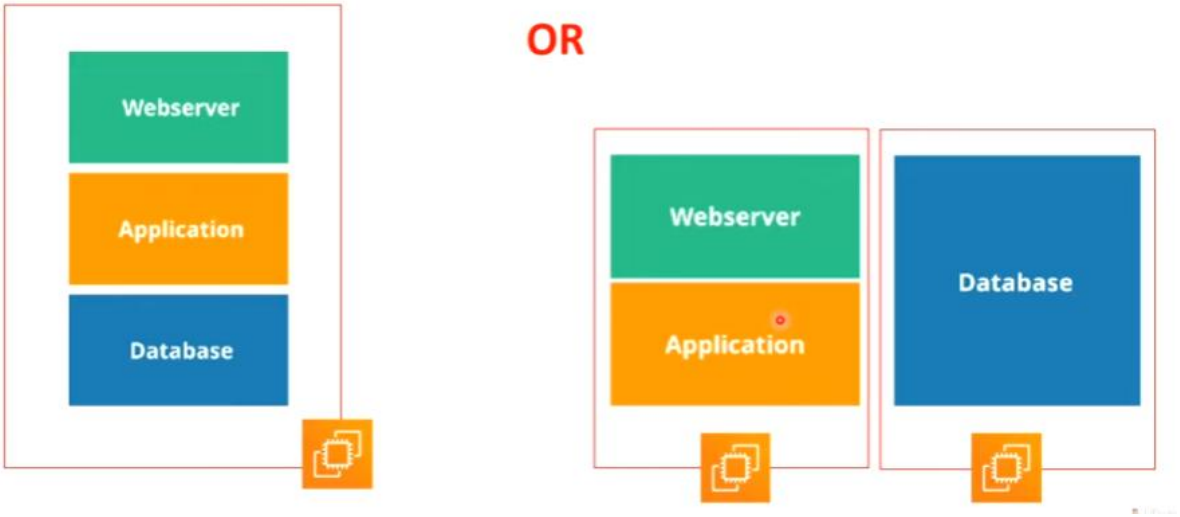
Experiment 10: AWS RDS

AWS RDS

- OVERVIEW OF AWS RDS
- CREATION OF DATABASE INSTANCE ON AWS RDS
- SECURITY GROUPS
- CONNECTING LOCAL PGADMIN TO CLOUD RDS

1. Overview of Database: SQL & NOSQL

DATABASES ON EC2 INSTANCE



3-tier Architecture 2-tier architecture

- **3-tier architecture**: where we can run all the 3 components in the same EC2 instance. **Not recommended**

- 2-tier architecture: Where application + webserver runs on one EC2 instance & Database runs on another EC2 instance & finally we can place both of these EC2 instances in one availability zone.

Why should you run DBs on EC2 instances?

1 Access to the DB instance OS.

2 Advanced DB option tuning (DBROOT)

3 Vendor demands.

4 DB version that AWS doesn't provide.

5. You might need a specific version of an OS and DB that AWS doesn't provide.

Why shouldn't you run DBs on EC2 instances?

1 Admin overhead

2 Backup and DR (Disaster Recovery)

3 EC2 is running in a single AZ.

4 Will miss out on features from AWS DB products.

5 Skills and setup time to monitor

6 Performance will be slower than AWS options.

EC2 has some restrictions and disadvantages so, we try not to use the EC2 instances for Databases.

Hence, we use RDS to overcome these disadvantages.

---------- AWS RDS----------

WHY AWS RDS: SCALABILITY, PERFORMANCE, ADVANCED SECURITY, COST-EFFECTIVENESS, MANAGEABILITY, AVAILABILITY, DURABILITY

- INTRODUCTION
- WHY CHOOSE RDS OVER EC2 OR ON-PREMISE SETUP?
- RDS CONCEPTS
- HANDS ON

Amazon RDS is a managed relational database service provided by AWS. It makes it easier to set up, operate, and scale relational databases in the AWS cloud. Some of its important features include Dashboard, Databases, Query Editor, Performance Insights, and Snapshots among many others.

The different Database engines that are supported by RDS are:

1. MySQL
2. SQL Server
3. PostgreSQL
4. Oracle
5. MariaDB
6. Amazon Aurora
7. Amazon RDS Custom

WHY CHOOSE RDS OVER EC2 & ON-PREMISES?

For a relational database in an on-premises server, you assume full responsibility for the server, operating system, and software. For a database on an Amazon EC2 instance, AWS manages the layers below the operating system. In this way, Amazon EC2 eliminates some of the burden of managing an on-premises database server.

