



DOCKER



Lorena Martín Tejera
4 Diciembre 2017

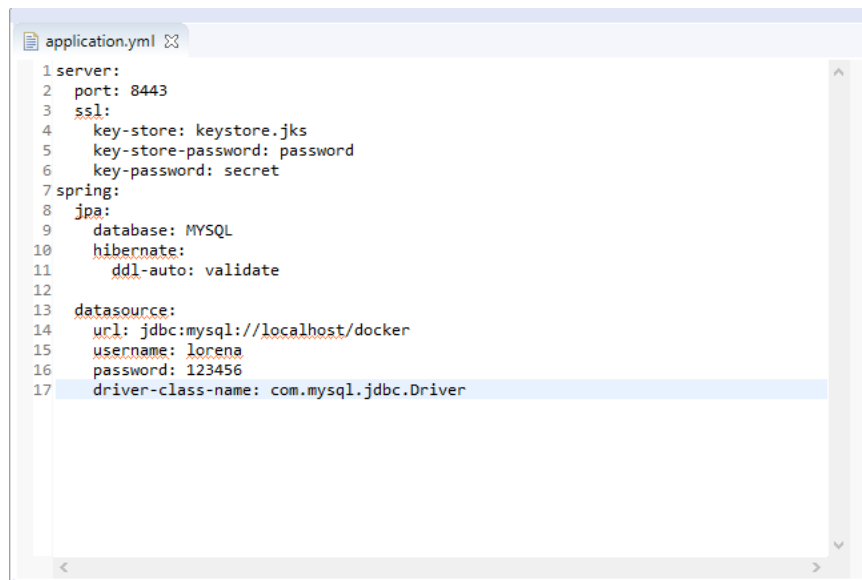
Los pasos para la realización de la práctica utilizando como sistema operativo Linux han sido los siguientes:

- Lo primero de todo, es necesario cambiar la base de datos de la aplicación. Para ello, hay que cambiar las dependencias de h2 por las de mysql como se muestra en la siguiente imagen (Ilustración 1).

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
</dependency>
```

Ilustración 1: dependencias de mysql

- Una vez hecho el paso anterior, hay que descargar e instalar Docker. Una vez hecho esto, se debe descargar la imagen de Docker, que en mi caso he utilizado mysql. Dicha imagen, se debe de descargar de github en el siguiente enlace: <https://github.com/docker-library/mysql>
- Para poder ejecutar Docker hay que ser el administrador mediante el comando “sudo -i” y mediante “service docker start” arrancamos Docker.
- Una vez arrancado, introducimos en la línea de comandos “docker build -t loresql --label loresql 5.7/” para crear el contenedor de mysql.
- Mediante el comando “docker run --name loresql -e MYSQL_ROOT_PASSWORD=123456 -e MYSQL_DATABASE=docker -e MYSQL_USER=lorena -e MYSQL_PASSWORD=123456 -p 3306:3306 -d loresql:latest” arranca el contenedor creado anteriormente con la imagen mysql y permitiendo tener el servidor activo y así conectarlo con la aplicación.
- El siguiente paso que se tiene que hacer es modificar los archivos “application.yml” tanto del cliente como del servidor como se muestra en las siguientes imágenes (Ilustración 2 y 3):

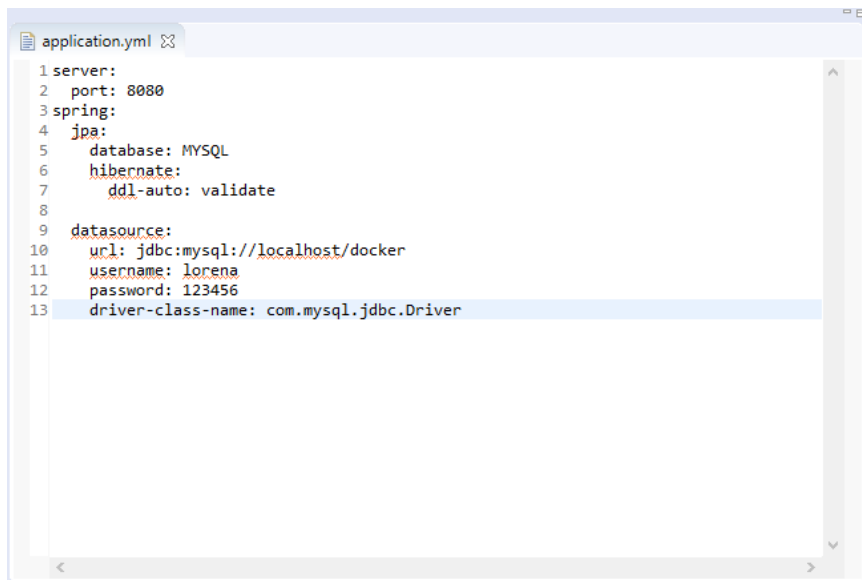


```

1 server:
2   port: 8443
3   ssl:
4     key-store: keystore.jks
5     key-store-password: password
6     key-password: secret
7 spring:
8   jpa:
9     database: MYSQL
10    hibernate:
11      ddl-auto: validate
12
13  datasource:
14    url: jdbc:mysql://localhost/docker
15    username: lorena
16    password: 123456
17    driver-class-name: com.mysql.jdbc.Driver

```

Ilustración 2: application.yml del cliente



```

1 server:
2   port: 8080
3 spring:
4   jpa:
5     database: MYSQL
6     hibernate:
7       ddl-auto: validate
8
9  datasource:
10    url: jdbc:mysql://localhost/docker
11    username: lorena
12    password: 123456
13    driver-class-name: com.mysql.jdbc.Driver

```

Ilustración 3: application.yml del servidor

NOTA: en un inicio, hay que poner en el “application.yml” del cliente “ddl-auto: create” para que se cree la tabla de usuarios con los datos de los usuarios. Esto se puede comprobar una vez que ejecutemos nuestra imagen que se explica más adelante.

- Se arrancan los proyectos en eclipse y el siguiente comando “docker exec -it mysql mysql -uroot -p” lo que hace es abrir la consola de mysql para poder realizar las consultas. Una vez dentro, se debe poner “use docker;” y “show tables;” este último para poder visualizar las tablas creadas que son 3: producto, usuario y roles_usuario.

NOTA: para que no se creen usuarios duplicados cada vez que se ejecuta la imagen, es necesario comentar en el controller del cliente el método “initDataBase()” y cambiar el fichero “application.yml” de “create” a “validate” en “ddl-auto”.

- Para visualizar los usuarios de la aplicación, se introduce en la consola de sql *“select * from docker.usuario”* y para probar que se almacena todos los productos introducidos utilizando la aplicación en la base de datos, mediante *“select * from docker.producto”*.