

UNIVERSIDAD DEL PAÍS VASCO



ADMINISTRACIÓN DE SISTEMAS - CURSO 2021 / 2022

Entrega individual

Autor:

Ander San Juan

https:

`//github.com/sanjuesc/ProyectoIndividualAS`

Índice

1	Introducción	1
1.1	Tomcat	1
1.2	Otros	1
2	Tareas realizadas	2
2.1	Crear una aplicación Docker	2
2.2	Crear una aplicación Kubernetes	3
2.3	Utilizar bind mounts o volúmenes Docker/volúmenes de Kubernetes para guardar los ficheros de forma persistente	4
3	Ficheros	5

1 Introducción

1.1 Tomcat

La imagen Docker que recibí fue la imagen oficial de Tomcat, pero en vez de usar la oficial me decidí por usar una imagen que usara Tomcat 9 y Java 8. Tomcat es un contenedor, un shell de ejecución que maneja e invoca servlets por cuenta del usuario, de Servlets con un entorno JSP.

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/.

1.2 Otros

Junto a la imagen de Tomcat, he decidido usar también la imagen oficial de MySQL para así poder almacenar y recuperar los datos referentes a mi aplicación.

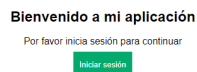
2 Tareas realizadas

2.1 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 ejecuta 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 (más 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 la opción de cerrar sesión), la web nos devolverá a la página de inicio.



Perdon por no currarme mas esta pagina

Figura 1: Página web inicial

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 más tablas, entonces se listarán 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 -pCONTRASENA BaseDeDatos > Fichero
```

Idealmente, me hubiera gustado no tener que hacer esto último ya que implica pasar la dirección del servidor MySQL, usuario y contraseña por línea 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, se debe a que la aplicación descarga un fichero WAR (que contiene todos los ficheros relacionados con la aplicación web de Tomcat) de un repositorio mio que puede encontrarse en GitHub.



id	name	pass
1	ulopezmovoa	unaipass
2	ander	probak

Hacer dump

Figura 2: Aspecto de la web con la tabla inicial

La imagen de mi aplicación Tomcat puede encontrarse en el repositorio de Dockerhub bajo el nombre de 'andersanjuan/sistemas', pero también se puede usar mediante el fichero Dockerfile. Se entregan dos ficheros docker-compose, uno usando la imagen de Dockerhub y otro construyendola con Dockerfile.

Los ficheros docker-compose se componen 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.

2.2 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.

2.3 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 volúmenes de Kubernetes para guardarlos cuando usamos Kubernetes.

3 Ficheros

Todos los ficheros pueden encontrarse en el siguiente repositorio de GitHub. El repositorio sigue la siguiente estructura:

- **Código:** Todo el código relacionado al servidor de Tomcat. Los archivos mas importantes son los siguientes:
 - **src/main/webapp/index.jsp:** Fichero JSP donde se ve como funciona la web que se usa para iniciar sesión.
 - **src/main/java/com/example/Prueba:** Servlet.java es el Servlet principal en el que podemos ver las tablas y clicar el botón que nos redirige al otro Servlet (Dump.java), el cual hace el volcado de la base de datos y nos devuelve al Servlet.
- **K8s:** Ficheros Kubernetes mencionados anteriormente.
- **docker-compose (usando Dockerfile):** Fichero docker-compose obteniendo la imagen desde el fichero Dockerfile.
- **docker-compose (usando version del repositorio):** Fichero docker-compose obteniendo la imagen desde el repositorio de Dockerhub.
- **Documentación.pdf:** Este mismo documento (viva la recursión).