

## Worksheet 9

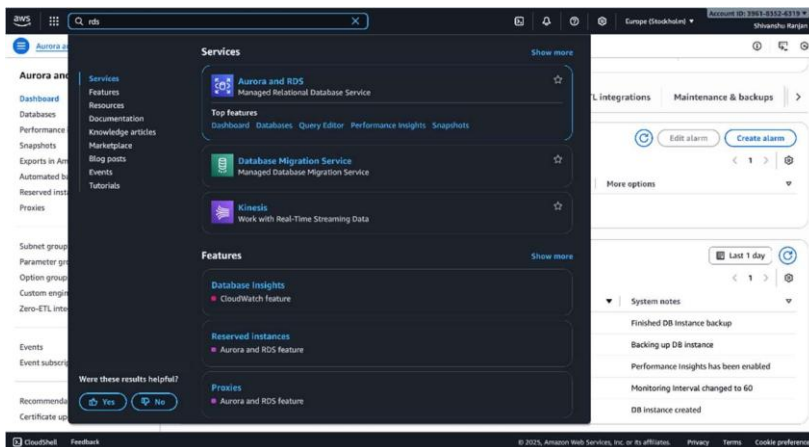
**Aim:** To understand and implement the setup of Amazon Relational Database Service (AWS RDS) by creating a database instance, configuring security groups, and establishing a secure connection between the local pgAdmin tool and the RDS instance hosted on the AWS Cloud.

### 1. Objective:

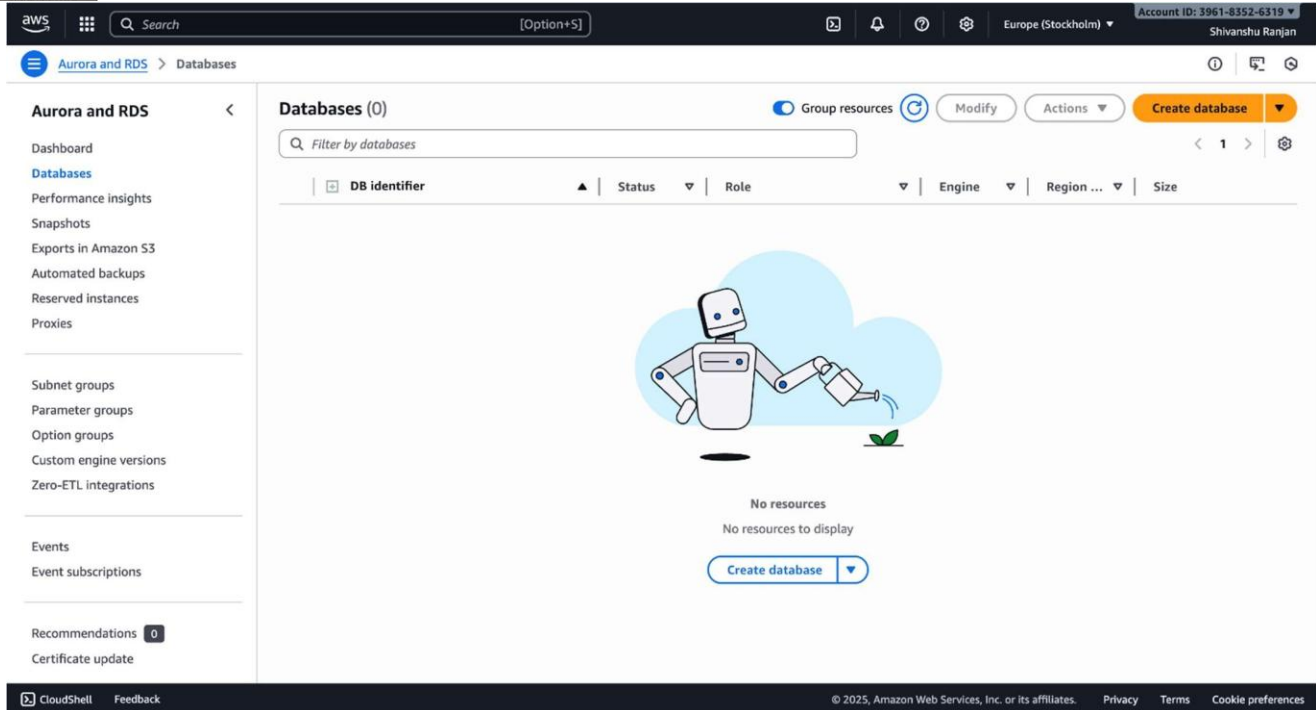
- To learn the basic concepts and features of Amazon Relational Database Service (AWS RDS).
- To create and configure a new RDS database instance on the AWS Management Console.
- To understand the role and configuration of security groups for controlling database access.
- To connect a local pgAdmin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



aws [Option+S] Search Europe (Stockholm) Account ID: 3961-8352-6319 Shivanshu Ranjan

