



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

**Ingeniería de Software 3
TRABAJO PRÁCTICO N°6**

Profesor: Ing Fernando Bono.

Alumno: Menel Angelo (1804789)

CONCEPTOS DE DOCKERFILE

- FROM: Se le indica el nombre de la imagen a utilizar.
- RUN: Corre un comando de consola.
- ADD: Agrega archivos o directorios a la imagen.
- COPY: Realiza la copia de un directorio a otro.
- EXPOSE: Le indica al contenedor que puerto tiene que estar escuchando.
- CMD: Se especifican los parámetros que utilizará el ENTRYPOINT.
- ENTRYPOINT: Especifica el ejecutable que usará el contenedor.

`docker run -p 8080:8080 test-spring-boot`

```

:: Spring Boot ::                (v2.0.2.RELEASE)

2022-09-15 15:21:04.149 INFO 1 --- [          main] s.actuator.SampleActuatorApplication : Starting SampleActuatorApplication v2.0.2 on 3db7c0
5d92f1 with PID 1 (/app/spring-boot-application.jar started by root in /app)
2022-09-15 15:21:04.152 INFO 1 --- [          main] s.actuator.SampleActuatorApplication : No active profile set, falling back to default pro
files: default
2022-09-15 15:21:04.194 INFO 1 --- [          main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.con
text.AnnotationConfigServletWebServerApplicationContext$567d299b: startup date [Thu Sep 15 15:21:04 GMT 2022]; root of context hierarchy
2022-09-15 15:21:05.078 INFO 1 --- [          main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.Pr
oxyTransactionManagementConfiguration' of type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpring
CGLIB$$364e9ebc] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2022-09-15 15:21:05.302 INFO 1 --- [          main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-09-15 15:21:05.322 INFO 1 --- [          main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-09-15 15:21:05.322 INFO 1 --- [          main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.31
2022-09-15 15:21:05.329 INFO 1 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which al
lows optimal performance in production environments was not found on the java.library.path: [/usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64/server:/usr
/lib/jvm/java-1.8-openjdk/jre/lib/amd64:/usr/lib/jvm/java-1.8-openjdk/jre/../lib/amd64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]
2022-09-15 15:21:05.392 INFO 1 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-09-15 15:21:05.392 INFO 1 --- [ost-startStop-1] o.s.w.config.ContextLoader      : Root WebApplicationContext: initialization complete
d in 1198 ms
2022-09-15 15:21:06.032 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
2022-09-15 15:21:06.035 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [*/*]
2022-09-15 15:21:06.035 INFO 1 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: /*

```

```
curl -v localhost:8080
```

```

❯ ~/Ing_3/Trabajo_Practico_6/spring-boot develop !1 curl -v localhost:8080
* Trying 127.0.0.1:8080...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET / HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.85.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Date: Thu, 15 Sep 2022 15:22:43 GMT
<
* Connection #0 to host localhost left intact
{"message":"Spring boot says hello from a Docker container"}%

```

Luego agrego al Dockerfile un multistage, con una imagen de maven y otra de openjdk.

```
FROM maven:3.5.2-jdk-8-alpine AS MAVEN_TOOL_CHAIN
COPY pom.xml /tmp/
RUN mvn -B dependency:go-offline -f /tmp/pom.xml -s /usr/share/maven/ref/settings-docker.xml
COPY src /tmp/src/
WORKDIR /tmp/
RUN mvn -B -s /usr/share/maven/ref/settings-docker.xml package

FROM java:8-jre-alpine

EXPOSE 8080

RUN mkdir /app
COPY --from=MAVEN_TOOL_CHAIN /tmp/target/*.jar /app/spring-boot-application.jar

ENV JAVA_OPTS="-Xms32m -Xmx128m"

ENTRYPOINT exec java $JAVA_OPTS -Djava.security.egd=file:/dev/./urandom -jar /app/spring-boot-application.jar

HEALTHCHECK --interval=1m --timeout=3s CMD wget -q -T 3 -s http://localhost:8080/actuator/health/ || exit 1
```

- Lo que estoy haciendo es definir 2 imágenes dentro del Dockerfile, donde uno sera el de maven. Copio el pom.xml al directorio TMP, corro maven, copio el directorio de nuestra app al TMP/SRC, le indico que el directorio de trabajo es TMP.
- Para java, le indico el puerto al que tiene que escuchar, le pido que crea el directorio app, luego en el COPY creo el enlace entre java y maven. El ENTRYPOINT indica el archivo jar a ejecutar.
- HEALTHCHECK es un comando para crear un endpoint que te indica si el contenedor esta levantado o no.

¿Para qué está la key build.context en el docker-compose.yml?

- El build.context se utiliza para poder utilizar la imagen para distintos contextos, como uno de producción y otro de desarrollo.
- El path al directorio que tiene el Dockerfile.