

一、镜像操作指令

FROM	指定基础镜像	FROM centos:6
ENV	设置环境变量，可在后面指令使用	ENV key value
COPY	拷贝本地文件到镜像的指定目录	COPY ./mysql-5.7.rpm /tmp
RUN	执行Linux的shell命令，一般是安装过程的命令	RUN yum install gcc
EXPOSE	指定容器运行时监听的端口，是给镜像使用者看的	EXPOSE 8080
ENTRYPOINT	镜像中应用的启动命令，容器运行时调用	ENTRYPOINT java -jar xx.jar

二. 基于CentOS基础镜像构建Java应用镜像

1. 上传JDK

创建文件夹，将jdk安装包上传到该目录下

```
mkdir -p /root/dockerfile/java
```

2. 编写dockerfile

在/root/dockerfile/java目录下创建文件dockerfile_java,内容如下：

```
FROM centos:7
# 配置环境变量，JDK的安装目录
ENV JAVA_DIR=/usr/local

# 拷贝jdk和java项目的包
COPY ./jdk-8u131-linux-x64.tar.gz $JAVA_DIR/
COPY ./docker-demo.jar /tmp/app.jar

# 安装JDK
RUN cd $JAVA_DIR \
  && tar -xf ./jdk-8u131-linux-x64.tar.gz \
  && mv ./jdk1.8.0_131 ./java8

# 配置环境变量
ENV JAVA_HOME=$JAVA_DIR/java8
ENV PATH=$PATH:$JAVA_HOME/bin

# 暴露端口
EXPOSE 8080

# 入口，java项目的启动命令
ENTRYPOINT java -jar /tmp/app.jar
```

3. 生成镜像

```
#docker build -f 文件名 -t 镜像名 .代表从当前目录
docker build -f dockerfile_java -t myjava:1.0 .
```

4. 验证镜像是否生成成功

```
docker images
```

5. 启动容器

```
docker run --name myjava -d -p 8080:8080 myjava:1.0
```

二. 通过java镜像构建应用镜像

1. 编写dockerfile

```
FROM java:8-alpine
COPY ./docker-demo.jar /tmp/app.jar
# 暴露端口
EXPOSE 8080
# 入口, java项目的启动命令
ENTRYPOINT java -jar /tmp/app.jar
```

3. 生成镜像

```
#docker build -f 文件名 -t 镜像名 .代表从当前目录
docker build -f dockerfile_java2 -t myjava:2.0 .
```

4. 验证镜像是否生成成功

```
docker images
```

5. 启动容器

```
docker run --name myjava2 -d -p 8081:8080 myjava:2.0
```