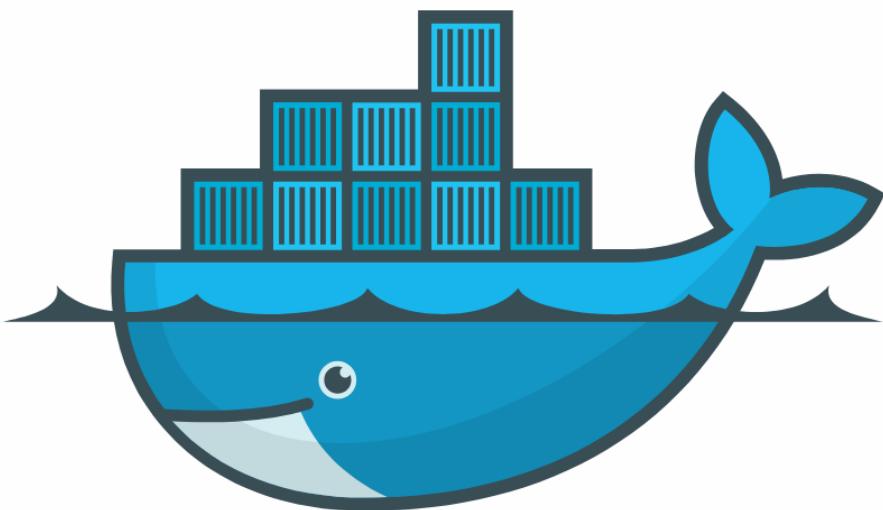


CONTENEDORES: CREACIÓN CON MYSQL



Ivana Sánchez Pérez

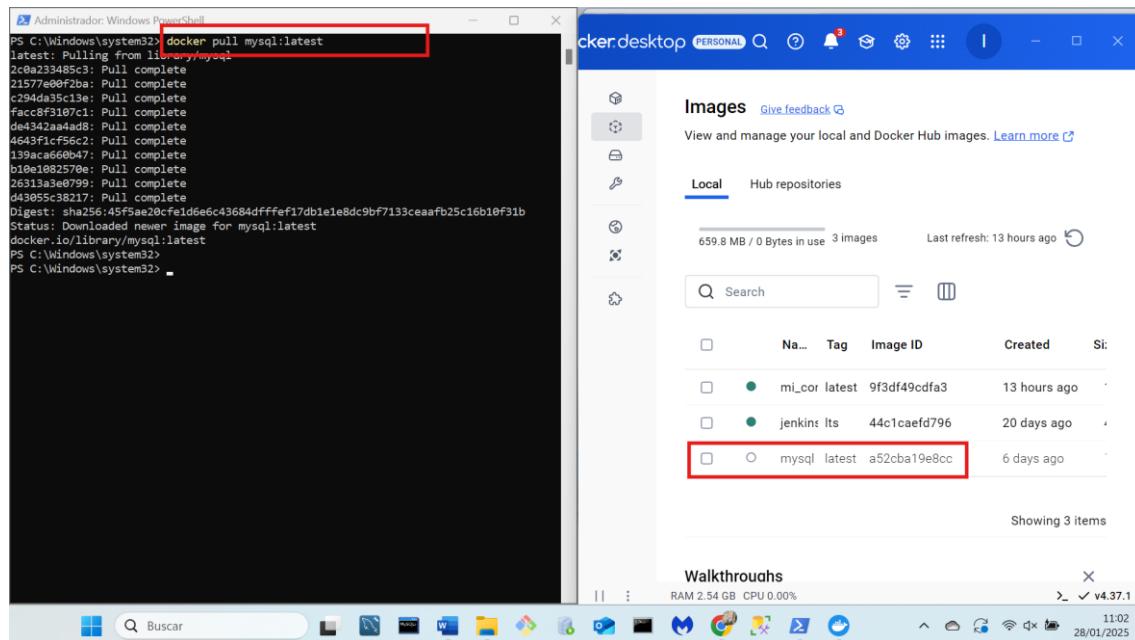
Investigar la imagen oficial de MySQL

La imagen oficial de MySQL está disponible en Docker Hub y por defecto usa el puerto 3306 para conexiones.

Esta imagen es compatible con MySQL 8.0, 5.7 y 5.6, y permite personalizar la configuración a través de variables de entorno.

