**Docker部署SpringBoot项目**

[雄关漫道从头越](https://www.jianshu.com/u/cd18446dfb5f)关注

0.9262019.01.20 02:46:03字数 390阅读 72,950

[使用docker部署SpringBoot项目](https://blog.csdn.net/junmoxi/article/details/80861199)  
[springboot整合docker部署（两种构建Docker镜像方式）](https://www.cnblogs.com/shamo89/p/9201513.html)

**1.创建springboot项目**

创建springboot项目

package com.eangulee.demo.controller;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

@Controller

public class HelloController {

@RequestMapping("/")

@ResponseBody

public String hello() {

return "Hello, SpringBoot With Docker";

}

}

**2.打包springboot项目为jar包**

**3. 编写Dockerfile文件**

# Docker image for springboot file run

# VERSION 0.0.1

# Author: eangulee

# 基础镜像使用java

FROM java:8

# 作者

MAINTAINER eangulee <eangulee@gmail.com>

# VOLUME 指定了临时文件目录为/tmp。

# 其效果是在主机 /var/lib/docker 目录下创建了一个临时文件，并链接到容器的/tmp

VOLUME /tmp

# 将jar包添加到容器中并更名为app.jar

ADD demo-0.0.1-SNAPSHOT.jar app.jar

# 运行jar包

RUN bash -c 'touch /app.jar'

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]

**解释下这个配置文件：**

VOLUME 指定了临时文件目录为/tmp。其效果是在主机 /var/lib/docker 目录下创建了一个临时文件，并链接到容器的/tmp。改步骤是可选的，如果涉及到文件系统的应用就很有必要了。/tmp目录用来持久化到 Docker 数据文件夹，因为 Spring Boot 使用的内嵌 Tomcat 容器默认使用/tmp作为工作目录  
项目的 jar 文件作为 “app.jar” 添加到容器的  
ENTRYPOINT 执行项目 app.jar。为了缩短 Tomcat 启动时间，添加一个系统属性指向 “/dev/./urandom” 作为 Entropy Source

如果是第一次打包，它会自动下载java 8的镜像作为基础镜像，以后再制作镜像的时候就不会再下载了。

**4. 部署文件**

在服务器新建一个docker文件夹，将maven打包好的jar包和Dockerfile文件复制到服务器的docker文件夹下



docker文件夹

**5. 制作镜像**

执行下面命令， 看好，最后面有个"."点！

docker build -t springbootdemo4docker .

docker build -t eurekasingle4docker .

docker build -f Dockerfile4eurekamaster -t eurekamaster4docker .

docker build -f Dockerfile4eurekabackup -t eurekabackup4docker .

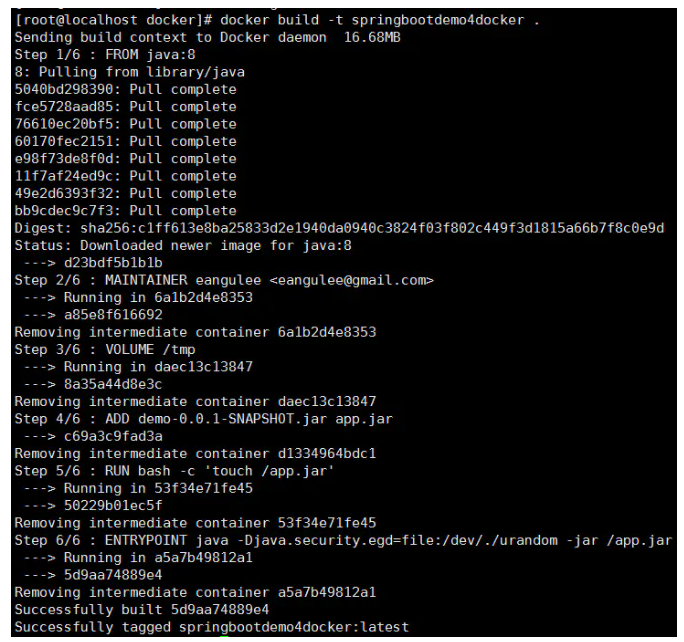
docker build -f Dockerfile4springboot-admin -t springboot-admin4docker .

docker build -f Dockerfile4papy -t papy4docker .

docker build -f Dockerfile4configprd -t configprd4docker .

docker build -f Dockerfile4zuul -t zuul4docker .

