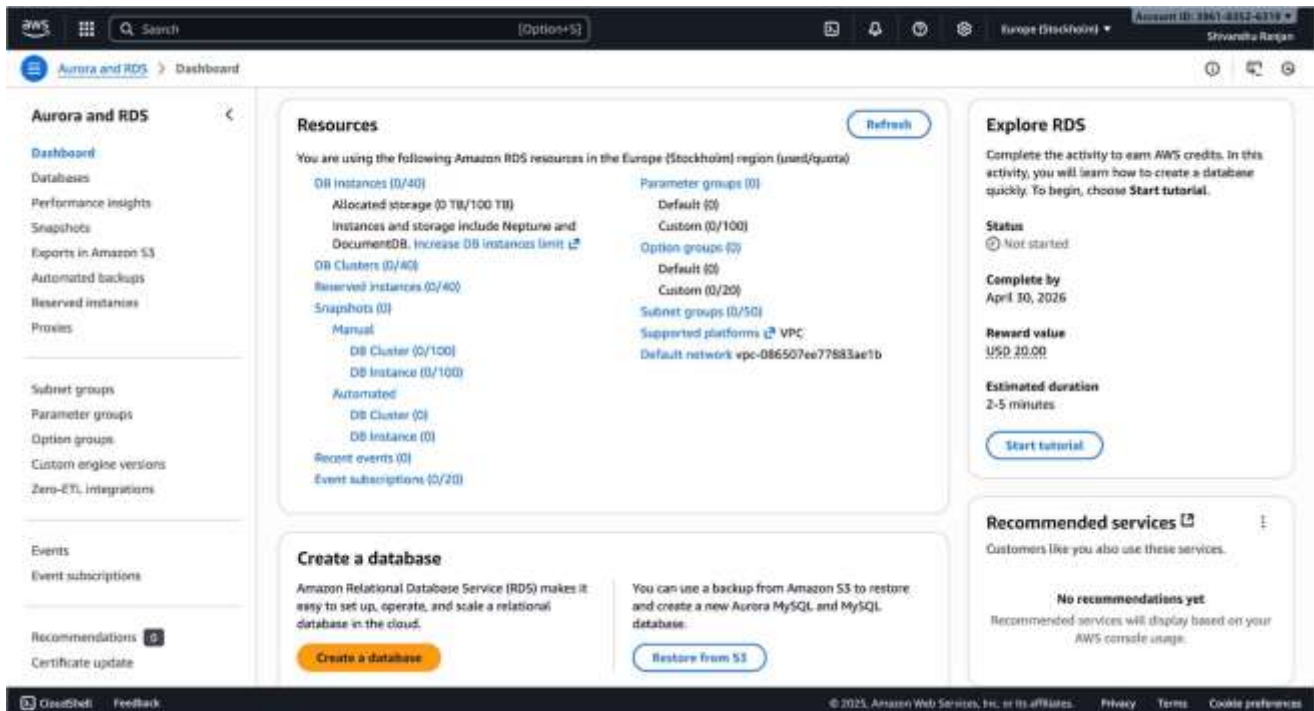
1. Sign-in



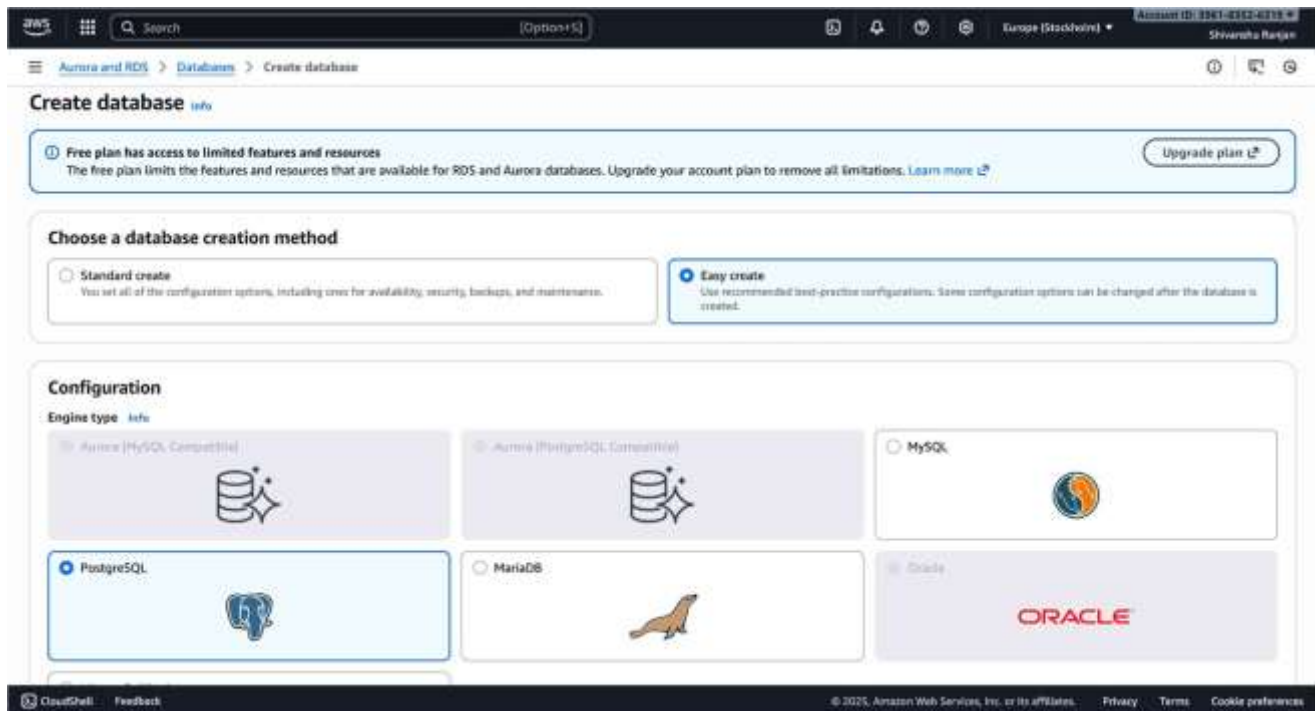
2. Navigating to RDS Service



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)
- ☐ MySQL
- ☒ PostgreSQL
- ☐ MariaDB
- ☐ 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

Master username [info](#)
