Provision a MySQL Server (Google Cloud Platform)

Contents

Overview

Google Cloud Platform (GCP) is a collection of modular Cloud computing services which run on the same internal infrastructure that Google uses for its end-user products, such as *YouTube*, *AdSense*, and its eponymous *Search*.

Amongst other services, a portable MySQL server can be provisioned directly within GCP itself, without needing to hook into, for instance, a Windows installation.

By using direct provisioning, this will devolve responsibility for managing database connections from the user to GCP.

This is an example of Platform-as-a-Service (PaaS) in-action: GCP takes control of configuration from the runtime level upwards.

Tutorial

Here, we'll provision an embedded MySQL server within Google Cloud Platform:

Provision a GCP MySQL Server

1. Navigate to Google Cloud Platform, and sign in using your usual credentials. (If you are already signed in, click the Console button in the top-right of the page.)
2. Once you have successfully signed in, click the burger icon in the top-left and navigate to the **Storage** -> **SQL** option.
3. In the **Cloud SQL** instances box, click **Create Instance**.
4. Within the **MySQL** option on the left panel, click the **Choose MySQL** button.
5. In the **Create a MySQL Second Generation** instance panel, change the Instance info to the following values:

   a. Set the **Instance ID** to `sakila`.

   b. Set the **Root password** to `root`.

   c. Set the **Region** to `europe-west1` (or whichever Region is closest to your geographic location).

   d. Leave all other options at their defaults, then click **Create**.

(*Note: This MySQL instance may take up to five minutes to be created.*)

Connecting to MySQL inside GCP

1. Once the MySQL instance has been successfully generated, click the Activate Cloud Shell button on the top navigation bar. This will open a command-line interface at the foot of the page. (You may see a Welcome panel; click Start if you do).

2. Inside the Cloud Shell, enter the following command:

```
gcloud sql connect sakila --user=root --quiet
```

   You should receive an output whitelisting your IP address so it can connect to the device:

```
Whitelisting your IP for incoming connection for 5 minutes… done.
```

3. You'll be prompted to enter your password; enter root and hit Enter. If successful, you'll see something like this:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 5283
Server version: 5.7.14-google-log (Google)
```

   At this point, you can open the **Cloud Shell** in a new tab, if you wish to, by clicking the **Open in new window** button on the top right of the **Cloud Shell** panel.

Uploading to the GCP MySQL Server Instance

The `sakila.sql` file is available by clicking the `Download Resources` button at the top of this page.

1. To use the Sakila database to your Cloud instance of MySQL, you will first need to create a database in which you will import its data. Enter the following SQL commands into the **Cloud Console**:

```
CREATE DATABASE sakila;
USE sakila;
```

   You should receive a `Database changed` output, and your active database should be changed to `sakila`.

2. You can now upload the database by clicking the **More** button (three dots) on the top right of the **Cloud Shell** panel, then clicking **Upload File** and navigating to wherever you stored the `sakila.sql` file.

   (*Note: This could take up to 5 minutes, depending on network speed.*)

   Once the file has finished uploading, you can close the **File Upload** panel.

3. The sakila.sql file contains all the SQL commands you will need to generate and populate the Sakila database. To build the database, run the files by entering the following commands:

```
SOURCE sakila.sql;
```

   If you have successfully followed these instructions, you should see several `Query OK` outputs.

Exercises

You should also have the `world.sql` file.

Follow the above instructions with the World database by uploading the `world.sql` file and using similar commands:

```
CREATE DATABASE IF NOT EXISTS world;
USE world;
SOURCE world.sql;
```