



DOCKER AVANZADO



4 DE DICIEMBRE DE 2017
EVA LÓPEZ PUENTE

DOCKER AVANZADO

Para la realización de la práctica, se ha elegido el sistema operativo Linux.

Se parten de los pasos realizados en la práctica anterior para tener el contenedor de la base de datos, y también, es necesario tener los permisos de administrador.

Para la parte de backend, es necesario tener un archivo .jar, para ello se modifica el *pom.xml* añadiendo la opción “<packaging>jar</packaging>”.

En el archivo *application.yml*, hay que cambiar el “ddl-auto: create” para poder crear los contenidos necesarios y en la url, ahora tenemos que conectarlo con la imagen de docker creada anteriormente, que será nuestra base de datos, “url: jdbc:mysql://ejersql/ejercicio”.

Finalmente, hay que crear un fichero Dockerfile, el cual nos permitirá construir la imagen de docker de nuestro código de backend.

```
FROM java:8
MAINTAINER Eva Lopez Puente
EXPOSE 8080
VOLUME /tmp
ADD /target/ServerSql-0.0.1-SNAPSHOT.jar practica.jar
ENTRYPOINT ["java", "-jar", "practica.jar"]
```

FROM → Indica la imagen base para nuestra aplicación

MAINTAINER → Indica quién es el creador de esa imagen

EXPOSE → Indica el puerto sobre el que correrá nuestra parte de backend

VOLUME → Indica el punto de montaje

ADD → Indica la ruta al fichero .jar desde el que parte la imagen, se encuentra en el directorio target y el nombre que lo identifica es el mismo nombre que aparece en <artifactId> del pom.xml. Para crear el .jar se ejecuta la aplicación como Maven build “clean install verify package”.

ENTRYPOINT → Indica el contenedor que será el ejecutable

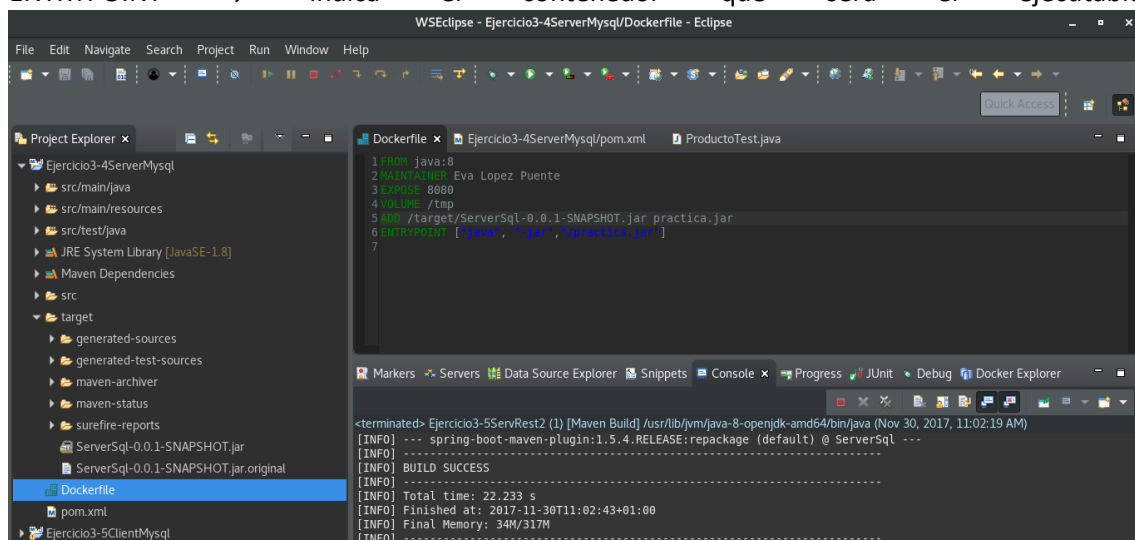


Ilustración 1

DOCKER AVANZADO

Para poder ejecutar este fichero, es necesario moverse desde la terminal, a la carpeta donde se encuentra.

