Lab 2 Contenedor Docker de MySQL y PostgreSQL

Objetivo

- Mostrar al participante el procedimiento para la creación de contenedores Docker para MySQL y PostgreSQL

Requisito

- Tener instalado Docker en Linux o Windows Pro

Procedimiento

1. Descarga la imagen de Docker Hub. Ejecuta el comando.

```
[admin@host1 ~]$ docker pull mysql:8
8: Pulling from library/mysql
0bb5c0c24818: Pull complete
cbb3106fbb5a: Pull complete
550536ae1d5e: Pull complete
33f98928796e: Pull complete
a341087cff11: Pull complete
0e26ac5b33f6: Pull complete
c883b83a7112: Pull complete
873af5c876c6: Pull complete
8fe8ebd061d5: Pull complete
7ac2553cf6b4: Pull complete
ad655e218e12: Pull complete
Digest: sha256:96439dd0d8d085cd90c8001be2c9dde07b8a68b472bd20efcbe3df78cff66492
Status: Downloaded newer image for mysql:8
docker.io/library/mysql:8
[admin@host1 ~]$
```

2. Muestra el listado de las imágenes y verifica que la imagen de mysql se haya descargado.

```
[admin@host1 ~]$ docker images
REPOSITORY                TAG      IMAGE ID       CREATED       SIZE
ccarrenovi/artemis        26       29506f286853   10 days ago   565MB
mysql                     8        3842e9cdffd2   12 days ago   538MB
apache/camel-k            1.10.3   84f891ee74ba   2 weeks ago   1.11GB
gcr.io/k8s-minikube/kicbase   v0.0.36   866c1fe4e3f2   4 weeks ago   1.11GB
openjdk                   12       e1e07dfba89c   3 years ago   470MB
[admin@host1 ~]$
```

3. Crea el contenedor de MySQL exponiendo el puerto interno **3306**. Asigna la clave "**password**" al usuario **root**, define la base de datos "**db_sales**" para ello utilizaremos variables de entorno.

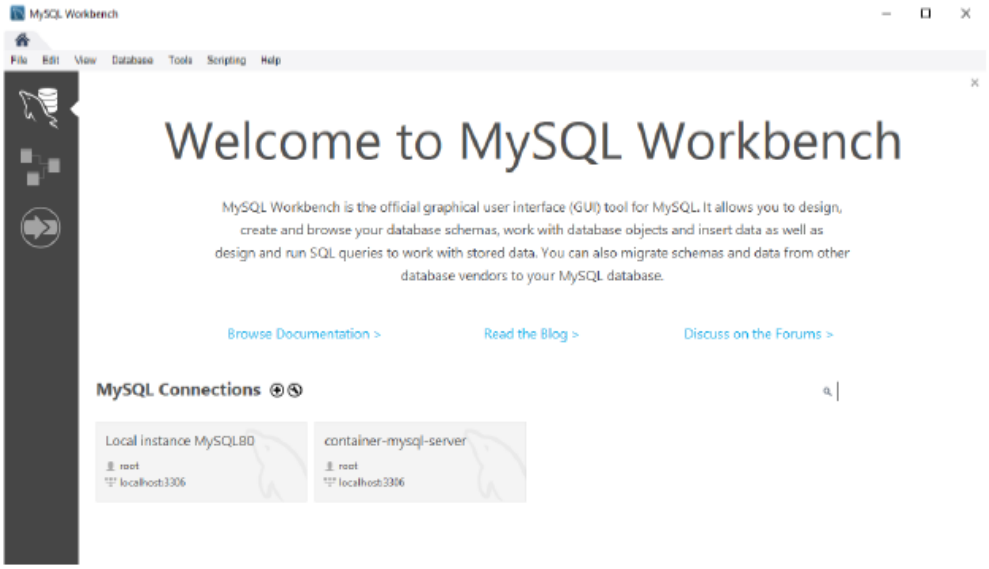
   [admin@host1 ~]$ **docker run -p 3306:3306 --name mysql-server -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=db_sales --network bridge -d mysql:8**
   f6599dc825f547bf4340f840f48c46c0e1bce74ef99161f47d611eec58a9d61a
   [
   [admin@host1 ~]$ **docker ps**
   CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
   f6599dc825f5 mysql:8 "docker-entrypoint.s…" 10 seconds ago Up 8 seconds 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp mysql-server
   3e253e97f87b gcr.io/k8s-minikube/kicbase:v0.0.36 "/usr/local/bin/entr…" About an hour ago Up About an hour 127.0.0.1:49162->22/tcp, 127.0.0.1:49161->2376/tcp, 127.0.0.1:49160->5000/tcp, 127.0.0.1:49159->8443/tcp, 127.0.0.1:49158->32443/tcp minikube-m02
   af66a765bbfe gcr.io/k8s-minikube/kicbase:v0.0.36 "/usr/local/bin/entr…" About an hour ago Up About an hour 127.0.0.1:49157->22/tcp, 127.0.0.1:49156->2376/tcp, 127.0.0.1:49155->5000/tcp, 127.0.0.1:49154->8443/tcp, 127.0.0.1:49153->32443/tcp minikube
   [admin@host1 ~]$

   Sugerencia: Revisa la documentación de DockerHub https://hub.docker.com/_/mysql sobre la imagen Docker de MySQL.

