Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
○ AWS Introductions
○ AWS Sign up
○ AWS Billing Alert
○ AWS EC2

AWS RDS

Contents

Overview

Amazon **R**elational **D**atabase **S**ervice (RDS) was designed to make it easier to set up, operate, and scale in the cloud.
This is very cost-efficient and has resizeable capacity, and the usual time-consuming
tasks are all automated.

RDS is similar to EC2, where you have the option to choose from different hardware and
software types depending on your needs. You can create a database that's optimized for
memory, performance or I/O. There are even options to
choose which database engine you want to use. These can be, MySQL, PostgreSQL and so on.

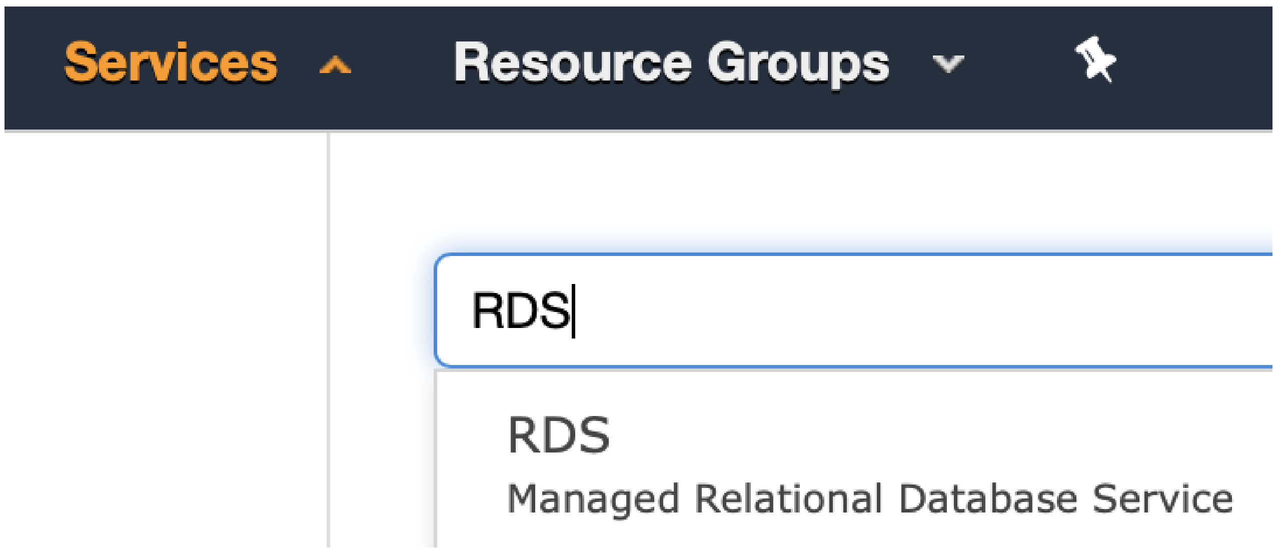
Amazon even offer a database migration service where you can migrate your current
on-premise database to the cloud and on RDS.

Tutorial

Web Console

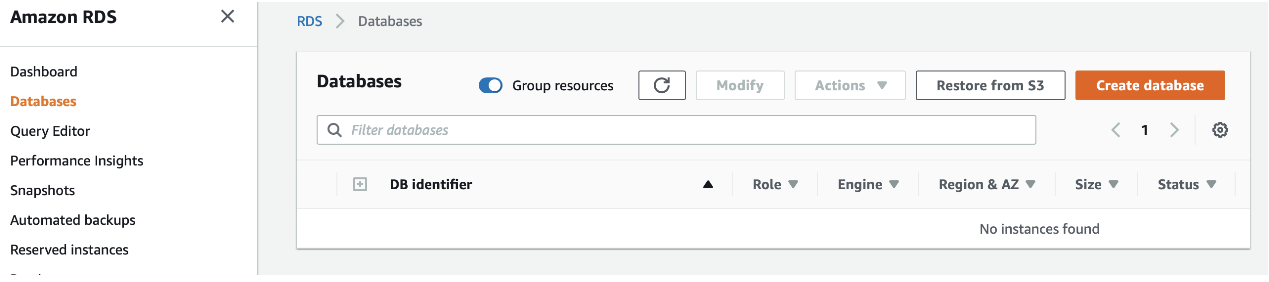
1. Navigate to the AWS Console and sign in here

