**10步学会容器**

**操作**

**欧阳**

**2019/03/11**

**【备注】Github：**

https://github.com/yangyangmyself/dockerapp/tree/master/ten-minute-case

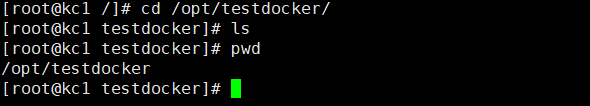
**//1创建目录**

$>mkdir /opt/testdocker



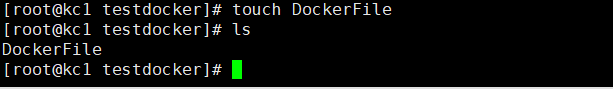
**//2进入目录**

$>cd /opt/testdocker



**//3创建DockerFile文件**

$>touch DockerFile



**//4 Maven打包好项目，拷贝至当前目录（即/opt/testdocker）**

dockerApp、startService-linux.sh为项目运行包及启动脚本



**//5 编缉DockerFile文件，添加以下内容**

FROM java:8

# Copying jdk from OS directory which DockerFile file directory to image directory

#COPY jdk1.8.0\_144 /usr/local/jdk1.8.0\_144

# Configure jdk enviroment

#ENV JAVA\_HOME /usr/local/jdk1.8.0\_144

#ENV PATH $PATH:$JAVA\_HOME/bin:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/jre/lib/rt.jar

# Copy application package include start application shell script

COPY dockerApp /usr/local/dockerApp

COPY startService-linux.sh /usr/local/

RUN chmod 770 /usr/local/startService-linux.sh

EXPOSE 8088

RUN pwd

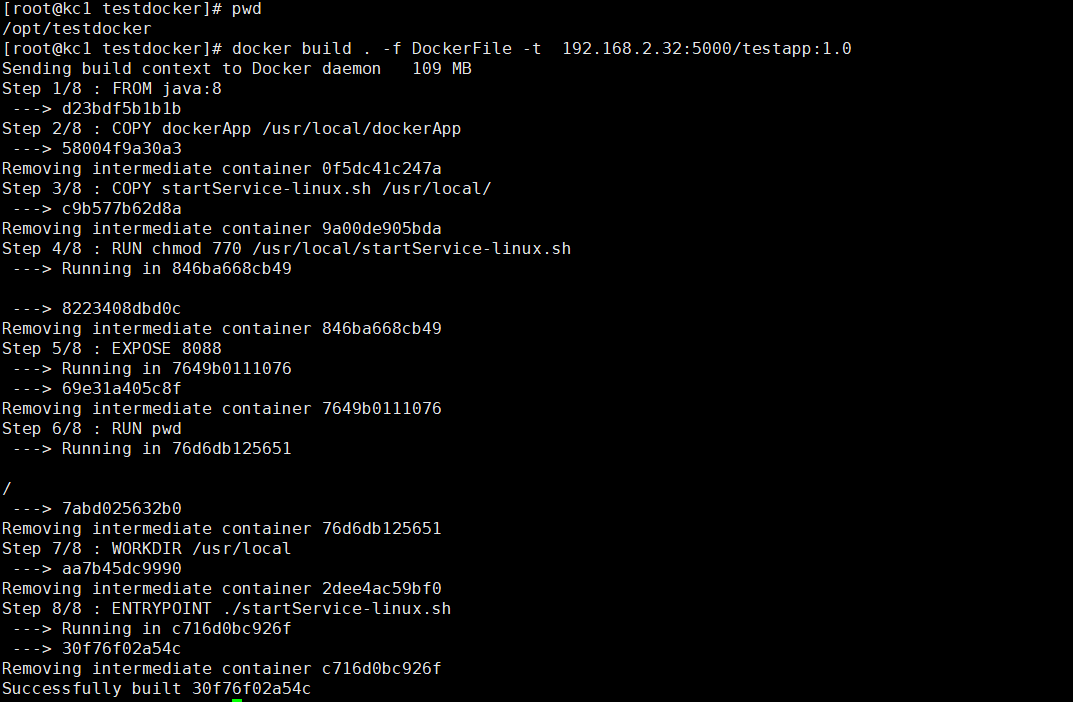
WORKDIR /usr/local

ENTRYPOINT ["./startService-linux.sh"]

