

Worksheet 9

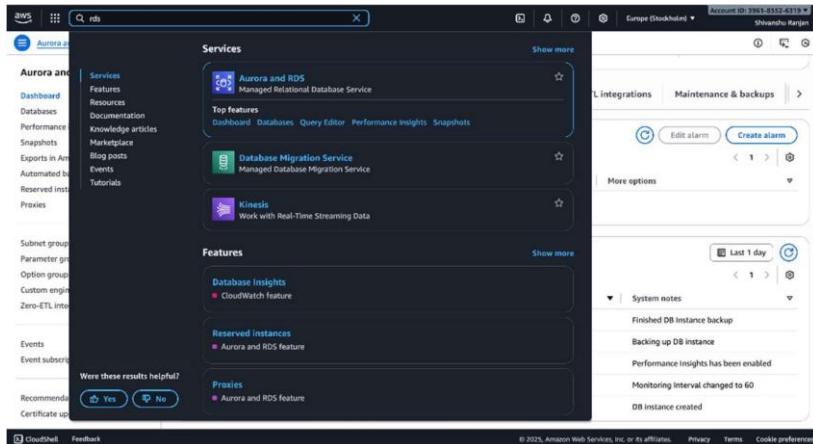
Aim: To understand and implement the setup of Amazon Relational Database Service (AWS RDS) by creating a database instance, configuring security groups, and establishing a secure connection between the local pgAdmin tool and the RDS instance hosted on the AWS Cloud.

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



Discover. Learn. Empower.

2. Navigating to RDS Service



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Aurora and RDS > Databases

Databases (0)

Filter by databases

DB identifier	Status	Role

No resources

No resources to display

Create database

Aurora and RDS

- 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

Certificate update

CloudShell Feedback

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

3. Amazon RDS Dashboard Overview

Aurora and RDS > Dashboard

Resources

You are using the following Amazon RDS resources in the Europe (Stockholm) region (used/quota)

DB Instances (0/40)	Parameter groups (0)
Allocated storage (0 TB/100 TB)	Default (0)
Instances and storage include Neptune and DocumentDB. Increase DB instances limit ↗	Custom (0/100)
DB Clusters (0/40)	Option groups (0)
Reserved instances (0/40)	Default (0)
Snapshots (0)	Custom (0/20)
Manual	Subnet groups (0/50)
DB Cluster (0/100)	Supported platforms ↗ VPC
DB Instance (0/100)	Default network vpc-086507ee77883ae1b
Automated	
DB Cluster (0)	
DB Instance (0)	
Recent events (0)	
Event subscriptions (0/20)	

Refresh

Create a database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Create a database

You can use a backup from Amazon S3 to restore and create a new Aurora MySQL and MySQL database.

Restore from S3

Explore RDS

Complete the activity to earn AWS credits. In this activity, you will learn how to create a database quickly. To begin, choose Start tutorial.

Status

Not started

Complete by

April 30, 2026

Reward value

USD 20.00

Estimated duration

2-5 minutes

Start tutorial

Recommended services

Customers like you also use these services.

No recommendations yet

Recommended services will display based on your AWS console usage.

CloudShell Feedback

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

Discover. Learn. Empower.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

4. Creating a New Database Instance

The screenshot shows the 'Create database' step in the AWS RDS console. At the top, there's a banner for the 'Free plan has access to limited features and resources'. Below it, two creation methods are shown: 'Standard create' (disabled) and 'Easy create' (selected). The 'Configuration' section lists several engine types: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL, PostgreSQL (selected), MariaDB, and Oracle. The PostgreSQL card is highlighted with a blue border. At the bottom, there are links for CloudShell, Feedback, and copyright information.

5. Selecting PostgreSQL as Database Engine

This screenshot shows the configuration details for a PostgreSQL database instance. It includes sections for DB instance identifier ('shivanshu-Db'), master username ('postgres'), credentials management (choosing 'Self managed'), and password settings. Two DB instance options are listed: one with 4 vCPUs and 32 GiB RAM, and another with 2 vCPUs and 1 GiB RAM. The 'Self managed' password field contains '*****'. At the bottom, there are links for CloudShell, Feedback, and copyright information.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

6. Choosing Deployment Option and Template

The screenshot shows the 'Create database' step in the AWS RDS wizard. The configuration details are as follows:

Setting	Value	Status
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

A note at the bottom states: "ⓘ 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."

