Relational Database (SQL) services (optional)

AWS has a managed relational database service called RDS. Ideally we would allow each student to configure their own database using RDS. However, AWS imposes a strict 100 database limit, which has been a significant problem in previous years. This year we have a new approach which will allow all students to use an SQL database in their projects, but you will no longer be able to configure it yourself.

The approach works like this. DBS has created a large PostgreSQL database on RDS and has configured separate a schema for each student, accessible with a preset username and password. Your application will use these credentials to access the database and create the tables etc. necessary. You can also access the database with the postgresql command line utility using these credentials.

Due to the shared nature of the database, DBS has also added connection time limits to ensure that nobody is hogging the shared resource.

Connecting to the shared PostgreSQL database

You will only be able to connect to the database from within the AWS environment, for example from an EC2 instance.

Connection information

Each student will receive a unique username and password via Canvas messages. Please make sure that you keep these credentials safe as you will need them for connecting to the database. The following information will be required for connecting to the database:

- *Endpoint (host):* `database-1-instance-1.ce2haupt2cta.ap-southeast-2.rds.amazonaws.com`
- *Port:* `5432`
- *Database:* `cohort_2025`
- *Engine:* PostgreSQL (RDS), server v16.x
- *SSL:* Required (`sslmode=require`)

Connecting with psql (CLI)

- If `psql` isn't installed on your EC2 instance, you may need to install the client first: `sudo apt install postgresql-client`
- Then connect with the following command, replacing `<username>` with the username we sent to you

```
psql "host=database-1-instance-1.ce2haupt2cta.ap-southeast-2.rds.amazonaws.com port=5432 dbname=cohort_2025 user=<username> sslmode=require"
```

- When prompted, enter the password that we sent you

Connecting from your app

We recommend that you use a standard library for accessing the database from within your application. You may wish to consider the following:

- Node/Javascript: **node-postgres (https://www.npmjs.com/package/pg)**
- Python: **Psycopg (https://www.psycopg.org/psycopg3/)**

Other database options

We recommend using the PostgreSQL database provided. However, if your application requires MariaDB/MySQL then you can deploy a database yourself. Unfortunately, due to the quota on the number of RDS databases, you won't be able to use RDS. Instead you can deploy the database on an EC2 instance.

We recommend deploying your database using a standard Docker container on a separate EC2 instance. Please refer to **mariadb on Dockerhub (https://hub.docker.com/_/mariadb)** for instructions on running MariaDB (which is compatible with MySQL). You can use other databases if you like.

Launch your EC2 instance for the database as for other EC2 instances in the unit, with the following differences:

- Add `CAB432DBSG` security group, which allows connections on ports 3306 (MariaDB/MySQL) and 22 (SSH) from AWS resources in the `CAB432SG` security group (eg. your application EC2 instance)
- If you need ssh access from other places, you can also add the EC2 instance to `CAB432SG` to allow SSH access from QUT networks, or use the session manager.
- It is appropriate to use a private subnet, as databases generally should *not* be accessible from the internet.

Please keep in mind the following:

- EC2 instances are stopped at 3AM every day. Hence you will need to restart your database EC2 instance for the day if you plan to use it.
- EC2 instances will get a new IP address when they restart. Hence you will need to update the IP address/DNS address of the database in your application whenever the instance is restarted
- It is possible to obtain the IP address of an EC2 instance using the `DescribeInstances` action in the AWS SDK or CLI. The instance ID is stable, so you can search for the instance

with its ID to obtain the IP address or DNS name. We leave it to you to explore this if you desire.

- While you might be tempted to run the database container on the same EC2 instance as your application, this will cause problems for you in assessment 3 when EC2 instance will be automatically created and terminated as your application scales out and back in. Your database will need to be in a separate EC2 instance to isolate it from the scaling process.