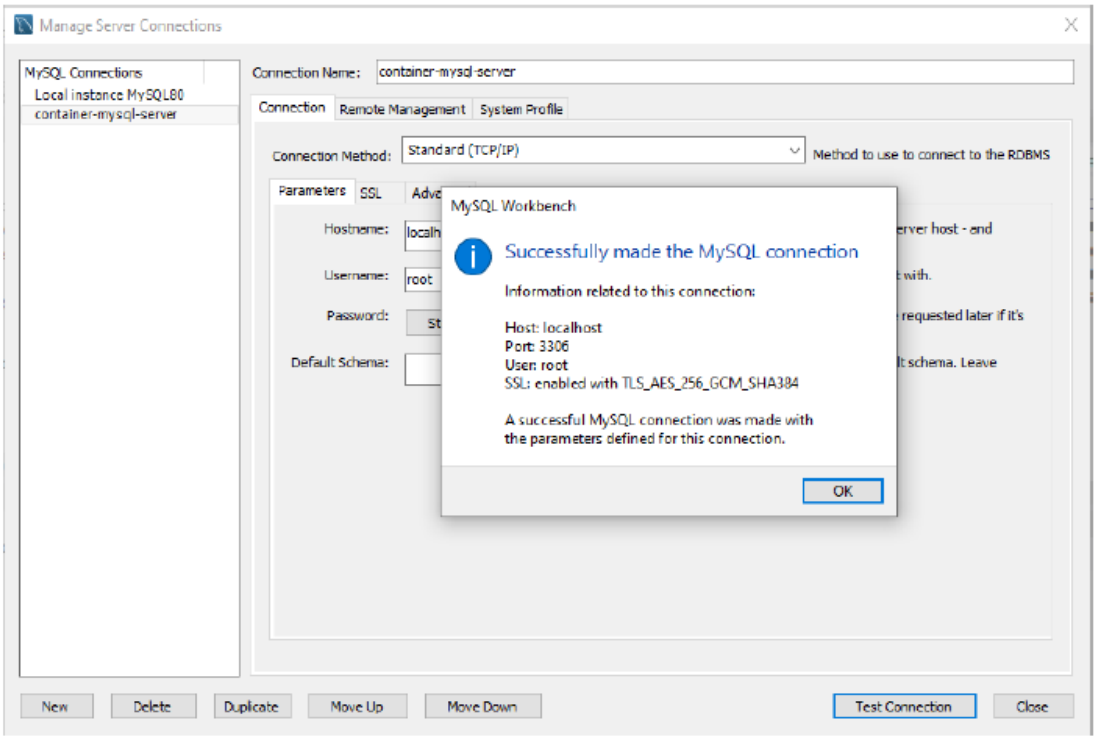
4. Muestra el log del contenedor mysql-server

   [admin@host1 ~]$ **docker logs -f mysql-server**
   2022-11-28 16:18:25+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
   2022-11-28 16:18:25+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
   2022-11-28 16:18:26+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
   2022-11-28 16:18:26+00:00 [Note] [Entrypoint]: Initializing database files
   2022-11-28T16:18:26.574714Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.

5. Instala MySQL Worbench en Windows.



6. Usando MySQL Workbench conéctate al servidor mysql desplegado en el contenedor.



7. Descarga de Docker hub la imagen de PostgreSQL.

[admin@host1 ~]$ **docker pull postgres:12-alpine**
12-alpine: Pulling from library/postgres
ca7dd9ec2225: Pull complete
126ca7716c09: Pull complete
72a45366ffb3: Pull complete
d8518ca0d726: Pull complete

4cd2fc6e1219: Pull complete
45d26ade401f: Pull complete
e8ee9d381ccf: Pull complete
a4df2282fbc2: Pull complete
Digest: sha256:cb4143d08a2d194819a5970c3ecda7825561cf3d91219e93f398a39bb7254dfc
Status: Downloaded newer image for postgres:12-alpine
docker.io/library/postgres:12-alpine

8. Crea el contenedor **postgresql-server** usando la imagen **postgresql:12-alpine**

[admin@host1 ~]$ **docker run -p 5432:5432 --name postgresql-server --network bridge -e POSTGRES_PASSWORD=password -e POSTGRES_DB=db_customer -d postgres:12-alpine**
b4bbf6306b398af76b847eeb45f852c3172f338849983ac28f33a0ddb5003620
[admin@host1 ~]$

9. Verifica que se haya desplegado el servidor postgresql en un contenedor docker.

[admin@host1 ~]$ **docker ps**
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b4bbf6306b39 postgres:12-alpine "docker-entrypoint.s…" 3 minutes ago Up 3 minutes 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp postgresql-server
f6599dc825f5 mysql:8 "docker-entrypoint.s…" 13 minutes ago Up 13 minutes 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp mysql-server
3e253e97f87b gcr.io/k8s-minikube/kicbase:v0.0.36 "/usr/local/bin/entr…" About an hour ago Up About an hour 127.0.0.1:49162->22/tcp, 127.0.0.1:49161->2376/tcp, 127.0.0.1:49160->5000/tcp, 127.0.0.1:49159->8443/tcp, 127.0.0.1:49158->32443/tcp minikube-m02
af66a765bbfe gcr.io/k8s-minikube/kicbase:v0.0.36 "/usr/local/bin/entr…" About an hour ago Up About an hour 127.0.0.1:49157->22/tcp, 127.0.0.1:49156->2376/tcp, 127.0.0.1:49155->5000/tcp, 127.0.0.1:49154->8443/tcp, 127.0.0.1:49153->32443/tcp minikube
[admin@host1 ~]$ **docker exec -it postgresql-server /bin/sh**
/ # **su postgres**
/ $ **psql**
psql (12.13)
Type "help" for help.

postgres=# **\l**
List of databases
Name	Owner	Encoding	Collate	Ctype	Access privileges
db_customer	postgres	UTF8	en_US.utf8	en_US.utf8
postgres	postgres	UTF8	en_US.utf8	en_US.utf8
template0	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres +
				postgres=CTc/postgres
template1	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres +
				postgres=CTc/postgres

(4 rows)

postgres=# **\q**
/ $ **exit**
/ # **exit**
[admin@host1 ~]$