Worksheet 9

Student Name: Parv Basnal **UID: 23BCS13701**
Branch: CSE **Section/Group: KRG 2-A**
Semester: 5th **Date of Performance:30/10/2025**
Subject Name: ADBMS **Subject Code: 23CSP-333**

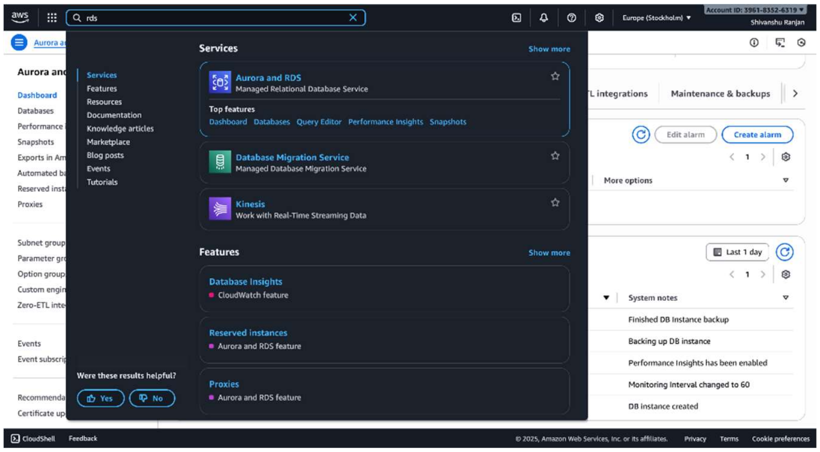
1. **Aim: To understand and implement the setup of Amazon Relational Database Service (AWS RDS) by creating a database instance, configuring security groups, and establishing a secure connection between the local pgAdmin tool and the RDS instance hosted on the AWS Cloud.**

2. Objective:

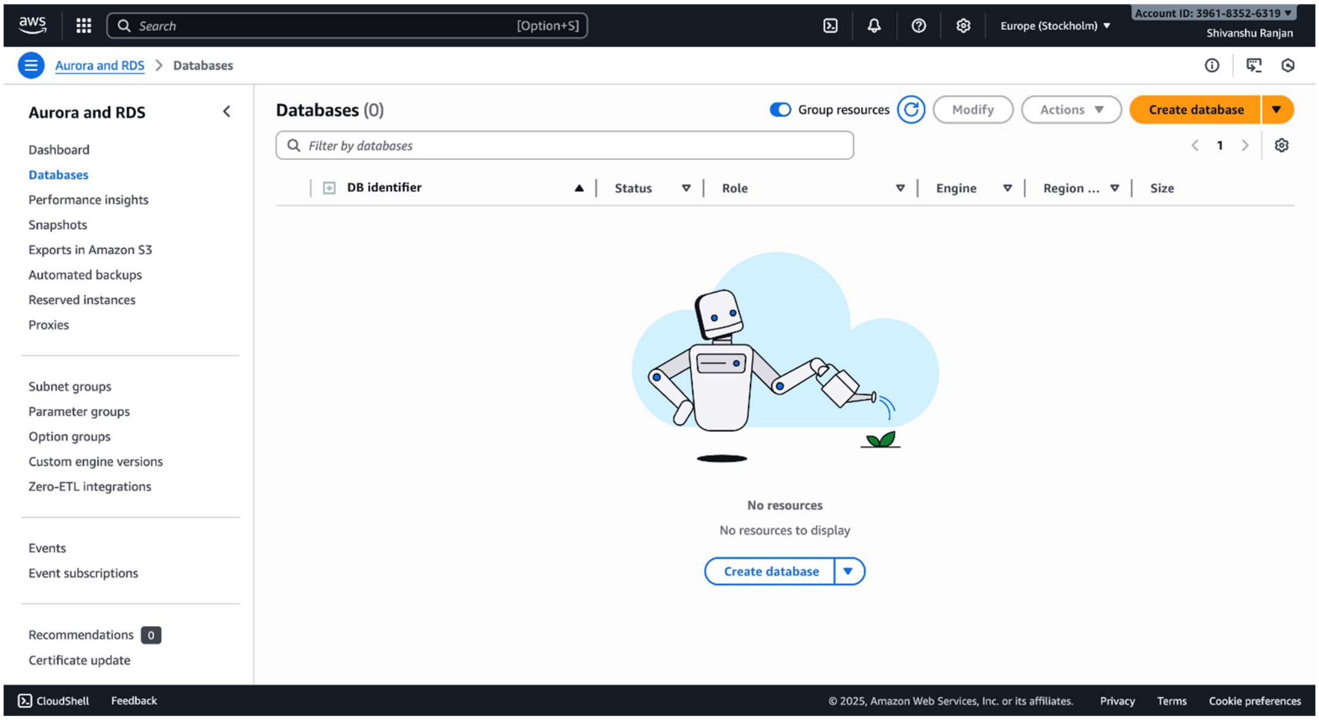
- To learn the basic concepts and features of Amazon Relational Database Service (AWS RDS).
- To create and configure a new RDS database instance on the AWS Management Console.
- To understand the role and configuration of security groups for controlling database access.
- To connect a local pgAdmin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