Feature	On-premises management	Amazon EC2 management
Application optimization	Customer	Customer
Scaling	Customer	Customer
High availability	Customer	Customer
Database backups	Customer	Customer
Database software patching	Customer	Customer
Database software install	Customer	Customer
Operating system (OS) patching	Customer	Customer
OS installation	Customer	Customer
Server maintenance	Customer	AWS
Hardware lifecycle	Customer	AWS
Power, network, and cooling	Customer	AWS

Feature	Description	Purpose
CloudWatch	Monitors RDS metrics, sends alarms, and stores logs	Performance & health monitoring
Automated Backups	Daily snapshots and transaction logs	Disaster recovery

Manual Snapshots	User-created backups	Long-term retention
Multi-AZ Deployment	Standby replica in another AZ	High availability
Read Replicas	Read-only copies	Load balancing
Security (IAM, KMS, SSL)	Data protection and access control	Compliance & safety
Performance Insights	SQL and load analysis tool	Performance tuning
Storage Auto Scaling	Grows storage automatically	Prevents space outages
Enhanced Monitoring	OS-level real-time metrics	Deep diagnostics
Cross-Region Replicas	Replication to other regions	Global availability
Parameter/Option Groups	DB configuration controls	Customization

----------EC2 VS RDS----------

Amazon RDS is a managed database service. It's responsible for most management tasks. By eliminating tedious manual tasks, Amazon RDS frees you to focus on your application and your users. Amazon themselves suggests customers use RDS over EC2 for relational databases.

Feature	Amazon EC2 management	Amazon RDS management
Application optimization	Customer	Customer
Scaling	Customer	AWS
High availability	Customer	AWS
Database backups	Customer	AWS
Database software patching	Customer	AWS
Database software install	Customer	AWS
OS patching	Customer	AWS
OS installation	Customer	AWS
Server maintenance	AWS	AWS
Hardware lifecycle	AWS	AWS
Power, network, and cooling	AWS	AWS

RDS CONCEPTS:

DB instances

A DB instance is an isolated database environment in the AWS Cloud. The basic building block of Amazon RDS is the DB instance.

DB engines

A DB engine is the specific relational database software that runs on your DB instance. Amazon RDS currently supports the following engines:

- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

DB instance classes

A DB instance class determines the computation and memory capacity of a DB instance. A DB instance class consists of both the DB instance type and the size. Each instance type offers different compute, memory, and storage capabilities.

DB instance storage

Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance. DB instance storage comes in the following types:

- General Purpose (SSD)
- Provisioned IOPS (PIOPS)
- Magnetic

BY DEFAULT GRAVITON 2 IS THE PROCESSOR PROVIDED BY DEFAULT DATABASE ON AWS.

SIMILARY BASED ON DB INSTANCE CLASSES WE CAN OPT FOR DIFFERENT PROCESSORS AS WELL

EBS IS NOTHING BUT A VIRTUAL HARD-DRIVE ATTACHED TO ALREADY RUNNING DATABASES INSTANCES.

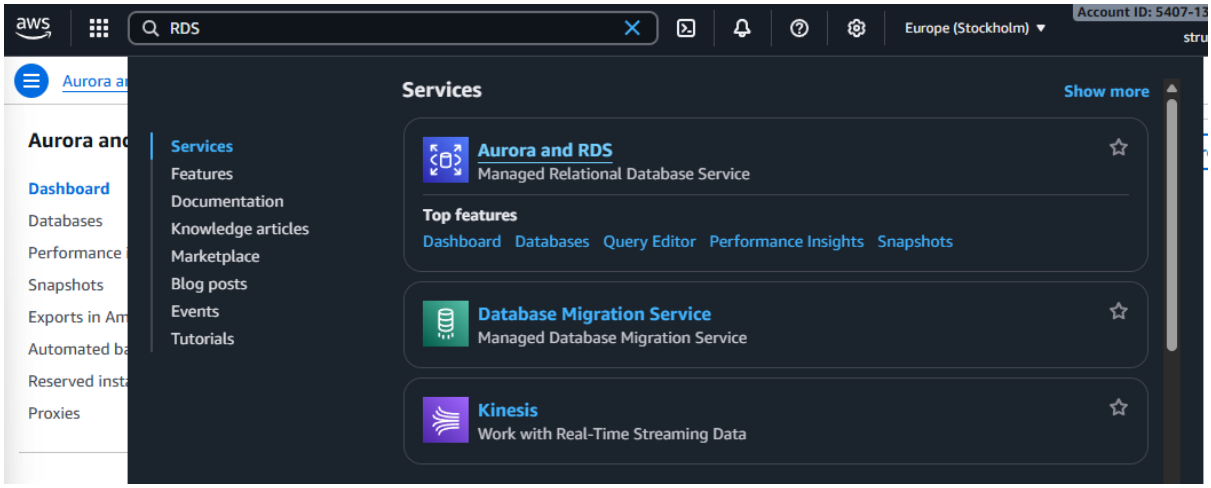
DATABASE INSTANCE STORAGE: MEANS HOW MUCH INPUT / OUPTUT WE WANT

AGAIN WE HAVE AUTO-SCALING AS WELL, MEANS IF YOUR DB SIZE GROWS, IT WILL AUTOMATICALLY SCALE. (BE CAUTIOUS WITH THE PRICES OF AWS)