2. Search for RDS under the services drop-down menu, and click on RDS

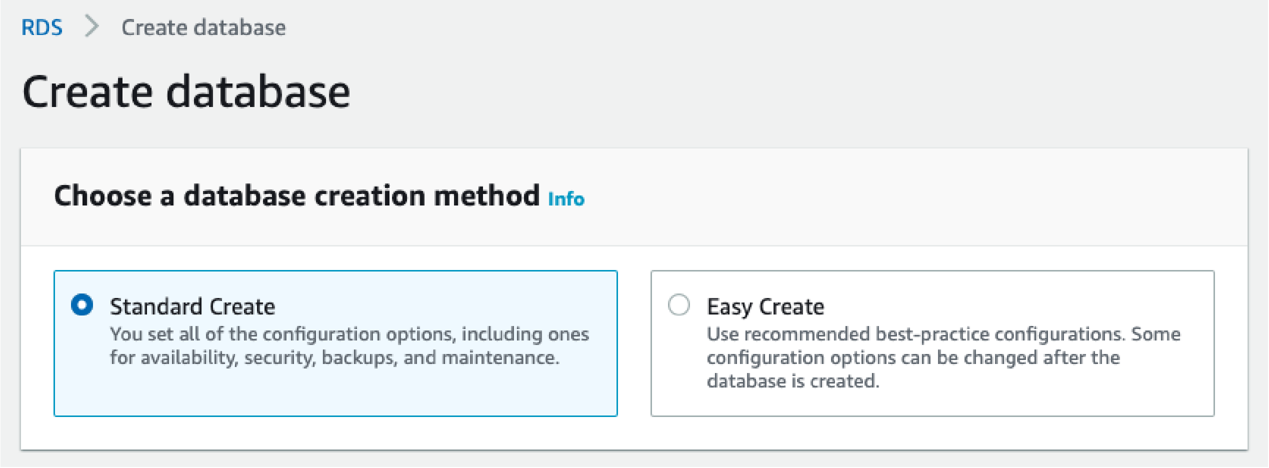


3. You will be redirected to the RDS homepage. On the left panel, click on **Databases**, this will show all of your running databases. Since you have no

running databases, we will need to create one. So click on **Create database**



4. By default, the **Standard Create** is checked, leave this as default otherwise we would have to configure options we don't need to.



