

Contenedores con Docker. Descarga de imágenes de aplicaciones

- Se trata de descargar la imagen del servicio mysql preconfigurado desde el Docker Hub (en lugar de descargar una imagen de Ubuntu y dentro de ella descargar mysql) para ello:

a) Descarga la imagen con **docker pull mysql**

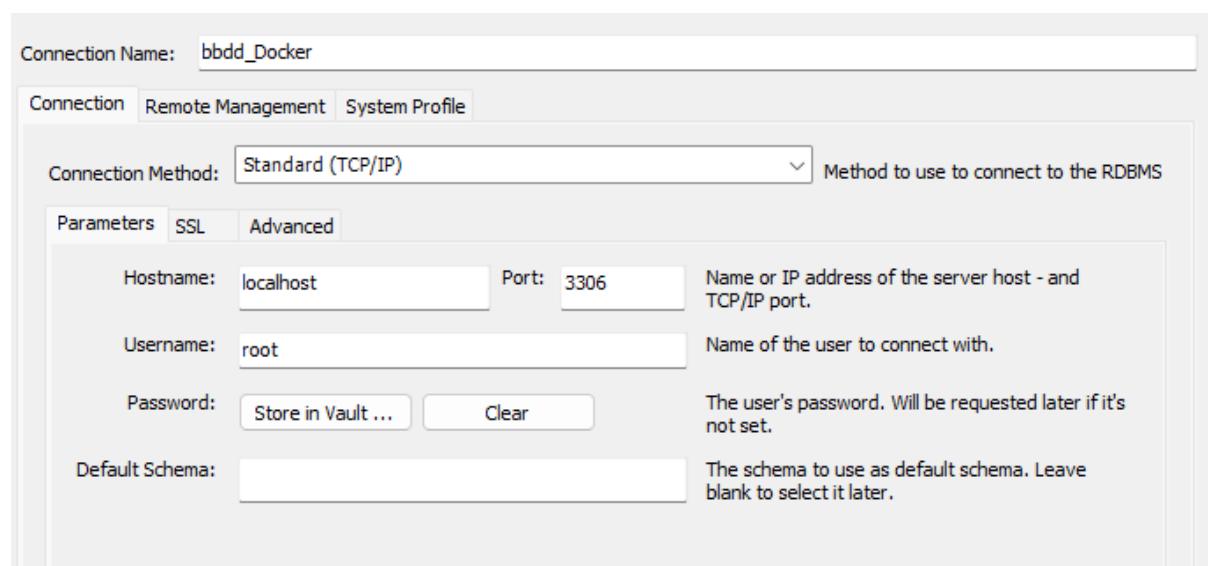
b) Ejecuta el comando:

```
docker run --name mysql -e MYSQL_ROOT_PASSWORD=<password> -d -p  
3306:3306 mysql
```

En el comando anterior la opción -e es de “entorno” y sirve para establecer variables para el contenedor y el -d sirve para ejecutar el contenedor y dejarlo en corriendo en background ya que se trata de un servicio.

```
PS C:\Users\fabri> docker pull mysql  
Using default tag: latest  
latest: Pulling from library/mysql  
f6a972aa365b: Download complete  
ac5a3aa7003a: Download complete  
57bd38a2d740: Download complete  
ba8d858f8b56: Download complete  
e4fd376e797b1: Download complete  
98d18218e1bb: Download complete  
55f85a7d691e: Download complete  
b843491434c1: Download complete  
aff72f8f4e98: Download complete  
7a5e1e917526: Download complete  
Digest: sha256:fe036967257bf11aab7184e371920c5b629f0dd36cbefdf4cccd2ae18cc900dbd  
Status: Downloaded newer image for mysql:latest  
docker.io/library/mysql:latest  
PS C:\Users\fabri> docker run --name mysql -e MYSQL_ROOT_PASSWORD=<password> -d -p 3306:3306 mysql  
47e9cf8286390a8f1415b6e01874b23c8b6f42ea072a67c2eb52707ee7b0f7a6  
PS C:\Users\fabri> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
47e9cf828639 mysql "docker-entrypoint.s..." 2 minutes ago Up 2 minutes 0.0.0.0:3306->3306/tcp, 33060/tcp mys  
ql  
PS C:\Users\fabri> |
```

c) Ejecuta el comando docker ps para comprobar que el contenedor está en ejecución