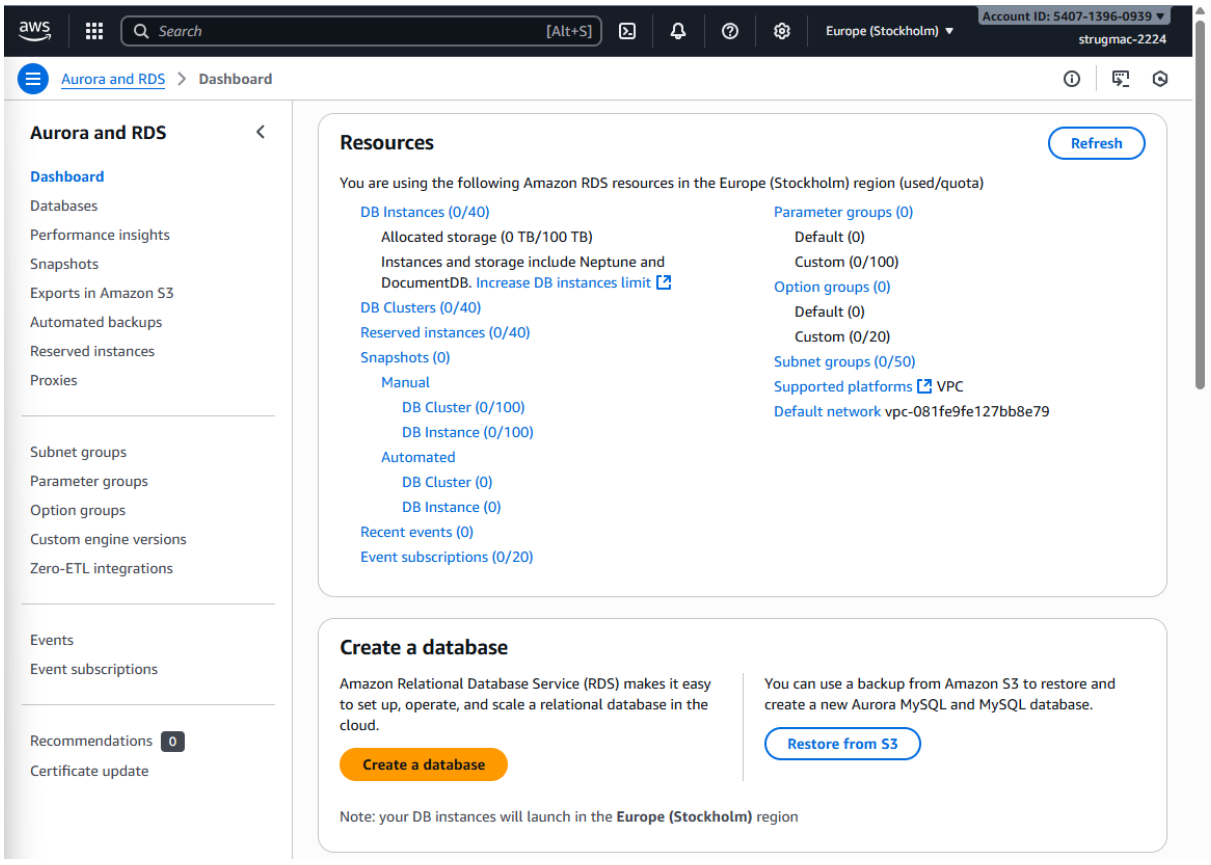
HANDSON;

