

使用Maven插件构建Docker镜像

原创 梦想de星空 macrozheng 2019-06-12 20:03

收录于合集

#mall学习教程（参考篇）

27个

本文主要介绍如何使用Maven插件将SpringBoot应用打包为Docker镜像，并上传到私有镜像仓库 Docker Registry的过程。

Docker Registry

Docker Registry 2.0搭建

```
docker run -d -p 5000:5000 --restart=always --name registry2 registry:2
```

如果遇到镜像下载不下来的情况，需要修改 /etc/docker/daemon.json 文件并添加上 registry-mirrors 键值，然后重启docker服务：

```
{  
  "registry-mirrors": ["https://registry.docker-cn.com"]  
}
```

Docker开启远程API

用vim编辑器修改docker.service文件

```
vi /usr/lib/systemd/system/docker.service
```

需要修改的部分：

```
ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

修改后的部分：

```
ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock
```

让Docker支持http上传镜像

```
echo '{ "insecure-registries":["192.168.3.101:5000"] }' > /etc/docker/daemon.json
```

重新启动Docker服务

```
systemctl stop docker  
systemctl start docker
```

开启防火墙的Docker构建端口

```
firewall-cmd --zone=public --add-port=2375/tcp --permanent  
firewall-cmd --reload
```

使用Maven构建Docker镜像

该代码是在mall-tiny-02的基础上修改的。

在应用的pom.xml文件中添加docker-maven-plugin的依赖

```
<plugin>  
  <groupId>com.spotify</groupId>  
  <artifactId>docker-maven-plugin</artifactId>  
  <version>1.1.0</version>  
  <executions>  
    <execution>  
      <id>build-image</id>  
      <phase>package</phase>  
      <goals>  
        <goal>build</goal>  
      </goals>  
    </execution>  
  </executions>  
  <configuration>  
    <imageName>mall-tiny/${project.artifactId}:${project.version}</imageName>  
    <dockerHost>http://192.168.3.101:2375</dockerHost>  
    <baseImage>java:8</baseImage>  
    <entryPoint>["java", "-jar", "/${project.build.finalName}.jar"]  
    </entryPoint>  
    <resources>  
      <resource>  
        <targetPath></targetPath>  
        <directory>${project.build.directory}</directory>  
        <include>${project.build.finalName}.jar</include>  
      </resource>  
    </resources>  
  </configuration>  
</plugin>
```

相关配置说明：

- `executions.execution.phase`: 此处配置了在maven打包应用时构建docker镜像;
- `imageName`: 用于指定镜像名称, `mall-tiny`是仓库名称, `${project.artifactId}`为镜像名称, `${project.version}`为镜像版本号;
- `dockerHost`: 打包后上传到的docker服务器地址;
- `baseImage`: 该应用所依赖的基础镜像, 此处为java;
- `entryPoint`: docker容器启动时执行的命令;
- `resources.resource.targetPath`: 将打包后的资源文件复制到该目录;
- `resources.resource.directory`: 需要复制的文件所在目录, maven打包的应用jar包保存在target目录下;
- `resources.resource.include`: 需要复制的文件, 打包好的应用jar包。

修改application.yml, 将localhost改为db

可以把docker中的容器看作独立的虚拟机, `mall-tiny-docker`访问localhost自然会访问不到mysql, docker容器之间可以通过指定好的服务名称db进行访问, 至于db这个名称可以在运行`mall-tiny-docker`容器的时候指定。

```
spring:
  datasource:
    url: jdbc:mysql://db:3306/mall?useUnicode=true&characterEncoding=utf-8&serverTimezone=
    username: root
    password: root
```

使用IDEA打包项目并构建镜像

注意: 依赖的基础镜像需要先行下载, 否则会出现构建镜像超时的情况, 比如我本地并没有java8的镜像, 就需要先把镜像pull下来, 再用maven插件进行构建。

- 执行maven的package命令:



- 构建成功:



- 镜像仓库已有该镜像:



运行mall-tiny-docker项目

启动mysql服务

- 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
```

- 进入运行mysql的docker容器：

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

- 使用mysql命令打开客户端：

```
mysql -uroot -proot
```



- 修改root帐号的权限，使得任何ip都能访问：

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



- 创建mall数据库：

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

- 将mall.sql文件拷贝到mysql容器的/目录下：

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

- 将sql文件导入到数据库：

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



启动mall-tiny-docker应用服务

- 使用docker命令启动（--link表示应用可以用db这个域名访问mysql服务）：

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



- 开启8080端口：

```
firewall-cmd --zone=public --add-port=8080/tcp --permanent
firewall-cmd --reload
```

- 进行访问测试，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>

推荐阅读

- [开发者必备Docker命令](#)
- [开发者必备Linux命令](#)
- [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
还在百度Docker命令？推荐一套我用起来特顺手的命令！

下一篇
使用Dockerfile为SpringBoot应用构建Docker镜像

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

