

University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT: 9

NAME : Chahal Goyal UID : 23BCS11983
SECTION : KRG_1B SEMESTER: 5TH
SUBJECT CODE: 23CSP-339 SUBJECT : ADBMS

Problem Statement:

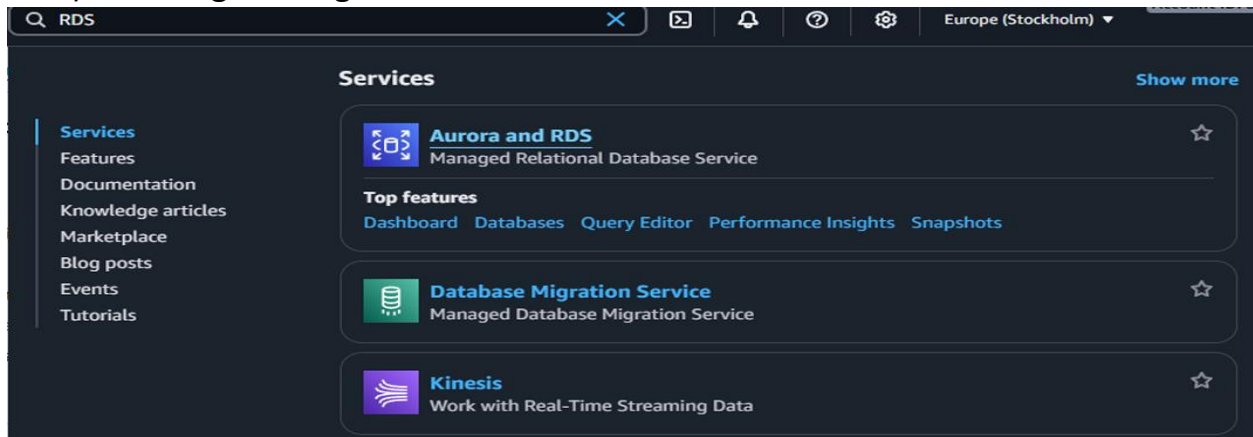
- To design and interact with cloud-based data storage systems using AWS RDS.
- Connect it with local database from your system.

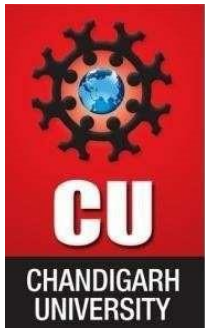
Hardware/Software Requirements

Component	Specification
Hardware	Personal computer
Cloud Services	AWS RDS
Tools	PGAdmin
Languages	PostgreSQL
OS	Windows

Steps:

- Go to aws homepage -> click on sign in-> enter user name with email address.
- After sign-in -> go to search bar -> search for rds -> hit enter.





University Institute of Engineering

Department of Computer Science & Engineering

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 [Refresh](#)

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-081fe9fe127bb8e79

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](#)

Note: your DB instances will launch in the **Europe (Stockholm)** region

iii) Click on create database.

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](#)

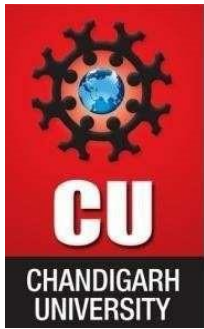
Choose a database creation method

- ☐ **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)
- ☒ PostgreSQL
- ☐ Microsoft SQL Server
- ☐ MariaDB
- ☐ MySQL
- ☐ Oracle



University Institute of Engineering

Department of Computer Science & Engineering

- iv) In the standalone create, we can set everything for our database, the incoming traffic, ip addresses to be used, backup.

DB instance size

☐ Production
db.r7g.xlarge
4 vCPUs
32 GiB RAM
400 GiB
1.946 USD/hour

☐ Dev/Test
db.r7g.large
2 vCPUs
16 GiB RAM
200 GiB
0.278 USD/hour

☒ Free tier
db.t4g.micro
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.