1. GO TO AWS HOMEPAGE -> CLICK ON SIGN IN-> ENTER USER NAME WITH EMAIL ADDRESS.

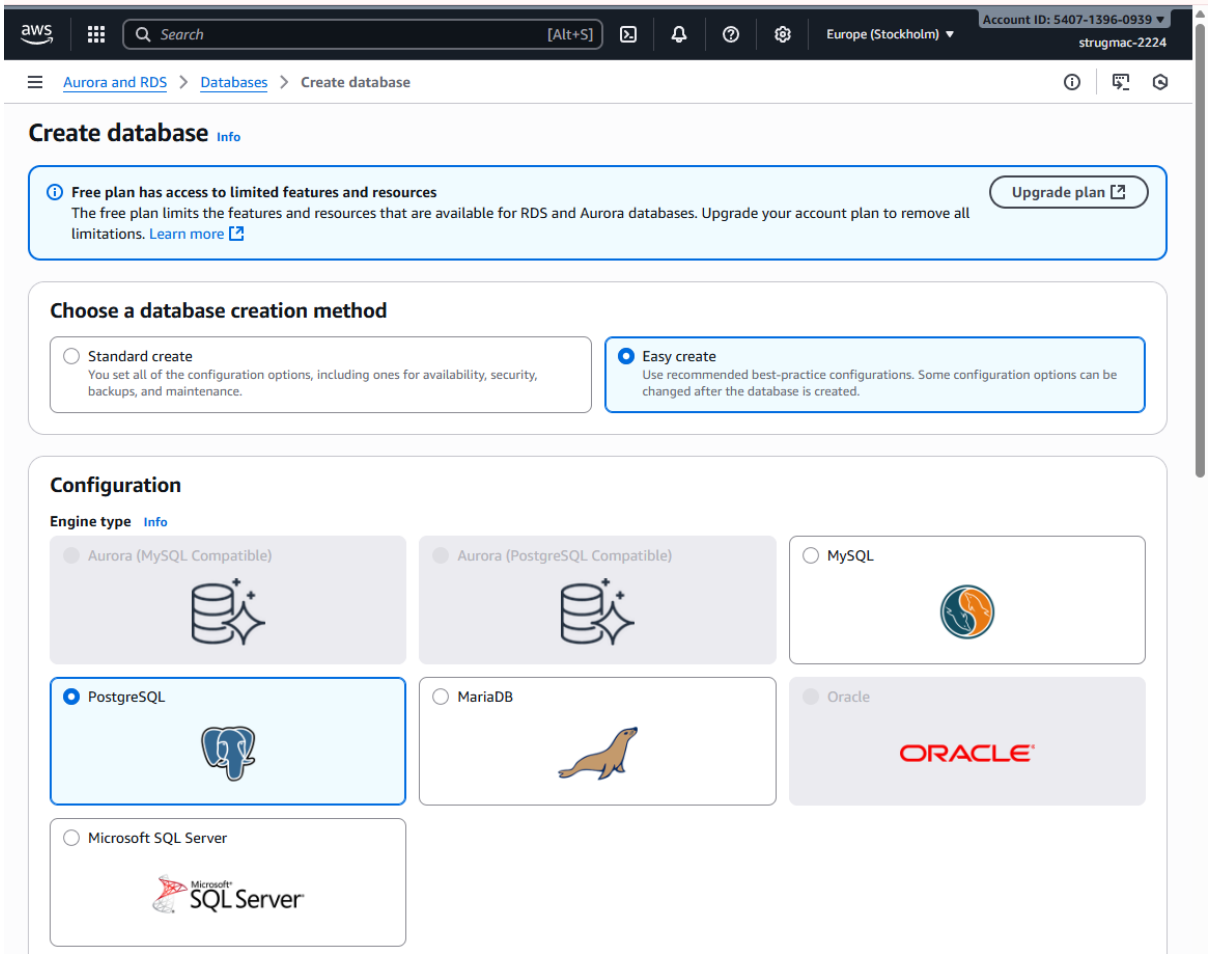
2. AFTER SIGN-IN -> GO TO SEARCH BAR -> SEARCH FOR RDS -> HIT ENTER



3. HOW TO CREATE MY SQL DATABASE INSTANCE ON AWS RDS?



4. CLICK ON CREATE DATABASE



IN THE STANDALONE CREATE, WE CAN SET EVERYTHING FOR OUR DATABASE, THE INCOMING TRAFFIC, IP ADDRESSES TO BE USED, BACKUP ETC.

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: / ' " @

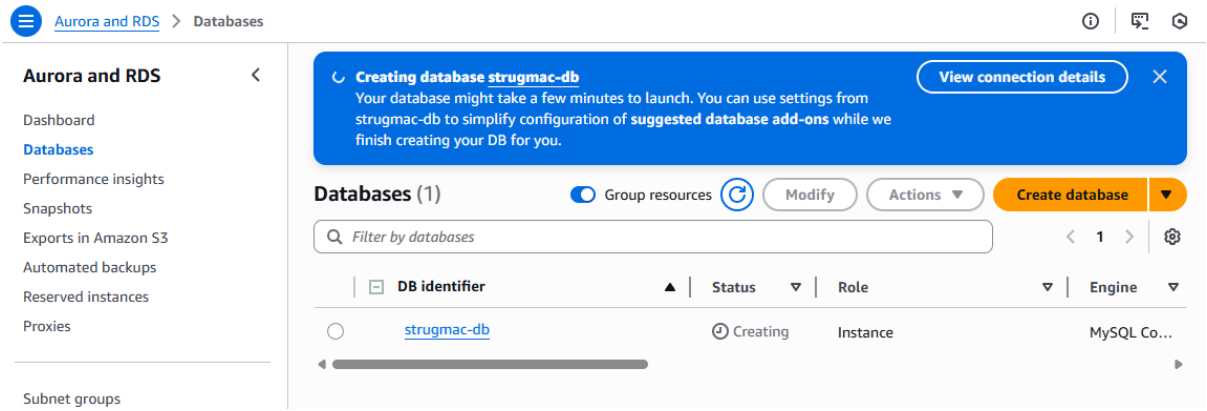
Confirm master password | Info

••••••••••••••••

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.