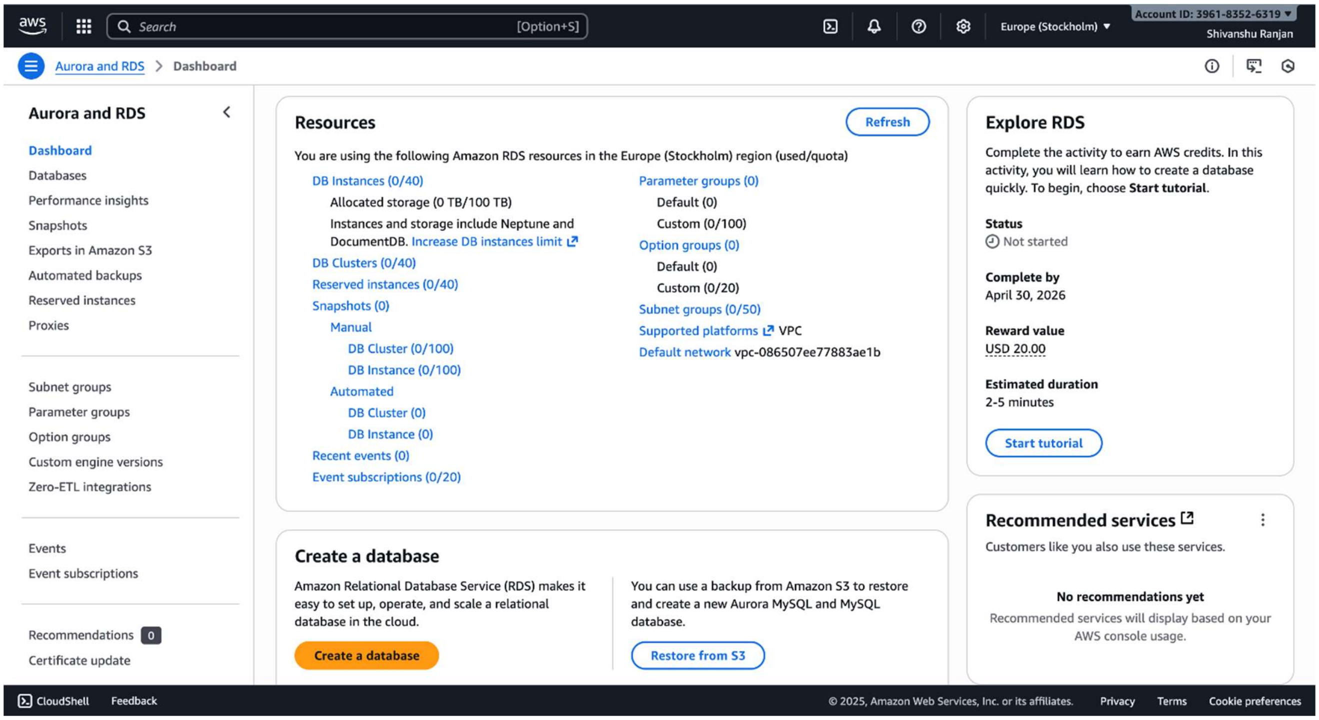


Databases (0)

No resources

No resources to display

3. Amazon RDS Dashboard Overview



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

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

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.

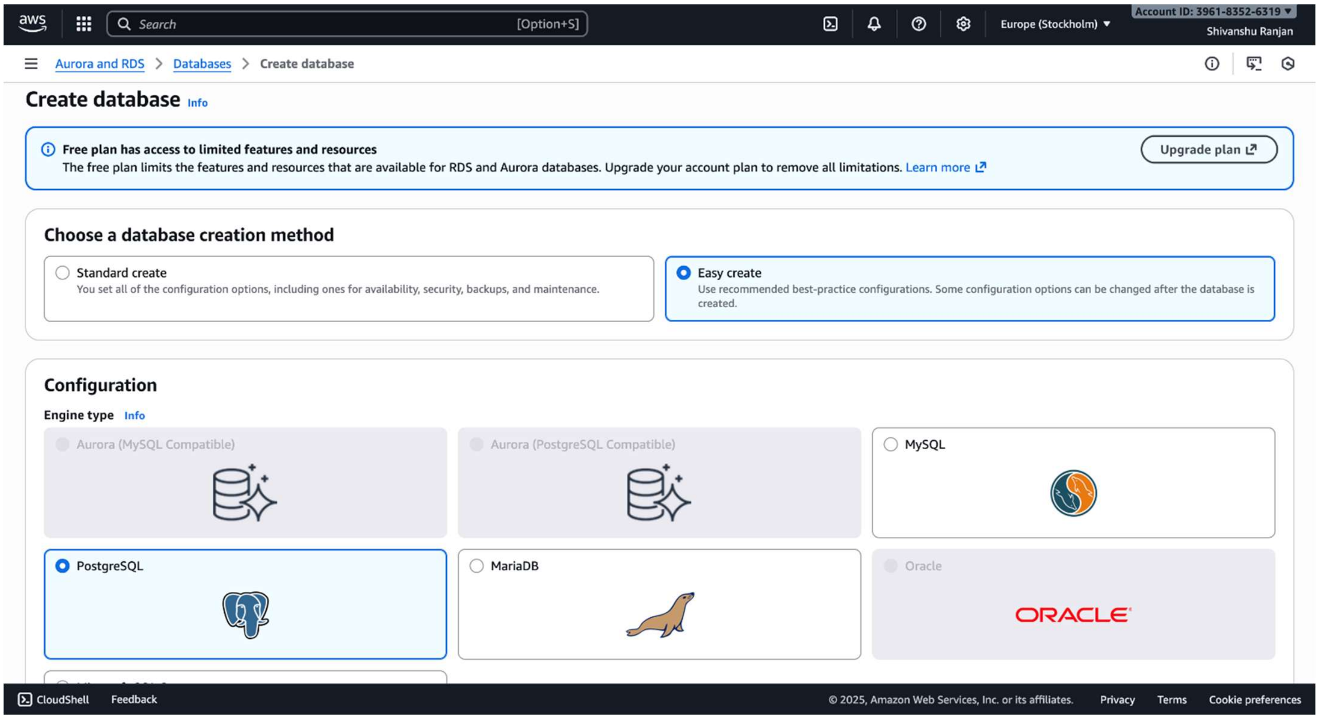
Recommended services

Customers like you also use these services.

No recommendations yet

Recommended services will display based on your AWS console usage.

4. Creating a New Database Instance



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.

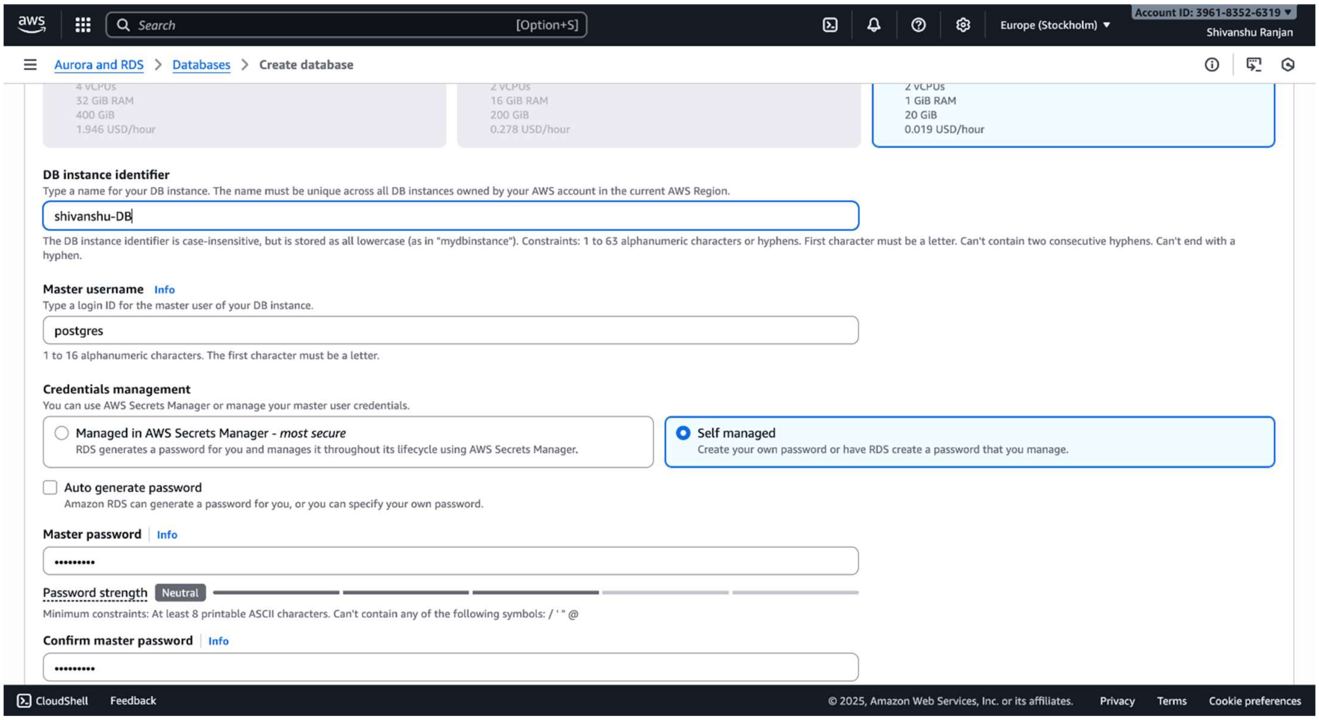
● Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Configuration

