用Docker发布SpringBoot应用

1 关于Docker

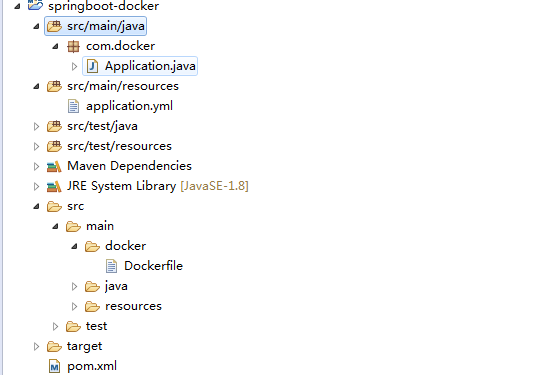
这个…… 关于Docker我就不说了，问问度娘吧。或者下载[《Docker技术入门与实战》](http://product.dangdang.com/23620853.html" \t "http://www.java1234.com/a/javabook/javabase/2016/0628/_blank)研究研究也是可以的https://pan.baidu.com/s/1bprrGD5

1. 关于SpringBoot

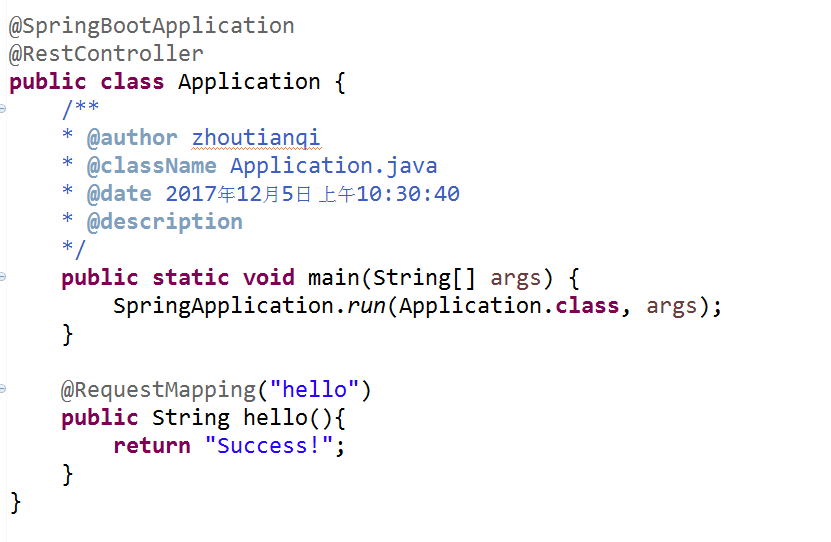
也不做赘述，推荐大家关注http://www.spring4all.com/

1. 言归正传

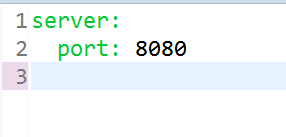
Eclipse新建Maven项目，结构如图。



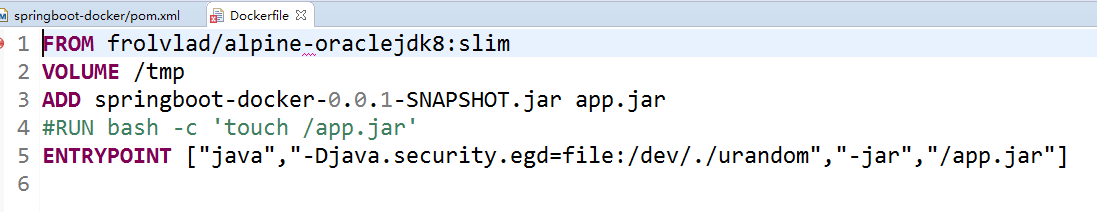
SpringBoot启动入口和控制层



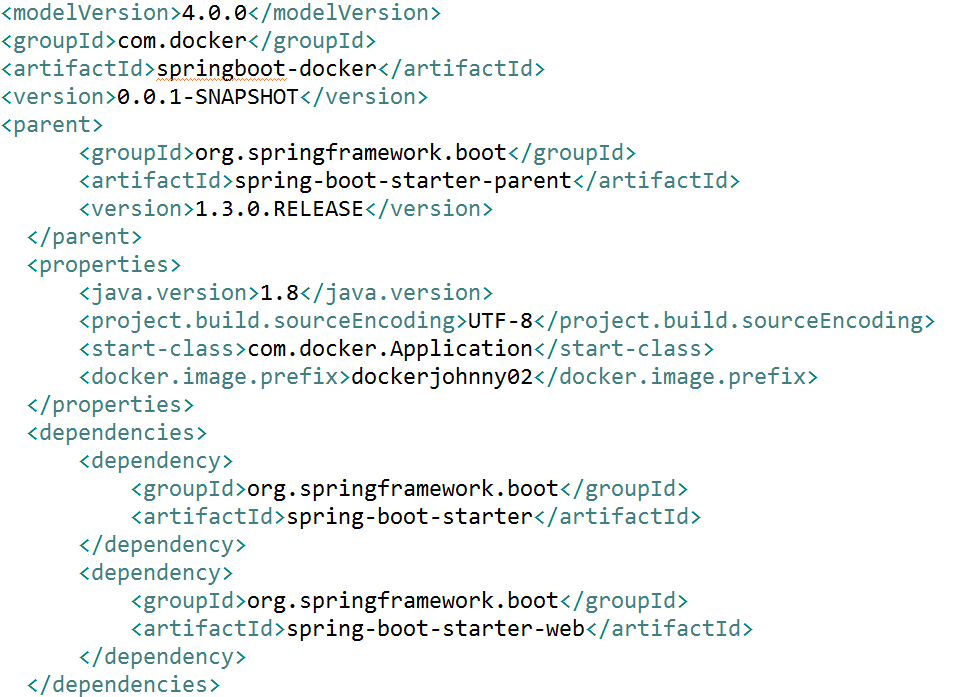
Yml配置文件