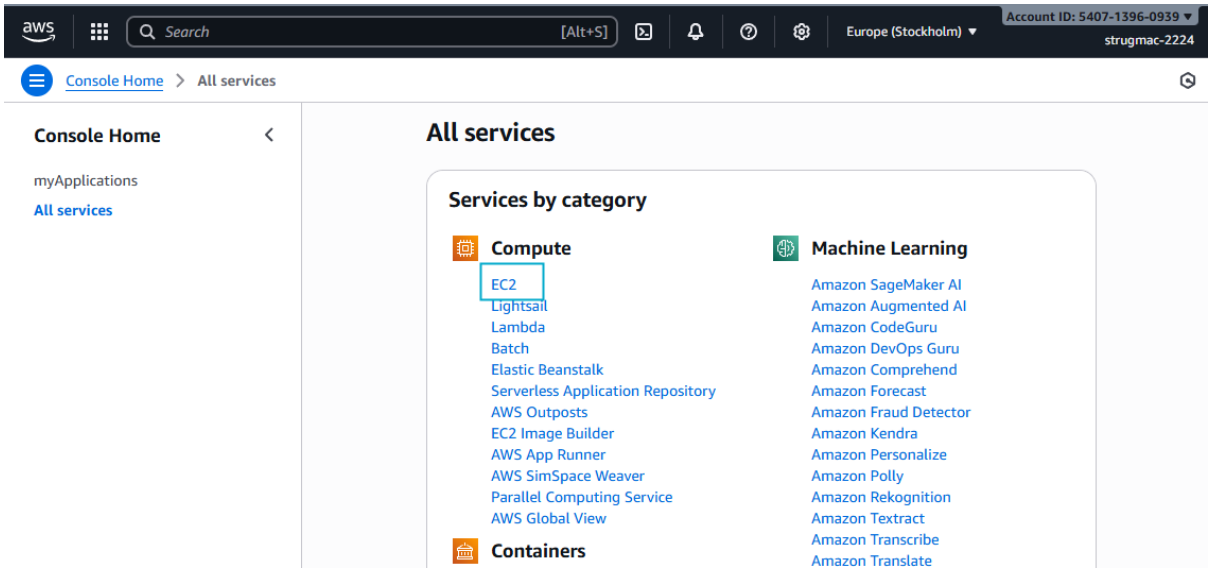
Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-081fe9fe127bb8e79)	No
Multi-AZ	No	Yes
Option group	default:mysql-8-0	Yes
Subnet group	Create new DB Subnet Group	Yes
Automatic backups	Enabled	Yes
VPC security group	default	Yes
Publicly accessible	No	Yes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