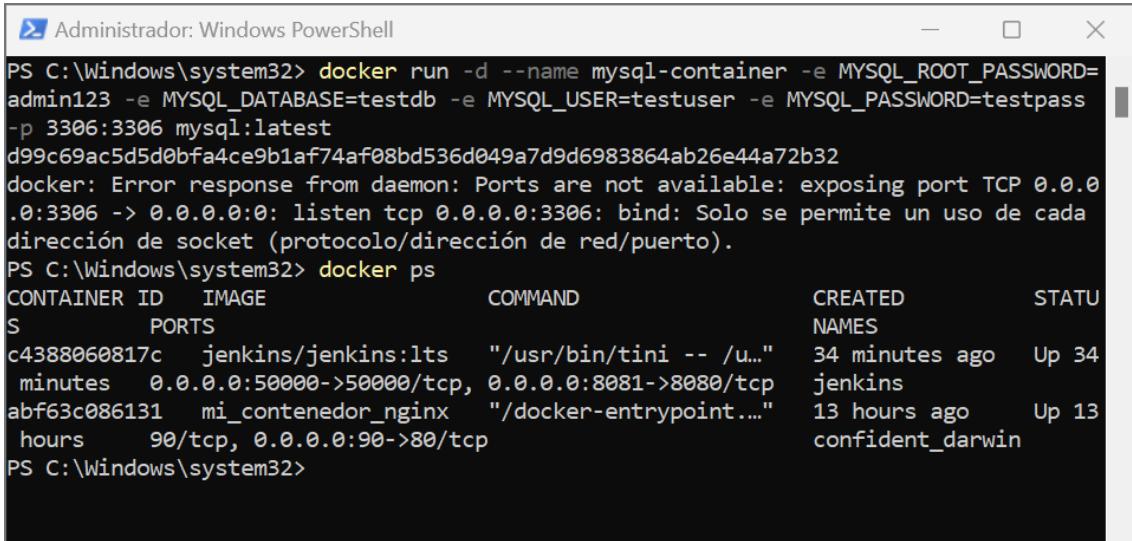
Descargar la imagen de MySQL

Ejecutamos el comando en el terminal de la PowerShell `docker pull mysql:latest` para descargar la imagen de MySQL



Iniciar un contenedor con MySQL

Creamos e iniciamos un contenedor configurando las credenciales básicas de acceso con `docker run -d --name mysql-container -e MYSQL_ROOT_PASSWORD=admin123 -e MYSQL_DATABASE=testdb -e MYSQL_USER=testuser -e MYSQL_PASSWORD=testpass -p 3306:3306 mysql:latest`



```
PS C:\Windows\system32> docker run -d --name mysql-container -e MYSQL_ROOT_PASSWORD=admin123 -e MYSQL_DATABASE=testdb -e MYSQL_USER=testuser -e MYSQL_PASSWORD=testpass -p 3306:3306 mysql:latest
d99c69ac5d5d0bfa4ce9b1af74af08bd536d049a7d9d6983864ab26e44a72b32
docker: Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:3306 -> 0.0.0.0:0: listen tcp 0.0.0.0:3306: bind: Solo se permite un uso de cada dirección de socket (protocolo/dirección de red/puerto).
PS C:\Windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
S PORTS NAMES
c4388060817c jenkins/jenkins:lts "/usr/bin/tini -- /u..." 34 minutes ago Up 34 jenkins
abf63c086131 mi_contenedor_nginx "/docker-entrypoint...." 13 hours ago Up 13 confident_darwin
PS C:\Windows\system32>
```

Explicación de los parámetros:

- **-d:** Inicia el contenedor en segundo plano.
- **--name mysql-container:** Asigna un nombre al contenedor.
- **-e MYSQL_ROOT_PASSWORD=admin123:** Establece la contraseña del usuario root.
- **-e MYSQL_DATABASE=testdb:** Crea una base de datos de llamada testdb.
- **-e MYSQL_USER=testuser:** Crea un usuario llamado testuser.
- **-e MYSQL_PASSWORD=testpass:** Establece la contraseña para testuser.
- **-p 3306:3306:** Asociación el puerto local 3306 al puerto 3306 del contenedor.