RDS > Create database

Create database

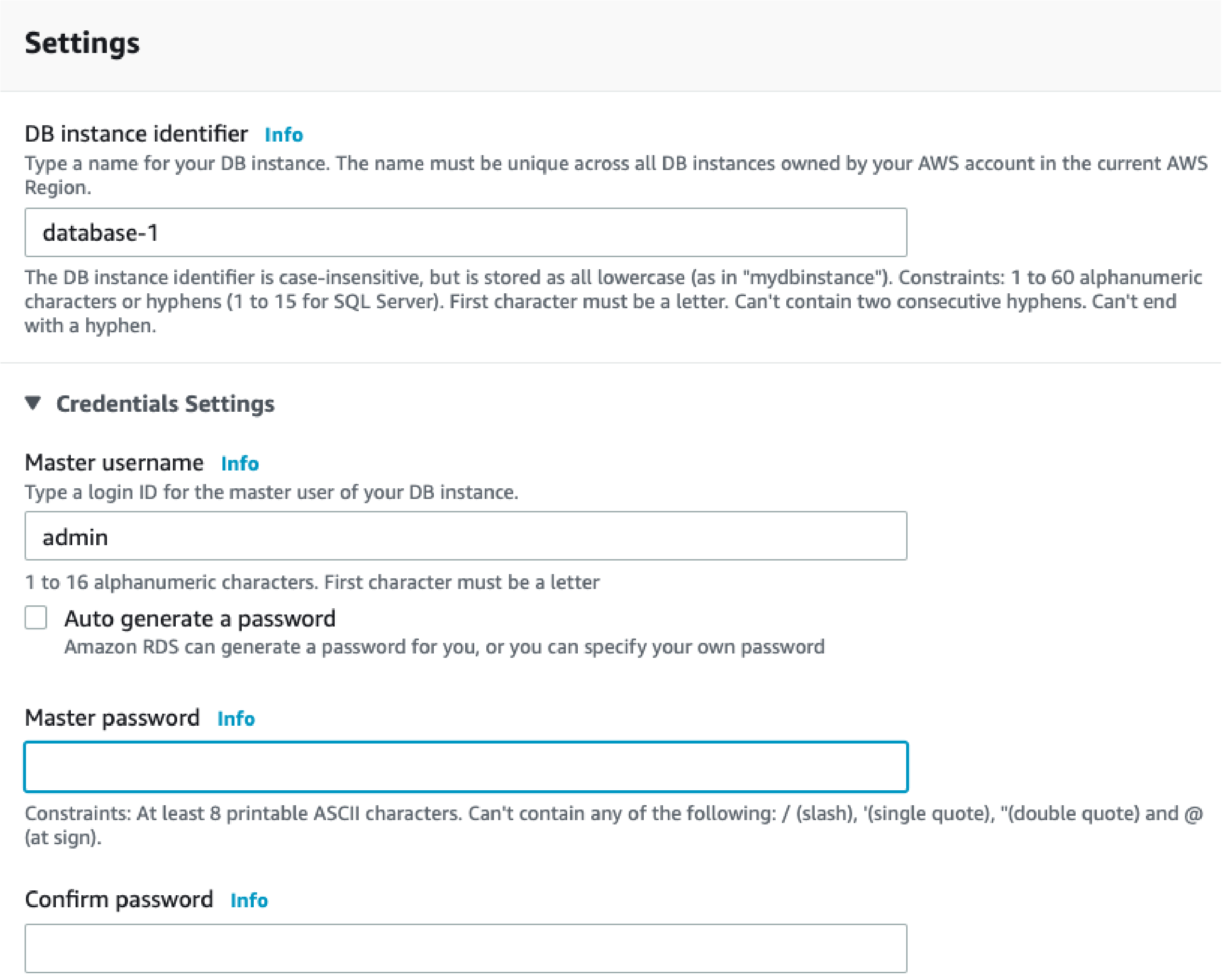
Choose a database creation method Info

● 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.

5. Select **MySQL** as your Database Engine tool, by default AWS will select the "Edition" and "Version". These will change in the future, due to MySQL constantly updating. Under the "Templates" section, select "Free tier".

6. Complete the *Settings* section which is configuring how you can connect to your RDS. The username, password and DB identifier are important to create and remember so that we can connect to this database.



Settings

DB instance identifier Info
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.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. First character must be a letter

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

Master password Info

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm password Info

7. Leave the rest of the configuration as default. This will result in making your RDS instance run in the

Default VPC, **20 GB Scalable Storage**, Hardware spec of **db.t2.micro**. Proceed to **Create database**.
The RDS instance may take up to 20 minutes to create.