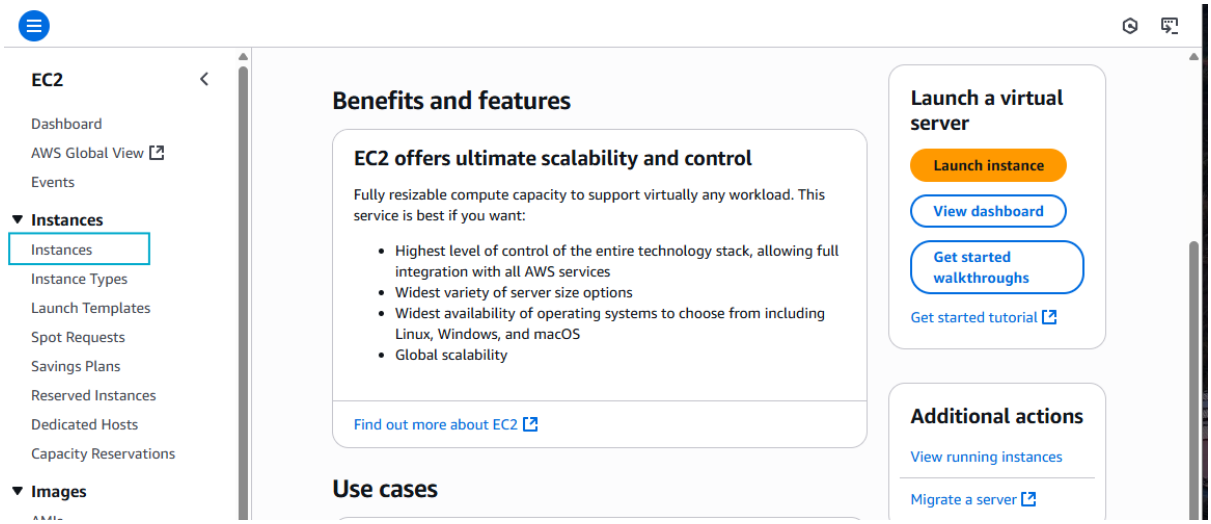


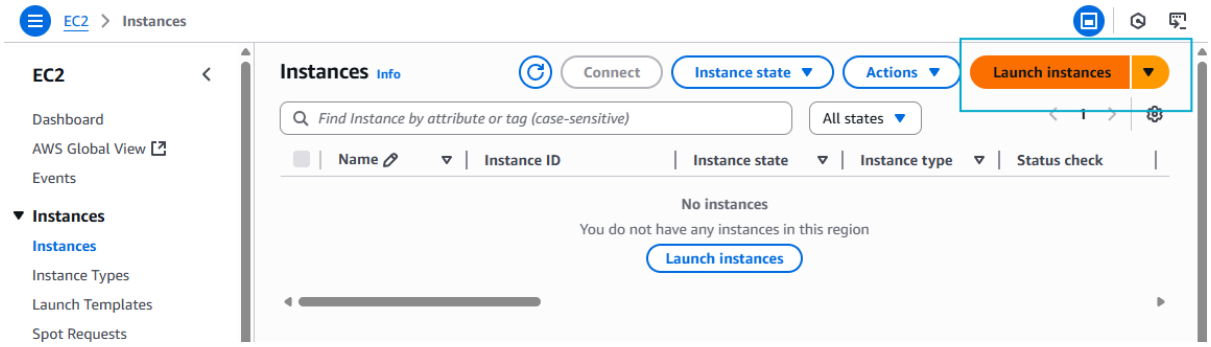
Now this will create a MySQL database to me, and we want to connect to RDS for which we have to launch a server which basically will have MySQL Client installed inside it.

For that we have to launch an EC2 instance,

Launching an EC2 instance







Application and OS Images (Amazon Machine Image) Info

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**.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Key pair (login) Info

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

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, amd6...read more
ami-0a716d3f3b16d290c