Buttons at the bottom right include 'Cancel' and 'Create database'.

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

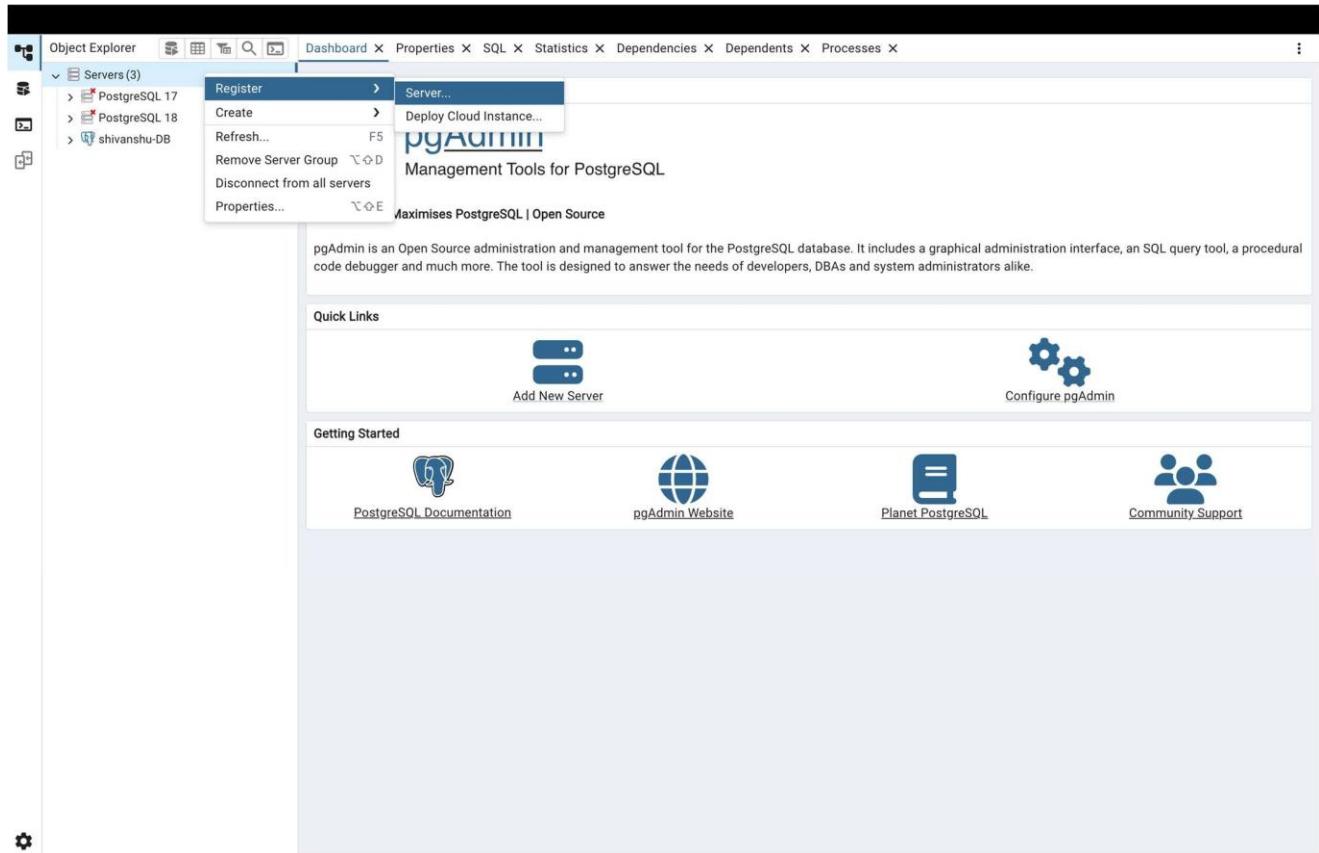
The screenshot shows the 'Creating database shivanshu-db' status page. A message indicates the database might take a few minutes to launch. The 'Databases (1)' section shows the following details:

DB identifier	Status	Role	Engine	Region	Size
shivanshu-db	Creating	Instance	PostgreSQL	-	db.t4g.micro