Type a login ID for the master user of your DB instance.
postgres

Credentials management
You can use AWS Secrets Manager to manage your master user credentials.

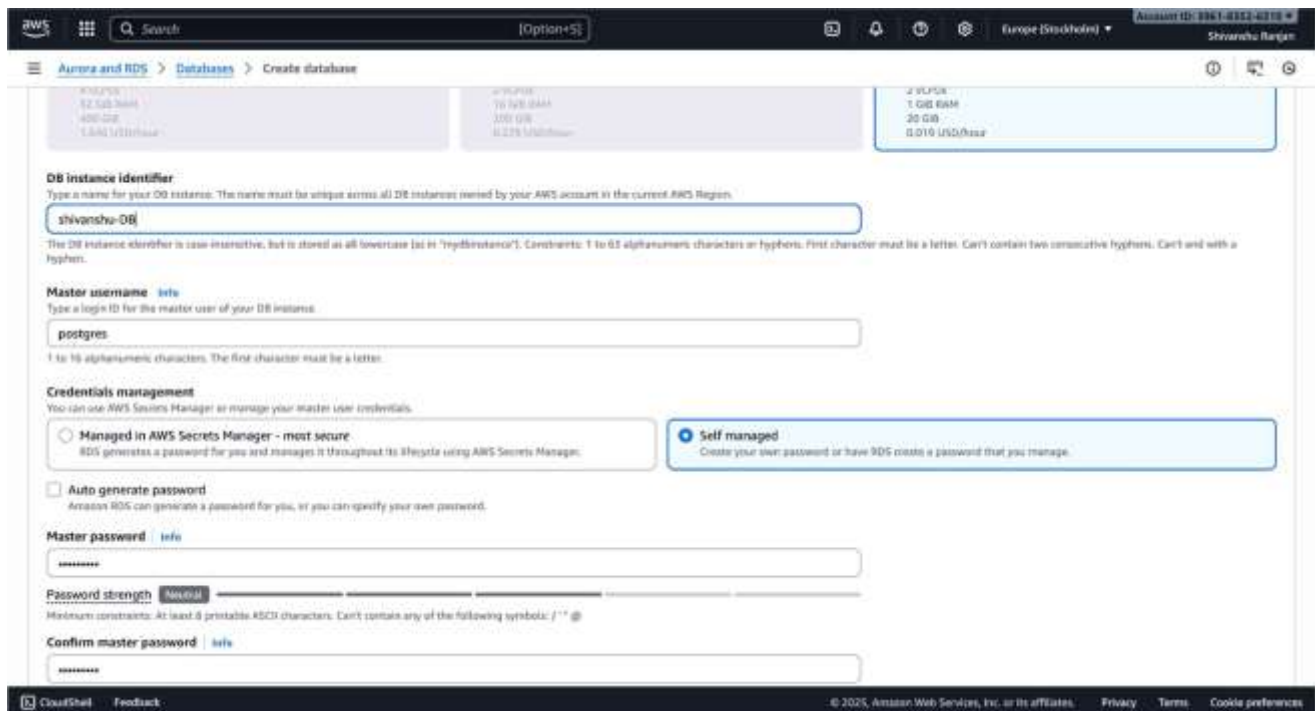
- ☐ Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
- ☒ Self managed
Create your own password or have RDS create a password that you manage.