Engine type Info

○ Aurora (MySQL Compatible)
○ Aurora (PostgreSQL Compatible)
○ MySQL
● PostgreSQL
○ MariaDB
○ Oracle

5. Selecting PostgreSQL as Database Engine



Aurora and RDS > Databases > Create database

4 vCPUs
32 GiB RAM
400 GiB
1.946 USD/hour

2 vCPUs
16 GiB RAM
200 GiB
0.278 USD/hour

2 vCPUs
1 GiB RAM
20 GiB
0.019 USD/hour

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.

□ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

•••••••••

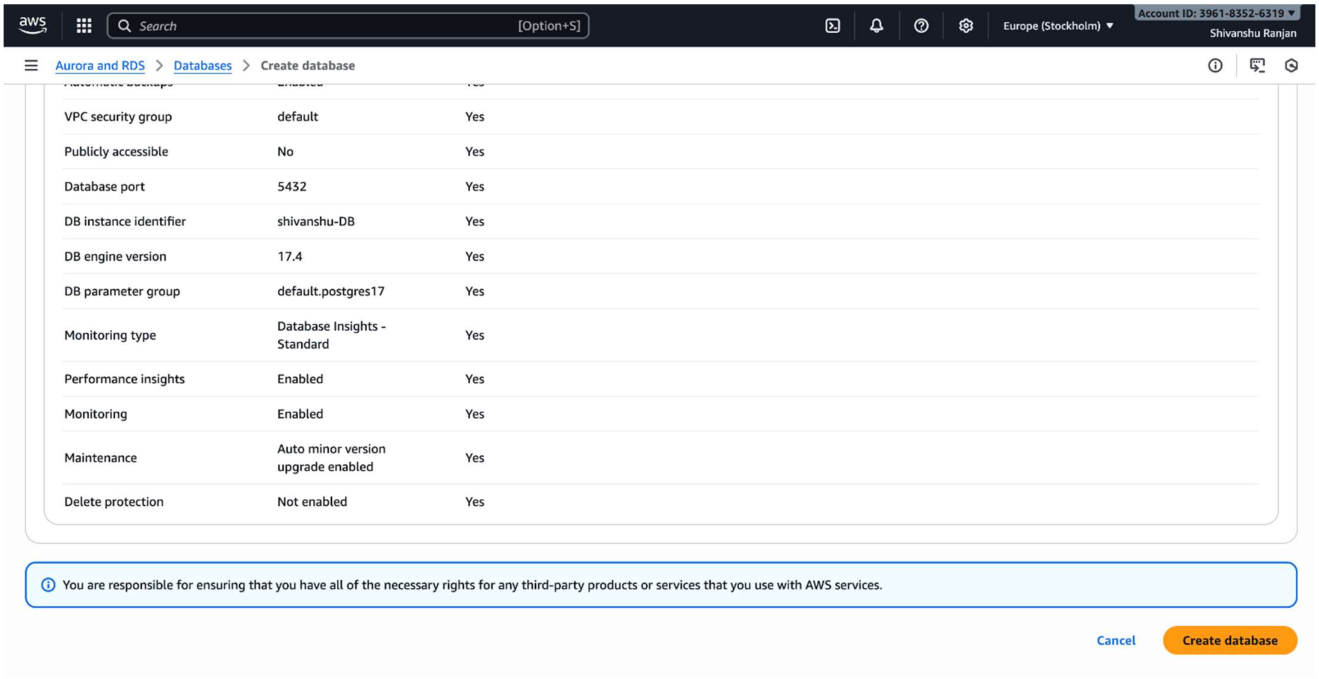
Password strength Neutral

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password Info

•••••••••

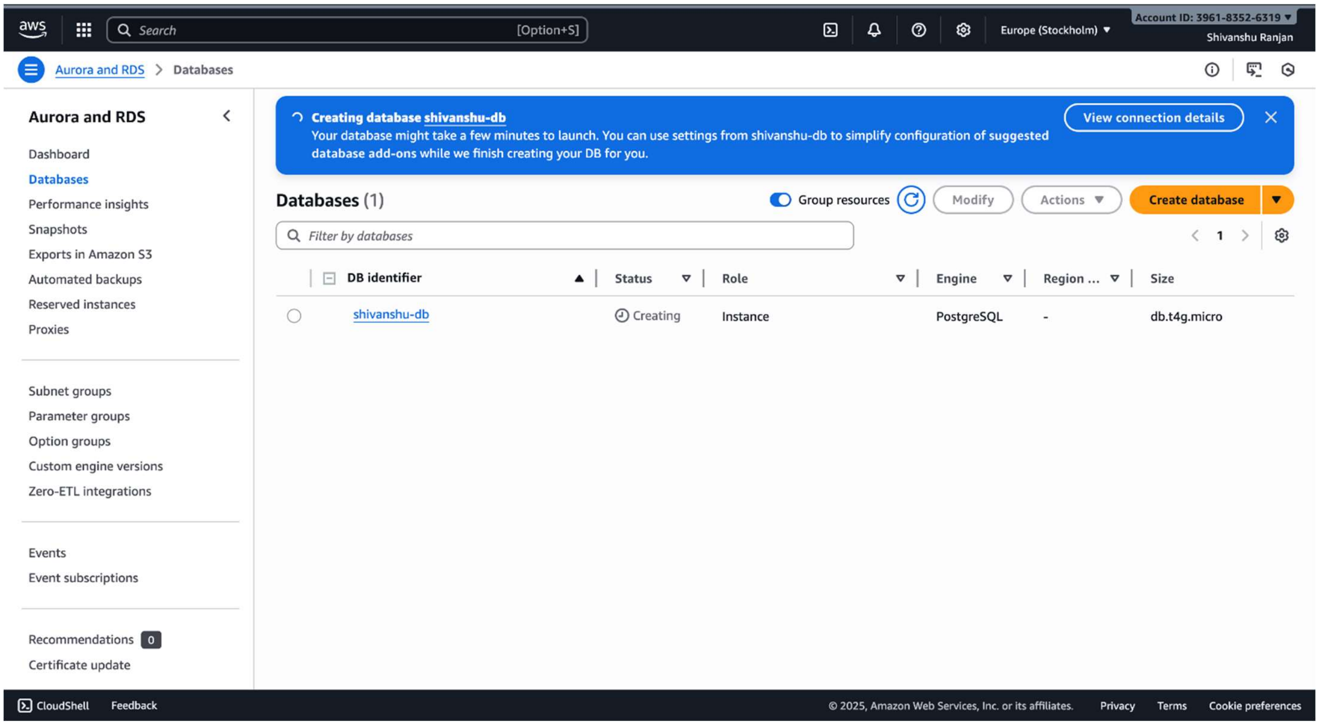
6. Choosing Deployment Option and Template



Automatic backups	Enabled	Yes
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

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.