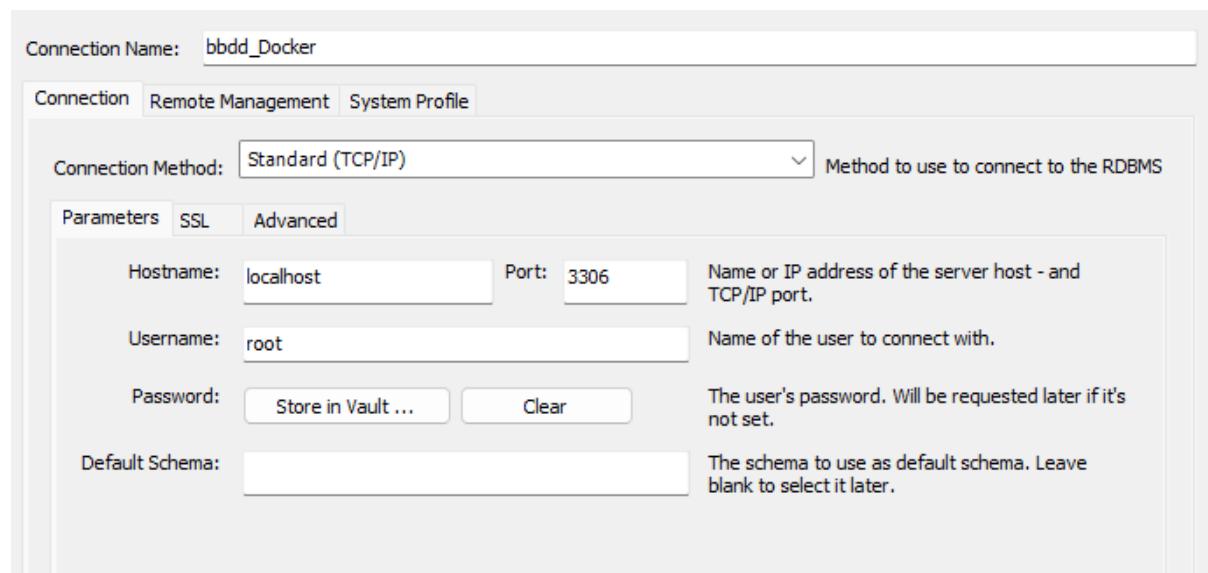
d) Vamos a probar el acceso al servidor MySQL y para ello en la maquina anfitrión instalamos el cliente de mysql con:

- Caso de maquina Debian: apt-get install mariadb-client
- Caso de maquina Windows: se debe instalar un cliente mysql como Mysql Workbench o bien phpmyadmin (accesible desde XAMPP)

e) Desde la maquina anfitrión accedemos al servicio MySQL del contenedor con:

- Caso de maquina Debian: mysql -h localhost -u root -p
- Caso de maquina Windows: acceso desde el cliente grafico Workbench (si se usa phpmyadmin debe ajustarse la contraseña del root de MySQL en el fichero config.inc.php)

Nos pedirá la contraseña de ROOT que establecimos en el apartado b) y ya estamos en la gestion de MySQL desde donde podemos teclear sentencias SQL como create database pruebas; use pruebas; create table.., etc.. y cuando terminemos salimos con quit a la consola del sistema anfitrión



Aqui hubo un fallo debido a que no se especificaba desde donde se comprendia la contraseña en la documentacion.

f) Se pide ahora crear un nuevo contenedor de mysql a partir de la imagen descargada anteriormente, pero esta vez debemos lanzarlo con un volumen gestionado por Docker que permita guardar los datos del directorio /var/lib/mysql.

