来源网址：https://blog.csdn.net/jiangyu1013/article/details/81988342

1. 把要部署的工程打成一个jar包。（我的工程叫 gentle ）

    打 jar 的方法：超简单方法： Intellij Idea 把 java 工程打成可运行的 jar

2. 把 jar 包上传到 服务器确定目录下。

   我是用 rz 命令直接选择 jar 上传到我存放 jar 的目录：/root/jiangyu/dockerfiles/jar

3.编写超级简单版本的 dockerfile 文件：

# jdk 是一个已有的包含 jdk 的镜像

FROM jdk

#  作者签名

MAINTAINER JY

#  简化 jar 的名字路径 （#：表示注释掉）

COPY jar/gentle-1.0-SNAPSHOT.jar /app.jar

#  执行 java -jar 命令 （CMD：在启动容器时才执行此行。RUN：构建镜像时就执行此行）

CMD java -jar /app.jar

#  设置对外端口为 8089

EXPOSE 8089

4. 通过dockerfile 文件生成本工程的镜像：

docker build -t gentle .

此行命令最后有一个点，表示使用当前上下文中的 dockerfile 文件（上图中当前路径下绿色的那个dockerfile） 。

docker images 查看已有镜像  ，docker build -t gentle .  构建镜像，-t ：给镜像取名为 gentle 。

从第 2 次查看镜像可以确认，新的镜像 gentle 已构建成功。

5. 通过镜像启动容器，命令如下：

#  -d 后台运行、

# 最后一个 gentle 是引用的镜像的名字、

#  --name jy\_gentle 给容器取名为 jy\_gentle （取名参数前面是两短横线）、

#  -p 8089:8089 端口映射，注意是小写 p

# 前一个 8089 是对外浏览器上访问的端口，后一个 8089 是容器内工程本身的端口，两者可不一样

docker run -d -p 8089:8089 --name jy\_gentle gentle

docker ps -a ：查看当前服务器上所有容器。

从第 2 次查看可知容器已成功启动。黄框中的是各个容器使用的镜像的名字，蓝框中的各个容器的名字。

我的工程设置的本地端口也是8089

6. postman 和浏览器 顺利请求到 gentle 工程中的接口，说明部署成功

7. gentle 工程中接口实现：

部署完啦 ，走，出门溜达溜达，找点吃的去  ...

--------------------------------

2018.11.28 后记，每次改代码都要重新部署依旧很麻烦，于是写了个很简单的脚本，运行就部署。

脚本内容只是把命令整合到一起了而已：

cd gentle\_code

# 备份原代码

tar -zcvf gentle.tar ./gentle

# 删除原代码文件夹

rm -rf gentle

# 拉取代码

echo -e "\033[32m\n从git拉取代码\n\033[0m"

git clone https://gitee.com/FJ\_WoMenDeShiJie/gentle.git

echo -e "\033[32m\n代码拉取完成\n\033[0m"

# 工程打包

cd gentle

rm -rf ./target/gentle-1.0-SNAPSHOT.jar

echo -e "\033[32m\n开始打jar包\n\033[0m"

mvn install -Dmaven.test.skip=true

rm -rf ../../jar/gentle-1.0-SNAPSHOT.jar

cp ./target/gentle-1.0-SNAPSHOT.jar ../../jar

echo -e "\033[32m\njar包完成\n\033[0m"

# 删除原镜像

docker rmi gentle &> /dev/null

echo -e "\033[32m\n删除原镜像\n\033[0m"

echo -e "\033[32m\n构建新镜像\n\033[0m"

cd ../../

docker build -t gentle .

echo -e "\033[32m\n新镜像构建成功\n\033[0m"

# 删除原容器

docker rm -f gentle

# 启动容器

docker run -d -p 8089:8089 --name gentle gentle

echo -e "\033[32m\n后端工程部署完成\n\033[0m"

 运行效果：

postman 请求，返回的已经有新接口数据了。

---------------------

作者：微风--轻许--

来源：CSDN

原文：https://blog.csdn.net/jiangyu1013/article/details/81988342

版权声明：本文为博主原创文章，转载请附上博文链接！