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

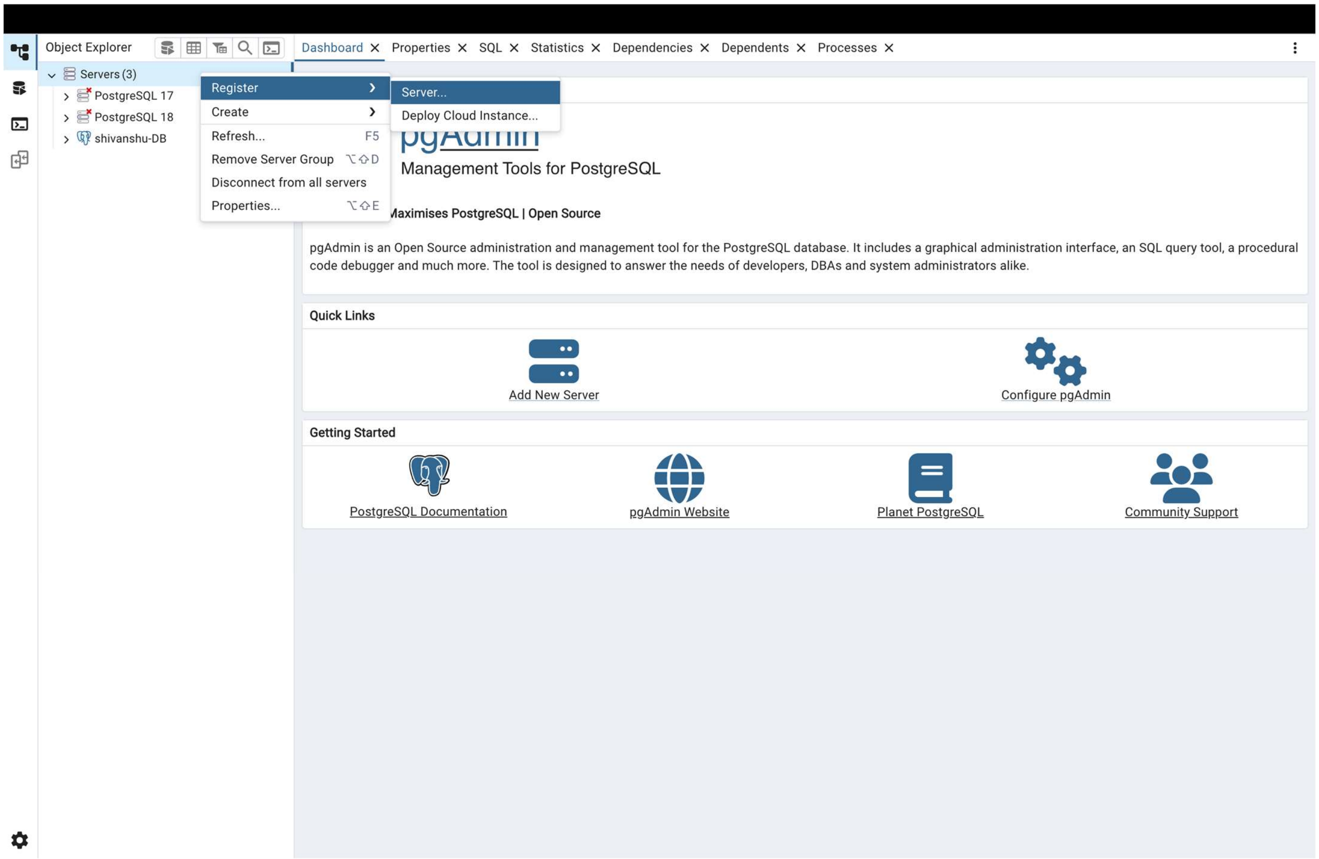


Creating database shivanshu-db
Your database might take a few minutes to launch. You can use settings from shivanshu-db to simplify configuration of suggested database add-ons while we finish creating your DB for you.

Databases (1)

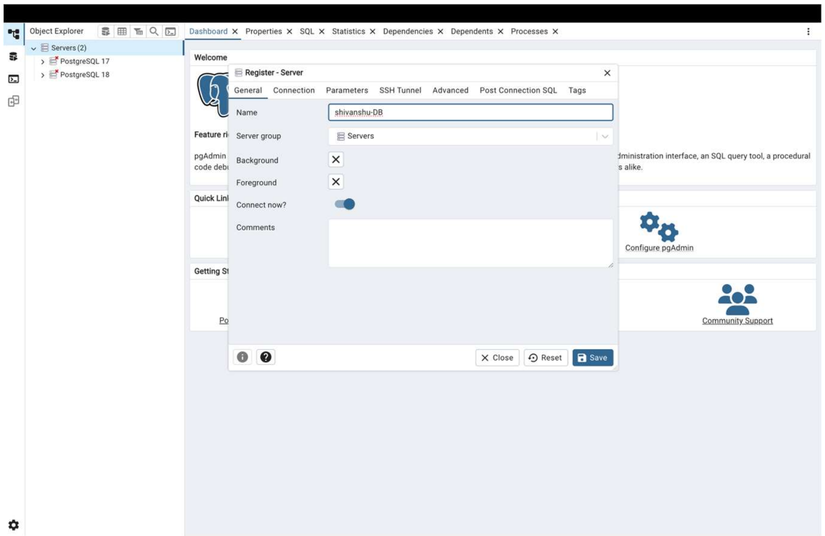
	DB identifier	Status	Role	Engine	Region ...	Size
○	shivanshu-db	Creating	Instance	PostgreSQL	-	db.t4g.micro

8. Setting Up Instance Size and Storage



pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

9. Configuring Connectivity and VPC Settings



pgAdmin
code debu

dministration interface, an SQL query tool, a procedural
s alike.