Aurora and RDS > Databases

**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

**Databases (0)**

Filter by databases

DB identifier Status Role Engine Region ... Size

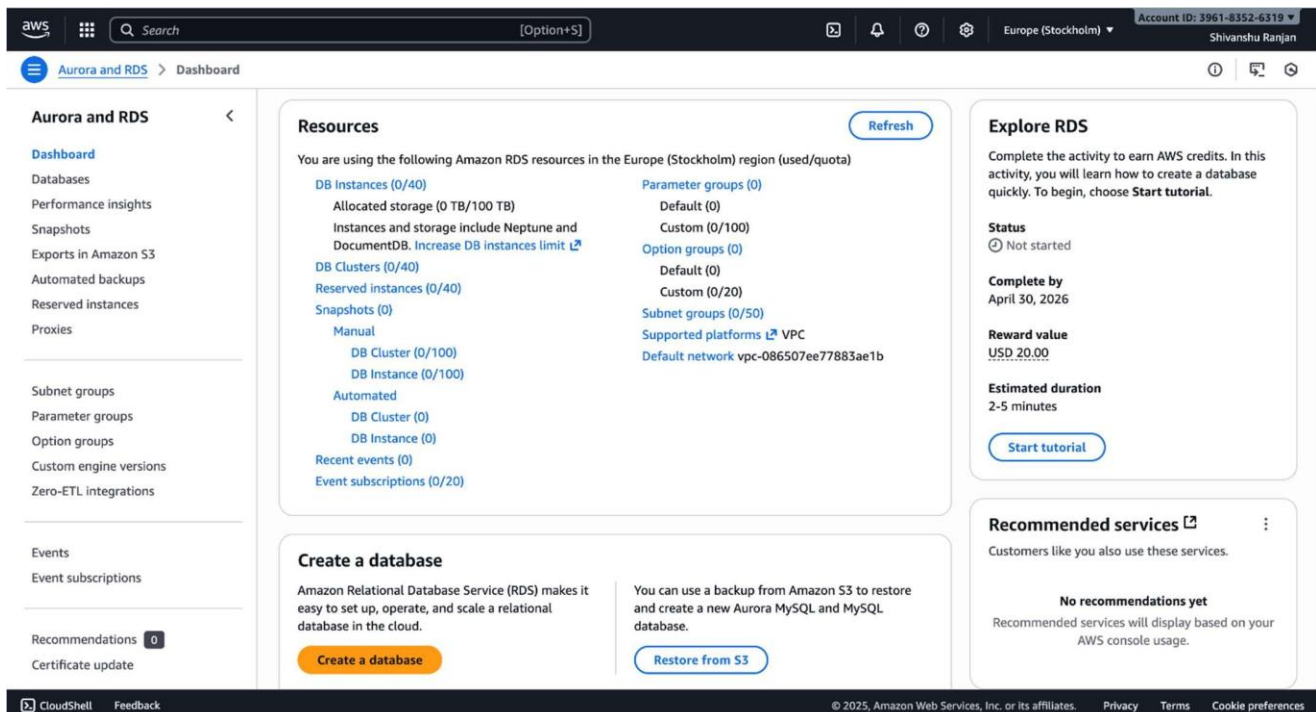
No resources

No resources to display

Create database

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

### 3. Amazon RDS Dashboard Overview



aws [Option+S] Search Europe (Stockholm) Account ID: 3961-8352-6319 Shivanshu Ranjan

Aurora and RDS > Dashboard

**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

**Resources**

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

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

**Create a database**

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

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

Create a 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**

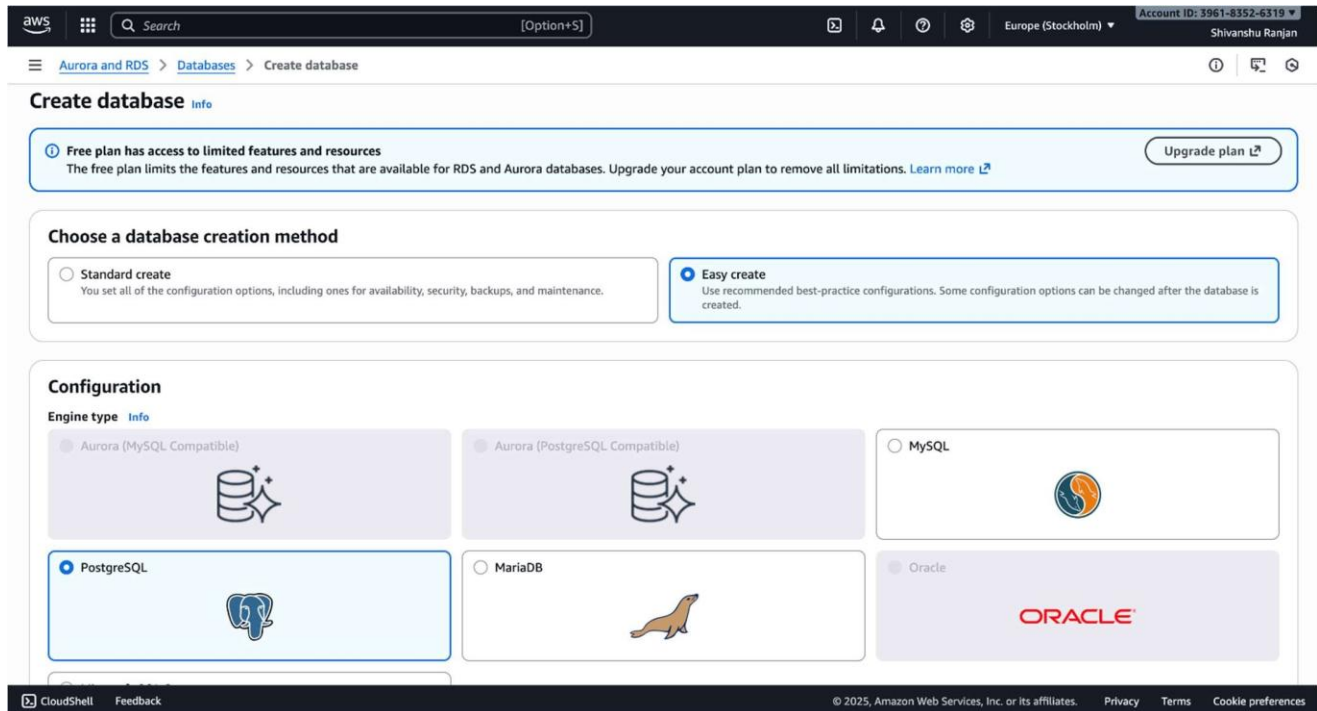
Customers like you also use these services.