☐ 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](#)

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

Master username [info](#)
Type a login ID for the master user of your DB instance.
postgres

Credentials management
You can use AWS Secrets Manager to 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:

Setting	Value	Yes/No
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 informational box stating: "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 "Cancel" and "Create database" buttons.

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

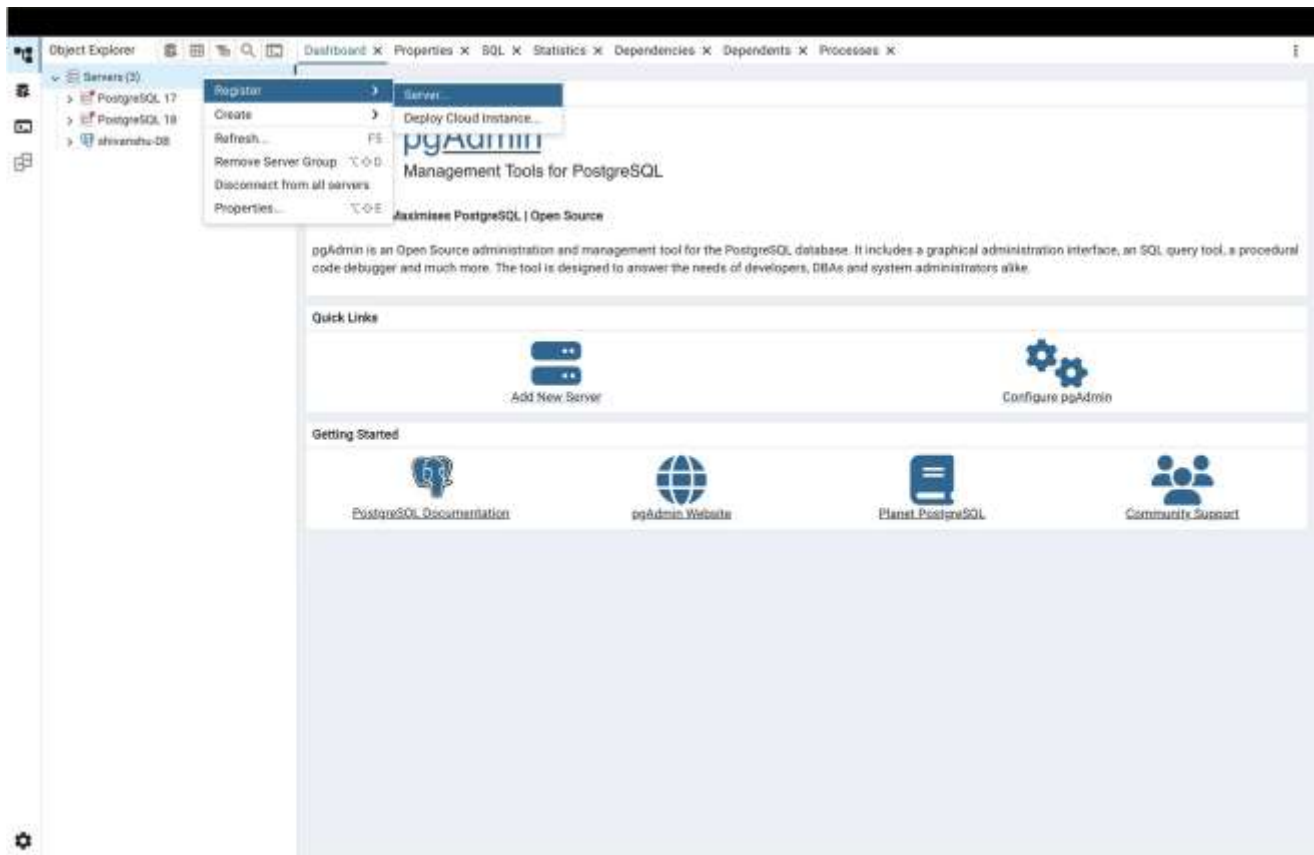
The screenshot shows the 'Databases' page in the AWS Management Console. A blue banner at the top indicates: "Creating database shivanshu-db. Your database might take a few minutes to launch. You can use settings from shivanshu-db to simplify configuration of suggested database add-ons while we finish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

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

DB identifier	Status	Role	Engine	Region	Size
shivanshu-db	Creating	Instance	PostgreSQL	Europe (Stockholm)	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.

