

使用Dockerfile为SpringBoot应用构建Docker镜像

原创 梦想de星空 macrozheng 2019-06-17 08:32

收录于合集

#mall学习教程（参考篇）

27个

上次写过一篇[使用Maven插件构建Docker镜像](#)，讲述了通过docker-maven-plugin来构建docker镜像的方式，此种方式需要依赖自建的Registry镜像仓库。本文将讲述另一种方式，使用Dockerfile来构建docker镜像，此种方式不需要依赖自建的镜像仓库，只需要应用的jar包和一个Dockerfile文件即可。

Dockerfile常用指令

ADD

用于复制文件，格式：

```
ADD <src> <dest>
```

示例：

```
# 将当前目录下的mall-tiny-docker-file.jar包复制到docker容器的/目录下
ADD mall-tiny-docker-file.jar /mall-tiny-docker-file.jar
```

ENTRYPOINT

指定docker容器启动时执行的命令，格式：

```
ENTRYPOINT ["executable", "param1", "param2" ...]
```

示例：

```
# 指定docker容器启动时运行jar包
ENTRYPOINT ["java", "-jar", "/mall-tiny-docker-file.jar"]
```

ENV

用于设置环境变量，格式：

```
ENV <key> <value>
```

示例：

```
# mysql运行时设置root密码  
ENV MYSQL_ROOT_PASSWORD root
```

EXPOSE

声明需要暴露的端口(只声明不会打开端口)，格式：

```
EXPOSE <port1> <port2> ...
```

示例：

```
# 声明服务运行在8080端口  
EXPOSE 8080
```

FROM

指定所需依赖的基础镜像，格式：

```
FROM <image>:<tag>
```

示例：

```
# 该镜像需要依赖的java8的镜像  
FROM java:8
```

MAINTAINER

指定维护者的名字，格式：

```
MAINTAINER <name>
```

示例：

```
MAINTAINER macrozheng
```

RUN

在容器构建过程中执行的命令，我们可以用该命令自定义容器的行为，比如安装一些软件，创建一些文件等，格式：

```
RUN <command>  
RUN ["executable", "param1", "param2"...]
```

示例：

```
# 在容器构建过程中需要在/目录下创建一个mall-tiny-docker-file.jar文件  
RUN bash -c 'touch /mall-tiny-docker-file.jar'
```

使用Dockerfile构建SpringBoot应用镜像

编写Dockerfile文件

```
# 该镜像需要依赖的基础镜像  
FROM java:8  
# 将当前目录下的jar包复制到docker容器的/目录下  
ADD mall-tiny-docker-file-0.0.1-SNAPSHOT.jar /mall-tiny-docker-file.jar  
# 运行过程中创建一个mall-tiny-docker-file.jar文件  
RUN bash -c 'touch /mall-tiny-docker-file.jar'  
# 声明服务运行在8080端口  
EXPOSE 8080  
# 指定docker容器启动时运行jar包  
ENTRYPOINT ["java", "-jar", "/mall-tiny-docker-file.jar"]  
# 指定维护者的名字  
MAINTAINER macrozheng
```

使用maven打包应用

在IDEA中双击package命令进行打包:



打包成功后展示:

```
[INFO] --- spring-boot-maven-plugin:2.1.3.RELEASE:repackage (repackage) @ mall-tiny-dock
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 30.749 s
[INFO] Finished at: 2019-06-16T14:11:07+08:00
[INFO] Final Memory: 43M/306M
[INFO] -----
```

将应用jar包及Dockerfile文件上传到linux服务器：



在Linux上构建docker镜像

在Dockerfile所在目录执行以下命令：

```
# -t 表示指定镜像仓库名称/镜像名称:镜像标签 .表示使用当前目录下的Dockerfile
docker build -t mall-tiny/mall-tiny-docker-file:0.0.1-SNAPSHOT .
```

输出如下信息：

```
Sending build context to Docker daemon 36.37MB
Step 1/5 : FROM java:8
---> d23bdf5b1b1b
Step 2/5 : ADD mall-tiny-docker-file-0.0.1-SNAPSHOT.jar /mall-tiny-docker-file.jar
---> c920c9e9d045
Step 3/5 : RUN bash -c 'touch /mall-tiny-docker-file.jar'
---> Running in 55506f517f19
Removing intermediate container 55506f517f19
---> 0727eded66dc
Step 4/5 : EXPOSE 8080
---> Running in d67a5f50aa7d
Removing intermediate container d67a5f50aa7d
---> 1b8b4506eb2d
Step 5/5 : ENTRYPOINT ["java", "-jar", "/mall-tiny-docker-file.jar"]
```

```
---> Running in 0c5bf61a0032
Removing intermediate container 0c5bf61a0032
---> c3614dad21b7
Successfully built c3614dad21b7
Successfully tagged mall-tiny/mall-tiny-docker-file:0.0.1-SNAPSHOT
```

查看docker镜像：



运行mysql服务并设置

1.使用docker命令启动：

```
docker run -p 3306:3306 --name mysql \
-v /mydata/mysql/log:/var/log/mysql \
-v /mydata/mysql/data:/var/lib/mysql \
-v /mydata/mysql/conf:/etc/mysql \
-e MYSQL_ROOT_PASSWORD=root \
-d mysql:5.7
```

2.进入运行mysql的docker容器：

```
docker exec -it mysql /bin/bash
```

3.使用mysql命令打开客户端：

```
mysql -uroot -proot --default-character-set=utf8
```

4.修改root帐号的权限，使得任何ip都能访问：

```
grant all privileges on *.* to 'root'@'%'
```

5.创建mall数据库：

```
create database mall character set utf8
```

6.将mall.sql文件拷贝到mysql容器的/目录下：

```
docker cp /mydata/mall.sql mysql:/
```

7. 将sql文件导入到数据库：

```
use mall;  
source /mall.sql;
```

运行mall-tiny-docker-file应用

```
docker run -p 8080:8080 --name mall-tiny-docker-file \  
--link mysql:db \  
-v /etc/localtime:/etc/localtime \  
-v /mydata/app/mall-tiny-docker-file/logs:/var/logs \  
-d mall-tiny/mall-tiny-docker-file:0.0.1-SNAPSHOT
```

访问接口文档地址<http://192.168.3.101:8080/swagger-ui.html>:



项目源码地址

<https://github.com/macrozheng/mall-learning/tree/master/mall-tiny-docker-file>

推荐阅读

- [使用Maven插件构建Docker镜像](#)
- [开发者必备Docker命令](#)
- [开发者必备Linux命令](#)
- [mall在Linux环境下的部署（基于Docker容器）](#)
- [mall在Windows环境下的部署](#)
- [mall整合SpringBoot+MyBatis搭建基本骨架](#)
- [mall整合Swagger-UI实现在线API文档](#)
- [mall整合SpringSecurity和JWT实现认证和授权（一）](#)
- [mall整合Elasticsearch实现商品搜索](#)
- [mall整合Mongodb实现文档操作](#)
- [mall整合RabbitMQ实现延迟消息](#)

■ [mall整合OSS实现文件上传](#)



欢迎关注，点个在看

收录于合集 #mall学习教程（参考篇） 27

上一篇
使用Maven插件构建Docker镜像

下一篇
使用Docker Compose部署SpringBoot应用

[阅读原文](#)

喜欢此内容的人还喜欢

项目中到底该不该用Lombok?
macrozheng