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 **Very strong**

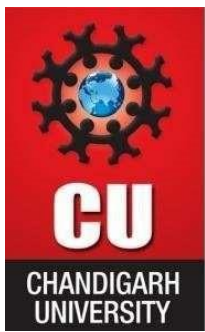
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / * @

Confirm master password [Info](#)

▼ View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use Standard create.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-081fe9fe127bb8e79)	No
Multi-AZ	No	Yes
Option group	default:mysql-8-0	Yes
Subnet group	Create new DB Subnet Group	Yes
Automatic backups	Enabled	Yes
VPC security group	default	Yes



University Institute of Engineering

Department of Computer Science & Engineering

Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	strugmac-DB	Yes
DB engine version	8.0.42	Yes
DB parameter group	default.mysql8.0	Yes
Monitoring type	Database Insights - Standard	Yes
Performance insights	Not enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes

v) Confirm all your selection .

Aurora and RDS > Databases

Creating database strugmac-db
Your database might take a few minutes to launch. You can use settings from strugmac-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
strugmac-db	Creating	Instance	MySQL Co...

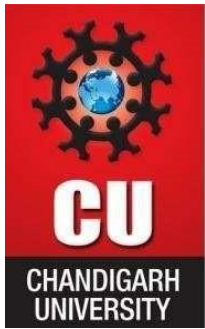
Now this will create a MySQL database to me, and we want to connect to RDS for which we have to launch a server which basically will have MySQL Client installed inside it. For that we have to launch an EC2 instance, Launching an EC2 instance.

Console Home > All services

All services

Services by category

- Compute**
 - [EC2](#)
 - [Lightsail](#)
 - [Lambda](#)
 - [Batch](#)
 - [Elastic Beanstalk](#)
 - [Serverless Application Repository](#)
 - [AWS Outposts](#)
 - [EC2 Image Builder](#)
 - [AWS App Runner](#)
 - [AWS SimSpace Weaver](#)
 - [Parallel Computing Service](#)
 - [AWS Global View](#)
- Containers**
- Machine Learning**
 - [Amazon SageMaker AI](#)
 - [Amazon Augmented AI](#)
 - [Amazon CodeGuru](#)
 - [Amazon DevOps Guru](#)
 - [Amazon Comprehend](#)
 - [Amazon Forecast](#)
 - [Amazon Fraud Detector](#)
 - [Amazon Kendra](#)
 - [Amazon Personalize](#)
 - [Amazon Polly](#)
 - [Amazon Rekognition](#)
 - [Amazon Textract](#)
 - [Amazon Transcribe](#)
 - [Amazon Translate](#)



University Institute of Engineering

Department of Computer Science & Engineering

EC2

Dashboard

AWS Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

EC2

Dashboard

AWS Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Benefits and features

EC2 offers ultimate scalability and control

Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Widest variety of server size options
- Widest availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

[Find out more about EC2](#)

Launch a virtual server

[Launch instance](#)

[View dashboard](#)

[Get started walkthroughs](#)

[Get started tutorial](#)

Additional actions

[View running instances](#)

[Migrate a server](#)

Use cases

Instances

Info

Connect

Instance state

Actions

Launch instances

All states

	Name	Instance ID	Instance state	Instance type	Status check
No instances					
You do not have any instances in this region					
Launch instances					

Application and OS Images (Amazon Machine Image)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

aws

Mac

ubuntu

Microsoft

Red Hat

SUSE

debian

[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Key pair (login)

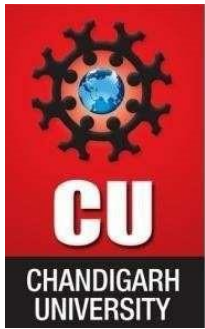
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Proceed without a key pair (Not recommended)

Default value

[Create new key pair](#)



University Institute of Engineering

Department of Computer Science & Engineering

▼ Network settings [Info](#) [Edit](#)