**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

## 4. Creating a New Database Instance



The screenshot shows the AWS Management Console 'Create database' page. The top navigation bar includes the AWS logo, a search bar, and account information (Europe (Stockholm), Account ID: 3961-8352-6319, Shivanshu Ranjan). The breadcrumb trail is 'Aurora and RDS > Databases > Create database'.

**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.

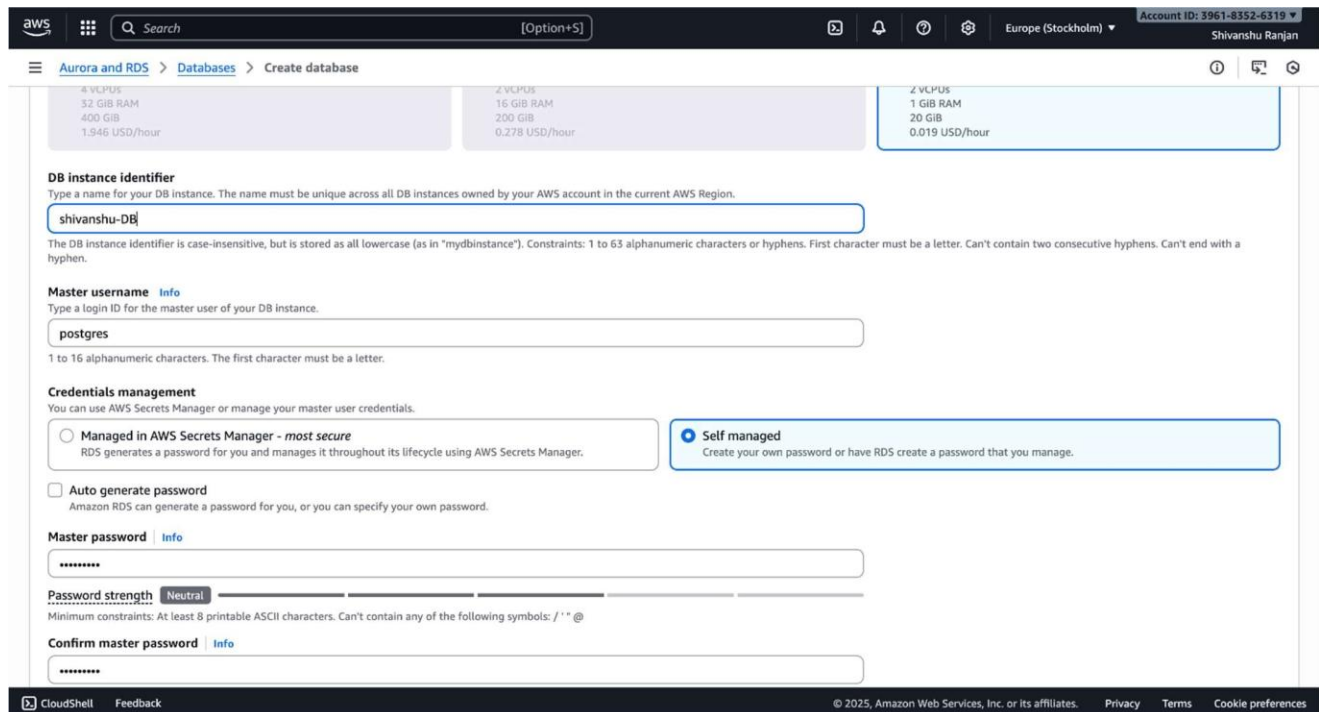
**Configuration**

**Engine type** info

- ☐ Aurora (MySQL Compatible)
- ☐ Aurora (PostgreSQL Compatible)
- ☐ MySQL
- ☒ PostgreSQL
- ☐ MariaDB
- ☐ Oracle

The bottom of the page shows a footer with 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

## 5. Selecting PostgreSQL as Database Engine



This screenshot shows the 'Create database' page with PostgreSQL selected as the engine type. The top navigation bar and breadcrumb trail are the same as in the previous screenshot.

