Experiment 9

Student Name: Gurshsaran singh **UID:** 23BAI70554

Branch: BE-AIT-CSE **Section/Group:** 23AML-1(A)

Semester: 5th **Date of Performance:** 29 Oct,2025

Subject Name: ADBMS **Subject Code:** 23CSP-333

Aim:

To understand and implement the creation and management of a relational database instance using **AWS RDS**, including setting up database connectivity (via pgAdmin or MySQL client), configuring security groups, and comparing RDS with EC2 database setups in terms of scalability, performance, and manageability.

Objective:

- To study the concept and features of **Amazon Web Services (AWS) Relational Database Service (RDS)**.
- To understand the **advantages of using RDS** over EC2 and on-premise database setups.
- To learn how to **create a database instance** on AWS RDS.
- To configure and manage **security groups** for secure database access.
- To learn how to **connect AWS RDS to local pgAdmin or MySQL client**.
- To explore various **RDS features** such as automated backups, monitoring, and scaling.
- To understand **Multi-AZ deployment**, **read replicas**, and **cross-region replication** for high availability.
- To gain hands-on experience in **launching and managing cloud-based databases** using AWS.

Theory:

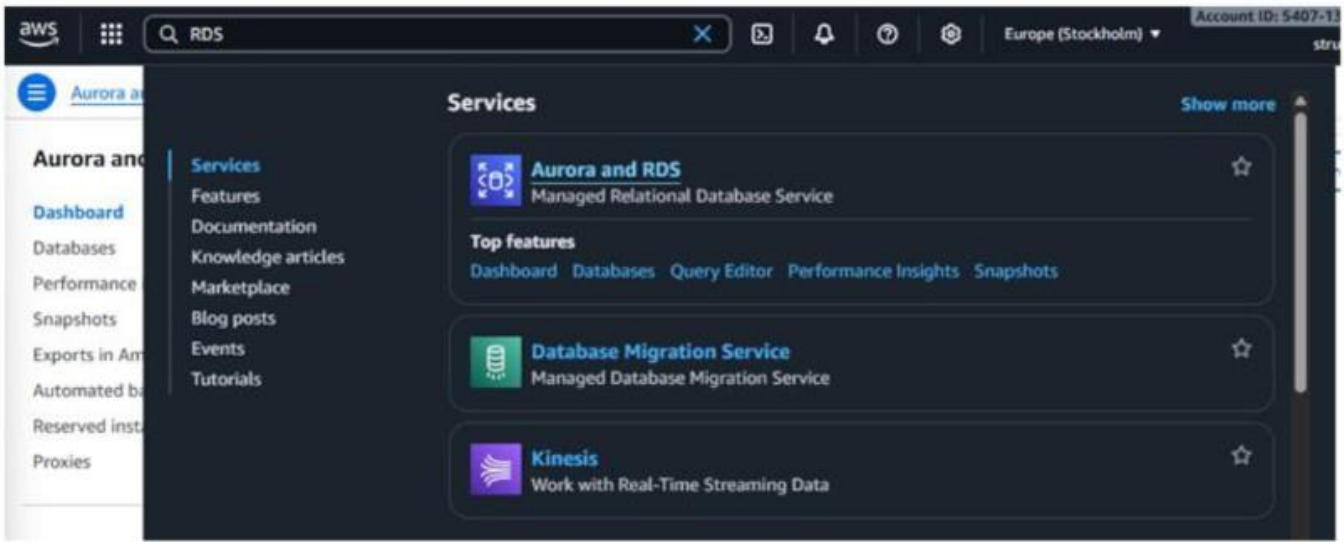
Amazon Web Services (AWS) Relational Database Service (RDS) is a **managed cloud database service** that simplifies the setup, operation, and scaling of relational databases. It automates key administrative tasks such as provisioning, patching, backups, and monitoring, allowing developers to focus on application logic rather than infrastructure management.