Mediante “*docker build -t practica .*”, especificamos el repositorio y la etiqueta con la que guardamos nuestra imagen. Es necesario escribir un punto al final para indicar que el fichero Dockerfile se encuentra en paralelo a la carpeta donde se está ejecutando el comando. Con “*-rm*” podemos eliminar la caché que se crea.

```
root@sephit:~/Programas/WSEclipse/Estasiquesique/Ejercicio3-5ServRest2# docker build -t practica .
Sending build context to Docker daemon 29.56MB
Step 1/6 : FROM java:8
--> d23bdf5b1b1b
Step 2/6 : MAINTAINER Eva Lopez Puente
--> Using cache
--> 1bdc03372be3
Step 3/6 : EXPOSE 8080
--> Using cache
--> 4dd15ad98df4
Step 4/6 : VOLUME /tmp
--> Using cache
--> e24ccd1d030
Step 5/6 : ADD /target/ServerSql-0.0.1-SNAPSHOT.jar practica.jar
--> b88e6693eb0f
Removing intermediate container 44feb61056a9
Step 6/6 : ENTRYPOINT java -jar /practica.jar
--> Running in 1a6b4bd5789f
--> 6ad8b40ea4f6
Removing intermediate container 1a6b4bd5789f
Successfully built 6ad8b40ea4f6
Successfully tagged practica:latest
root@sephit:~/Programas/WSEclipse/Estasiquesique/Ejercicio3-5ServRest2#
```

Ilustración 2

A continuación, “*docker run --name practica -p 8080:8080 --link ejersql -d practica*”. Con esto realizamos la conexión de la imagen practica con la base de datos ejersql en el puerto 8080.

```
root@sephit:~/Programas/WSEclipse/pMySQL/Ejercicio3-5ServerMySQL# docker run --name practica -p 8080:8080 --link ejersql -d practica
32bba04dd856580c78c44f13dab59e0a1f35294cdf7cc3739149cf0877f8b628
docker: Error response from daemon: Cannot link to a non running container: /ejersql AS /practica/ejersql.
root@sephit:~/Programas/WSEclipse/pMySQL/Ejercicio3-5ServerMySQL# docker run --name practica -p 8080:8080 --link ejersql -d practica
7docker: Error response from daemon: Conflict. The container name "/practica" is already in use by container "32bba04dd856580c78c44f13dab59e0a1f35294cdf7cc3739149cf0877f8b628". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
root@sephit:~/Programas/WSEclipse/pMySQL/Ejercicio3-5ServerMySQL# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
32bba04dd856        practica            "java -jar /practi..." 56 seconds ago      Up 24 seconds      0.0.0.0:8080->8080/tcp   practica
dcaae965b0f        ejersql:latest      "docker-entrypoint..." 22 minutes ago      Up 30 seconds      0.0.0.0:3306->3306/tcp   ejersql
root@sephit:~/Programas/WSEclipse/pMySQL/Ejercicio3-5ServerMySQL# docker images
REPOSITORY          TAG               IMAGE ID            CREATED             SIZE
practica             latest           ce5a9ab9e021       About a minute ago  673MB
```

Ilustración 3

DOCKER AVANZADO

Como anteriormente, en la consola de MySQL se pueden ver las tablas que nos crea la aplicación:

Primero es necesario seleccionar la base de datos “*use ejercicio*”, a continuación, se muestran “*show tables;*”

Para ver el contenido de una tabla concreta: “*SELECT * FROM ejercicio.producto*”, “*SELECT * FROM ejercicio.usuario*”, “*SELECT * FROM ejercicio.usuario_rol*”

```
mysql> use ejercicio
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> use ejercicio;
Database changed
mysql> show tables;
+-----+
| Tables_in_ejercicio |
+-----+
| producto             |
+-----+
1 row in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_ejercicio |
+-----+
| producto            |
| usuario             |
| usuario_rol         |
+-----+
3 rows in set (0.03 sec)

mysql> select * from ejercicio.usuario;
+-----+-----+-----+-----+
| id | cont | nombre |
+-----+-----+-----+-----+
| 1 | $2a$10$d/wM0B6/UcwRuPf.vQzEu.JRytYwAI4naa5W3mpnlKAGhNFboUYCC | user |
| 2 | $2a$10$13f2rlGchi7I34wrmqo9MexiYgQL./AnaUdz/nFeSwuM0fupkxENq | root |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |
```