**//6 编译生成镜像文件**

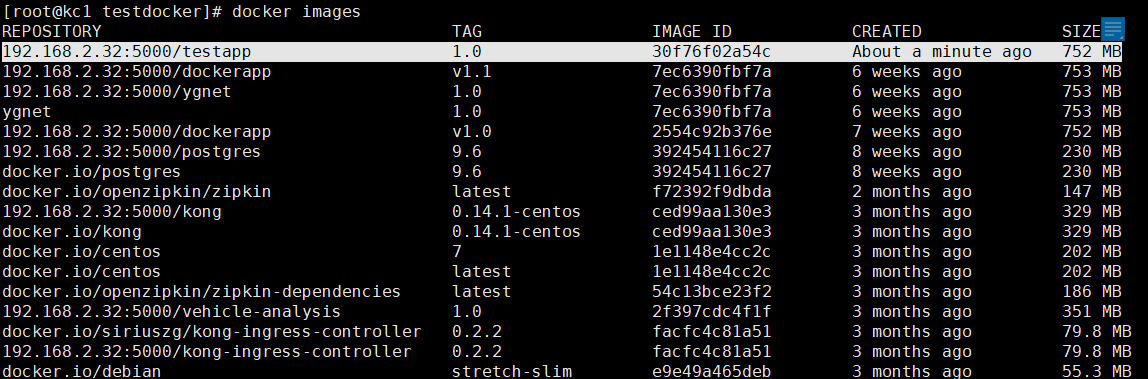
执行：

$>docker build . -f DockerFile -t 192.168.2.32:5000/testapp:1.0



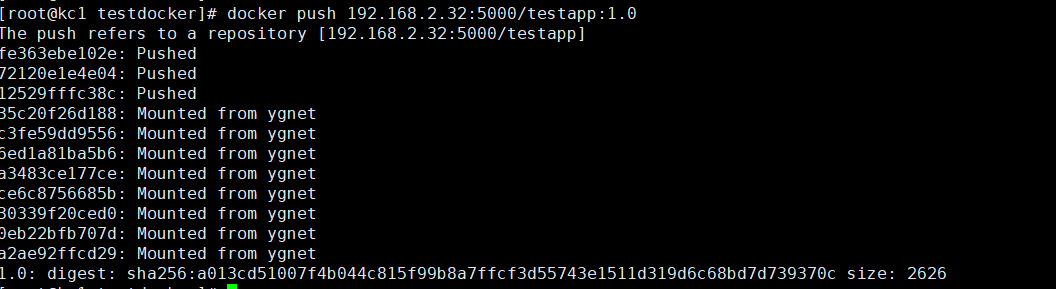
**//7 查看本地镜像库**

$>docker images



**//7 上传私有镜像仓库**

$>docker push 192.168.2.32:5000/testapp:1.0



**//8 在当前目当下，生成K8部署文件**

$>touch kube-springboot-app.yaml

$>vim kube-springboot-app.yaml

拷贝如下内容至kube-springboot-app.yaml

kind: Deployment

apiVersion: apps/v1beta1

metadata:

labels:

k8s-app: mics-app

name: misc-app

namespace: kong

spec:

replicas: 1

template:

metadata:

labels:

app: demoapp

spec:

containers:

- name: demoapp

image: 192.168.2.32:5000/testapp:v1.0

ports:

- containerPort: 8088

---

#------------------------springboot service------------------------------

kind: Service

apiVersion: v1

metadata:

labels:

k8s-app: mics-app

name: mics-app

namespace: kong

spec:

selector:

app: demoapp

ports:

- protocol: TCP

port: 8088

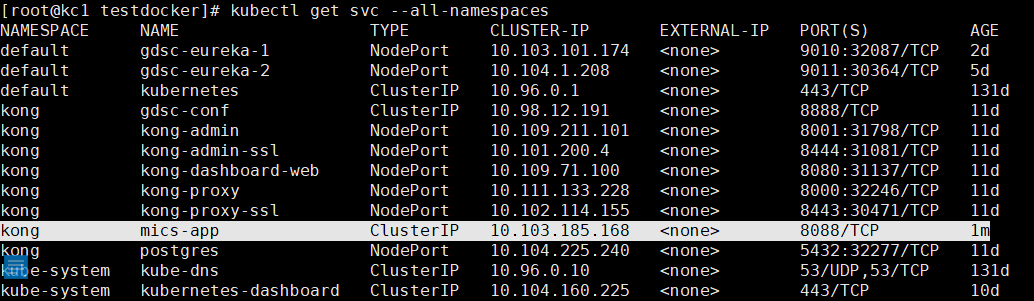
targetPort: 8088

**//9 部署K8集群**

$>kubectl create -f kube-springboot-app.yaml



**//10 查服务部署是否成功**



完成！