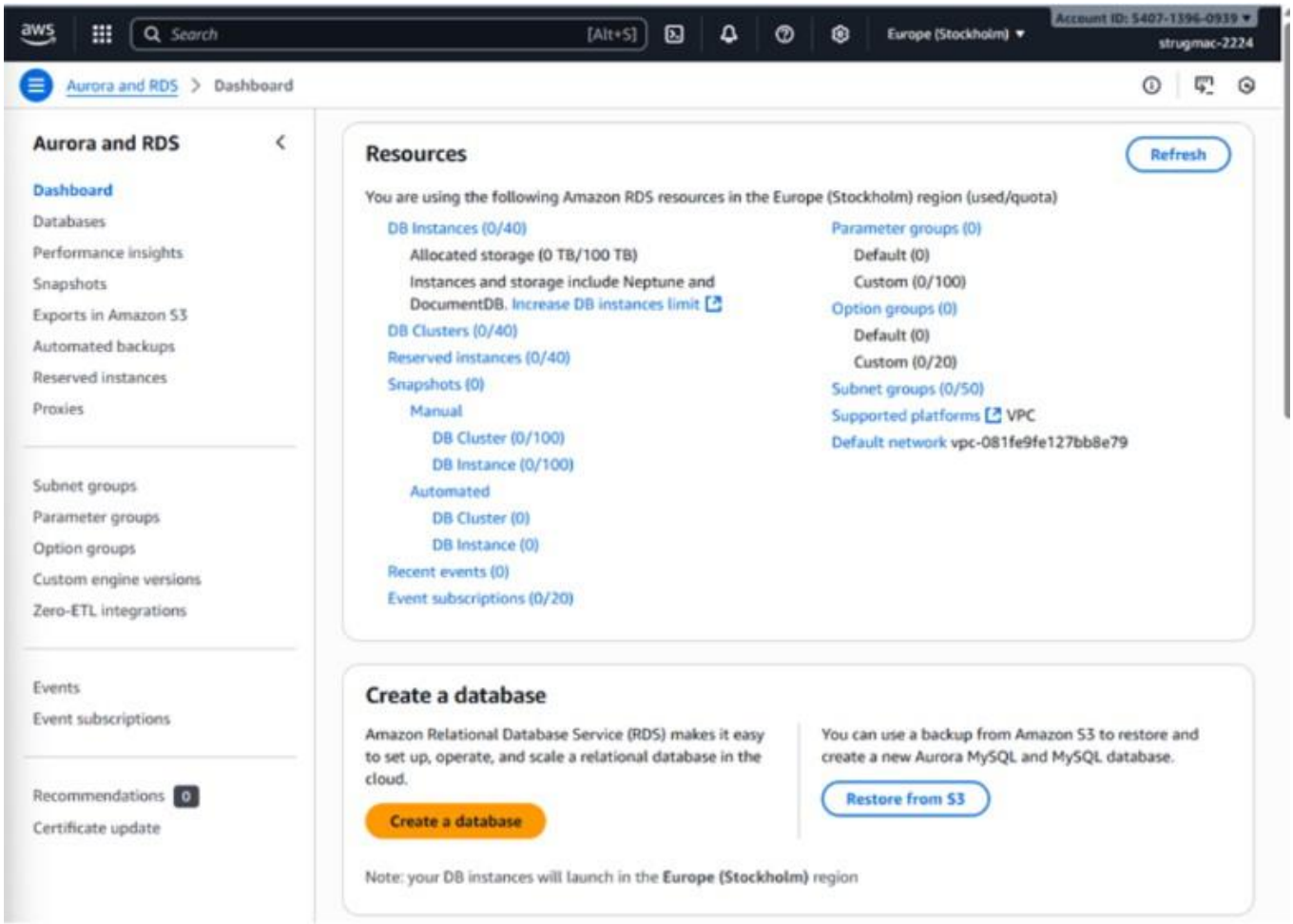
AWS RDS supports multiple database engines, including **MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server**, and provides features like **automated backups**, **multi-AZ deployment**, and **read replicas** to ensure high availability and reliability. Security is managed through **IAM**, **KMS encryption**, and **VPC security groups**, which protect databases from unauthorized access.

