Francesc
de Borja Moll
Centre Integrat de
Formació Professional

P2: DOCKERIZED DATABASE IN THE CLOUD

Albert Perelló Puertas
IFC31W Group

INDEX

1. Use google's cloud shell that has docker installed on it - to run a mysql docker database in detached mode. Use an environment variable to set the password.

In this practice we will create a database with MySQL. To do this, we will start by checking that we have docker already installed in the Google Cloud Shell.

```
aperellop@cloudshell:~$ docker -v
Docker version 19.03.13, build 4484c46d9d
```

The next thing to do is find a docker hub (a container that interests us) and install it with the indicated command.

```
docker pull mysql
```

```
aperellop@cloudshell:~$ docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
bb79b6b2107f: Pulling fs layer
49e22f6fb9f7: Pull complete
842b1255668c: Pull complete
9f48d1f43000: Pull complete
c693f0615bce: Pull complete
8a621b9dbed2: Pull complete
0807d32aef13: Pull complete
a56aca0feb17: Pull complete
de9d45fd0f07: Pull complete
1d68a49161cc: Pull complete
d16d318b774e: Pull complete
49e112c55976: Pull complete
Digest: sha256:8c17271df53ee3b843d6e16d46cff13f22c9c04d6982eb15a9a47bd5c9ac7e2d
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
aperellop@cloudshell:~$
```

Once the download is finished, we will start the container with the command

sudo docker run --name "NombreDelContenedor" -e MYSQL_ROOT_PASSWORD="Password" -d mysql:latest

```
aperellop@cloudshell:~$ sudo docker run --name database -e MYSQL_ROOT_PASSWORD=database -d mysql
:latest
91ff0e7e669f6ee1aa85b152065bc166f55e18067c704a92e21a5fd5ec0d3612
aperellop@cloudshell:~$
```

2. Following the class exercise University - with its university database schema - that you can check in our Google Drive:

- **Create this database.**
- **Use this database.**
- **Run the SQL script that you can find here in order to create a functional university organization database.**
- **Describe the database information organized in tables.**
- **Using some SQL instructions from https://www.w3schools.com/sql/default.asp**
- **inspect the data from the university database.**

If after executing the start command of the container, we want to verify that it is active, we can do it with

docker ps

If it is active, it will return a list of the containers that are being executed, where we will find the one we just created.

```
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
        PORTS                  NAMES
91ff0e7e669f        mysql:latest        "docker-entrypoint.s…"   52 seconds ago      Up 50 secon
ds       3306/tcp, 33060/tcp   database
```

Now, we will access the container terminal.

```
aperellop@cloudshell:~$ docker exec -it database mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Once we have accessed the MySQL console we will create and access a database using:

CREATE DATABASE “NombreDeLaBaseDeDatos”
USE “NombreDeLaBaseDeDatos”

```
mysql> CREATE DATABASE university;
Query OK, 1 row affected (0.01 sec)
```

Once we have accessed the MySQL console we will create and access a database using:

CREATE DATABASE “NombreDeLaBaseDeDatos”
USE “NombreDeLaBaseDeDatos”

Next, we will copy the table provided by the teacher and paste it in the console. This will begin to execute the SQL instructions of the text that we have pasted, we give enter and we will have finished.

In the case of wanting to check that it has worked and investigate it thoroughly, we use the following commands

SHOW TABLES;
SELECT * FROM “NombreDeLaTabla”;

```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_university |
+----------------------+
| advisor              |
| classroom            |
| course               |
| department           |
| instructor           |
| prereq               |
| section              |
| student              |
| takes                |
| teaches              |
| time_slot            |
+----------------------+
11 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM course;
+-----------+-----------------------------+------------+----------+
| course_id | title                       | dept_name  | credits |
+-----------+-----------------------------+------------+----------+
| BIO-101   | Intro. to Biology           | Biology    |       4 |
| BIO-301   | Genetics                    | Biology    |       4 |
| BIO-399   | Computational Biology       | Biology    |       3 |
| CS-101    | Intro. to Computer Science  | Comp. Sci. |       4 |
| CS-190    | Game Design                 | Comp. Sci. |       4 |
| CS-315    | Robotics                    | Comp. Sci. |       3 |
| CS-319    | Image Processing            | Comp. Sci. |       3 |
| CS-347    | Database System Concepts    | Comp. Sci. |       3 |
| EE-181    | Intro. to Digital Systems   | Elec. Eng. |       3 |
| FIN-201   | Investment Banking          | Finance    |       3 |
| HIS-351   | World History               | History    |       3 |
| MU-199    | Music Video Production      | Music      |       3 |
| PHY-101   | Physical Principles         | Physics    |       4 |
+-----------+-----------------------------+------------+----------+
13 rows in set (0.00 sec)
```

3. Repeat the two previous points (1 and 2) in a Ubuntu virtual machine from Microsoft Azure using your Azure for Students subscription. Install all the docker software that you need.

We create a virtual machine in microsoft azure.

We start the virtual machine.