Ilustración 4

Para que todo funcione correctamente hay que modificar de nuevo el archivo *application.yml* creando la tabla de usuarios con “*ddl-auto: create*” y una vez ya la tengamos activa, paramos la ejecución y lo cambiamos a “*ddl-auto: validate*”, de esta manera, no crea el contenido de las tablas de usuario y usuario_rol cada vez que se ejecuta la aplicación. También hay que indicarle el tipo de la base de datos, la url “*jdbc:mysql://localhost/ejercicio*”, el nombre y la contraseña.

Si ahora se ejecuta la aplicación, se puede ver que todo funciona correctamente y se puede acceder a la url “*localhost:8080/principal/list*” y ver los productos de la aplicación.

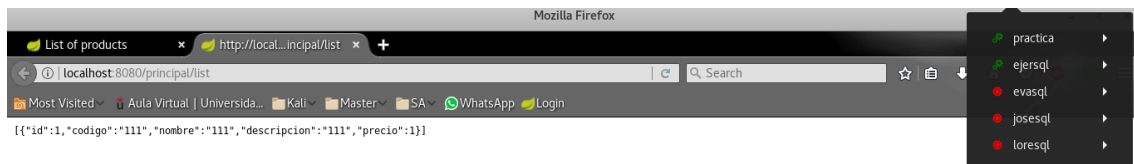


Ilustración 5

DOCKER AVANZADO

En la siguiente imagen, se pueden ver las imágenes de docker y los puertos en los que se ejecutan, así como los links entre un contenedor y otro.