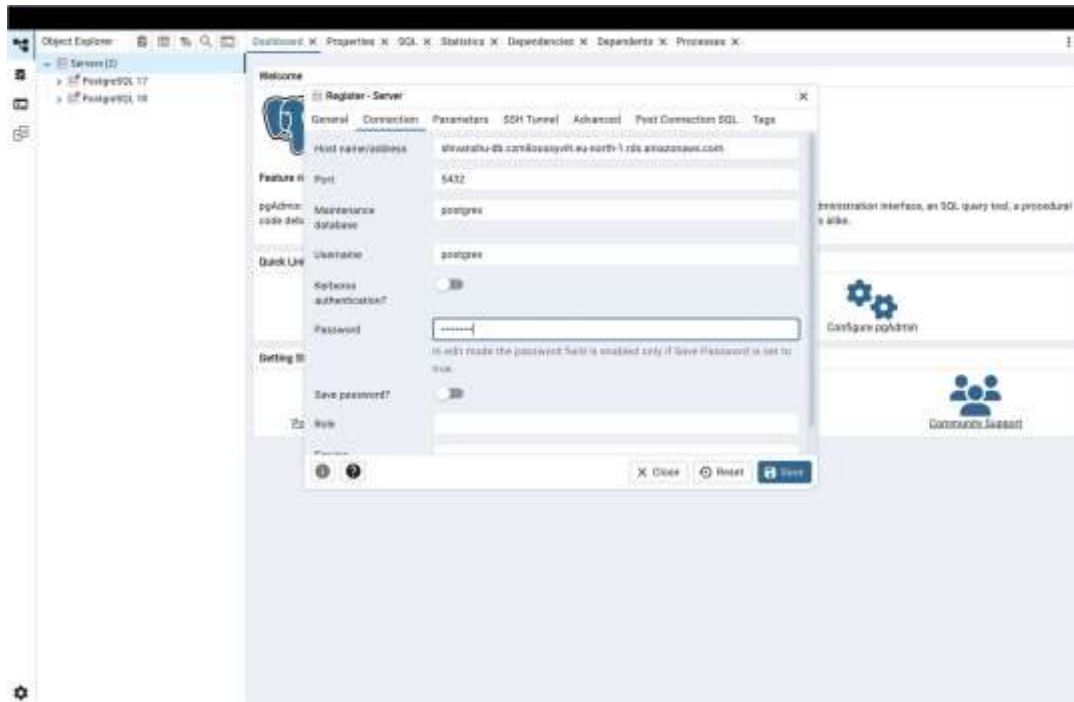
8. Setting Up Instance Size and Storage



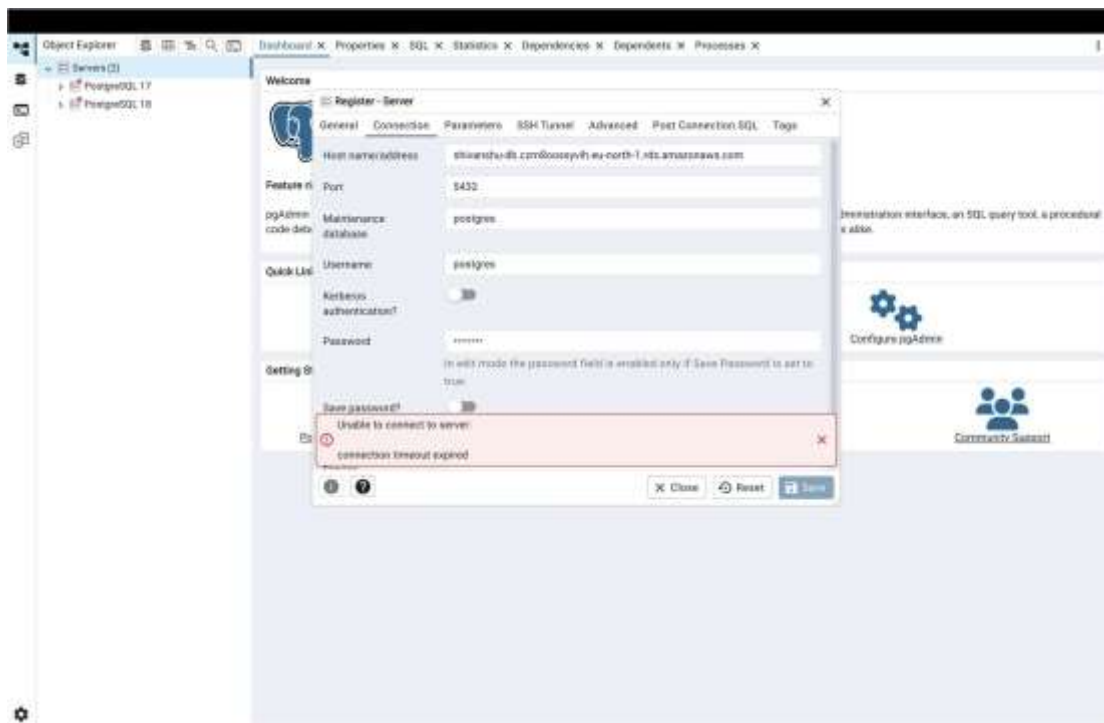
9. Configuring Connectivity and VPC Settings