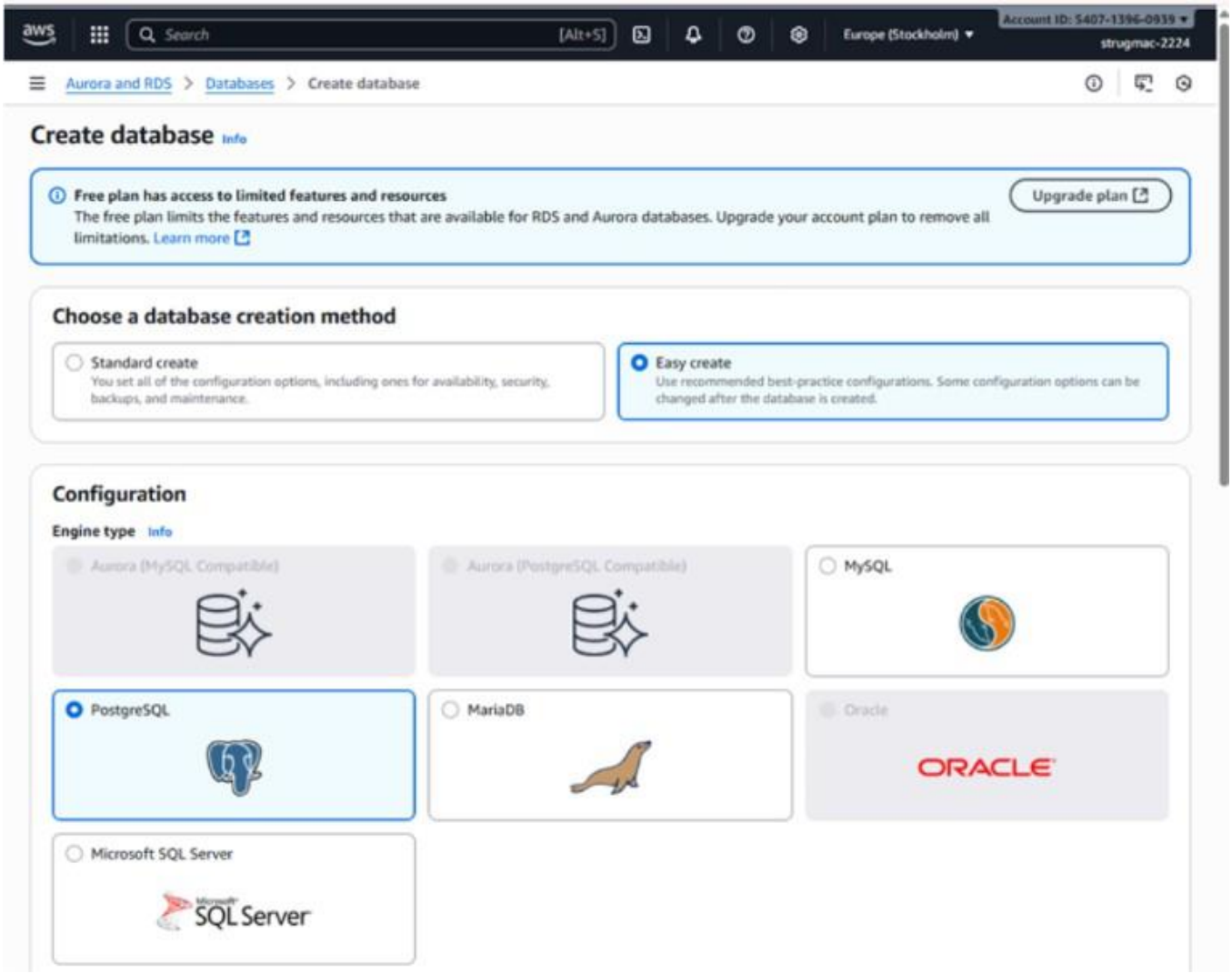
Additionally, AWS RDS integrates with **CloudWatch** for performance monitoring and offers **storage auto-scaling** to handle growing data needs efficiently. Compared to running databases on EC2 instances, RDS provides greater scalability, reduced administrative overhead, and enhanced performance, making it a cost-effective and reliable choice for cloud-based database management.

Procedure:

- Log in to the AWS Management Console using your credentials.
- Search for and open the **RDS** service from the AWS dashboard.
- Click on **Create Database** to start a new RDS instance setup.
- Choose the **Standard Create** option for manual configuration.
- Select the required **database engine** (MySQL or PostgreSQL).
- Enter the **DB instance name**, **master username**, and **password**.
- Choose the **instance class** and configure **storage settings**.
- Enable **storage auto-scaling** if needed.
- Configure **VPC** and **security groups** for database connectivity.
- Set the database to be **publicly accessible** (if connecting locally).
- Enable **automated backups** and optional **Multi-AZ deployment**.
- Review all settings and click **Create Database**.
- Wait for the instance status to become **Available** in the RDS dashboard.
- Copy the **endpoint** (host name) of the created database.
- Open **pgAdmin** or **MySQL Workbench** on your local machine.
- Create a new connection using the endpoint, username, and password.
- If connection fails, modify **inbound rules** in the security group to allow your local IP.
- Test the connection and perform basic SQL operations to verify setup.


RDS
Europe (Stockholm)
Services
Show more
Aurora and RDS
Managed Relational Database Service
Top features
Dashboard Databases Query Editor Performance Insights Snapshots
Database Migration Service
Managed Database Migration Service
Kinesis
Work with Real-Time Streaming Data
Services
Features
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials
Dashboard
Databases
Performance
Snapshots
Exports in Am
Automated ba
Reserved inst
Proxies


Search
[Alt+S]
Europe (Stockholm)
strugmac-2224
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


Search
[Alt+S]
Europe (Stockholm)
Account ID: 5407-1396-0835
strugmac-2224
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
ORACLE
Microsoft SQL Server
SQL Server

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.

strugmac-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.