-t 参数是指定此镜像的tag名



制作完成后通过**docker images**命令查看我们制作的镜像

**6.启动容器**

[root@localhost docker]# docker run -d -p 8080:8085 springbootdemo4docker

-d参数是让容器后台运行

-p 是做端口映射，此时将服务器中的8080端口映射到容器中的8085(项目中端口配置的是8085)端口

**docker run -d -p 1157:7998 eurekamaster4docker**

**docker run --**name=eurekasingle **-d -p 1157:7998** eurekasingle4docker

**docker run --**name=eurekamaster **-d -p 7998:7998** eurekamaster4docker

#add-host要在前面

docker run --add-host=backupIP:127.0.0.1 --name=eurekamaster -d -p 7998:7998 eurekamaster4docker

**docker run** --add-host=masterIP:127.0.0.1 **--**name=eurekabackup **-d -p 7999:7999** eurekabackup4docker

**#有用的， hostname eureka服务显示的时候会显示 名称，name docker用的名字 networkd 后面接自定义（只有自定义才能用别名） network-alias 别名 -p主机才能用商品访问到（网络组内是可以看到，但主机查不到）**

**~~docker run --hostname master --restart=always --name=eurekamaster --network eureka\_net --network-alias masterIP -d -p 7998:7998 eurekamaster4docker~~**

**~~docker run --hostname backup --restart=always --name=eurekabackup --network eureka\_net --network-alias backupIP -d -p 7999:7999 eurekabackup4docker~~**

**~~docker run --hostname zuul --restart=always --name=zuul --network eureka\_net --network-alias zuulIP -d -p 7680:7680 zuul4docker~~**

**~~docker run --hostname springboot-admin --restart=always --name=springboot-admin --network eureka\_net --network-alias springboot-adminIP -d -p 1002:1002 springboot-admin4docker~~**

**docker run --name nginx --restart=always --network eureka\_net --network-alias nginxIP -p 1157:1157 -v /usr/local/docker/nginx.conf:/etc/nginx/nginx.conf -d nginx**

**docker run --name nginx --restart=always --network eureka\_net --network-alias springboot-adminIP -p 1157:1157 -v /usr/local/docker/nginx.conf:/etc/nginx/nginx.conf -v /usr/local/docker/nginxtcp.conf:/etc/nginx/tcp.d/nginxtcp.conf -d nginx**

**docker run --name nginxtest --restart=always --network eureka\_net --network-alias springboot-adminIP -p 1157:1157 -p 9000:9000 -v /usr/local/docker/nginx.conf:/etc/nginx/nginx.conf -v /usr/local/docker/nginxtcp.conf:/etc/nginx/tcp.d/nginxtcp.conf -d nginx**

**~~docker run --hostname configprd --restart=always --name=configprd --network eureka\_net --network-alias configprdIP -d -p 7897:~~** ~~7897~~ **~~configprd4docker~~**

**~~docker run --hostname papy --restart=always --name=papy --network eureka\_net --network-alias papyIP -d -p 801:801 papy4docker~~**

**~~docker run --name=testmysql --network eureka\_net --network-alias testmysqlIP -d -p 33601:33601 mysql4docker~~**

**~~docker run --name=mysql --restart=always --network eureka\_net --network-alias mysqlIP -d -p 33061:33061 -e MYSQL\_ROOT\_PASSWORD=wosiboy2 mysql:5.7~~**

**docker run --name=mysqlbase --restart=always --network eureka\_net --network-alias mysqlIP -d -p 33061:33061 -v /data/mysql:/var/lib/mysql -v /usr/local/docker/my.cnf:/etc/my.cnf -e MYSQL\_ROOT\_PASSWORD=wosiboy2 mysql**

**--rm 前端运行参数 和 restart冲突**

**~~docker run --name jenkinsci-blueocean --network eureka\_net --network-alias jenkIP -u root --rm -d -p 7005:8080 -p 50000:50000 -v /data/jenkins:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkinsci/blueocean~~**

**~~docker run --name jenkinsci-blueocean --restart=always --network eureka\_net --network-alias jenkIP -u root -d -p 7005:8080 -p 50000:50000 -v /data/jenkins:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkinsci/blueocean~~**

**-----jenkins使用docker-compose 运行【1。修改WEB的目录 增加/项目访问，2去掉**CSRF】 目录文件 /usr/local/docker/docker-compose-jenkins

--- docker stop jenkinsci-blueoceantest

--p 多个文件的时候要使用P来区别名称吧

docker-compose stop #停止运行的service

**docker-compose rm -f [serviceName]**：删除已停止的所有service

**docker-compose build**：构建/重新构建所有镜像（images）

**docker-compose start [serviceName]**：启动已存在但停止的所有窗口

docker-compose up -d

# Pull image from Docker Hub.

docker pull gogs/gogs

# Create local directory for volume.

mkdir -p /var/gogs

# Use `docker run` for the first time.

**docker run --restart=always --network eureka\_net --network-alias gogsIP -d --name=gogs -p 10022:22 -p 10080:3000 -v /data/gogs:/data gogs/gogs**

**7. 访问网站**

直接浏览器访问： http://你的服务器ip地址:8080/