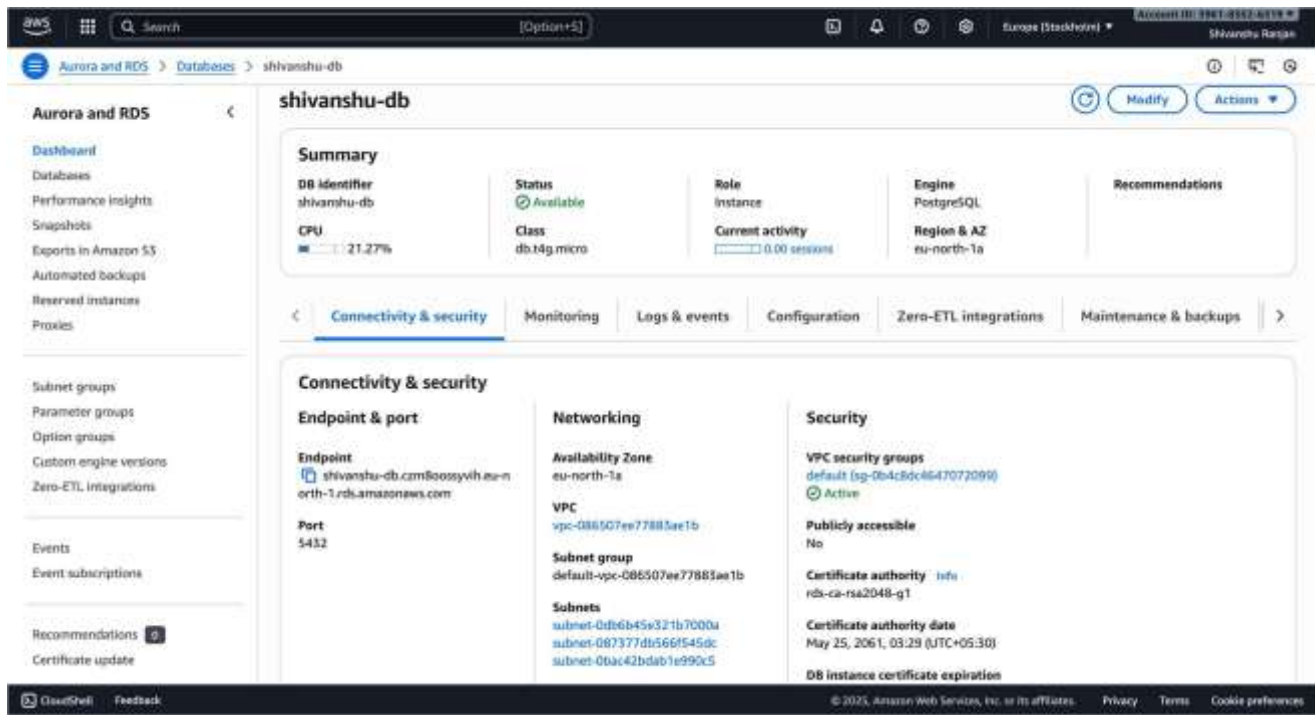
10. Gr Setting Up Security Groups for RDS Access



