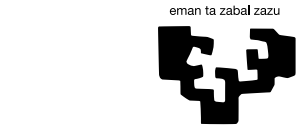
UNIVERSIDAD DEL PAÍS VASCO



ADMINISTRACIÓN DE SISTEMAS - CURSO 2021 / 2022

Entrega individual

Autor:
Ander San Juan

`https://github.com/sanjuesc/ProyectoIndividualAS`

2021

1 Entrega Individual

La imagen Docker que recibí fue Tomcat, pero en vez de usar la oficial me decidí por usar una imagen que usara Tomcat 9 y Java 8.

Por como funciona Tomcat, para acceder a la aplicación no basta con acceder a la URL donde se aloja la misma mediante el puerto adecuado, si no que hay que ir a la ruta del proyecto. En este caso, si la aplicación se aloja en local, habría que acceder mediante el enlace 127.0.0.1/Prueba-1.0-SNAPSHOT/.

He decidido usar también la imagen oficial de MySQL y he realizado las siguientes tareas:

1.1 Crear una aplicación Docker

Mi aplicación es una pequeña página web dinámica (aprovechando las opciones que ofrece Tomcat) que usa el controlador JDBC para poder conectarse a la base de datos MySQL que se ejecutar en el otro contenedor Docker.

Para hacer esto he tenido que crear mi propia imagen de Docker partiendo de la imagen de Tomcat, exponer el puerto 8080, descargar el cliente MySQL (mas adelante se explica el por qué), sudo (es necesario para instalar el cliente MySQL) y otros ficheros necesarios.

Primero debemos iniciar sesión en la web misma con nuestro nombre de usuario y contraseña, los cuales son ´ulopeznovoa´ y ´unaipass´ respectivamente, y si los datos coinciden con los de la base de datos, la web nos redireccionará a un Servlet. Si intentamos acceder a este Servlet sin usar el formulario de inicio de sesión o no tenemos una sesión creada anteriormente (lo cual es posible, ya que no he implementado al opción de cerrar sesión), la web nos devolverá a la página de inicio.

Una vez estemos dentro del Servlet se nos listarán las tablas que se encuentran en la base de datos y si hacemos clic en ellas se desplegará una tabla donde podremos ver su contenido. Si acabamos de iniciar la aplicación, solo encontraremos una tabla llamada USER, que se usa para iniciar sesión, pero si accedemos al contenedor Docker y añadimos mas tablas, entonces se listaran las demás también.

Además de eso habrá también un botón para generar un volcado de la misma base de datos (y gracias a las tablas mencionadas ya, podremos comprobar si se ha hecho bien el volcado). Ya que el controlador que he usado no da la opción de hacer volcados, he tenido que instalar el cliente MySQL en el contenedor Tomcat y ejecutar un comando de bash desde java para crearlo.

```
mysqldump --column-statistics=0 -h IP -u USUARIO -pCONTRASEKA Base_de_datos > Fichero
```

Idealmente, me hubiera gustado no tener que hacer esto último ya que implica pasar la dirección del

servidor MySQL, usuario y contraseña por linea de comandos, pero a falta de una opción mejor, he decidido dejarlo así por ahora. Este volcado quedará guardado en el volumen que he creado para la aplicación.

Si se ve algo mas además de lo ya mencionado, por ejemplo, un texto que dice ´This is a simple java servlet´, es posible que la imagen que hay subida en Dockerhub no sea la mas reciente (probablemente habré estado haciendo cambios en la web, pero no haya subido la ultima versión).

Todo esto se ca

El fichero docker-compose se compone de dos servicios, uno para la imagen de Tomcat con sus variables de entorno, puertos volúmenes y *links* y otra para la de MySQL con sus variables de entorno. Además de eso también se define el volumen que se usa para guardar ficheros de manera persistente.

1.2 Crear una aplicación Kubernetes

Además de el fichero docker-compose, existe también la opción de ejecutar la aplicación usando Kubernetes. Para esto tendremos que usar los ficheros que se encuentran dentro de la carpeta K8s, yo recomiendo usar el comando kubectl apply -f K8s (el nombre de la carpeta donde se encuentran los ficheros), ya que ejecutará el mismo comando sobre cada uno de los archivos de la carpeta indicada.

Estos archivos se componen de la siguiente manera:

- **db-depl.yaml**: El objeto Deployment que uso para la base de datos. Dentro de este se indica la imagen a usar, abrir el puerto 3306 y las variables de entorno necesarias.
- **db-serv.yaml**: El servicio que permite el acceso al puerto 3306 de la base de datos mediante el puerto 3306 del host.
- **servidor-web-depl.yaml**: El objeto Deployment que uso para la aplicación cliente. Dentro de este se indica la imagen a usar, abrir el puerto 8080, las variables de entorno necesarias y el volumen que se usará para guardar el volcado de la base de datos.
- **load-balancer8080.yml**: Un objeto de tipo LoadBalancer para poder acceder al puerto 8080 de la aplicación cliente mediante el puerto 80 del host.
- **volumenbasededatos.yaml**: El objeto usado para definir el volumen donde se guardará el volcado.

1.3 Utilizar bind mounts o volúmenes Docker/volúmenes de Kubernetes para guardar los ficheros de forma persistente

Como ya he indicado, he usado volúmenes de Docker para poder guardar ficheros de forma persistente cuando usamos la aplicación mediante Docker, y volumenes de Kubernetes para guardarlos cuando usamos Kubernetes.

2 Ficheros

Todos los ficheros pueden encontrarse en el siguiente repositorio de GitHub. Nada mas entrar pueden verse los ficheros Dockerfile y docker-compose y dos carpetas, en K8s se encuentran todos los archivos relacionados con Kubernetes y en Código se encuentra todo el proyecto de java relacionado con Tomcat.