Virtual server type (instance type)
t3.micro

Firewall (security group)
default

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

Cancel

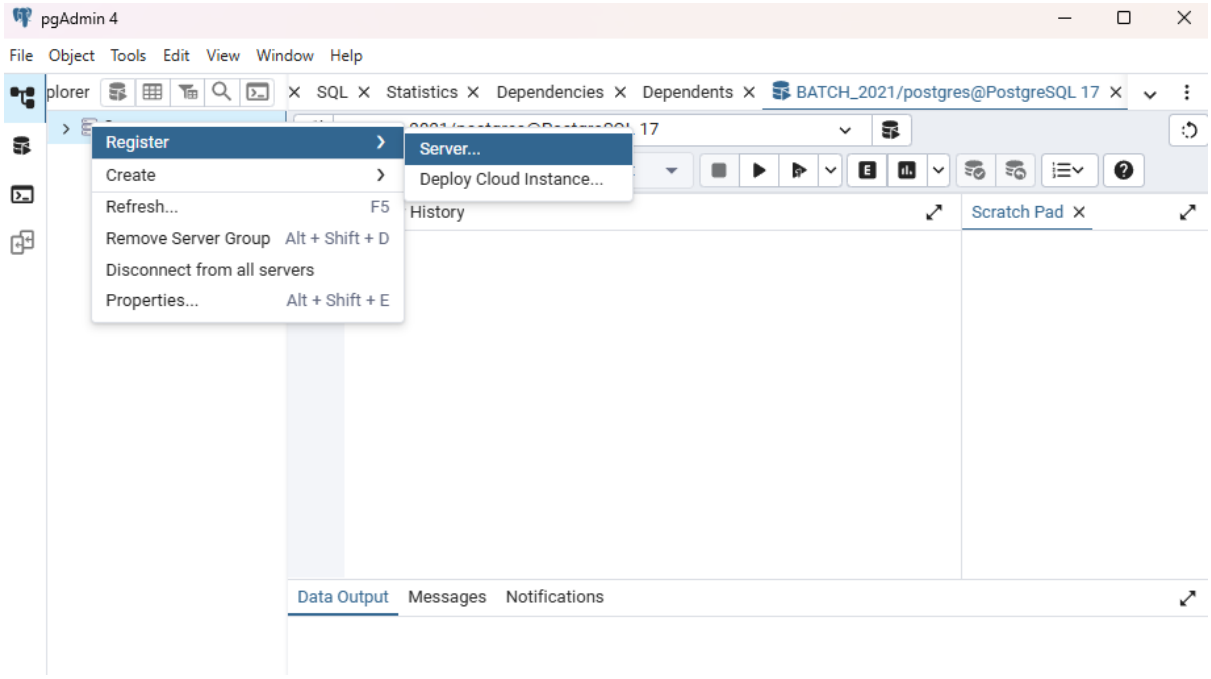
Launch instance

Preview code

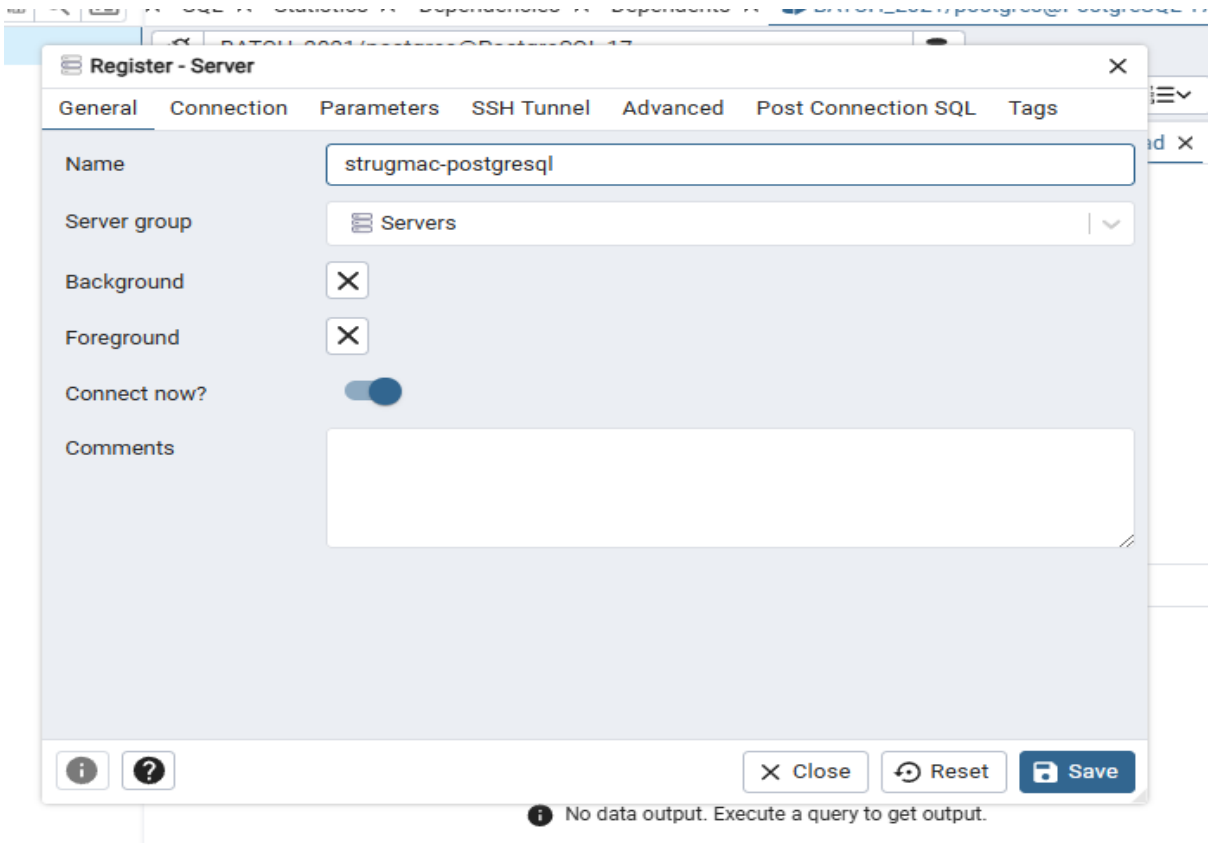
Other option is that we can connect the Postgres AWS RDS to our local machine.

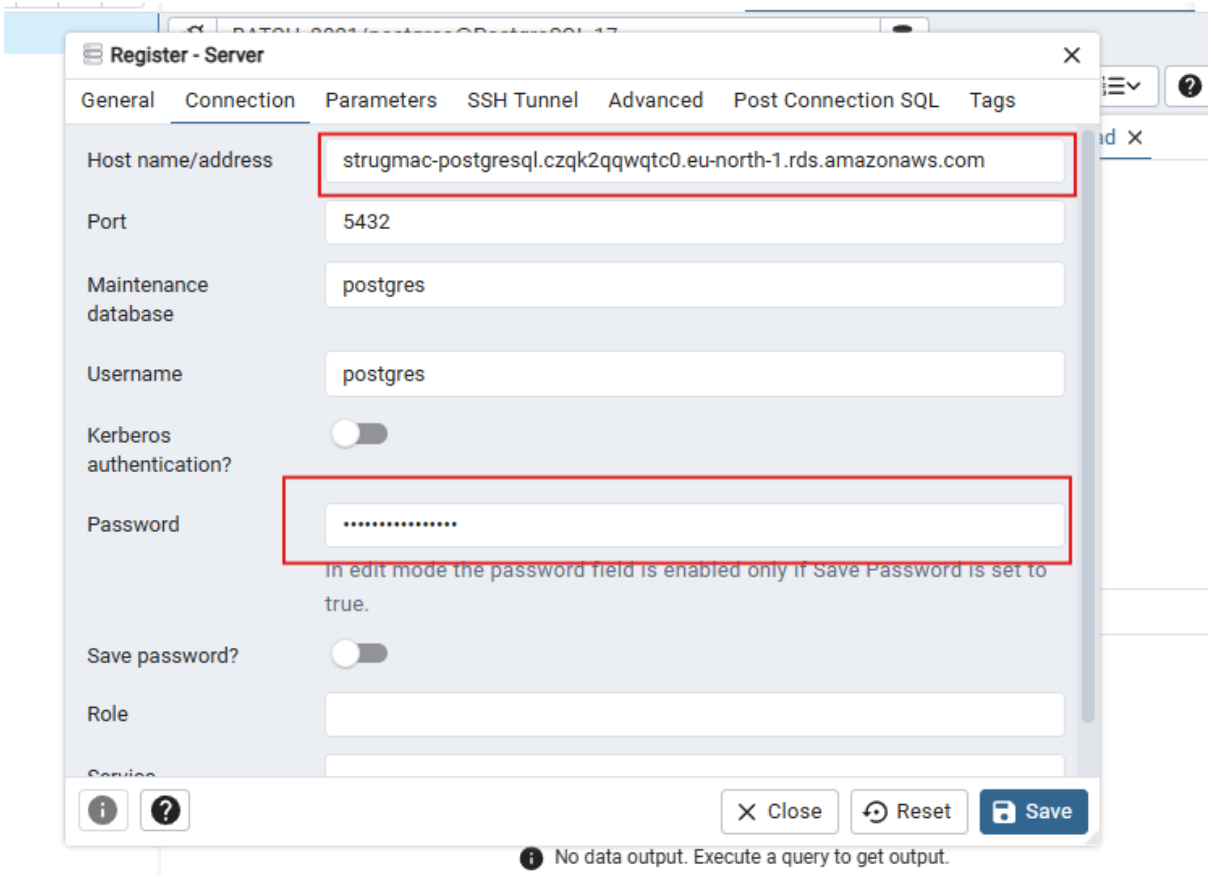
1. Create AWS RDS database for PostgresSQL