Network | [Info](#)
vpc-081fe9fe127bb8e79

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable

Firewall (security groups) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
☐ Create security group ☒ Select existing security group

Common security groups | [Info](#)

Select security groups

default sg-0e67db7abaff84225 ×
VPC: vpc-081fe9fe127bb8e79

[Compare security group rules](#)
Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Summary

Number of instances | [Info](#)

1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0a716d3f3b16d290c

Virtual server type (instance type)
t3.micro

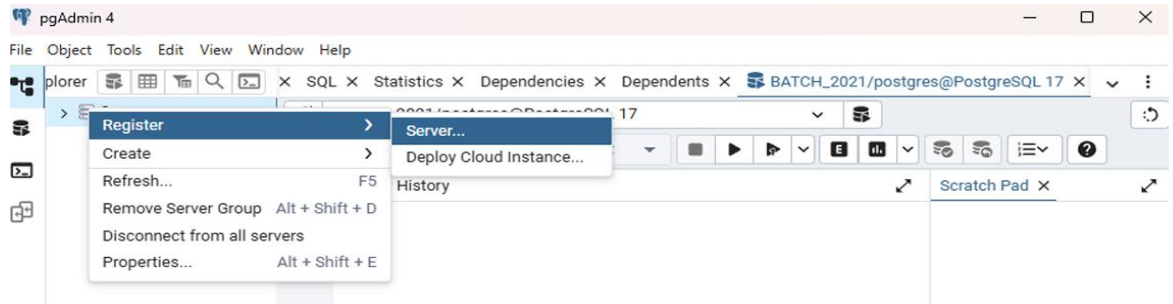
Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

vi) Connect the Postgres AWS RDS to our local machine.

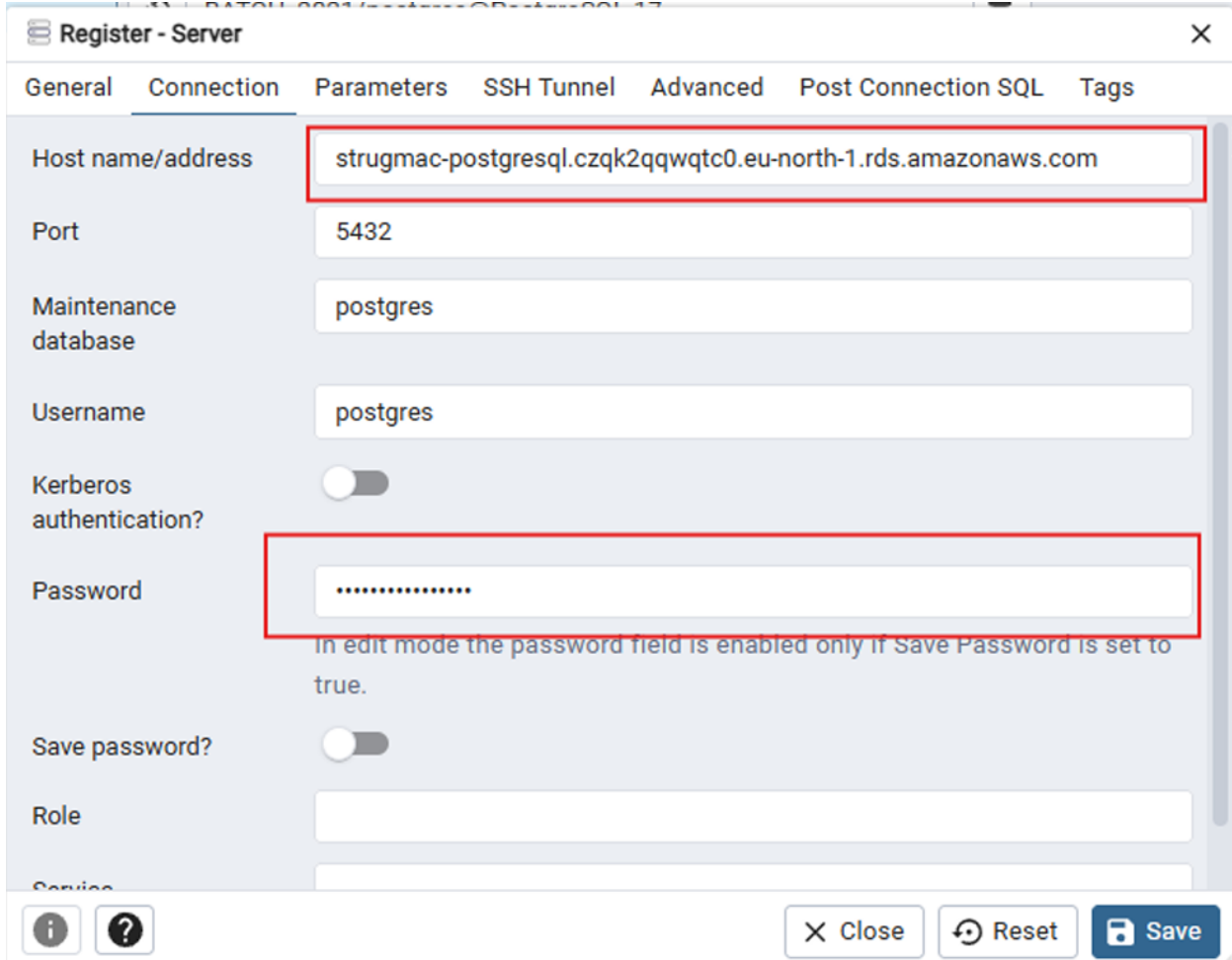
- Create AWS RDS database for PostgreSQL
- Connect from PgAdmin.