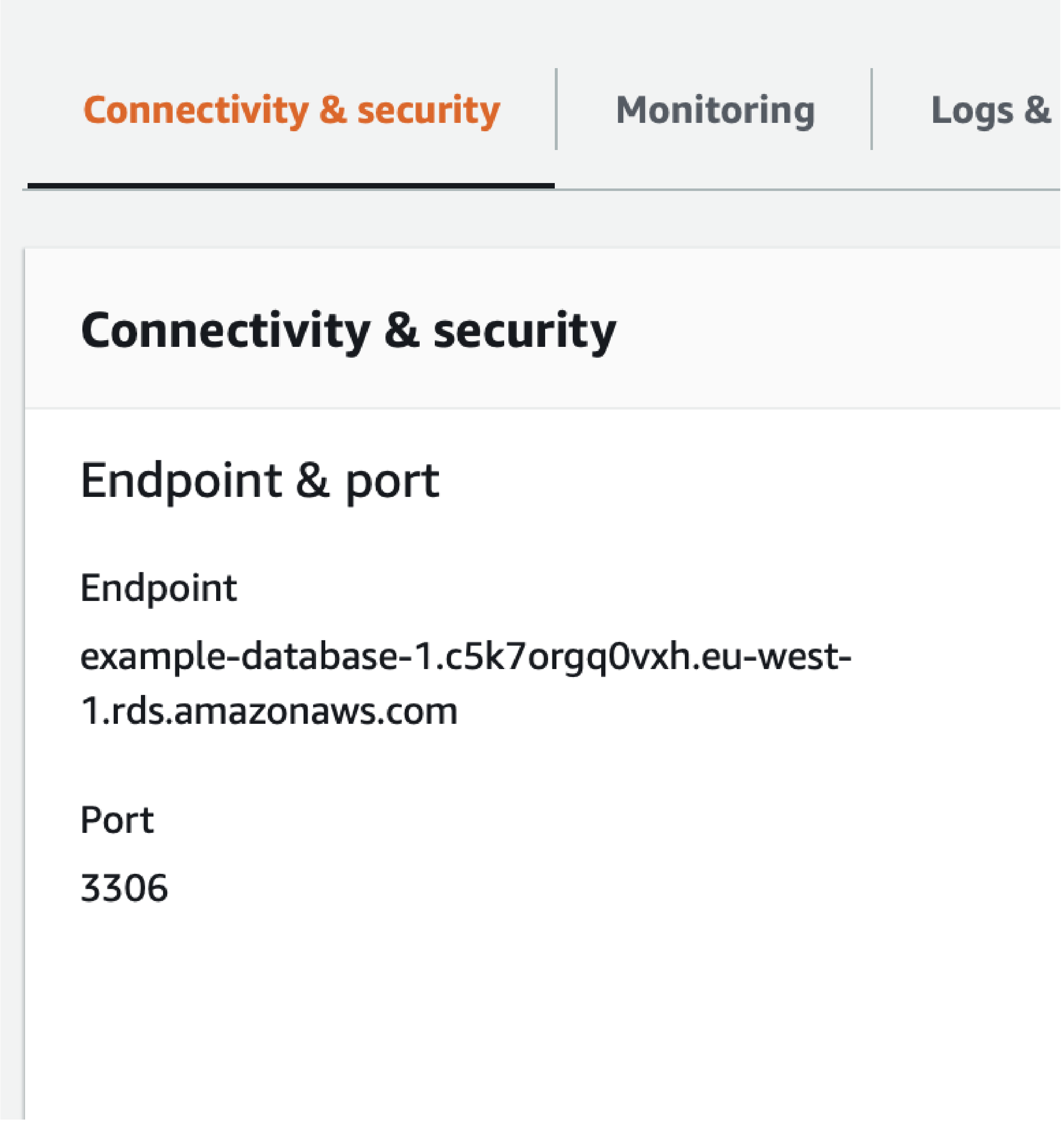
8. Create an EC2 instance in the **Default VPC**. Make sure the OS of this EC2 instance is Ubuntu 18.04LTS.
SSH into the new EC2 instance and execute the following commands:

```
sudo apt update -y && sudo apt install mysql-client-core-5.7 -y
```

9. Now you can run **mysql** commands in the terminal. To connect to your database, you will need to run the following command:

```
mysql -h (your RDS endpoint) -P 3306 -u admin -p
```

The RDS endpoint is located in your RDS Connectivity and Security section. Navigate to your list of RDS -> Select the RDS you want to connect to -> Under *Connectivity and security* -> **Endpoint**. Copy this endpoint and paste it to the command.