**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.  
  
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.  
  
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.

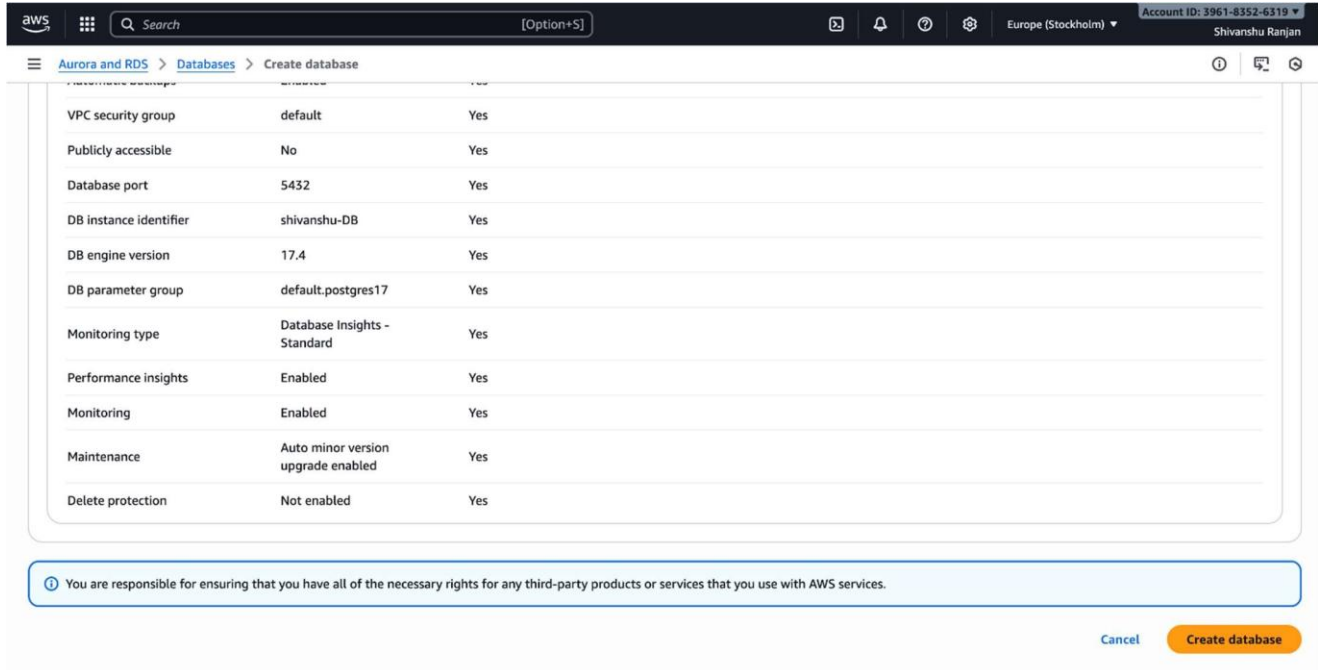
**Master password** info

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

**Confirm master password** info

The bottom of the page shows a footer with 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