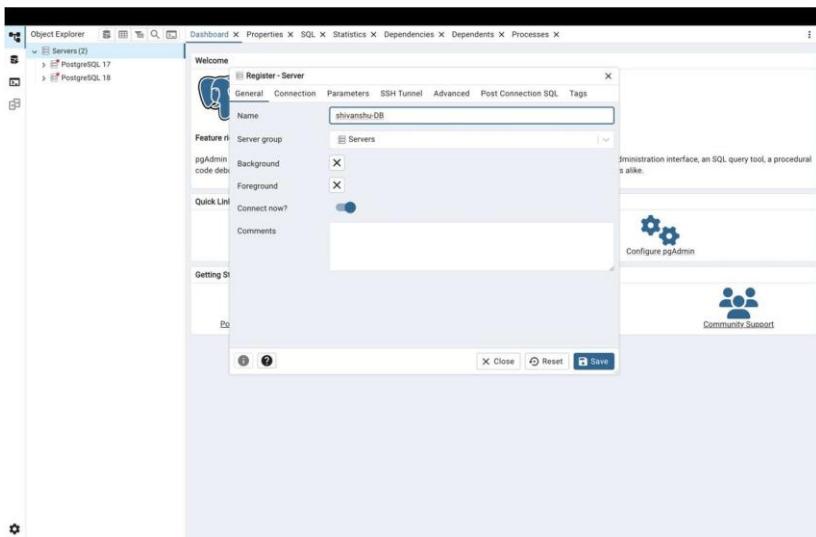
Other sections visible on the left include 'Dashboard', 'Databases', 'Performance insights', 'Schemas', 'Exports in Amazon S3', 'Automated backups', 'Reserved instances', 'Proxies', 'Subnet groups', 'Parameter groups', 'Option groups', 'Custom engine versions', 'Zero-ETL integrations', 'Events', 'Event subscriptions', 'Recommendations (0)', and 'Certificate update'.