11. Additional Database Configuration Options

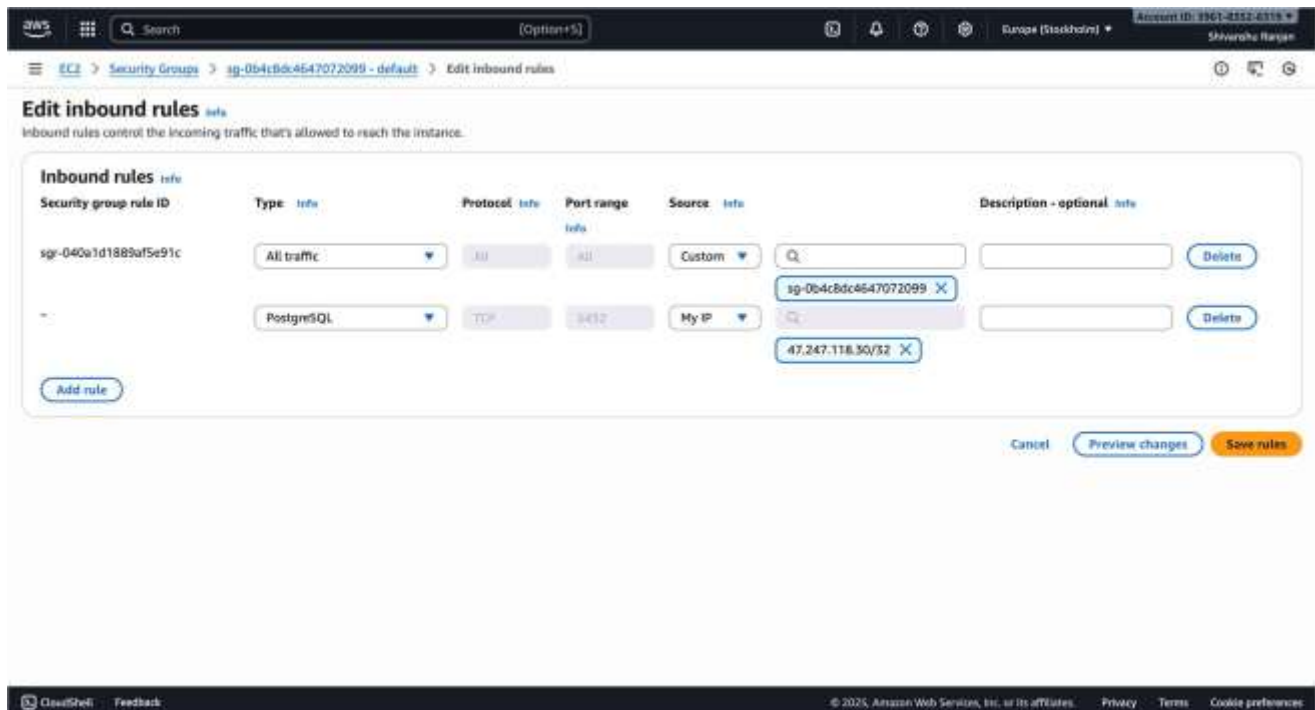


12. Reviewing and Creating the Database Instance



The screenshot displays the AWS Management Console for an Amazon RDS instance named 'shivanshu-db'. The console shows the instance's status as 'Available' and its class as 'db.t4g.micro'. The instance is located in the 'eu-north-1a' region. The 'Connectivity & security' tab is selected, showing the endpoint 'shivanshu-db.czm8oosyvh.az-north-1.rds.amazonaws.com' on port 5432. The VPC is 'default-vpc-086507ee77883ae1b' and the subnet group is 'default-vpc-086507ee77883ae1b'. The security group is 'sg-0b4c8dc4647072099'.

13. RDS Instance Creation in Progress



The screenshot displays the AWS Management Console for the 'Edit inbound rules' page of a security group. The page shows a table of inbound rules with columns for 'Security group rule ID', 'Type', 'Protocol', 'Port range', 'Source', and 'Description - optional'. There are two rules listed: one for 'All traffic' and another for 'PostgreSQL'. The 'PostgreSQL' rule is selected, and its details are shown in the 'Source' field, including 'My IP' and '47.247.118.30/32'.

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.czm8oossyviu.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