admin

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: / ' " @

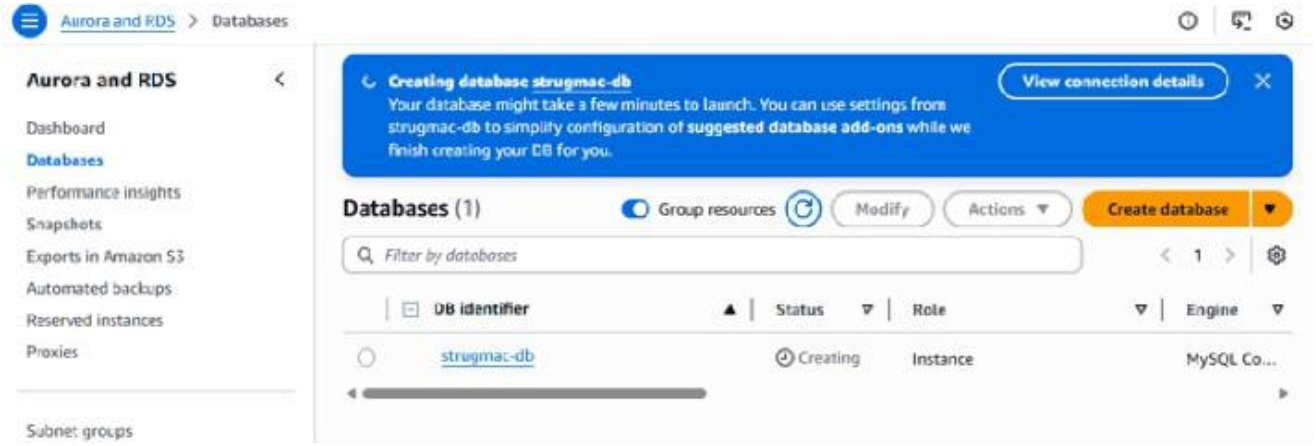
Confirm master password Info

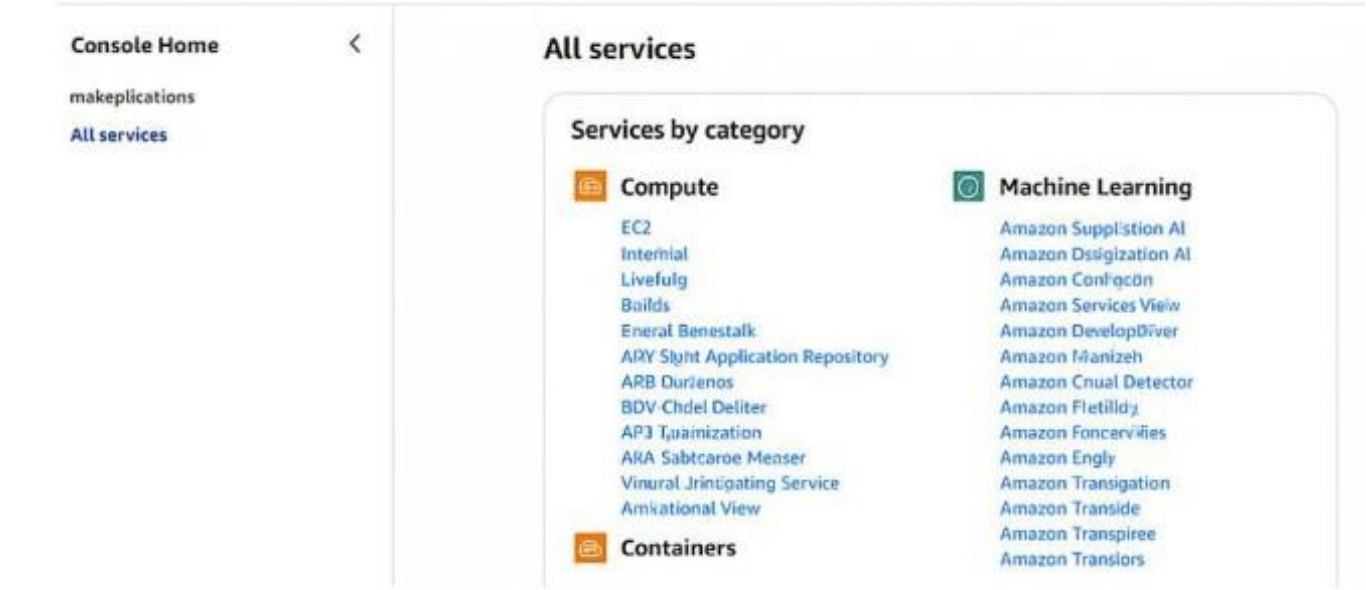
••••••••••••••••

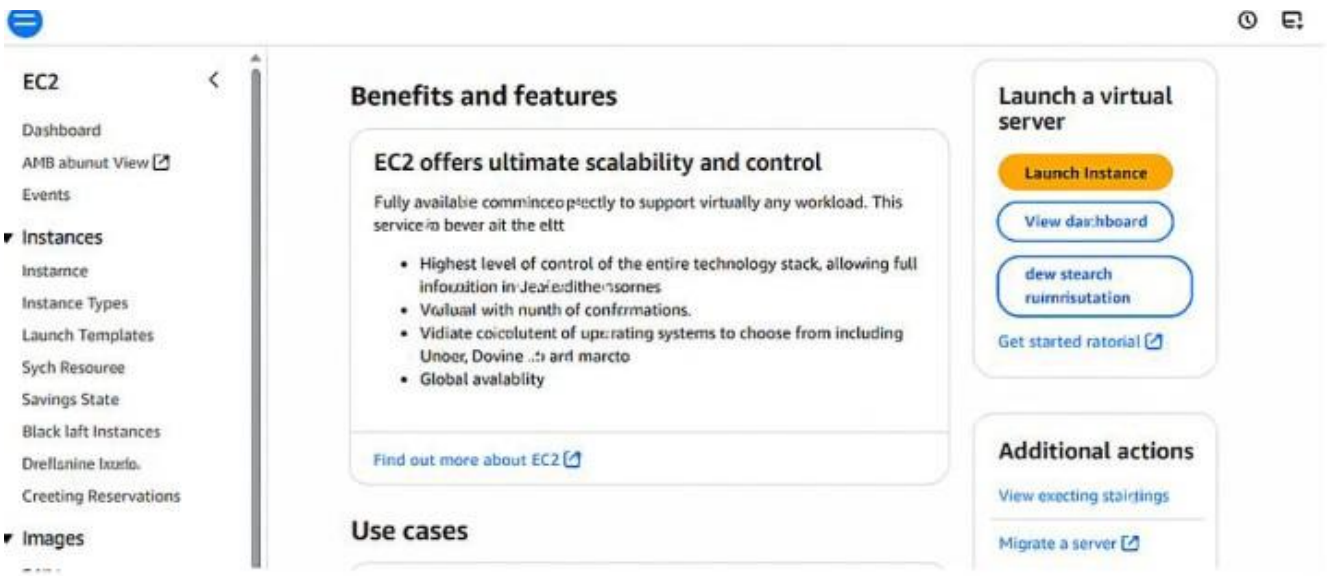
View default settings for Easy create

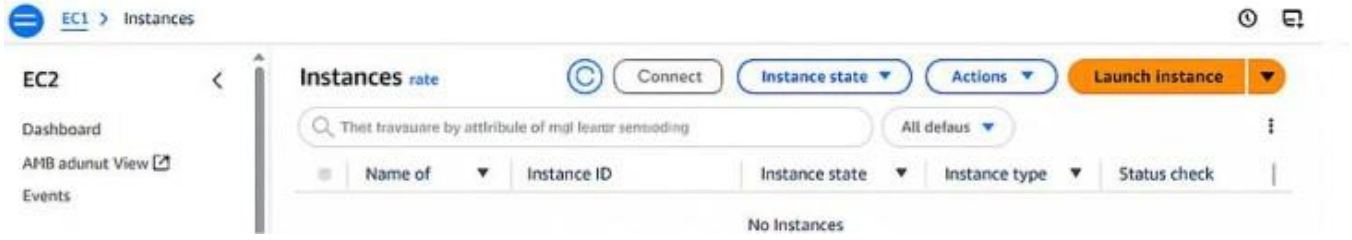
Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use Standard create.

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
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




Console Home
makeplications
All services
All services
Services by category
Compute
EC2
Machine Learning
Containers


EC2
Dashboard
Events
Instances
Instance Types
Launch Templates
Savings State
Images
Benefits and features
EC2 offers ultimate scalability and control
Find out more about EC2
Use cases
Launch a virtual server
Launch Instance
Get started
Additional actions
Migrate a server


EC1 > Instances
EC2
Dashboard
Events
Instances
Connect
Instance state
Actions
Launch instance
Name of
Instance ID
Instance state
Instance type
Status check
No Instances

Application and OS Images (Amazon Machine Image)

An AMI contains operating systems, application servers, and applications lhat for your usr instance. If you we're still, unit g orvistance Blowse more. more.

Search our fult.cotting /helpairiehcluding 1000s of applications and OS images

Search more

Amazon

aws

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

SUSE

Debian

Key pair (login) info

You can use a key pair to secure with to your instance. 1lr that t have access to t the selected key pair you launch the instance.

Key pair recommended

Proceed without a key pair (Not recommennded) Default value

▼ Network settings info

Edit

Network info

vpc. 65f01/eh72teb7h9

Subnet info

No preference: (Default subnel in anvailability zone)

Auto-assign public IP info

Enable

Firev alt (security groups) info

Cuuntoto seacurg prouphave best creatid that cantrol to parfic a rom the instants. You hitures allows spents a pfic treffic to reach instances.