Vemos que nos da un error, y esto es debido a que el puerto 3306 está siendo utilizado por otra instancia de MySQL, por lo que procedemos a detener el proceso.

Para ello, primero comprobaremos si el puerto está en uso con el comando **netstat -aon | findstr :3306**, con el número PID que nos da hacemos otra búsqueda con el comando **tasklist | findstr num.PID** para identificar el proceso y de este modo detenerlo con el comando **taskkill /PID num /F**

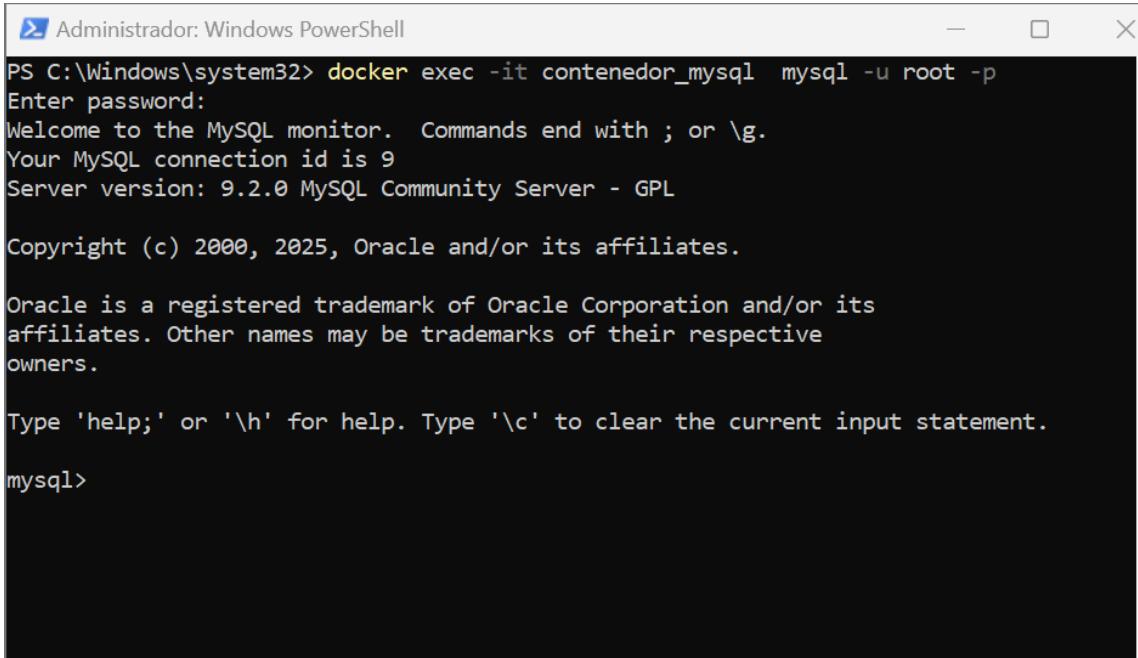
```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker run -d --name mysql-container -e MYSQL_ROOT_PASSWORD=admin123 -e MYSQL_DATABASE=testdb -e MYSQL_USER=testuser -e MYSQL_PASSWORD=testpass -p 3306:3306 mysql:latest
docker: Error response from daemon: Conflict. The container name "/mysql-container" is already in use by container "4de1b09ae8349c6b1be9b8384434be920cb963c34e9b006faecf e49d36e446ff". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
PS C:\Windows\system32> netstat -an | findstr :3306
  TCP    0.0.0.0:3306          0.0.0.0:0              LISTENING      5924
  TCP    [::]:3306           [::]:0              LISTENING      5924
PS C:\Windows\system32> tasklist | findstr 5924
mysqld.exe            5924 Services             0       6.584 KB
PS C:\Windows\system32> taskkill /PID 5924 /F
Correcto: se terminó el proceso con PID 5924.
```