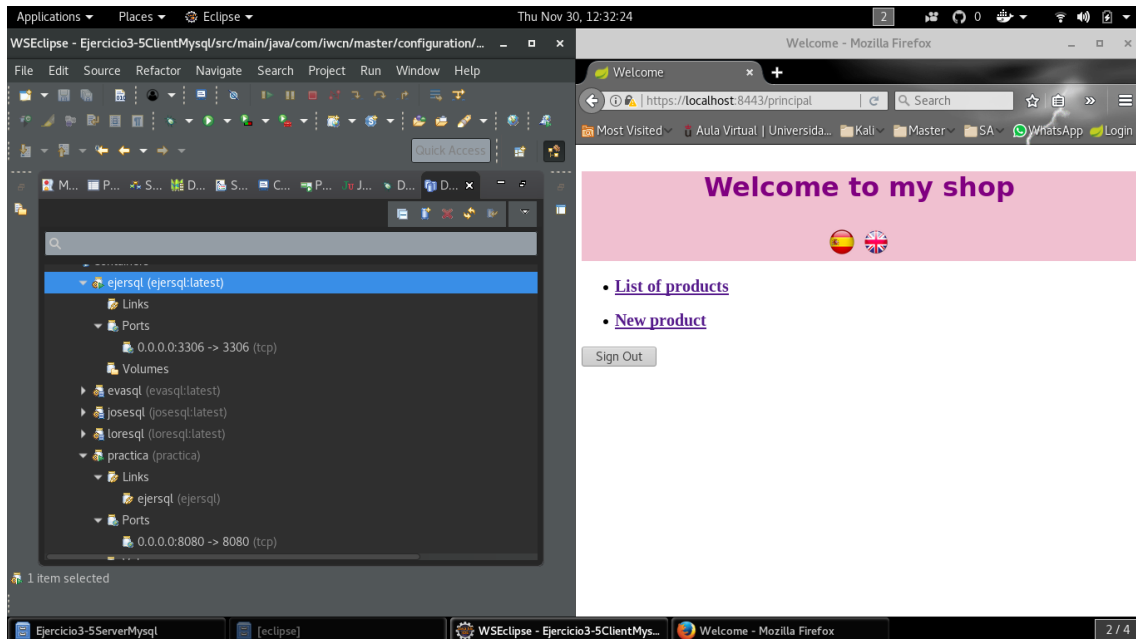


Ilustración 6

DOCKER AVANZADO

Para subir esta imagen a Docker Hub, hay que crearse una cuenta. En este caso, hemos creado una cuenta común entre Jose, Víctor, Rubén, Eva y Lorena.

Lo primero es iniciar sesión en docker mediante *"docker login"* e introducir el nombre y la contraseña de la cuenta.

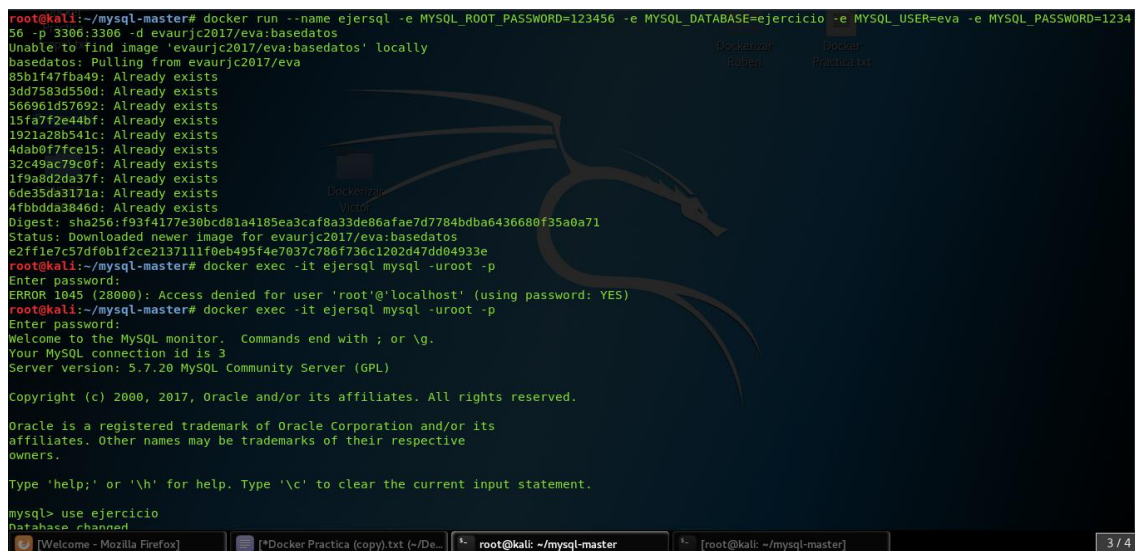
Para subir la imagen de la parte backend escribimos *"docker tag practica evaurjc2017/eva:practica"*, lo que nos crea un tag practica en el repositorio evaurjc2017/eva.

Para subir la imagen de la base de datos de MySQL escribimos *"docker tag ejersql evaurjc2017/eva:basedatos"* y nos crea un tag basedatos en el repositorio evaurjc2017/eva.

A continuación, subimos la imagen mediante *"docker push evaurjc2017/eva"* y para comprobar que se ha subido correctamente *"docker search evaurjc2017"* y nos aparece un listado con los repositorios que tenemos.

Finalmente, para descargar la imagen y poder probarlo realizamos *"docker pull evaurjc2017/eva:basedatos"*, y *"docker pull evaurjc2017/eva:practica"*. En el caso de no realizar el pull y continuar con los siguientes pasos, la imagen se descarga automáticamente desde Docker Hub.

Para comprobar que todo funciona correctamente, es necesario abrir una consola en la que arrancaremos la imagen de la base de datos: *"docker run --name ejersql -e MYSQL_ROOT_PASSWORD=123456 -e MYSQL_DATABASE=ejercicio -e MYSQL_USER=eva -e MYSQL_PASSWORD=123456 -p 3306:3306 -d evaurjc2017/eva:basedatos"*.