rds-ca-rsa2048-g1

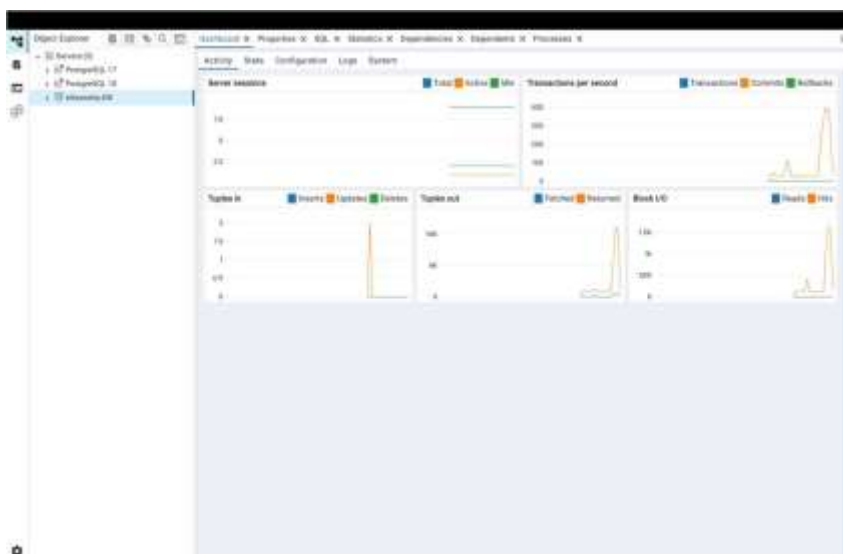
Certificate authority date

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

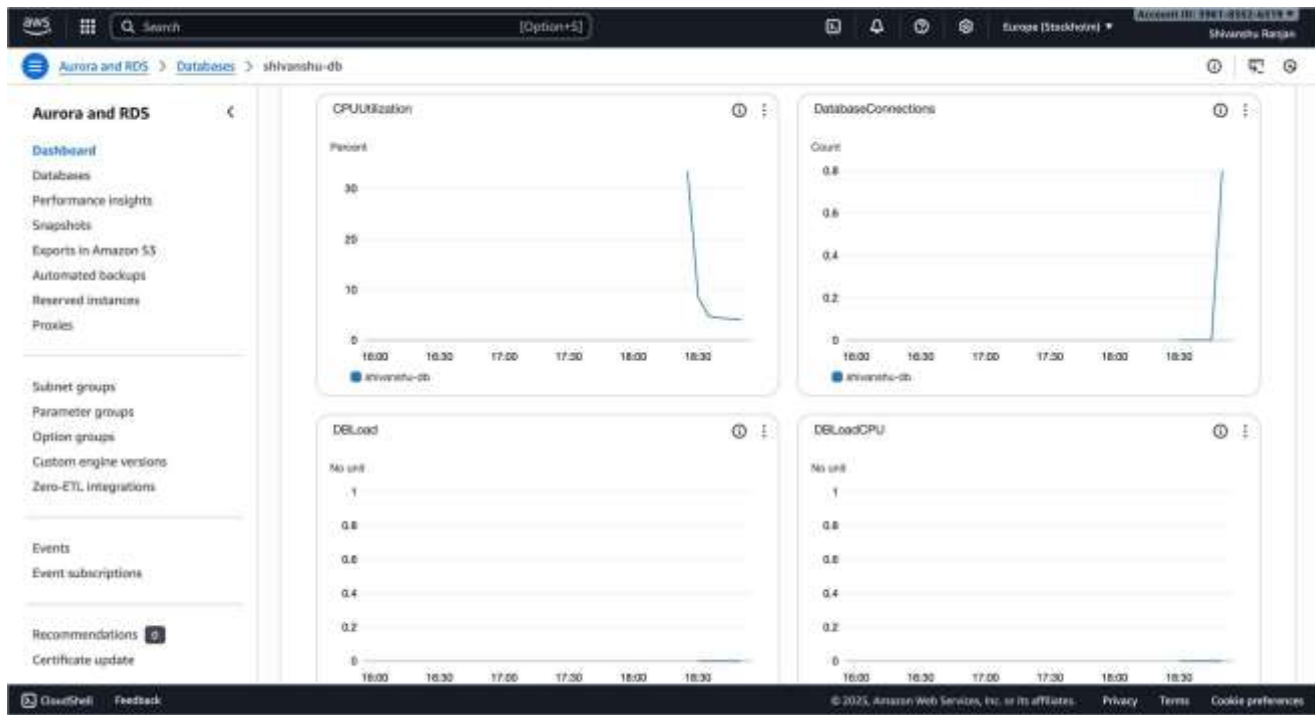
DB instance certificate expiration date

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

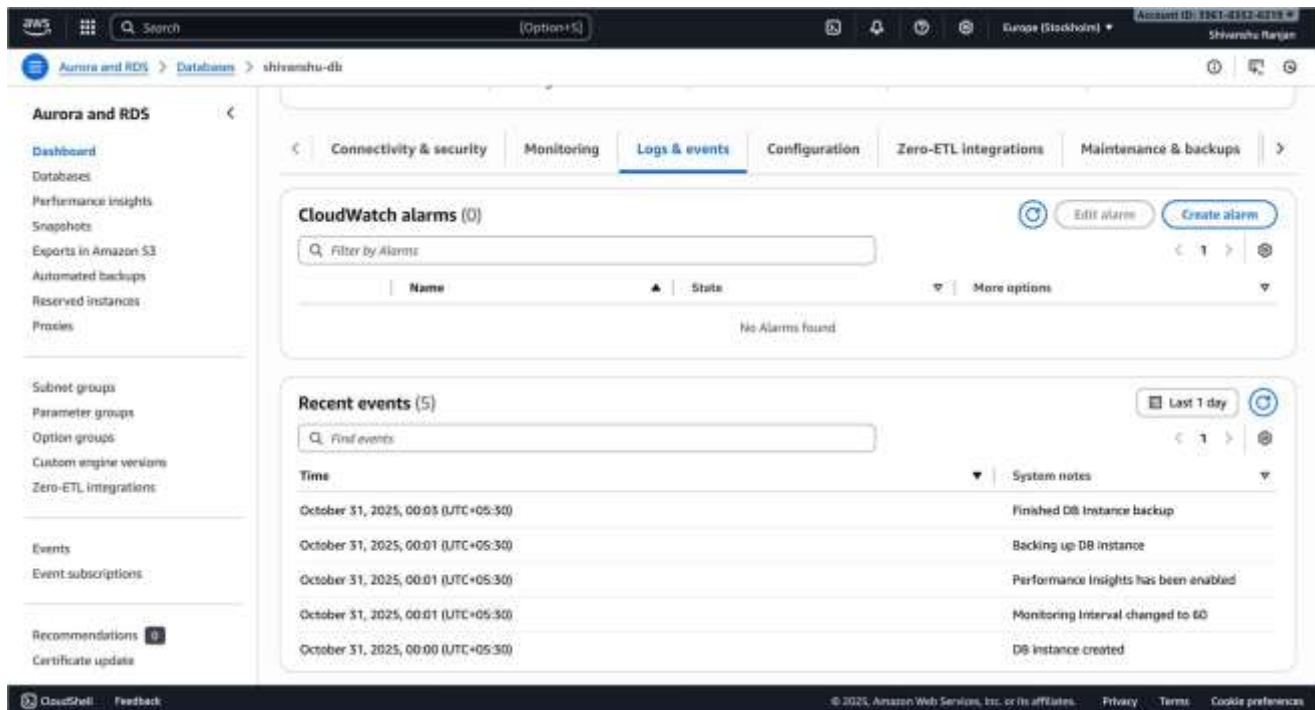