## 6. Choosing Deployment Option and Template

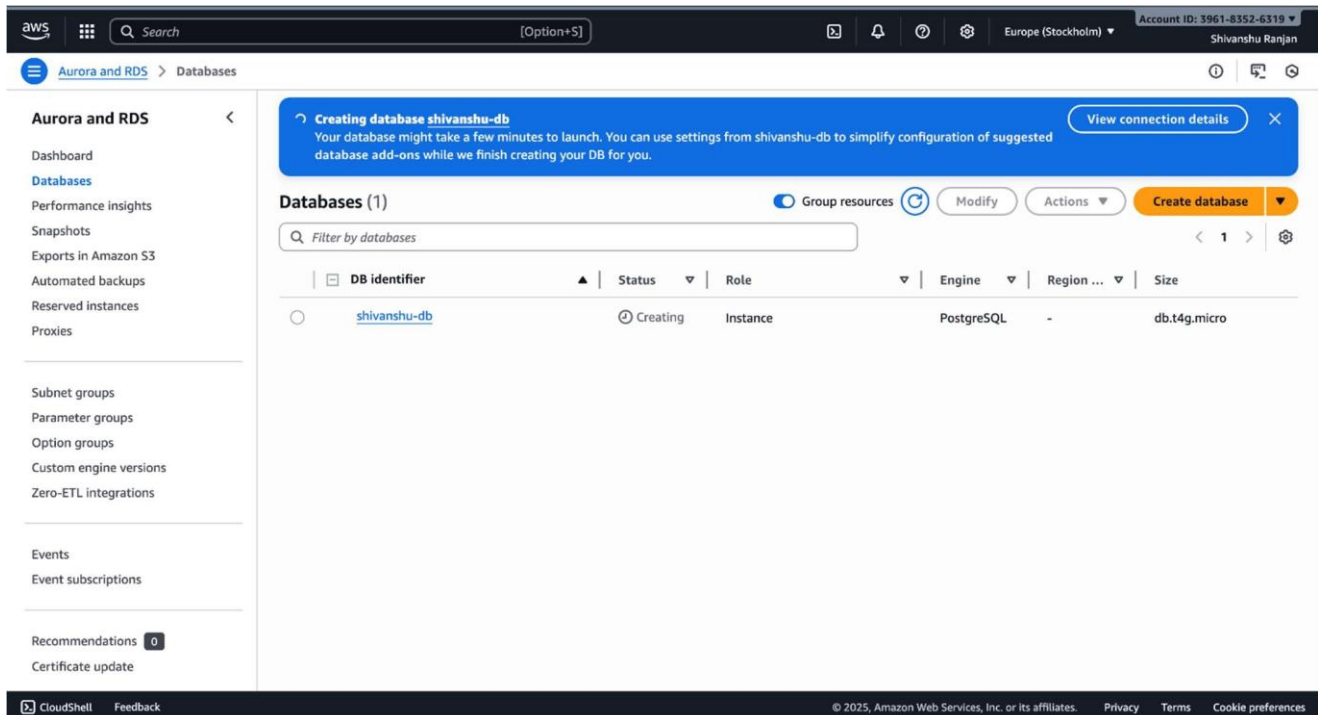


Setting	Value	Required
VPC security group	default	Yes
Publicly accessible	No	Yes
Database port	5432	Yes
DB instance identifier	shivanshu-DB	Yes
DB engine version	17.4	Yes
DB parameter group	default.postgres17	Yes
Monitoring type	Database Insights - Standard	Yes
Performance insights	Enabled	Yes
Monitoring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

**Important:** 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.

[Cancel](#) [Create database](#)

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



**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.

[View connection details](#)

**Databases (1)** [Group resources](#) [Modify](#) [Actions](#) [Create database](#)

Filter by databases

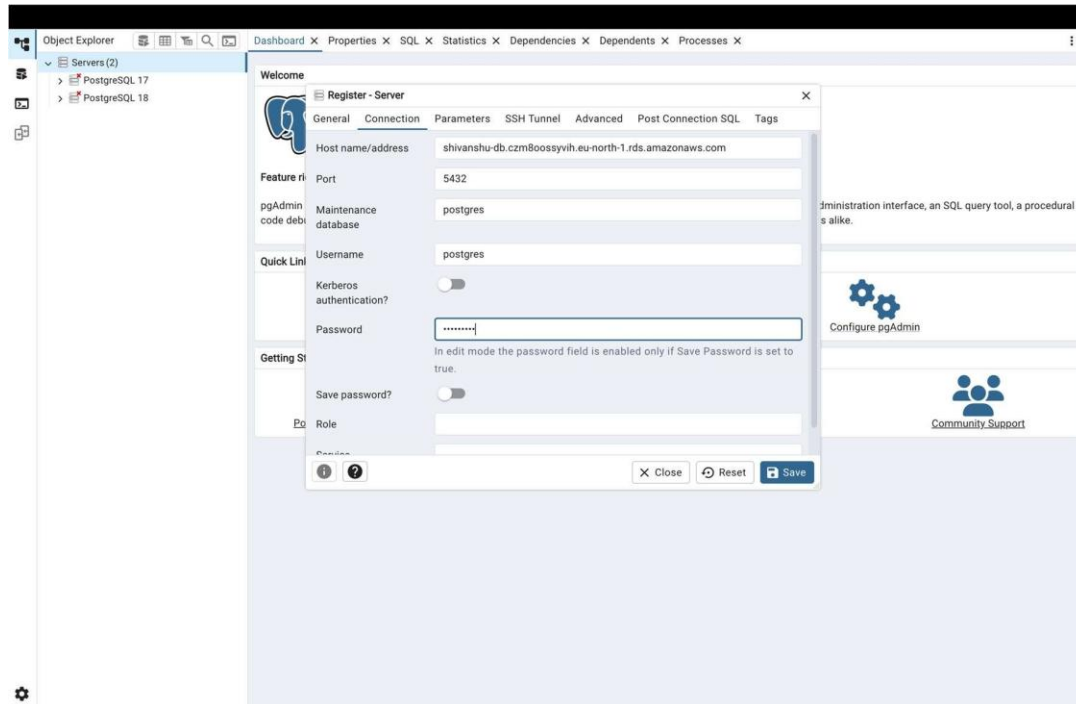
DB identifier	Status	Role	Engine	Region	Size
<a href="#">shivanshu-db</a>	Creating	Instance	PostgreSQL	-	db.t4g.micro