At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

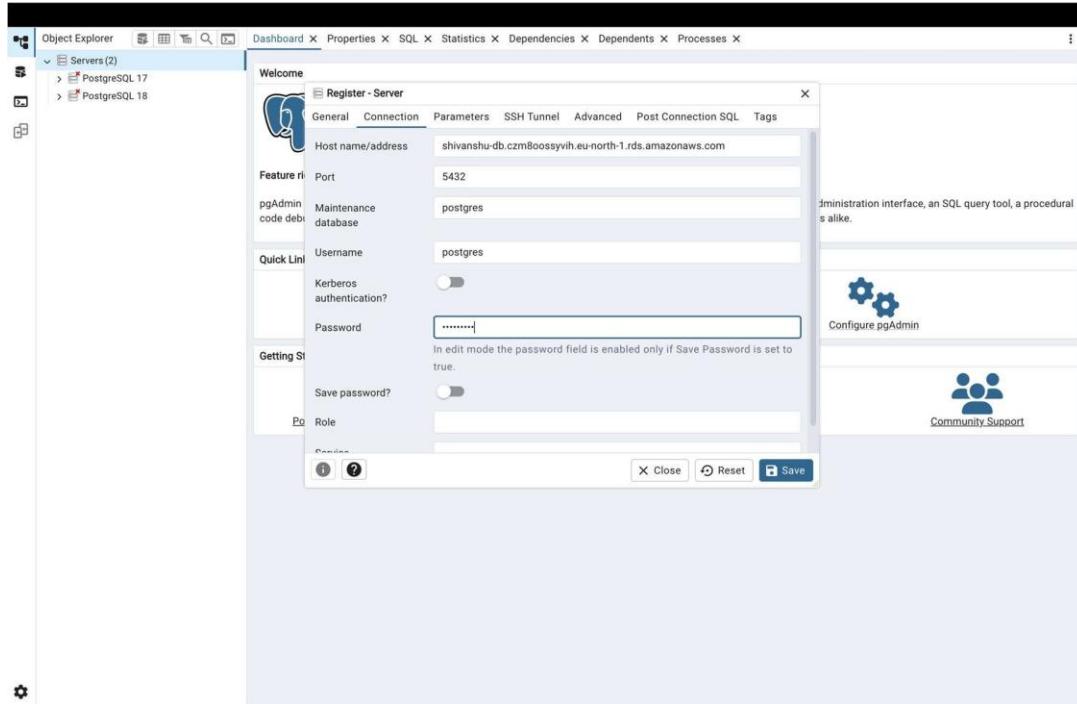
8. Setting Up Instance Size and Storage



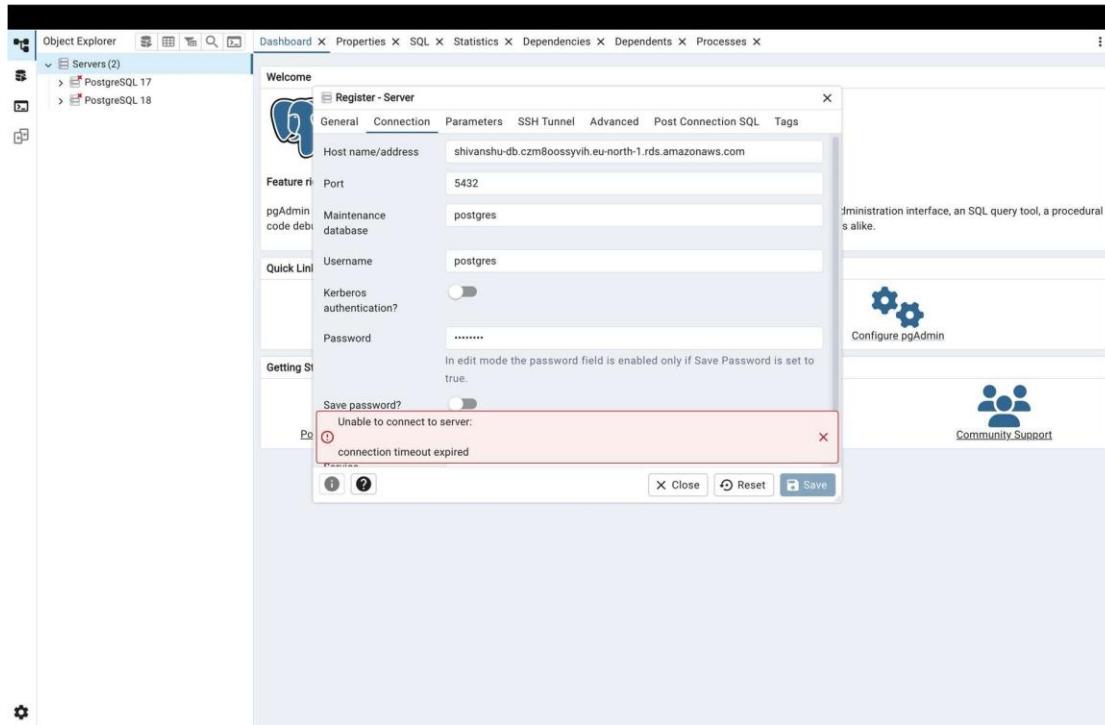
9. Configuring Connectivity and VPC Settings



10. Gr Setting Up Security Groups for RDS Access



11. Additional Database Configuration Options





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

12. Reviewing and Creating the Database Instance

The screenshot shows the AWS Aurora and RDS console. The main view displays the 'Summary' of the 'shivanshu-db' database instance. Key details include:

- DB identifier:** shivanshu-db
- Status:** Available
- Role:** Instance
- Engine:** PostgreSQL
- Region & AZ:** eu-north-1a

The CPU usage is shown as 21.27%. Below the summary, there are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Zero-ETL integrations', and 'Maintenance & backups'. On the left sidebar, there are links for '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'. At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information.

13. RDS Instance Creation in Progress

The screenshot shows the AWS EC2 Security Groups console. The user is editing the inbound rules for the default security group ('sg-0b4c8dc4647072099'). The current rule table is as follows:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-040a1d1889af5e91c	All traffic	All	All	Custom	sg-0b4c8dc4647072099
-	PostgreSQL	TCP	5432	My IP	47.247.118.30/32

At the bottom, there are buttons for 'Cancel', 'Preview changes', and 'Save rules'.

Discover. Learn. Empower.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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](#)