16. Launching pgAdmin on Local Machine



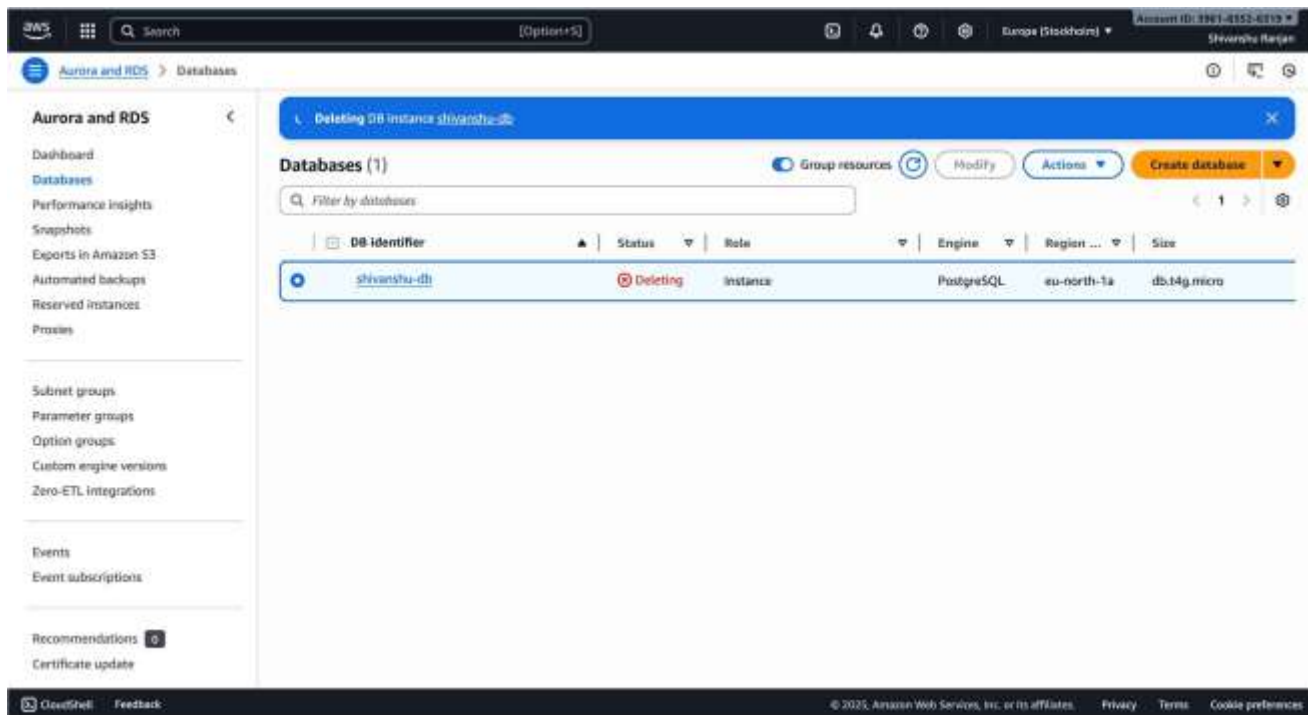
17. Adding a New Server in pgAdmin



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



19. Successful Connection to AWS RDS Database via pgAdmin



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.