

maven构建docker镜像

准备环境:

软件	版本
docker	18.09.2 (其他版本也可以)
maven	3.0.5 (3.0以上版本)
jdk	1.8

- 一、maven配置
- 二、项目配置
- 三、Dockerfile配置
- 四、编译生成镜像并推送
- 五、命令参数

一、maven配置

```
settings.xml<docker-maven-plugin>
  <mirror>
    <id>alimaven</id>
    <mirrorOf>central</mirrorOf>
    <name>aliyun maven</name>

    <url>http://maven.aliyun.com/nexus/content/repositories/central/</url>
  </mirror>

  <mirror>
    <id>nexus-aliyun</id>
    <mirrorOf>central</mirrorOf>
    <name>Nexus aliyun</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public</url>
  </mirror>

  <mirror>
    <id>central</id>
    <name>Maven Repository Switchboard</name>
    <url>http://repo1.maven.org/maven2/</url>
    <mirrorOf>central</mirrorOf>
  </mirror>

  <mirror>
    <id>repo2</id>
    <mirrorOf>central</mirrorOf>
    <name>Human Readable Name for this Mirror.</name>
    <url>http://repo2.maven.org/maven2/</url>
  </mirror>
```

```
<mirror>
  <id>ibiblio</id>
  <mirrorOf>central</mirrorOf>
  <name>Human Readable Name for this Mirror.</name>
  <url>http://mirrors.ibiblio.org/pub/mirrors/maven2/</url>
</mirror>

<mirror>
  <id>jboss-public-repository-group</id>
  <mirrorOf>central</mirrorOf>
  <name>JBoss Public Repository Group</name>
  <url>http://repository.jboss.org/nexus/content/groups/public</url>
</mirror>
```

docker

```
<pluginGroups>
  <pluginGroup>com.spotify</pluginGroup>
</pluginGroups>
```

```
<server>
  <id>my-docker-registry</id>
  <username>admin</username>
  <password>Harbor12345</password>
  <configuration>
    <email>houfei@sunline.cn</email>
```

```
    </configuration>
</server>
```

二、项目配置

```
pom.xml
docker-maven-plugin

<plugin>
  <groupId>com.spotify</groupId>
  <artifactId>docker-maven-plugin</artifactId>
  <version>0.4.13</version>
  <configuration>
    <imageName>10.22.60.25/test/${project.artifactId}</imageName>
    <dockerDirectory>./</dockerDirectory>
    <!-- <dockerDirectory>/data/docker</dockerDirectory> -->
    <resources>
      <resource>
        <targetPath>/</targetPath>
        <directory>${project.build.directory}</directory>
        <include>${project.build.finalName}.war</include>
      </resource>
    </resources>
    <serverId>my-docker-registry</serverId>          ##
  </configuration>
</plugin>
```

三、Dockerfile配置

Dockerfile和pom.xml文件放在同一级目录

此处使用的dubbo-admin的源代码进行的测试

```
FROM 10.22.60.25/basic/centos-jdk1.8:7.2.1511 wardockertomcatwarwebapps
ADD target/docker/dubbo-admin-2.5.7.war /dubbo-admin-2.5.7.war
EXPOSE 9000
CMD /bin/bash
```

四、编译生成镜像并推送

```
$ mvn clean install docker:build -DdockerImageTags=v3 -DpushImage
```

五、命令参数

```
-DdockerImageTagstag
-DpushImage
-DskipDockerBuild
-DskipDockerTag
-DskipDockerPush
-DskipDockerDocker

mvn docker:build
mvn docker:start
mvn docker:stop
mvn -Ddocker.removeAll docker:remove
mvn docker:logs
```