2. Connect from PgAdmin.

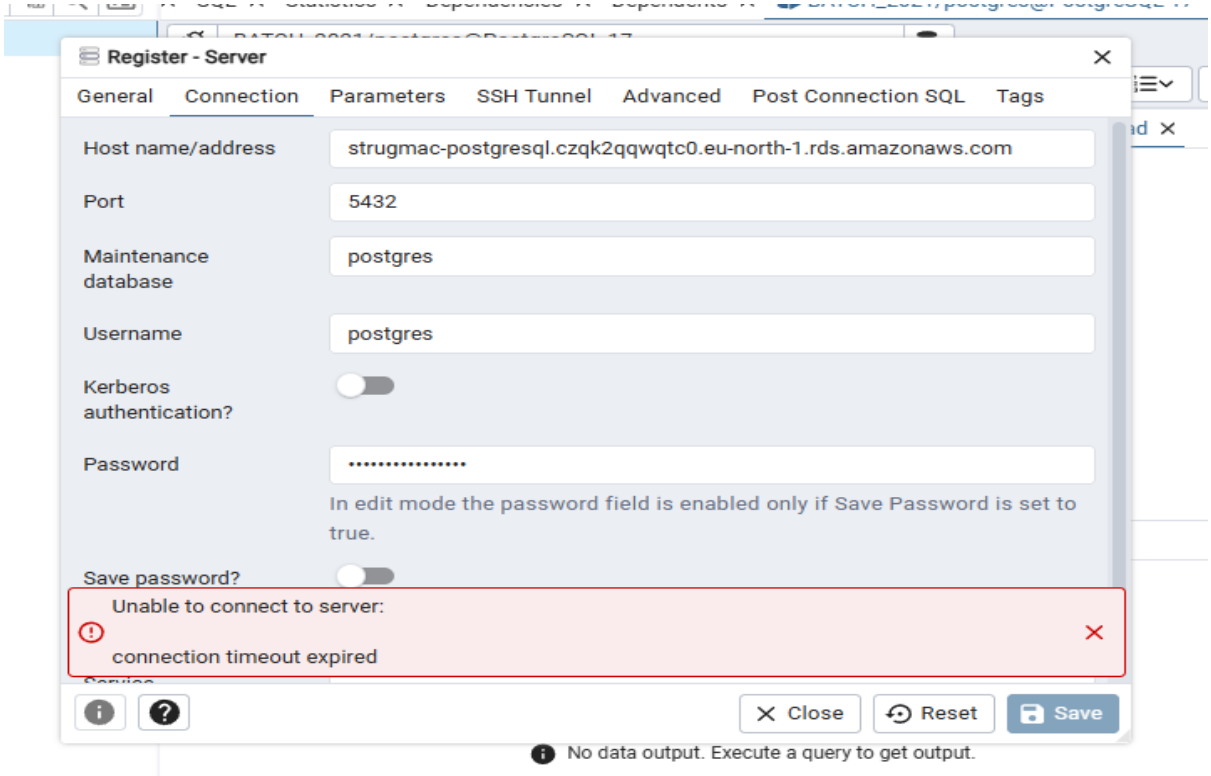


Copy the API Endpoints from the dashboard of AWS RDS Database instance.





Click on Save



Might give this error as this DB instance is not available locally.

Change the INBOUND RULES of DB Instance from the AWS Console