- ○ Create security group
- ◉ Select existing security group

Common security groups info

Select security groups ▼

default - sg 067lettb482425 ×
default

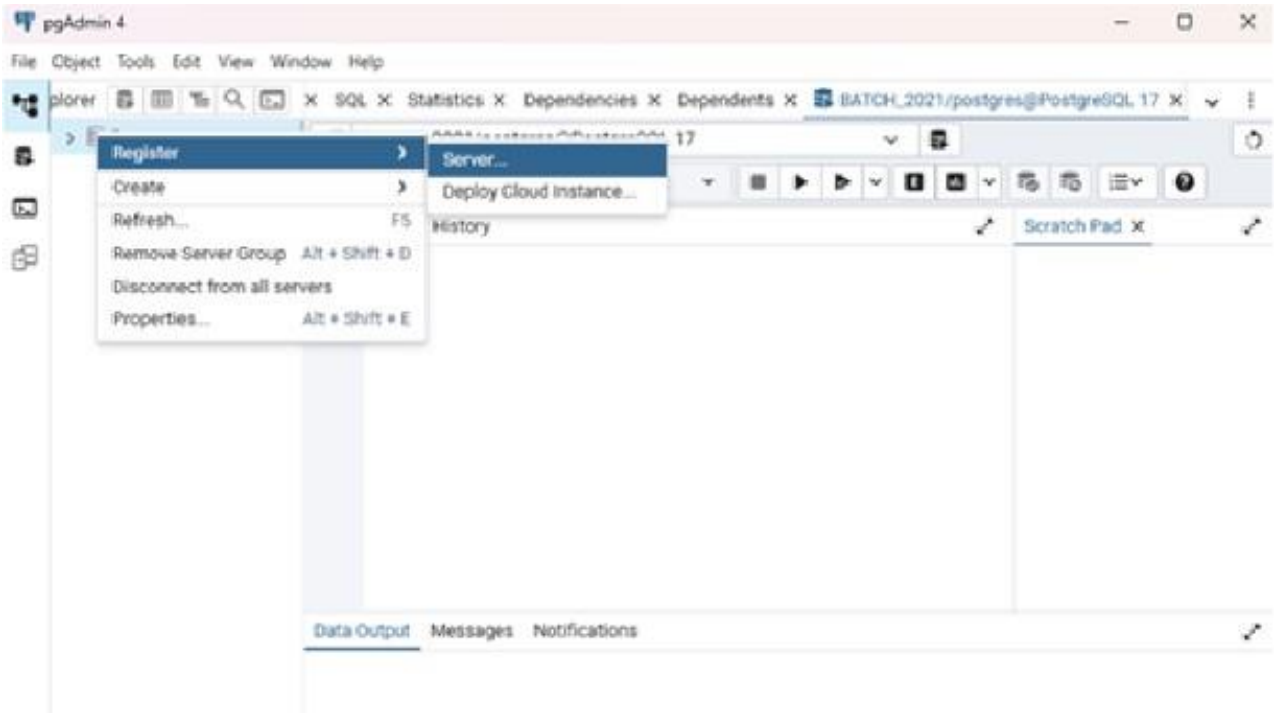
Security groups that youl add or remove fromwill be fied or removed from all your network interfaces.

