**使用daockerfile-maven-plugin将maven项目打包发布为docker镜像**

**环境配置**

Maven

Dockertoolbox

**操作步骤**

1. 创建一个简单的maven项目
2. 为项目添加插件，在项目根目录下pom.xml添加：

<properties>   
<docker.image.prefix>demo</docker.image.prefix> <!--镜像前缀，随便取-->   
</properties>   
<build>   
<plugins>   
<plugin>   
<groupId>com.spotify</groupId>   
<artifactId>dockerfile-maven-plugin</artifactId>   
<version>1.3.7</version>   
<!--放开这块注释，则运行mvn deploy命令就会打包镜像<executions>-->   
<!--<execution>-->   
<!--<id>default</id>-->   
<!--<goals>-->   
<!--<goal>build</goal>-->   
<!--<goal>push</goal>-->   
<!--</goals>-->   
<!--</execution>-->   
<!--</executions>-->   
<configuration>   
<repository>${docker.image.prefix}/${project.artifactId}</repository>   
<buildArgs>   
<JAR\_FILE>target/${project.build.finalName}.jar</JAR\_FILE>   
</buildArgs>   
</configuration>   
</plugin>   
</plugins>   
</bulid>

1. 配置dockerfile，在项目根目录下添加名为dockerfile的文件。项目打包成什么样的镜像由此文件中配置决定：

#选择基础镜像openjdk

FROM openjdk

#指定外部文件夹挂载到容器内的/tmp文件夹

VOLUME /tmp

#设置参数名为JAR\_FILE，在build镜像时可以使用--build-arg<varname>=<value>来指定参数值，本次将打包好的项目文件为变量值

ARG JAR\_FILE

#将${JAR\_FILE}文件，即打包好的项目文件，复制为镜像中的app.jar文件

ADD ${JAR\_FILE} app.jar

#在镜像中运行指定的指令

RUN bash -c 'touch /app.jar'

#暴露端口8800

EXPOSE 8800

#设置启动时的默认命令java -Djava.security.egd=file:/dev/./urandom -jar /app.jar

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]

1. 在maven项目根目录下执行以下命令打包项目：

mvn package -DskipTests

mvn dockerfile:build

**效果**

·在本地docker中可以找到打包好的项目镜像



·运行镜像后项目成功启动



**未解决的问题**

·没有自动push镜像

**附：**

配置好的pom.xml和dockerfile

放于项目根目录下