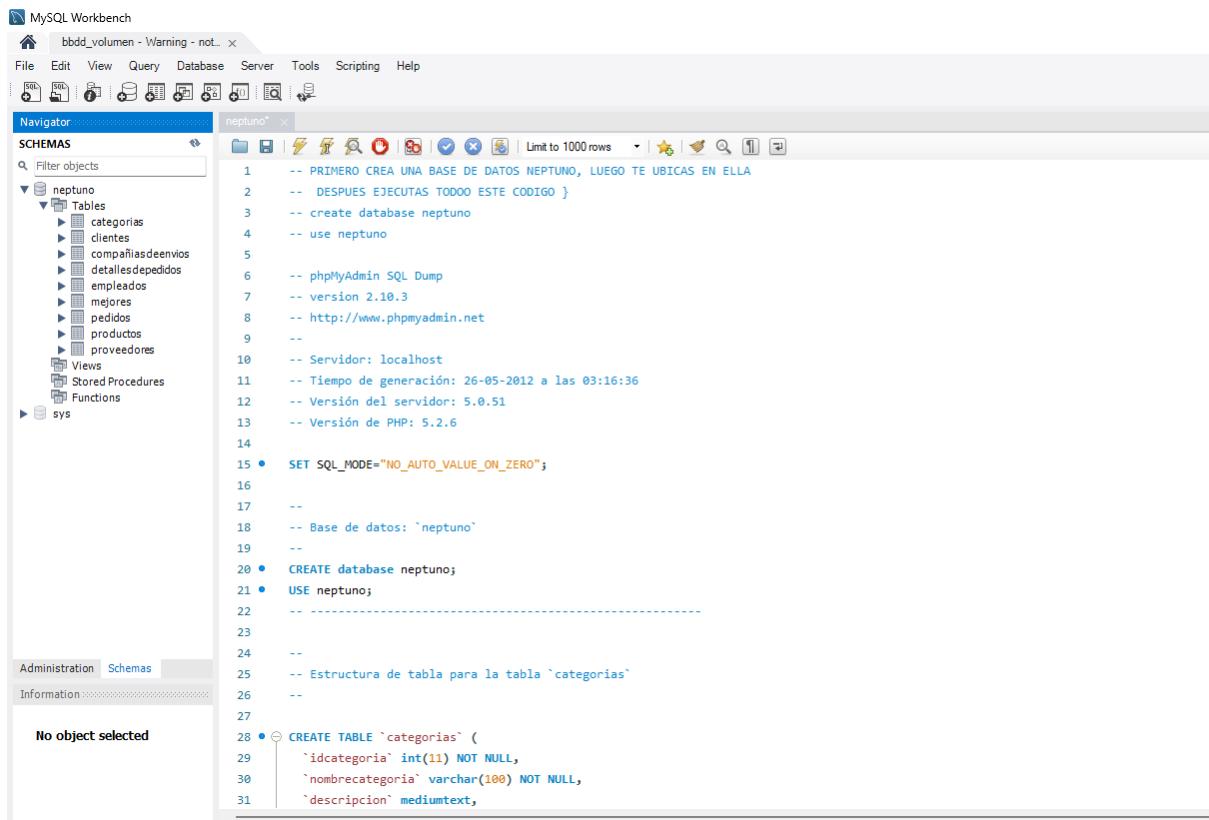
```
PS C:\Users\fabri> docker run --name mysqlvolu -v bbdd_act3:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=<password> -d -p 3308:3306 mysql
8368f5ea1368ea166fb297c961a30d7f1467f4f40b6f33c775bcb9696e998c7c
```

g) Una vez creado el contenedor se debe comprobar desde Docker Desktop que el volumen esta creado y en uso. Visualizar los datos inicialmente creados en el volumen

□	○	mysqlvolu	c43b9e9efb1d	mysql	3308:3306	0%	2 minutes ago
□	●	3ac8d013d96ef6fc8d519aeb587b1f64387144aae9c7c9a1020ff996a8c3ed8f			43 minutes ago	195.4 MB	⋮ ⏺ ⏻
□	●	bbdd_act3			6 minutes ago	0 Bytes	⋮ ⏺ ⏻
□	●	c13eb67777145c7f525ade1d30ad14f766b2ed34b20d8f3e4fd717dbae6d39a3			8 months ago	212.7 MB	⋮ ⏺ ⏻

A

h) Ahora, empleando la aplicación Mysql Workbench se debe ejecutar el script SQL Neptuno.sql que se entrega (este script crea la BBDD Neptuno, las tablas y varios datos de ejemplo)



```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
File Edit View Query Database Server Tools Scripting Help
Navigator neptuno*
SCHEMAS
Filter objects
neptuno
Tables
categorias
clientes
compañiasdeenvios
detallesdepedidos
empleados
mejores
pedidos
productos
proveedores
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
No object selected
1 -- PRIMERO CREA UNA BASE DE DATOS NEPTUNO, LUEGO TE UBICAS EN ELLA
2 -- DESPUES EJECUTAS TODO ESTE CODIGO
3 -- create database neptuno
4 -- use neptuno
5
6 -- phpMyAdmin SQL Dump
7 -- version 2.10.3
8 -- http://www.phpmyadmin.net
9 --
10 -- Servidor: localhost
11 -- Tiempo de generación: 26-05-2012 a las 03:16:36
12 -- Versión del servidor: 5.0.51
13 -- Versión de PHP: 5.2.6
14
15 • SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
16
17 --
18 -- Base de datos: `neptuno`
19 --
20 • CREATE database neptuno;
21 • USE neptuno;
22 -----
23
24 --
25 -- Estructura de tabla para la tabla `categorias`
26 --
27
28 • CREATE TABLE `categorias` (
29   `idcategoria` int(11) NOT NULL,
30   `nombrecategoría` varchar(100) NOT NULL,
31   `descripción` mediumtext,
```

BBDD Funcionando en el nuevo docker con volumen

i) Borramos ahora el contenedor y creamos un nuevo contenedor enlazándolo con el mismo volumen anterior

j) Nos conectamos con la aplicación y verificamos que los datos de la BBDD Neptuno siguen estando disponibles

