

Creando un servidor de MySQL local con Docker

Juan David Bedoya Tobon
Mateo Rodriguez Toro
Andres Posada Echavarria



Docker es un software de manejo de contenedores que permite exportar tu código junto con la configuración del entorno en donde se encuentra de manera sencilla y liviana sin preocuparse por los cambios que puedan haber en el sistema operativo, mientras que MySQL es la base de datos relacional Open Source más popular.

Origen : <https://hub.docker.com/r/mysql/mysql-server/>

objetivo

El objetivo de esta presentación es explicar el paso a paso de la creación de un contenedor de **MySQL** de manera local que se pueda acceder desde otras aplicaciones instaladas en la misma máquina.

Descargando la Imagen desde Docker

El tener la imagen del servidor de MySQL en nuestro computador nos permitirá crear todos los contenedores que queramos de MySQL en nuestra máquina. Podríamos llegar a tener uno por cada proyecto que estemos desarrollando para no mezclar las bases de datos, para tener un orden lógico, o simplemente para que cada una de ellas no sea muy pesada.

Para obtener la imagen simplemente hacemos **pull** desde el repositorio oficial.

```
$ sudo docker pull mysql/mysql-server:latest
```

En este caso estaré trabajando con la última versión de MySQL, sin embargo se puede **especificar la versión** que desees utilizar en vez de **latest**, por ejemplo **5.7**, **5.7.24**, **5.7..**

Ejecutando el contenedor a partir de la imagen

Ya teniendo la imagen estamos listos para ejecutarla. En este caso como queremos que además se pueda ejecutar desde fuera de los contenedores, y posiblemente desde Internet, debemos hacer un forward del puerto para poder acceder.

```
sudo docker run --name=MyMySQLServer -d -p 6603:3306  
mysql/mysql-server:latest
```

En este caso llamamos al servidor **MyMySQLServer** en modo daemon (**-d** hace que se ejecute en el background), y este es el nombre que usaremos de ahora en adelante. En este caso además de ejecutarlo le instruimos a docker que haga un forward del puerto **6603** al **3306** de dentro de docker, por lo que si queremos acceder desde fuera del contenedor lo haremos en el puerto **6603** de la máquina **host**.

Si bien el Servidor MySQL ya está corriendo, no sabemos cual es la clave de ROOT de MySQL, como este contenedor la genera de manera automática, simplemente debemos revisar los logs para ver cual es la clave de administrador.

```
sudo docker logs MyMySQLServer | grep GENERATED
```

Con **docker logs** obtenemos los logs de inicio del contenedor con MySQL, mientras que con **grep** filtramos el contenido para poder ver cual fue la clave de root generada al inicio. En este caso :

```
[Entrypoint] GENERATED ROOT PASSWORD: IMWop4h#0J0Bk@hLAk@n1uKuSUs
```

Ahora solo nos falta modificar la clave y permitir conexiones desde fuera del contenedor.

Modificando el servidor MySQL.

Si bien no podemos acceder aún al servidor MySQL desde afuera, si podemos hacerlo desde el mismo contenedor.

```
sudo docker exec -it MyMySQLServer mysql -uroot -p
```

Al ejecutar el comando nos preguntará la clave de root que es la que obtuvimos en el punto anterior. Simplemente copia y pega la clave obtenida anteriormente.

Ahora estamos dentro del servidor MySQL del contenedor.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 119

Server version: 8.0.13 MySQL Community Server - GPL

Copyright (c) 2000, 2018, 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>

Antes que todo, cambiaremos la clave por una más fácil de recordar.

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'password';
```

Recomiendo no usar la clave password :P.

Ahora especificaremos al servidor que el usuario root puede acceder desde cualquier lado. Insisto, en mi caso es un servidor de desarrollo local, no recomiendo hacer esto en producción.

```
mysql> UPDATE mysql.user set host="%" where user="root";
```

En este momento tenemos todo para poder acceder desde fuera del contenedor a nuestra base de datos MySQL, lo único que debemos hacer es reiniciar el servicio. Siéntete libre de cerrar la ventana o simplemente escribir **exit**.

Reiniciando el contenedor.

Si bien podríamos acceder con bash directamente a la máquina, reiniciar el servicio utilizando docker es mucho más sencillo.

```
sudo docker restart MyMySQLServer
```

Ahora, para probar si todo está bien, puedes usar tu cliente mysql preferido para poder acceder al servidor. En este caso usaremos la línea de comandos para hacerlo más sencillo.

```
mysql -u root -ppassword -h 127.0.0.1 -P 6603
```

Si realizamos todo de manera correcta podemos acceder al servidor MySQL de manera local en el puerto 6603 usando el usuario **root** y password **password**.



Conclusión

Tener uno o más servidores MySQL de manera local es bastante sencillo, y lo mejor es que no consumen recursos mientras no se estén ejecutando. Esto es perfecto para los que tenemos un laptop para desarrollar y queremos que la batería dure lo más posible, además de tener un orden lógico y poder diferenciar las bases de datos por proyectos.