```
PS C:\Users\apere> ssh aperellop@51.103.49.10
The authenticity of host '51.103.49.10 (51.103.49.10)' can't be established.
ECDSA key fingerprint is SHA256:zMF6cLikrNIB1evUm8+aGLNGu5+/beRddrnqmKpIco8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '51.103.49.10' (ECDSA) to the list of known hosts.
aperellop@51.103.49.10's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1031-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.comWelcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1031-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Oct 28 14:04:57 UTC 2020

  System load:  0.4               Processes:           113
  Usage of /:   4.4% of 28.90GB   Users logged in:     0
  Memory usage: 14%               IP address for eth0: 10.0.0.6
  Swap usage:   0%

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

aperellop@albert-ubuntu: $
```

Install docker.

```
aperellop@albert-ubuntu: $ sudo apt install docker docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done

No apt package "docker", but there is a snap with that name.
Try "snap install docker"

E: Unable to locate package docker
E: Unable to locate package docker.io
E: Couldn't find any package by glob 'docker.io'
E: Couldn't find any package by regex 'docker.io'
aperellop@albert-ubuntu: $ snap install docker
error: access denied (try with sudo)
aperellop@albert-ubuntu: $ sudo !!
sudo snap install docker
Download snap "core" (10185) from channel "stable"                    100% 11.9MB/s 0.0ns
```

We change the permissions of the file in the docker.

```
aperellop@albert-ubuntu:~$ sudo chmod 666 /var/run/docker.sock
```

We download and start the MySQL content.

```
aperellop@albert-ubuntu:~$ docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
bb79b6b2107f: Extracting [==================================================>]  27.09MB/27.09MB
49e22f6fb9f7: Download complete
842b1255668c: Download complete
9f48d1f43000: Download complete
c693f0615bce: Download complete
8a621b9dbed2: Download complete
0807d32aef13: Download complete
a56aca0feb17: Download complete
de9d45fd0f07: Downloading [=================>                                 ]  41.85MB/112.8MB
1d68a49161cc: Download complete
d16d318b774e: Download complete
49e112c55976: Downloading [==================================================>]     121B/121B
```

```
aperellop@albert-ubuntu:~$ sudo docker run --name database -e MYSQL_ROOT_PASSWORD=database -d mysql:latest
3716d66950ddd9f34ea9ea49eb4b033f30333d4baf89906ca89306dc94108eae
```

We do the verification.

```
aperellop@albert-ubuntu:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
          NAMES
3716d66950dd        mysql:latest        "docker-entrypoint.s…"   51 seconds ago      Up 50 seconds       3306/tcp, 33
060/tcp   database
```

We access the MySQL console.

```
aperellop@albert-ubuntu:~$ docker exec -it database mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

We create and access the database.

```
mysql> CREATE DATABASE university;
Query OK, 1 row affected (0.05 sec)

mysql> USE university;
Database changed
```

And finally we check and inquire about the table.

```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_university |
+----------------------+
| advisor              |
| classroom            |
| course               |
| department           |
| instructor           |
| prereq               |
| section              |
| student              |
| takes                |
| teaches              |
| time_slot            |
+----------------------+
11 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM student;
+-------+----------+------------+----------+rs' doesn't exist
| ID    | name     | dept_name  | tot_cred |
+-------+----------+------------+----------+
| 00128 | Zhang    | Comp. Sci. |      102 |
| 12345 | Shankar  | Comp. Sci. |       32 |
| 19991 | Brandt   | History    |       80 |
| 23121 | Chavez   | Finance    |      110 |
| 44553 | Peltier  | Physics    |       56 |
| 45678 | Levy     | Physics    |       46 |
| 54321 | Williams | Comp. Sci. |       54 |
| 55739 | Sanchez  | Music      |       38 |
| 70557 | Snow     | Physics    |        0 |
| 76543 | Brown    | Comp. Sci. |       58 |
| 76653 | Aoi      | Elec. Eng. |       60 |
| 98765 | Bourikas | Elec. Eng. |       98 |
| 98988 | Tanaka   | Biology    |      120 |
+-------+----------+------------+----------+
13 rows in set (0.00 sec)
```

4. Write your feedback about the difficulties you encountered and the things you have learned.

In this second practice I have encountered the difficulty that I had never worked with virtual machines before, and that is why I did not know how it works, but with a little help I have been able to carry it out, and I have learned what virtual machines are and how they work, more specifically how to create a virtual machine from scratch using Microsoft Azure, and how to use a MySQL docker to create a table and navigate through it with Google Cloud Shell and Windows PowerShell.