Ahora, vuelvo a instalarlo y a darle un nombre distinto al contendor para que no haya más conflictos.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker pull mysql:latest
latest: Pulling from library/mysql
2c0a233485c3: Pull complete
21577e00f2ba: Pull complete
c294da35c13e: Pull complete
facc8f3107c1: Pull complete
de4342aa4ad8: Pull complete
4643f1cf56c2: Pull complete
139aca660b47: Pull complete
b10e1082570e: Pull complete
26313a3e0799: Pull complete
d43055c38217: Pull complete
Digest: sha256:45f5ae20cfe1d6e6c43684dffef17db1e1e8dc9bf7133ceaafb25c16b10f31b
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
PS C:\Windows\system32> docker run -d --name contenedor_mysql -e MYSQL_ROOT_PASSWORD=admin123 -e MYSQL_DATABASE=testdb -e MYSQL_USER=testuser -e MYSQL_PASSWORD=testpass -p 3306:3306 mysql:latest
3b4029cee0013dcda1d1b31712d111060b87f5a36cb398274d7db1891d57b524
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS
PORTS          NAMES
3b4029cee001   mysql:latest   "docker-entrypoint.s..."   10 seconds ago   Up 9 seconds
  0.0.0.0:3306->3306/tcp, 33060/tcp   contenedor_mysql
PS C:\Windows\system32>
```

Acceder al contenedor de MySQL

Para conectarnos al contenedor utilizamos el comando **docker exec -it contenedor_mysql mysql -u root -p**, que al ejecutarlo nos pedirá la contraseña root que configuramos anteriormente (admin123), y ya estamos dentro de MySQL.



```
PS C:\Windows\system32> docker exec -it contenedor_mysql mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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>
```

Trabajar con la base de datos

- a. Nos conectamos a la base de datos creada con el comando **USE testdb;**
- b. Creamos una tabla:

```
CREATE TABLE usuarios (
    Id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    email VARCHAR(100)
);
```

```
mysql>
mysql> USE testdb;
Database changed
mysql> CREATE TABLE usuarios (
    ->     Id INT AUTO_INCREMENT PRIMARY KEY,
    ->     nombre VARCHAR(100),
    ->     email VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql>
```

c. Insertamos datos

```
INSERT INTO usuarios (nombre, email) VALUES
('Ivana', 'ivana@ejemplo.com'),
('Raquel', 'raquel@ejemplo.com');
```

```
mysql>
mysql>
mysql> INSERT INTO usuarios (nombre, email) VALUES
-> ('Ivana', 'ivana@ejemplo.com'),
-> ('Raquel', 'raquel@ejemplo.com');
Query OK, 2 rows affected (0.03 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
```

d. Consultamos datos

```
SELECT * FROM usuarios;
```

```
mysql> SELECT * FROM usuarios;
+----+-----+-----+
| Id | nombre | email          |
+----+-----+-----+
| 1  | Ivana  | ivana@ejemplo.com |
| 2  | Raquel | raquel@ejemplo.com|
+----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

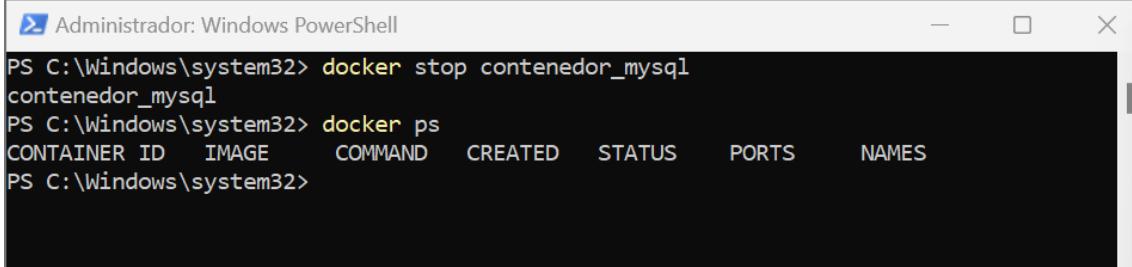
Salida del cliente MySQL

Exit;

```
mysql> exit  
Bye  
PS C:\Windows\system32>
```

Detener el contenedor de MySQL

Docker stop contenedor_mysql



A screenshot of a Windows PowerShell window titled "Administrador: Windows PowerShell". The window shows the command "docker stop contenedor_mysql" being run, followed by "docker ps" which displays a table of containers. The table has columns: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, PORTS, and NAMES. There is one entry: PS C:\Windows\system32>.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
						PS C:\Windows\system32>

Eliminar completamente el contenedor

Docker rm contenedor_mysql