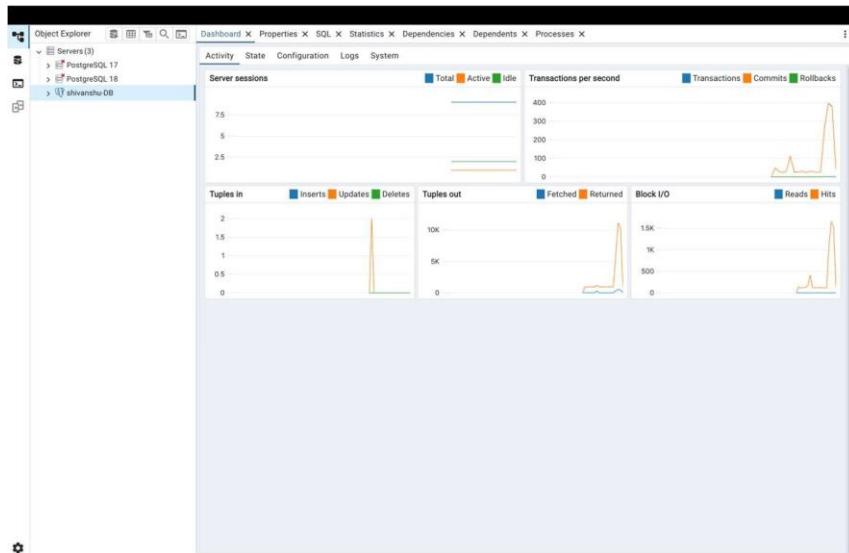
5432

15. Copying the RDS Endpoint for Connection

Connectivity & security

Endpoint & port	Networking	Security
Endpoint shivanshu-db.czm8oossyvih.eu-north-1.rds.amazonaws.com	Availability Zone eu-north-1a	VPC security groups default (sg-0b4c8dc4647072099) <input checked="" type="checkbox"/> Active
Port 5432	VPC vpc-086507ee77883ae1b	Publicly accessible Yes
	Subnet group default-vpc-086507ee77883ae1b	Certificate authority Info rds-ca-rsa2048-g1
	Subnets subnet-0db6b45e321b7000a subnet-087377db566f545dc subnet-0bac42bdab1e990c5	Certificate authority date May 25, 2061, 03:29 (UTC+05:30)
	Network type IPv4	DB instance certificate expiration date October 30, 2026, 23:59 (UTC+05:30)

