

**IES JACARANDÁ**



**Despliegue de  
Aplicaciones Web**

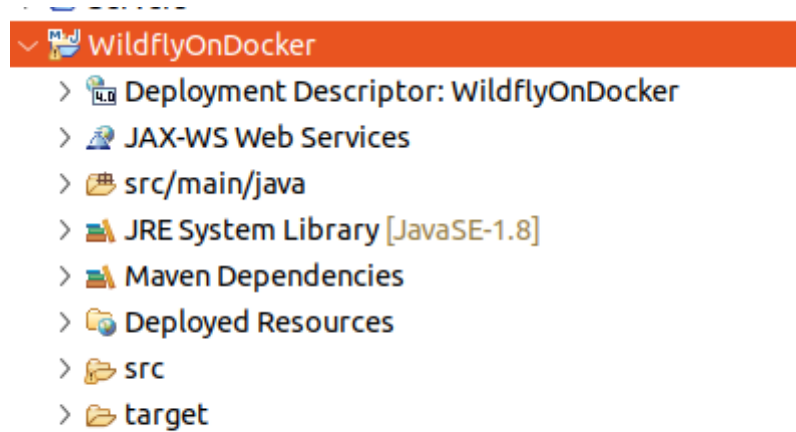
**Desplegar una aplicación  
con Wildfly**

**Por: Chisela Marien Colás Gil**

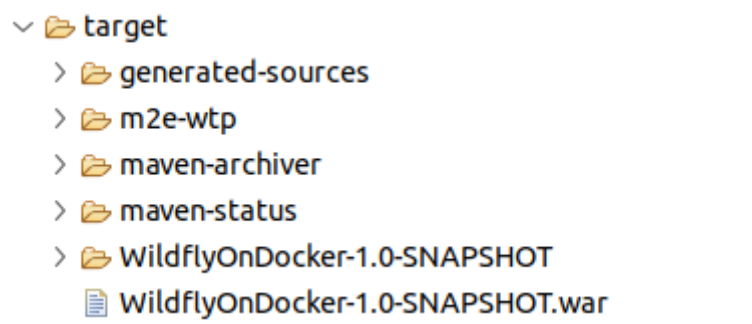
**2 DAW**

En esta guía vamos a crear un contenedor Wildfly que despliegue una sencilla aplicación que muestra “Hello World”. Para ello crearemos nuestra propia imagen a través de un *Dockerfile* partiendo de la imagen oficial y necesitaremos un proyecto Java con un archivo .war.

El primer paso será obtener el archivo .war en nuestro proyecto. A través de un IDE, en este caso Eclipse, importamos el proyecto en nuestro espacio de trabajo.



Sobre este proyecto hacemos click derecho > Run as > Maven build... y en la pestaña que aparecerá escribimos en la sección *Goals* **clean install**. Después de hacer esto, desplegamos el directorio *target* y encontramos el fichero .war.



El siguiente paso es crear nuestra propia imagen para el contenedor utilizando este fichero. Creamos un archivo llamado *Dockerfile*(sin extensión) a la misma altura que el directorio *target* y añadimos las siguientes líneas:

```
FROM jboss/wildfly
```

```
ADD target/WildflyOnDocker-1.0-SNAPSHOT.war /opt/jboss/wildfly/standalone/deployments/
```

Con esto hemos indicado que partimos de la imagen oficial de Wildfly y que la aplicación que se va a desplegar es la nuestra, gracias al fichero que hemos generado.

Ahora que tenemos este fichero podemos crear nuestra propia imagen con el comando *docker build --tag=wildflyapp* . en la ruta donde se encuentra el archivo *Dockerfile*, con el nombre que queramos gracias a la opción *--tag*.

```
estudiante@DAW1:~/Chisela/Despliegue web/wildfly/WildflyOnDocker$ docker build --tag=wildflyapp .
Sending build context to Docker daemon 56.32kB
Step 1/2 : FROM jboss/wildfly
--> 856694040847
Step 2/2 : ADD target/WildflyOnDocker-1.0-SNAPSHOT.war /opt/jboss/wildfly/standalone/deployments/
--> e3fdc8858944
Successfully built e3fdc8858944
Successfully tagged wildflyapp:latest
```

Y para acabar, con la imagen que acabamos de generar, creamos el contenedor con el siguiente comando:

```
$ docker run -p 8080:8080 -it wildflyapp /opt/jboss/wildfly/bin/standalone.sh -b 0.0.0.0
```

```
estudiante@DAW1:~/Chisela/Despliegue web/wildfly/WildflyOnDocker$ docker run -p 8080:8080 -it wildflyapp /opt/jboss/wildfly/bin/standalone.sh -b 0.0.0.0
=====

JBoss Bootstrap Environment

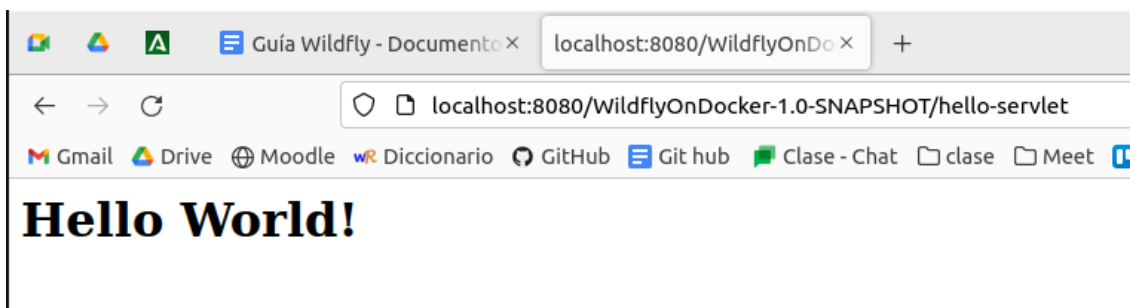
JBoss_HOME: /opt/jboss/wildfly

JAVA: /usr/lib/jvm/java/bin/java

JAVA_OPTS: -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -D
java.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.he
adless=true --add-exports=java.desktop/sun.awt=ALL-UNNAMED --add-exports=java.naming/com
sun.jndi.idap=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.b
```

Para comprobar que nuestro despliegue funciona correctamente accedemos a la siguiente dirección:

<http://localhost:8080/WildflyOnDocker-1.0-SNAPSHOT/hello-servlet>



Y como podemos ver, el despliegue ha sido un éxito.

## Referencias web

- [Repositorio original de la aplicación desplegada](#)
- [Tutorial para desplegar una aplicación con wildfly](#)
- [Crear un archivo.war a partir de un proyecto en Eclipse](#)