Name: shivanshu-DB

Server group: Servers

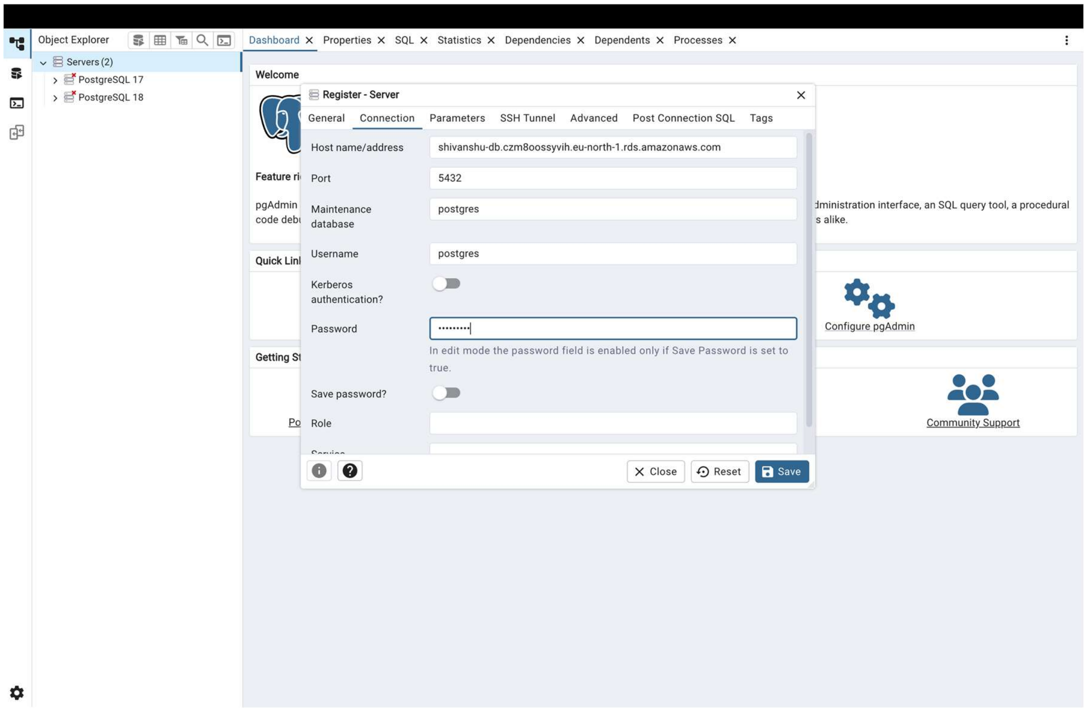
Background: ☒

Foreground: ☒

Connect now? ●

Comments:

10. Gr Setting Up Security Groups for RDS Access



Host name/address: shivanshu-db.czm8oossyvih.eu-north-1.rds.amazonaws.com

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication?

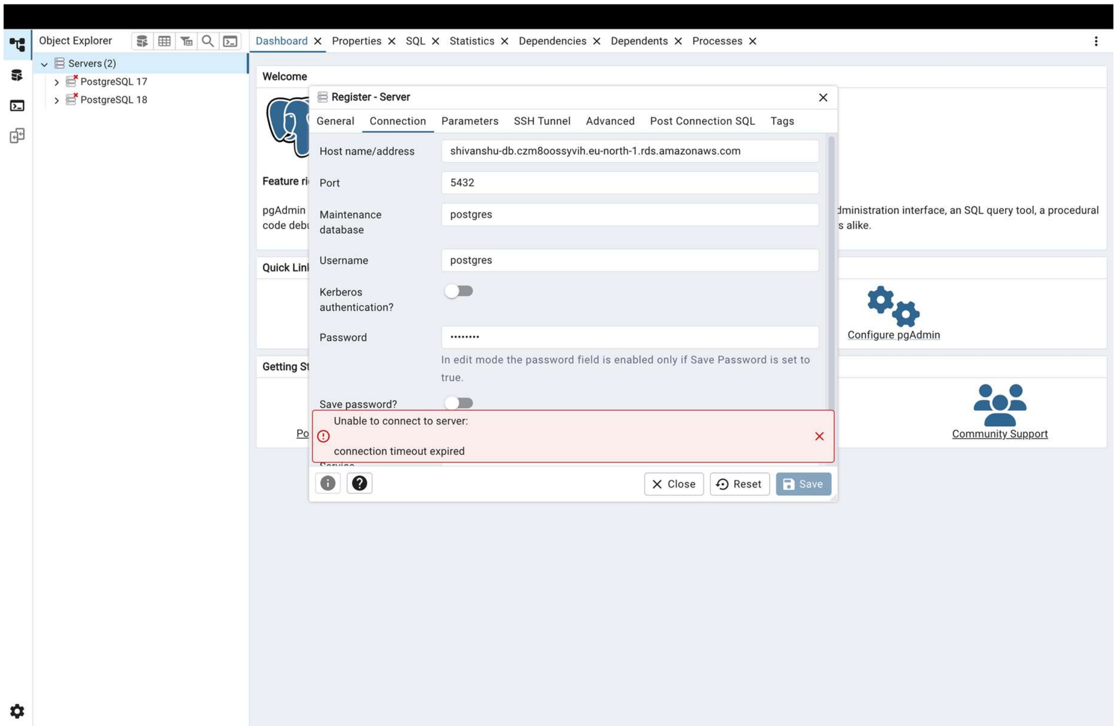
Password: •••••••••

In edit mode the password field is enabled only if Save Password is set to true.

Save password?

Role

11. Additional Database Configuration Options



Host name/address: shivanshu-db.czm8oossyvih.eu-north-1.rds.amazonaws.com

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication?