Connectivity & security

Endpoint & port

Endpoint

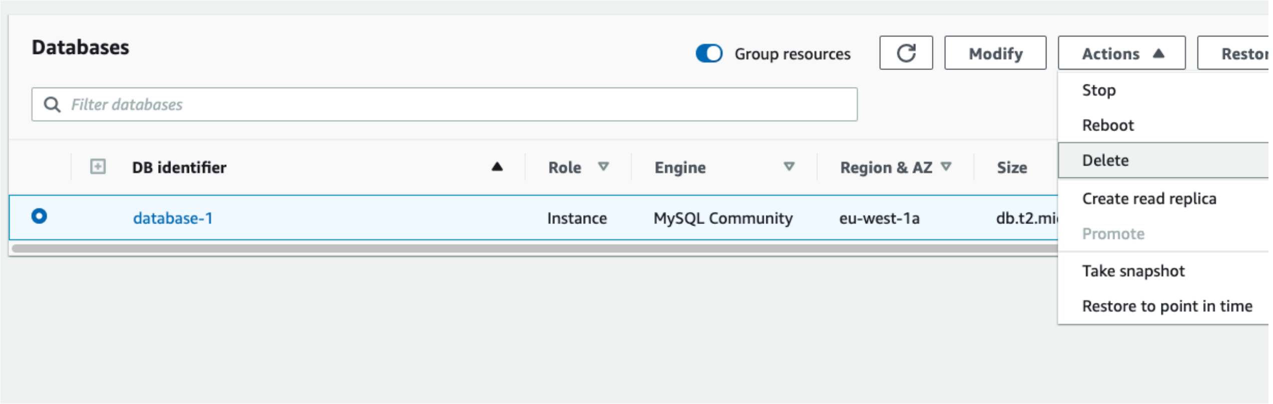
example-database-1.c5k7orgq0vxh.eu-west-1.rds.amazonaws.com

Port

3306

10. Clean up your AWS environment. Delete your database by selecting the database you want to delete -> Click on

Actions -> Delete.



	DB identifier	Role	Engine	Region & AZ	Size
●	database-1	Instance	MySQL Community	eu-west-1a	db.t2.mi

11. You will be prompted with options if you want to create a snapshot of your instance. Un-check this option and check the third option that you **"Acknowledge you wont be able to recover the database after deletion"**. Then confirm by typing in "delete me" in the empty text field, then click on the **Delete** button.

AWS CLI

We will be creating RDS instances using the AWS CLI. There are some concepts to understand regarding the parameters that will be used to create these instances.

DB Name

This is where you define the name for your database. Quite simple, and allows you to identify when looking at your list of database.

DB Instance Identifier

This is needed as this is the unique identifier which is part of the endpoint used to connect to the rds instance.

Allocated Storage

This is to define the size of your RDS instance in GB. When using the web console, this is set to being 20 GB by default. When using the AWS CLI to create an RDS instance you need to define the size manually.

Engine

This is needed to define which type of database engine you want to run, whether it be MySQL, PostgreSQL or other SQL types.

Database Instance Class

This is defining the hardware specifications of the database. Similar to defining the hardware specifications of an EC2 instance with predefined RAM, CPU Cores and so on. **db.t2.micro** is the free-tier use instance size along with MySQL engine.

Master Username

You will be required to make a master account for your RDS instance, this account will have the highest privilege.

Master User Password

This is the password associated with the master account.

1. Create your first RDS instance through the AWS CLI using the following command:

```
aws rds create-db-instance --db-name (define your own rds name) --db-instance-identifier (define your own identifier) --allocated-storage 20 --engine mysql --db-instance-class db.t2.micro --master-username admin --master-user-password (define your own password for your master username)
```

2. Create an EC2 instance running the following command:

```
aws ec2 run-instances --image-id ami-04137ed1a354f54c4 --count 1 --instance-type t2.micro --key-name (key-pair-name)
```

This will create an instance in the Default VPC, which your RDS instance should also be located.

3. SSH into the EC2 instance and run the following command to install the mysql client:

```
sudo apt update -y && sudo apt install mysql-client-core-5.7 -y
```

4. You will need to make sure you have the endpoint of the instance you want to connect to. Run the following command to get all the information about your RDS instances:

```
aws rds describe-db-instances
```

This will output the value in JSON format. Copy the **Endpoint Address** of the instance you want to connect to.

5. Connect to the RDS instance by running the following command in the EC2 instance:

```
mysql -h (your rds instance endpoint address) -P 3306 -u (your rds instance master username) -p
```

6. Clean your AWS environment. Delete both the EC2 instance and the RDS instance.

```
# Get instance id
aws ec2 describe-instances
```

Copy the instance id you want to delete

```
# Delete the instance
aws ec2 terminate-instances --instance-ids (your instance id)
```

```
# Delete RDS instance
aws rds delete-db-instance --db-instance-identifier (your db indentifier) --skip-final-snapshot
```

Exercises

There are no exercises for this module.