University Institute of Engineering

Department of Computer Science & Engineering

vii) Copy the API Endpoints from the dashboard of AWS RDS Database instance.



Register - Server

General Connection Parameters SSH Tunnel Advanced Post Connection SQL Tags

Host name/address: strugmac-postgresql.czqk2qqwqtc0.eu-north-1.rds.amazonaws.com

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication? ☐

Password:

Save password? ☐

Role:

Service:

viii) Change the INBOUND RULES of DB Instance from the AWS Console.

Connectivity & security

Endpoint & port

Endpoint
strugmac-postgresq
l.czqk2qqwqtc0.eu-north
-1.rds.amazonaws.com

Port
5432

Networking

Availability Zone
eu-north-1c

VPC
vpc-081fe9fe127bb8e79

Subnet group
default-vpc-
081fe9fe127bb8e79

Subnets
subnet-
00bf0147db6493492
subnet-
0aa3f608f07d8cecc
subnet-
0f9ee2b6eb9698f78

Security

VPC security groups
default (sg-
0e67db7abaff84225)
Active

Publicly accessible
No

Certificate authority

Info
rds-ca-rsa2048-g1

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

University Institute of Engineering

Department of Computer Science & Engineering

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-0d9f21030174e69aa	All traffic ▼	All	All	C... ▼	Q	Delete
-	PostgreSQL ▼	TCP	5432	M... ▼	Q	Delete

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

ix) This allows the local editor to access the cloud db via the ip of local host over the internet provided in the inbound rules.

Learning outcomes:

- I learned the core benefits of AWS RDS over EC2-hosted databases, including scalability, manageability, and built-in high availability.
- I practiced creating a database instance on AWS RDS and configuring parameters like engine type, storage, and backup settings.
- I understood how security groups control access to RDS instances and how to modify inbound rules for external connectivity.
- I learned how to retrieve the RDS endpoint and use it to connect a PostgreSQL database to my local PgAdmin client.
- I troubleshooted common connectivity issues and understood how to resolve them by adjusting AWS Console settings.
- I gained hands-on experience integrating cloud-hosted databases with local development tools for real-world applications.