Password: ••••••••

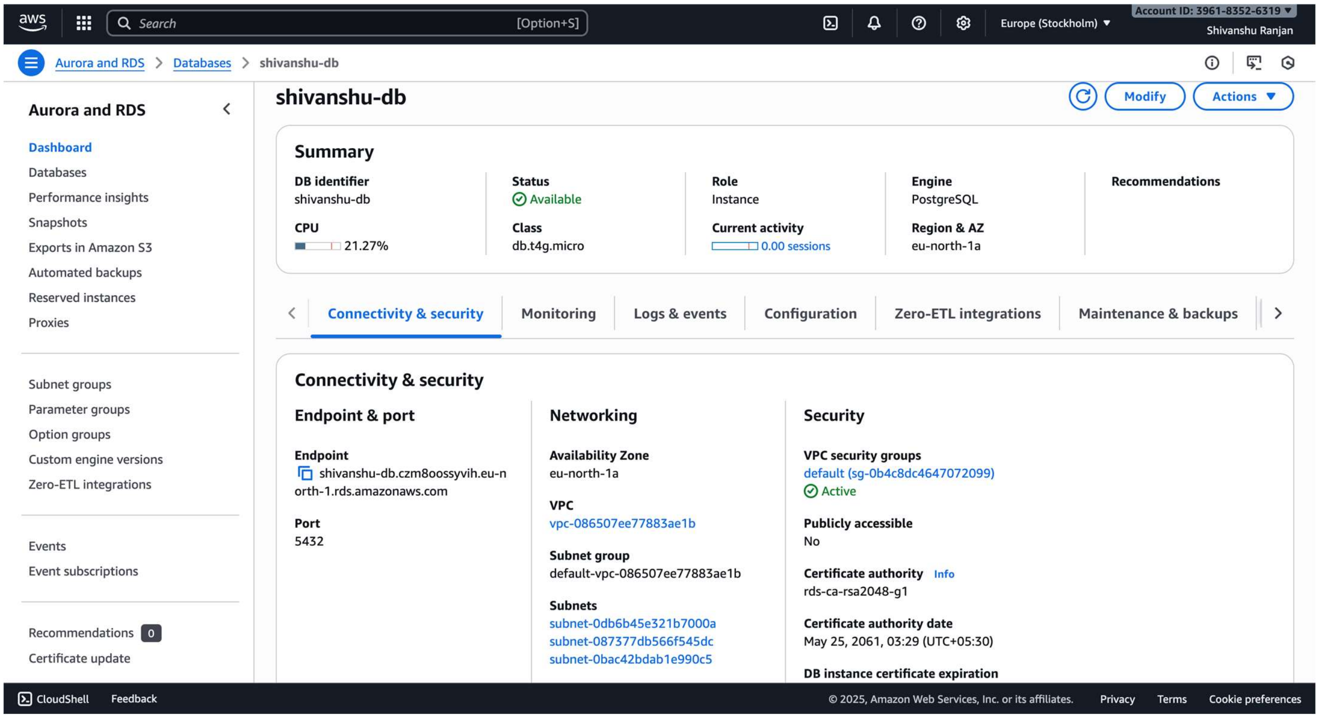
In edit mode the password field is enabled only if Save Password is set to true.

Save password?

Unable to connect to server:

connection timeout expired

12. Reviewing and Creating the Database Instance



shivanshu-db

Summary

DB identifier	Status	Role	Engine	Recommendations
shivanshu-db	Available	Instance	PostgreSQL	
CPU	**Class**	**Current activity**	**Region & AZ**	
21.27%	db.t4g.micro	0.00 sessions	eu-north-1a	

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

Security

VPC security groups
default (sg-0b4c8dc4647072099)
Active

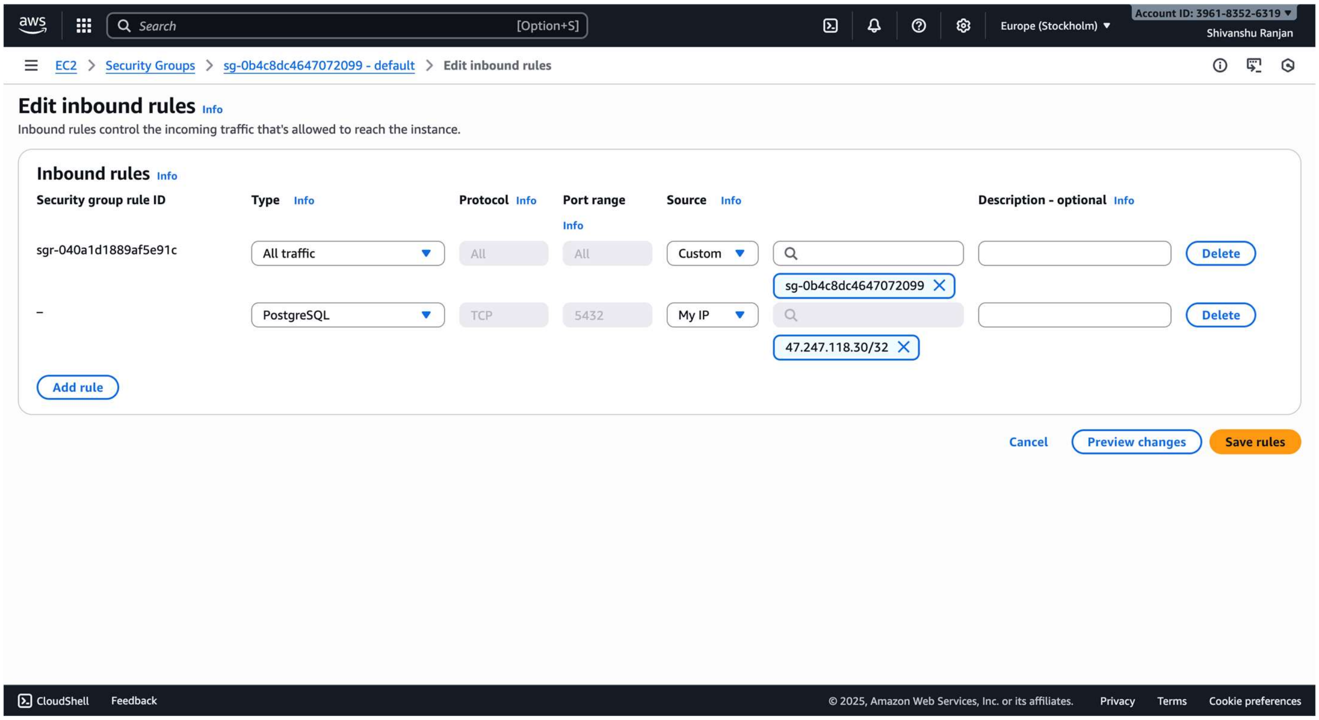
Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

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

DB instance certificate expiration

13. RDS Instance Creation in Progress



aws
Search [Option+S]
Europe (Stockholm) ▼
Account ID: 3961-8352-6319 ▼
Shivanshu Ranjan
EC2 > Security Groups > sg-0b4c8dc4647072099 - default > Edit inbound rules
Add rule
Cancel
Preview changes
Save rules
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