CloudShell Feedback

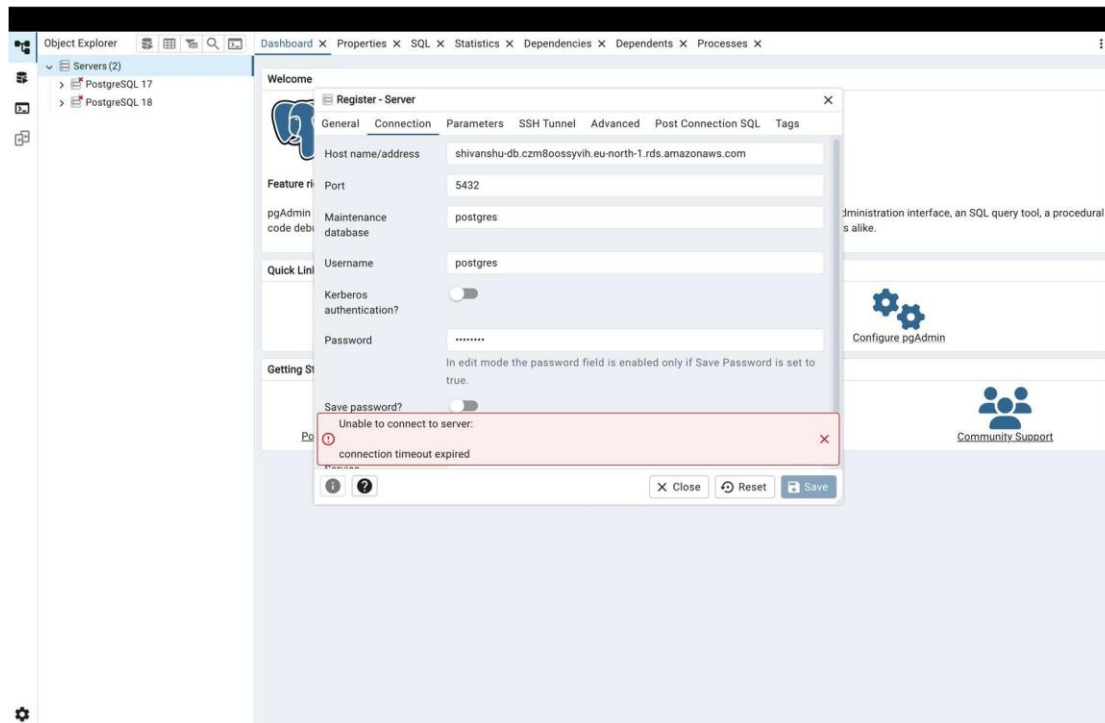
© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane displays a tree of servers under the name 'shivanshu-DB'. A right-click context menu is open, showing options such as 'Register', 'Create', 'Refresh...', 'Remove Server Group', 'Disconnect from all servers', and 'Properties...'. The 'Register' option is highlighted, and a sub-menu is visible with 'Server...' and 'Deploy Cloud Instance...'. The main panel shows the pgAdmin dashboard with the title 'pgAdmin! Management Tools for PostgreSQL'. Below the title, there is a paragraph describing pgAdmin as an open-source administration and management tool. Further down, there are two sections: 'Quick Links' with buttons for 'Add New Server' and 'Configure pgAdmin', and 'Getting Started' with links to 'PostgreSQL Documentation', 'pgAdmin Website', 'Planet PostgreSQL', and 'Community Support'.

## 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 displays the AWS Management Console for the 'shivanshu-db' RDS instance. The left sidebar shows the navigation menu with 'Aurora and RDS' selected. The main content area shows the instance summary and the 'Connectivity & security' tab.

**Summary**

DB identifier	Status	Role	Engine	Recommendations
shivanshu-db	Available	Instance	PostgreSQL	

**Connectivity & security**

Endpoint & port	Networking	Security
<b>Endpoint</b> shivanshu-db.czm8oossyviu.eu-north-1.rds.amazonaws.com <b>Port</b> 5432	<b>Availability Zone</b> eu-north-1a <b>VPC</b> vpc-0b6507ee77883ae1b <b>Subnet group</b> default-vpc-0b6507ee77883ae1b <b>Subnets</b> subnet-0db6b45e321b7000a subnet-087377db566f545dc subnet-0bac42bdab1e990c5	<b>VPC security groups</b> default (sg-0b4c8dc4647072099) Active <b>Publicly accessible</b> No <b>Certificate authority</b> rds-ca-rsa2048-g1 <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 displays the AWS Management Console for the 'sg-0b4c8dc4647072099' security group. The left sidebar shows the navigation menu with 'EC2' selected. The main content area shows the 'Edit inbound rules' page.

**Edit inbound rules**

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules**

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

**Add rule**

**Cancel** **Preview changes** **Save rules**

Discover. Learn. Empower.

## 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.czm8oossyvih.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](#)