▼ Summary

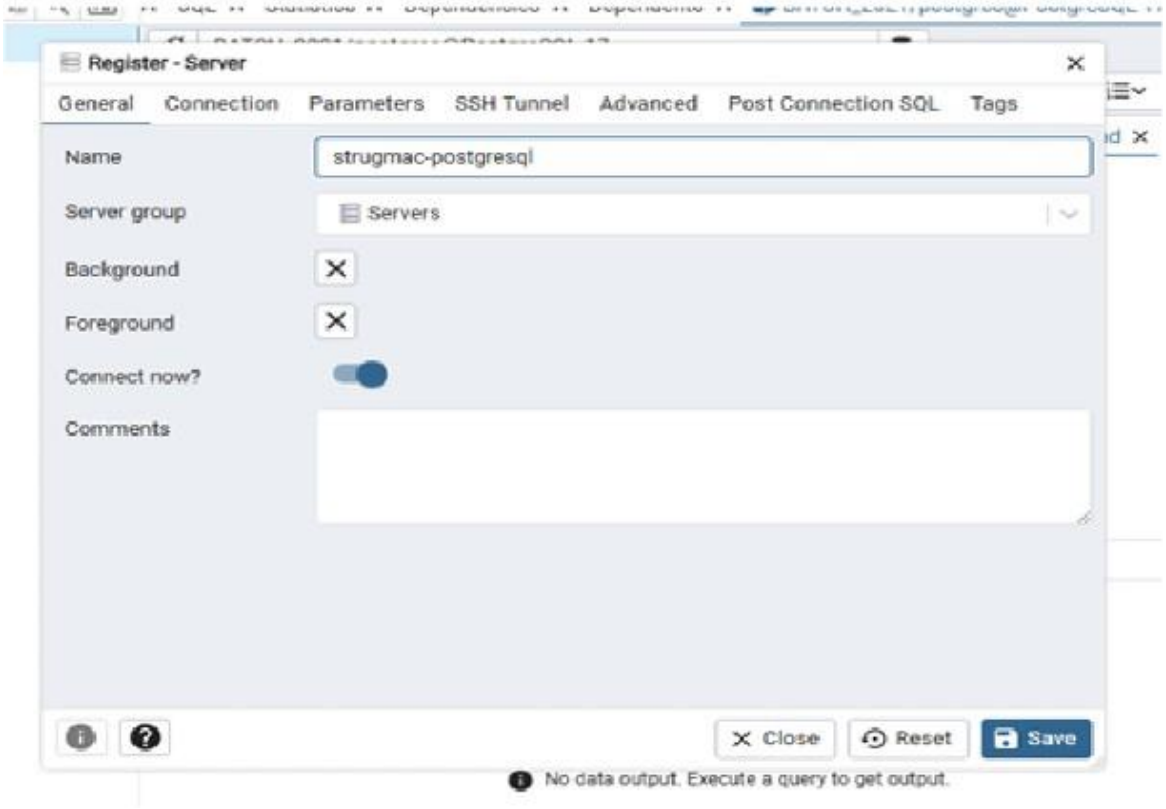
Number of instances: 1

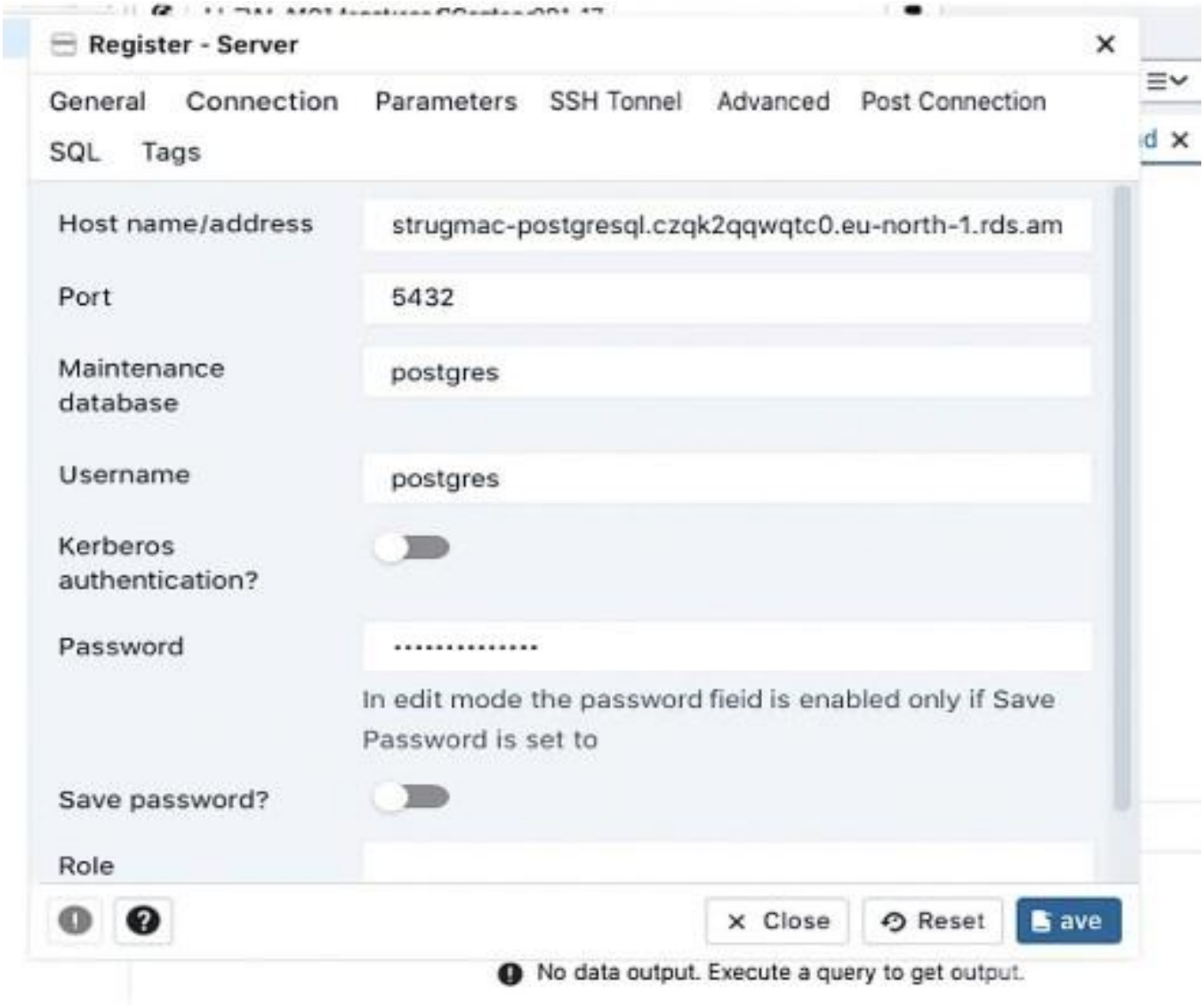
Cancel

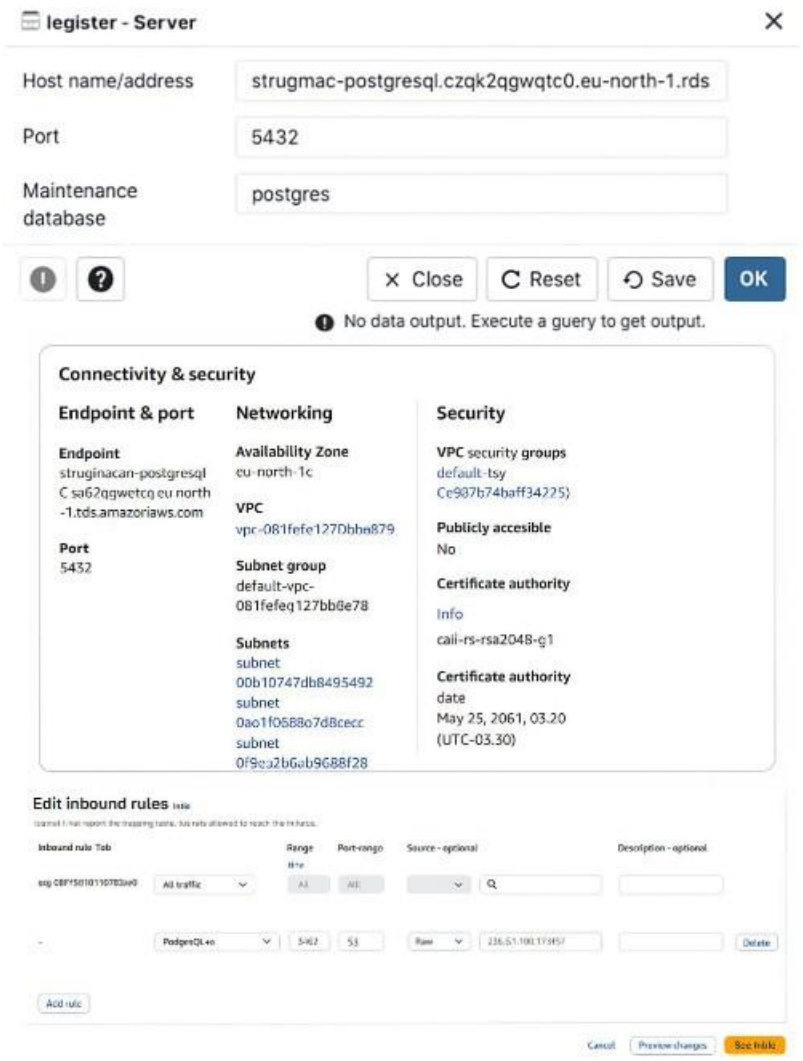
Launch instance



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




Register - Server
General
Connection
Parameters
SSH Tonnel
Advanced
Post Connection
SQL
Tags
Host name/address
strugmac-postgresql.czqk2qqwqtc0.eu-north-1.rds.am
Port
5432
Maintenance database
postgres
Username
postgres
Kerberos authentication?
Password
In edit mode the password fieid is enabled only if Save Password is set to
Save password?
Role
Close
Reset
ave
No data output. Execute a query to get output.



Learning Outcome:

- Understood the concept and functionality of **Amazon RDS (Relational Database Service)**.
- Learned how to **create and configure a database instance** on AWS RDS.
- Gained practical knowledge of **connecting AWS RDS with local pgAdmin/MySQL Workbench**.
- Understood how to **set up and modify VPC security groups** to manage inbound and outbound traffic.
- Learned to **edit inbound rules** to allow database access from specific IP addresses securely.
- Understood the importance of **network configuration and security** in cloud database management.
- Learned to **troubleshoot common connectivity errors** like "connection timeout expired."
- Gained experience in **launching and configuring EC2 instances** for database connectivity.
- Understood **differences between databases on EC2 and AWS RDS** in terms of scalability and maintenance.
- Developed the ability to **deploy, manage, and secure cloud-based relational databases** effectively.