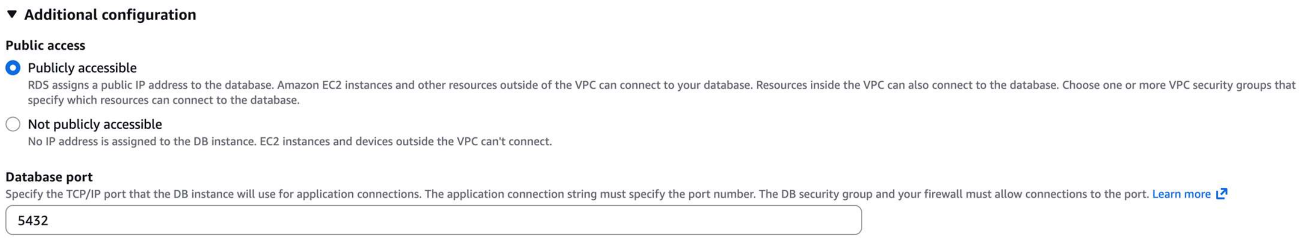
Edit inbound rules Info

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

Inbound rules Info

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

14. Viewing Database Instance Details



▼ **Additional configuration**

Public access

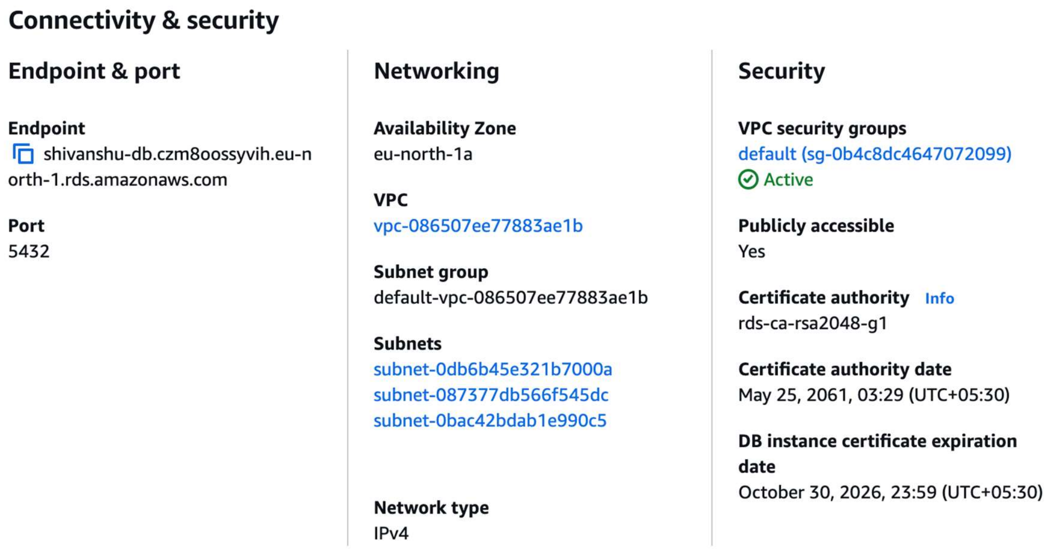
● Publicly accessible
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to 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

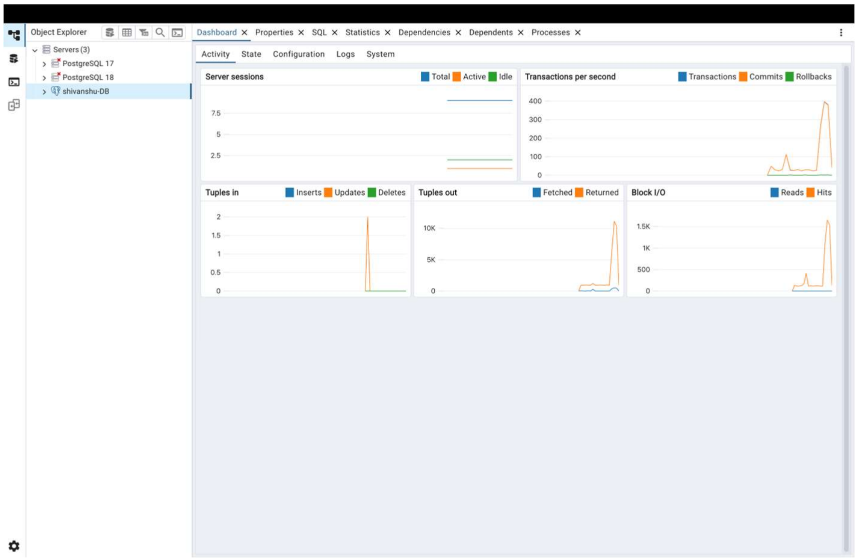
Publicly accessible
Yes

Certificate authority Info
rds-ca-rsa2048-g1

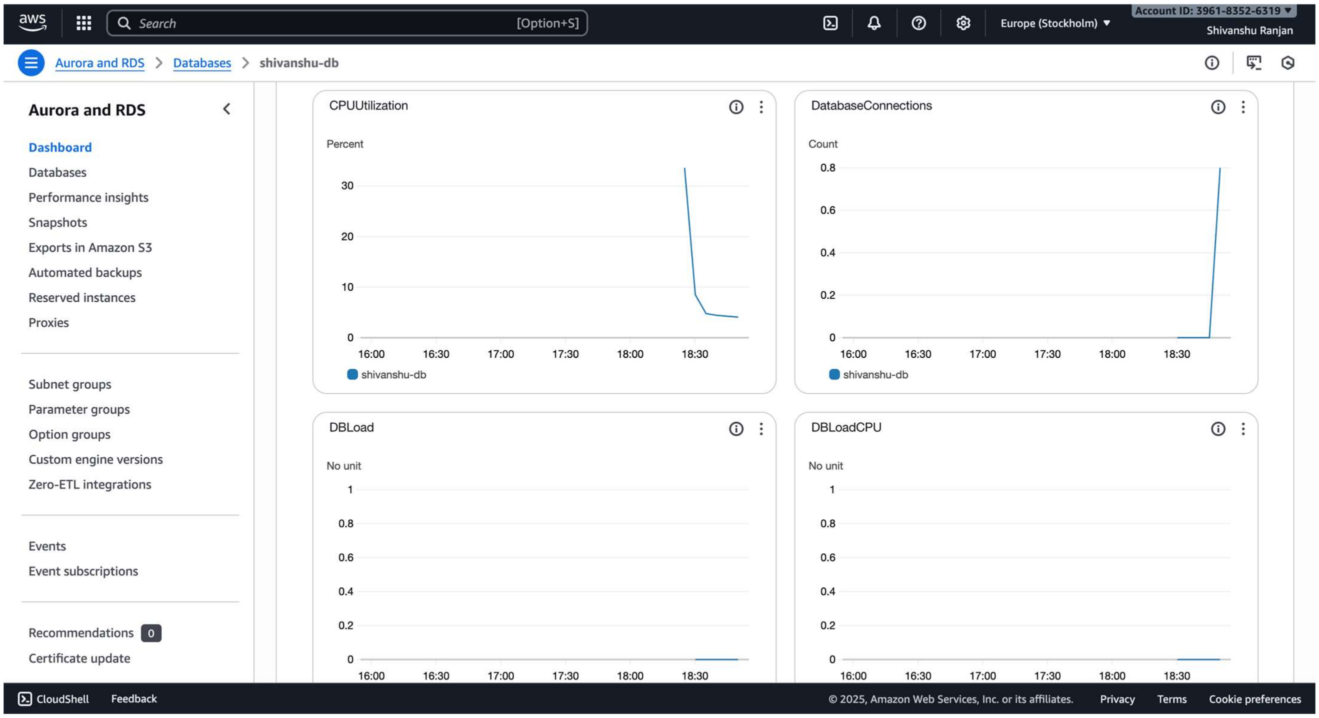
Certificate authority date
May 25, 2061, 03:29 (UTC+05:30)

DB instance certificate expiration date
October 30, 2026, 23:59 (UTC+05:30)