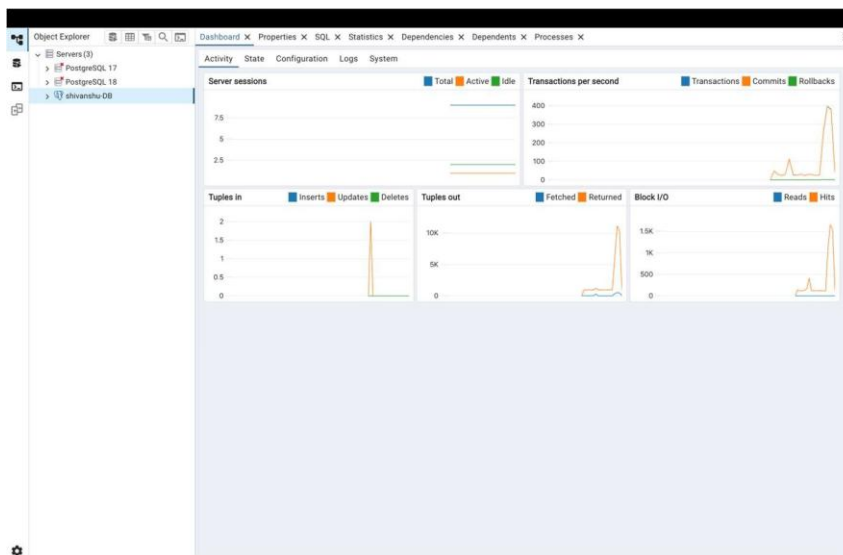
##### Certificate authority date

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

##### DB instance certificate expiration date

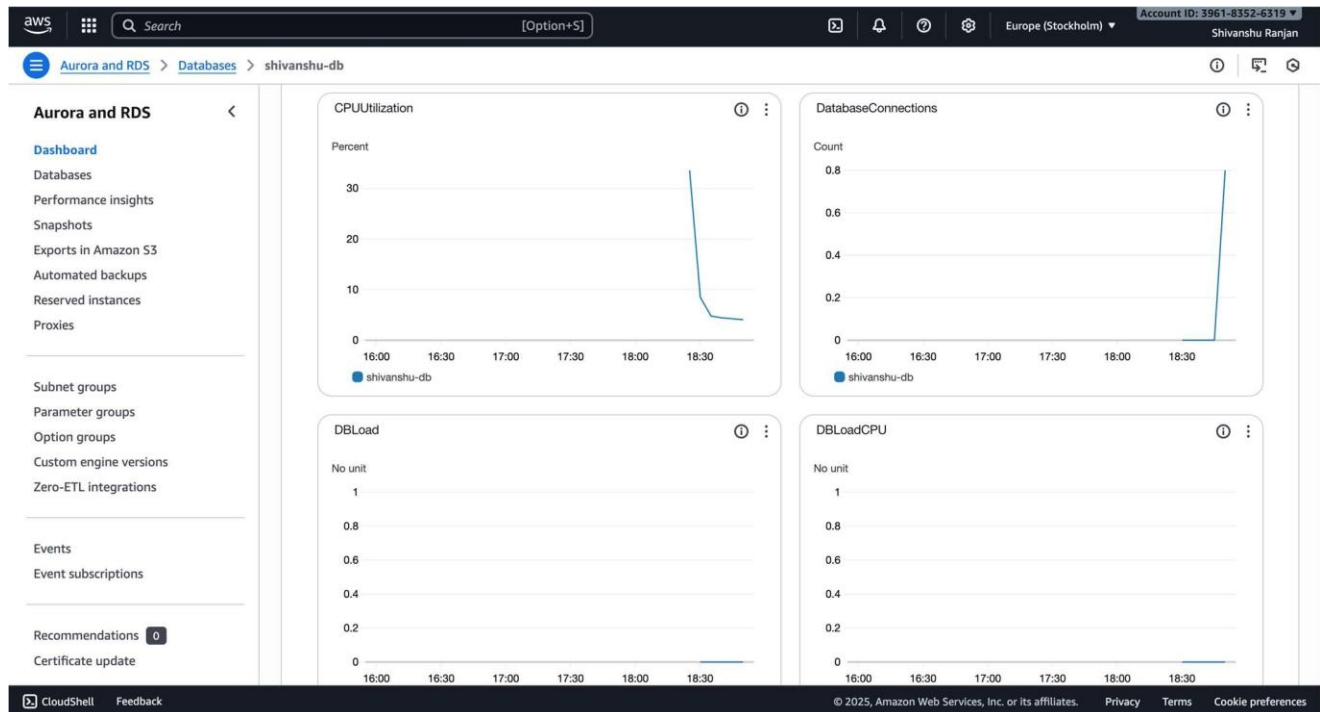
October 30, 2026, 23:59 (UTC+05:30)