A terminal window with a dark background and a dragon logo in the center. The terminal shows the following commands and output:

```
root@kali:~/mysql-master# docker run --name ejersql -e MYSQL_ROOT_PASSWORD=123456 -e MYSQL_DATABASE=ejercicio -e MYSQL_USER=eva -e MYSQL_PASSWORD=123456 -p 3306:3306 -d evaurjc2017/eva:basedatos
Unable to find image 'evaurjc2017/eva:basedatos' locally
basedatos: Pulling from evaurjc2017/eva
85b1f47fba49: Already exists
3d07583d550d: Already exists
566961d57692: Already exists
15fa7f2e44bf: Already exists
1921a28b541c: Already exists
4dab0f7fce15: Already exists
32c49ac79c0f: Already exists
1f9a8d2da37f: Already exists
6de35da3171a: Already exists
4fbbdda3846d: Already exists
Digest: sha256:f93f4177e30bcd81a4185ea3caf8a33de86afae7d7784bdba6436680f35a0a71
Status: Downloaded newer image for evaurjc2017/eva:basedatos
e2ff1e7c57df0b1f2ce2137111f0eb495f4e7037c786f736c1202d47d084933e
root@kali:~/mysql-master# docker exec -it ejersql mysql -uroot -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
root@kali:~/mysql-master# docker exec -it ejersql mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.20 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, 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> use ejercicio
Database changed
```

Ilustración 7

DOCKER AVANZADO

A continuación, ejecutamos la consola de MySql mediante “*docker exec -it ejersql mysql -uroot -p*”.

```

+-----+
| Tables_in_ejercicio |
+-----+
| producto            |
+-----+
1 row in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_ejercicio |
+-----+
| producto            |
| usuario01          |
| usuario_rolas       |
+-----+
3 rows in set (0.06 sec)

mysql> select * from ejercicio.usuario;
+-----+-----+-----+
| id | cont | nombre |
+-----+-----+-----+
| 1  | 2a$10$z8KrEoFbTBefqN8oHi9y8u15PhX.q.jgLSNhhwwNEa1z4lQbpe7LnG | user |
| 2  | 2a$10$gW0ym5Xdss9BHV09I9CPJ08Sr/vgLL7uF3UeQHdMXDY4Lq//tQ6a0 | root |
+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> select * from ejercicio.producto;
+-----+-----+-----+-----+
| id | codigo | descripcion | nombre | precio |
+-----+-----+-----+-----+
| 1  | 111    | 111         | 111    | 111    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

Ilustración 8

Por otro lado, se activa la consola para la parte del servidor en la que realizaremos la conexión con la base de datos `"docker run --name practica -p 8080:8080 --link ejersql -i evaurjc2017/eva:practica"`. Con `"-i"` indicamos que es interactivo.

```
root@kali:~# mysql-master# docker run --name practica p 8080:8080 --link ejersql -i evaurjc2017/eva:practica
Unable to find image 'evaurjc2017/eva:practica' locally
practica: Pulling from evaurjc2017/eva
Digest: sha256:g18520831fc0731fbc92481ac476bf43393999e611fbab3f1744cbcc0532c965
Status: Downloaded newer image for evaurjc2017/eva:practica
Digest: sha256:f93d477fe30dc01af1be3ec8fba3deeda7ae7774dbbbac430eb175da471
Root FS Mounted From Local Image or Container
WARNING: Please restart affected container(s)
---
[+] Running 1 container
Container ID   Name       Requested     Created             Status
-----
(VM) [C] [D] [V] [O] [Y] [N] [E] [S] [I] [T] [M] [A] [I] [N] [G] [ ] ejersql:mysql-servet-p
[+] Root FS Mounted From Local Image or Container
[+] Root FS Mounted From Local Image or Container
[[[#####]]] root@0:localhost (using password: YES)
mysql> eject mysql -servet-p
:: Spring Boot :: (v1.5.4.RELEASE)