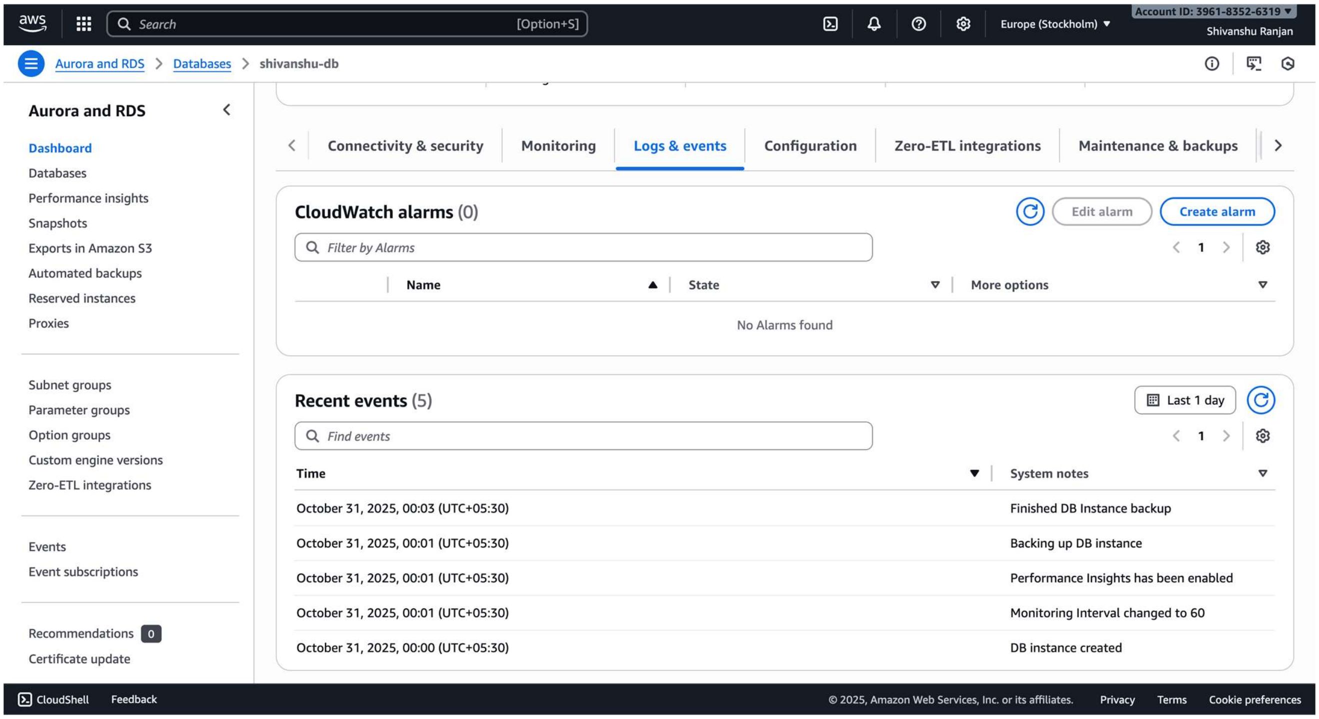
16. Launching pgAdmin on Local Machine



17. Adding a New Server in pgAdmin



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



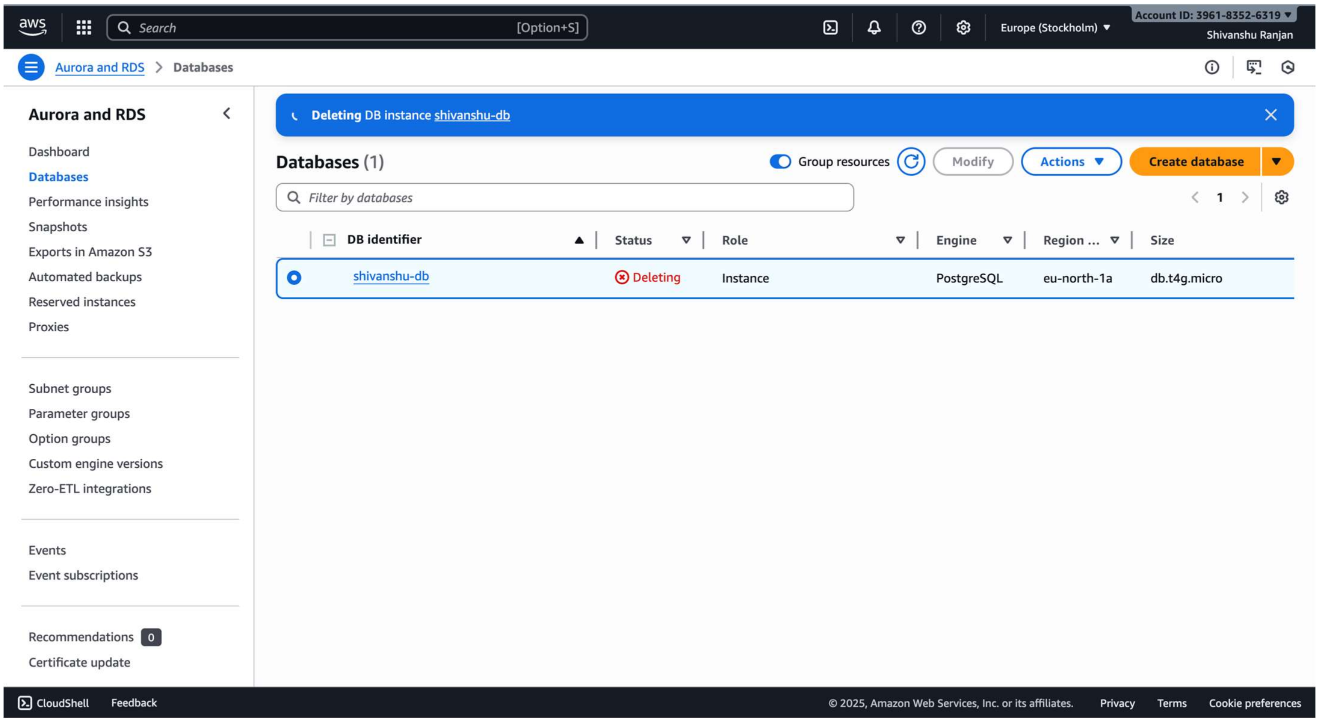
CloudWatch alarms (0)

Name	State	More options
No Alarms found		

Recent events (5)

Time	System notes
October 31, 2025, 00:03 (UTC+05:30)	Finished DB Instance backup
October 31, 2025, 00:01 (UTC+05:30)	Backing up DB instance
October 31, 2025, 00:01 (UTC+05:30)	Performance Insights has been enabled
October 31, 2025, 00:01 (UTC+05:30)	Monitoring Interval changed to 60
October 31, 2025, 00:00 (UTC+05:30)	DB instance created

19. Successful Connection to AWS RDS Database via pgAdmin



Deleting DB instance shivanshu-db

Databases (1)

	DB identifier	Status	Role	Engine	Region ...	Size
●	shivanshu-db	⊗ Deleting	Instance	PostgreSQL	eu-north-1a	db.t4g.micro

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.