## 16. Launching pgAdmin on Local Machine

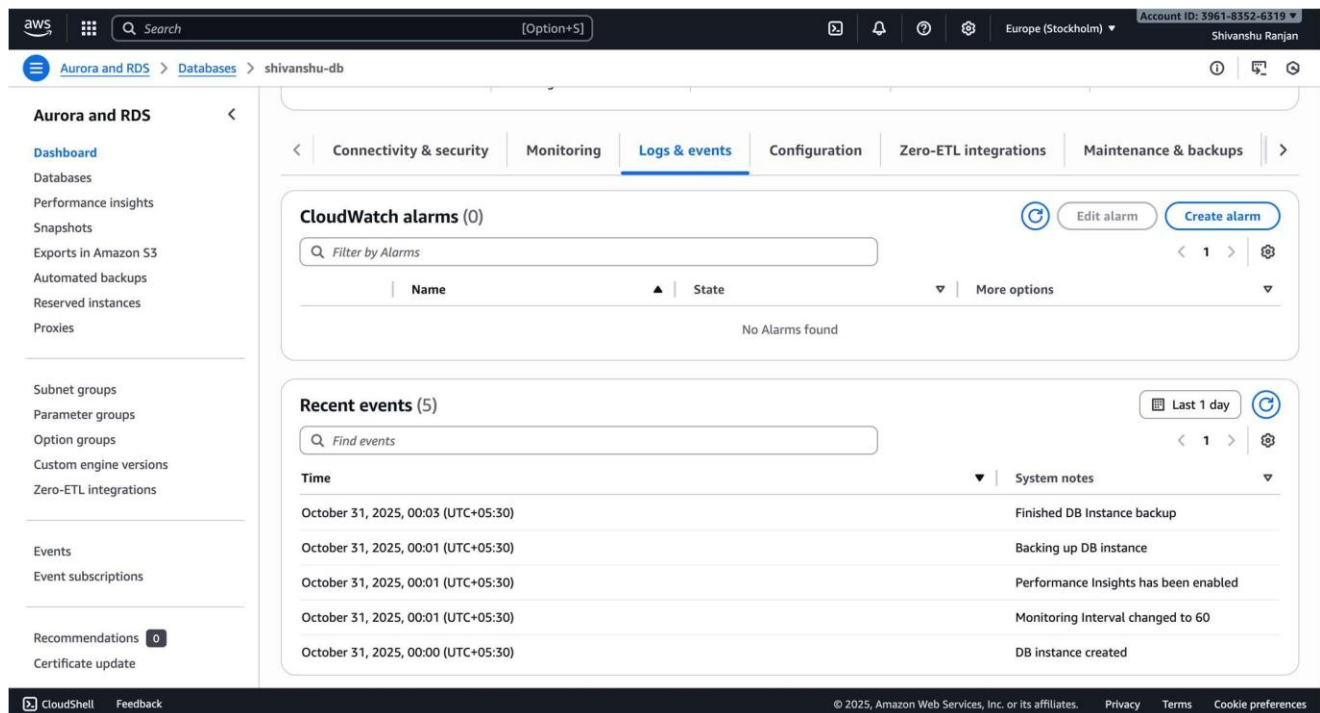




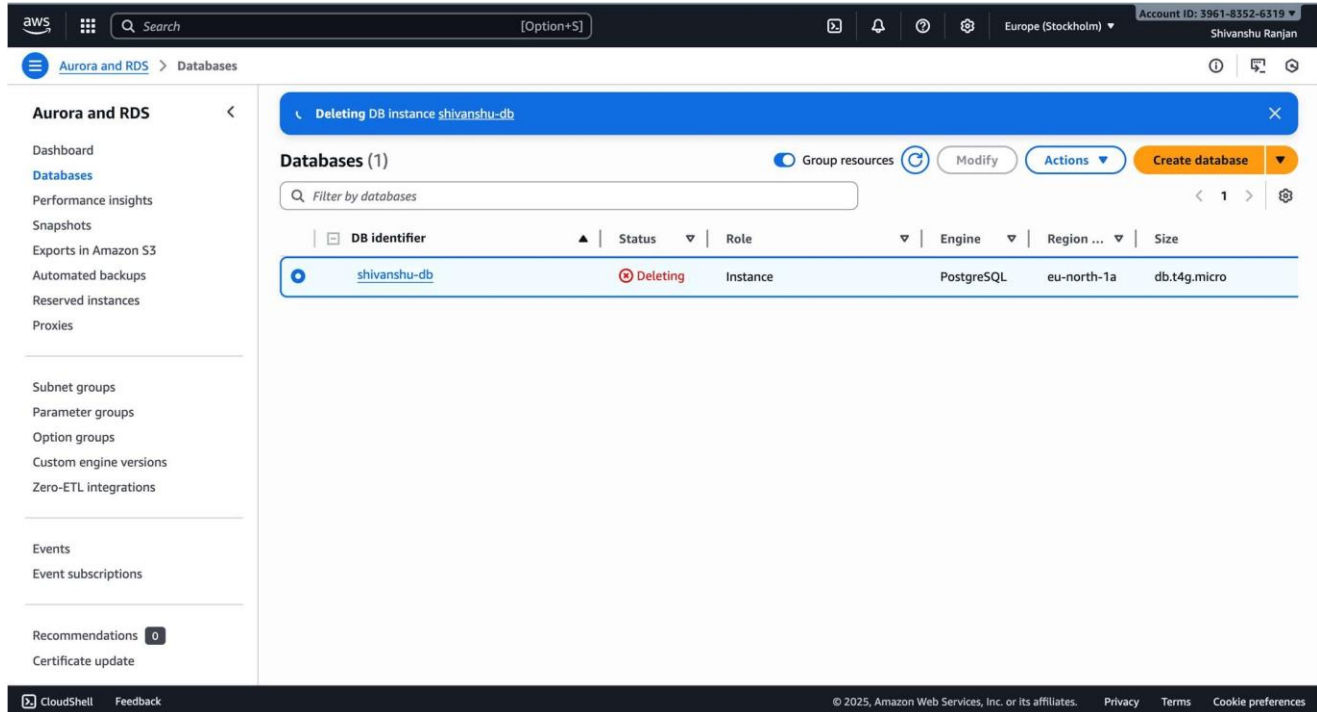
## 17. Adding a New Server in pgAdmin



## 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:

DB identifier	Status	Role	Engine	Region ...	Size
<a href="#">shivanshu-db</a>	Deleting	Instance	PostgreSQL	eu-north-1a	db.t4g.micro

The left sidebar contains various navigation options under 'Aurora and RDS', including 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 bottom of the console shows the CloudShell icon, a feedback link, and the copyright notice: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

## 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.