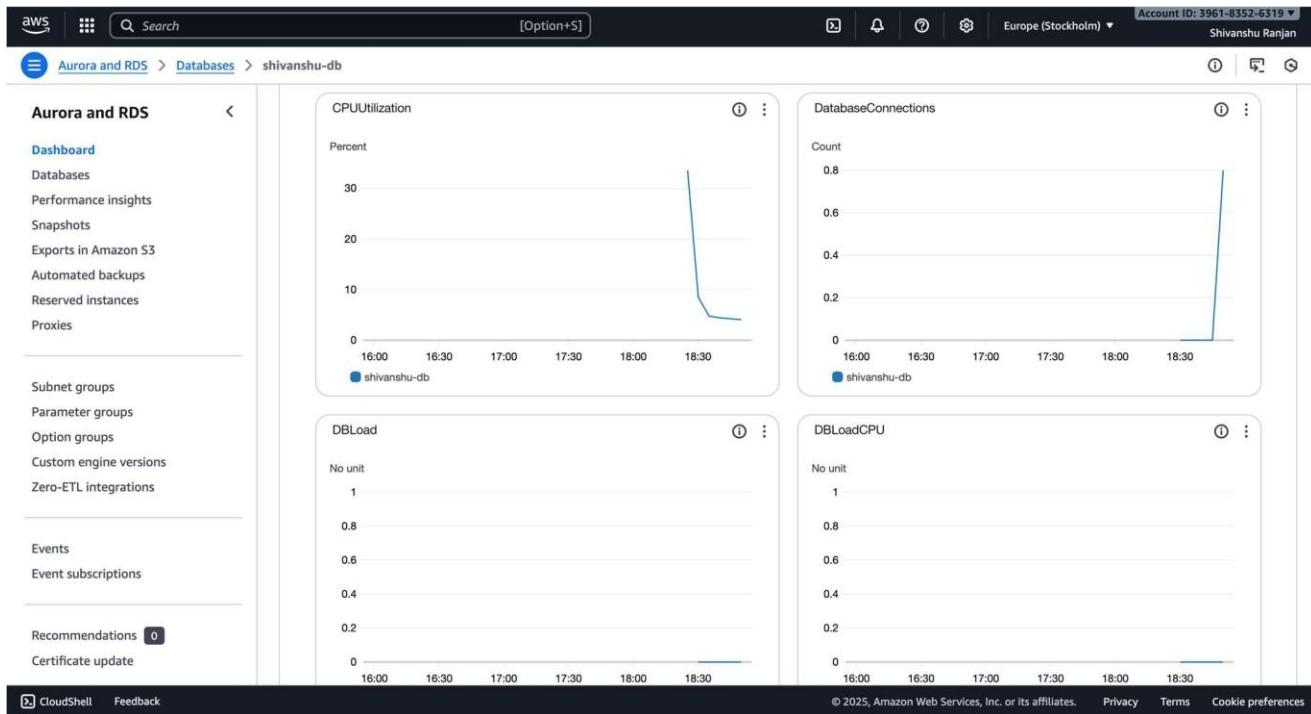
16. Launching pgAdmin on Local Machine



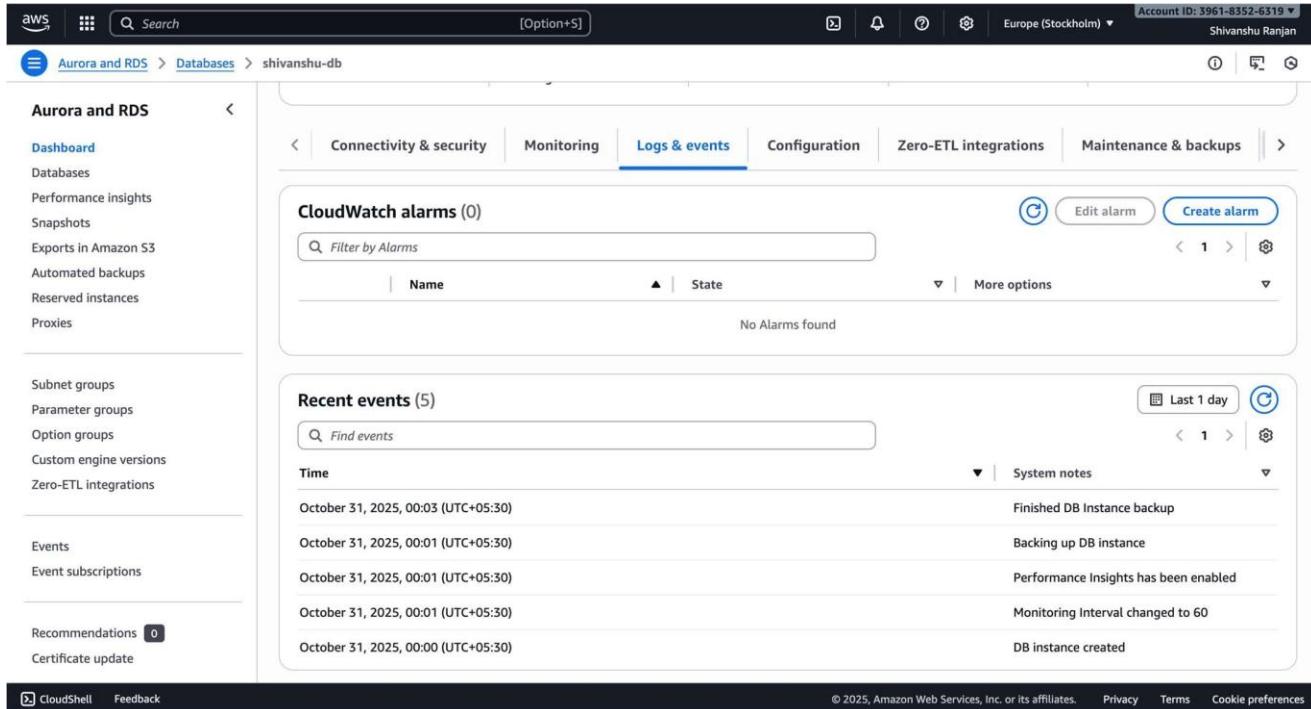


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

17. Adding a New Server in pgAdmin



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



Discover. Learn. Empower.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

19. Successful Connection to AWS RDS Database via pgAdmin

The screenshot shows the AWS RDS (Aurora and RDS) console. On the left, there's a sidebar with navigation links 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 area is titled 'Deleting DB instance shivanshu-db' and shows a table of databases. The table has columns for DB identifier, Status, Role, Engine, Region ..., and Size. One row is selected, showing 'shivanshu-db' with a status of 'Deleting', 'Instance' type, PostgreSQL engine, eu-north-1a region, and db.t4g.micro size. At the top right of the main area, there are buttons for Group resources, Modify, Actions (with a dropdown menu), Create database, and other options. The top bar also displays the account ID (3961-8352-6319), region (Europe (Stockholm)), and user (Shivanshu Ranjan). The bottom of the screen includes CloudShell, Feedback, and standard footer links.

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.