The MySQL monitor commands will help you
Type 'help;' for help; type '?' for help.
17-12-04 18:22:20.993 INFO 1 --- [ main] c.i.m.Ejercicio3yThymeleafSpringBootApplication : Starting Ejercicio3yThymeleafSpringBootApplication v0.0.1-SNAPSHOT on 45e07bc24ac4 with PID 1 (/practica.jar started by root in /)
17-12-04 18:22:20.997 INFO 1 --- [ main] c.i.m.Ejercicio3yThymeleafSpringBootApplication : No active profile set, falling back to default profile es: default
17-12-04 18:22:21.339 INFO 1 --- [ main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext[startup date [Mon Dec 04 18:22:21 UTC 2017]; root of context hierarchy [/practica.jar]]
17-12-04 18:22:27.160 INFO 1 --- [ main] o.apache.catalina.core.StandardService : Tomcat initialized with port(s): 8080 (http)
17-12-04 18:22:27.180 INFO 1 --- [ main] o.apache.catalina.core.StandardEngine : Starting service [Tomcat]
17-12-04 18:22:27.181 INFO 1 --- [ main] o.org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.15
17-12-04 18:22:27.598 INFO 1 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
17-12-04 18:22:27.598 INFO 1 --- [ost-startStop-1] o.s.web.context.ContextLoader      : Root WebApplicationContext: initialization completed
17-12-04 18:22:27.737 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Mapping servlet: 'dispatcherServlet' to [/]
```

Ilustración 9

DOCKER AVANZADO

A continuación, ejecutamos en eclipse la parte del cliente y en el navegador, ya podremos lanzar la aplicación y añadir productos.

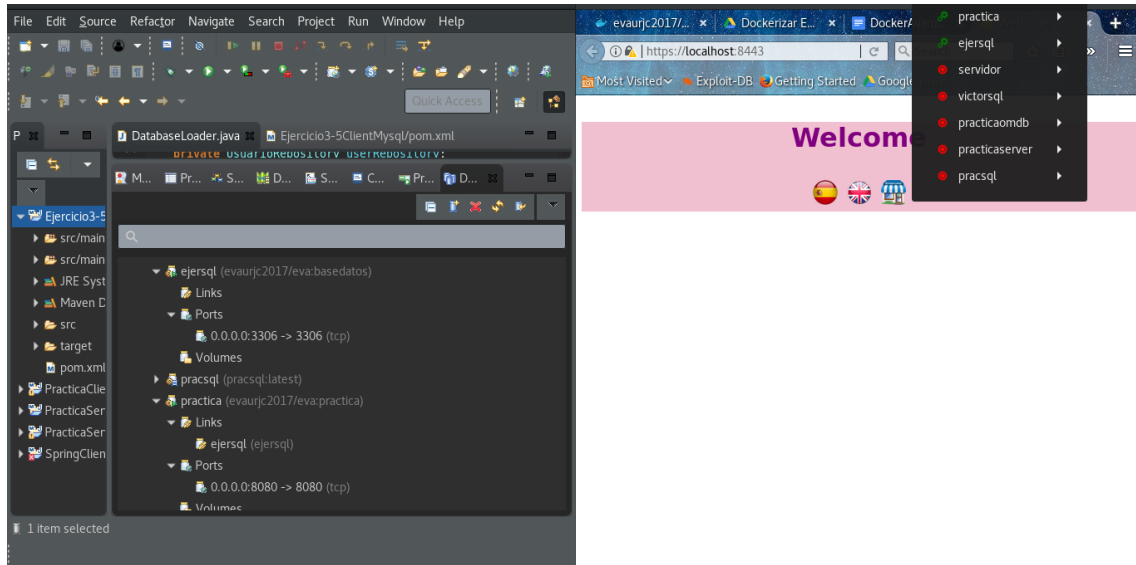


Ilustración 10