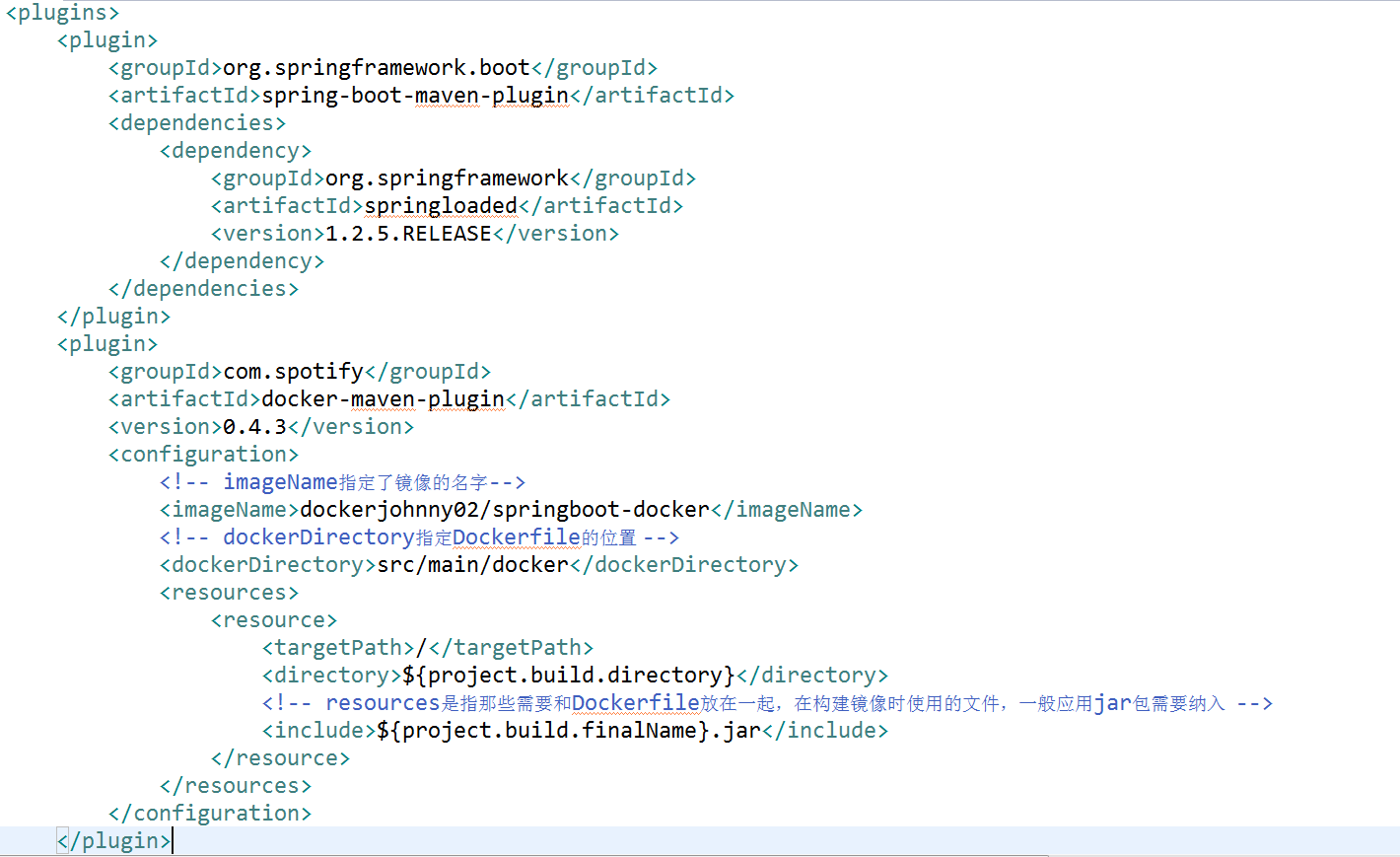


Dockerfile



Pom依赖





实际上与我们通常建的boot工程差不多，区别点有两个

A:增加了dockerfile文件 -> 用来创建一个自定义的镜像，包含了用户指定的软件依赖

# B:pom文件新增了docker插件docker-maven-plugin -> 将项目编译为docker镜像

1. dockerfile语法命令推荐一篇文章很不错

http://www.docker.org.cn/dockerppt/114.html

1. POM插件

Spring Boot Maven plugin 提供了很多方便的功能：

它收集的类路径上所有 jar 文件，并构建成一个单一的、可运行的jar，这使得它更方便地执行和传输服务。它搜索的 public static void main() 方法来标记为可运行的类。提供了一个内置的依赖解析器，用于设置版本号以匹配 Spring Boot 的依赖。您可以覆盖任何你想要的版本，但它会默认选择的 Boot 的版本集。

Spotify 的 docker-maven-plugin 插件：

用于构建 Maven 的 Docker Image。

imageName指定了镜像的名字

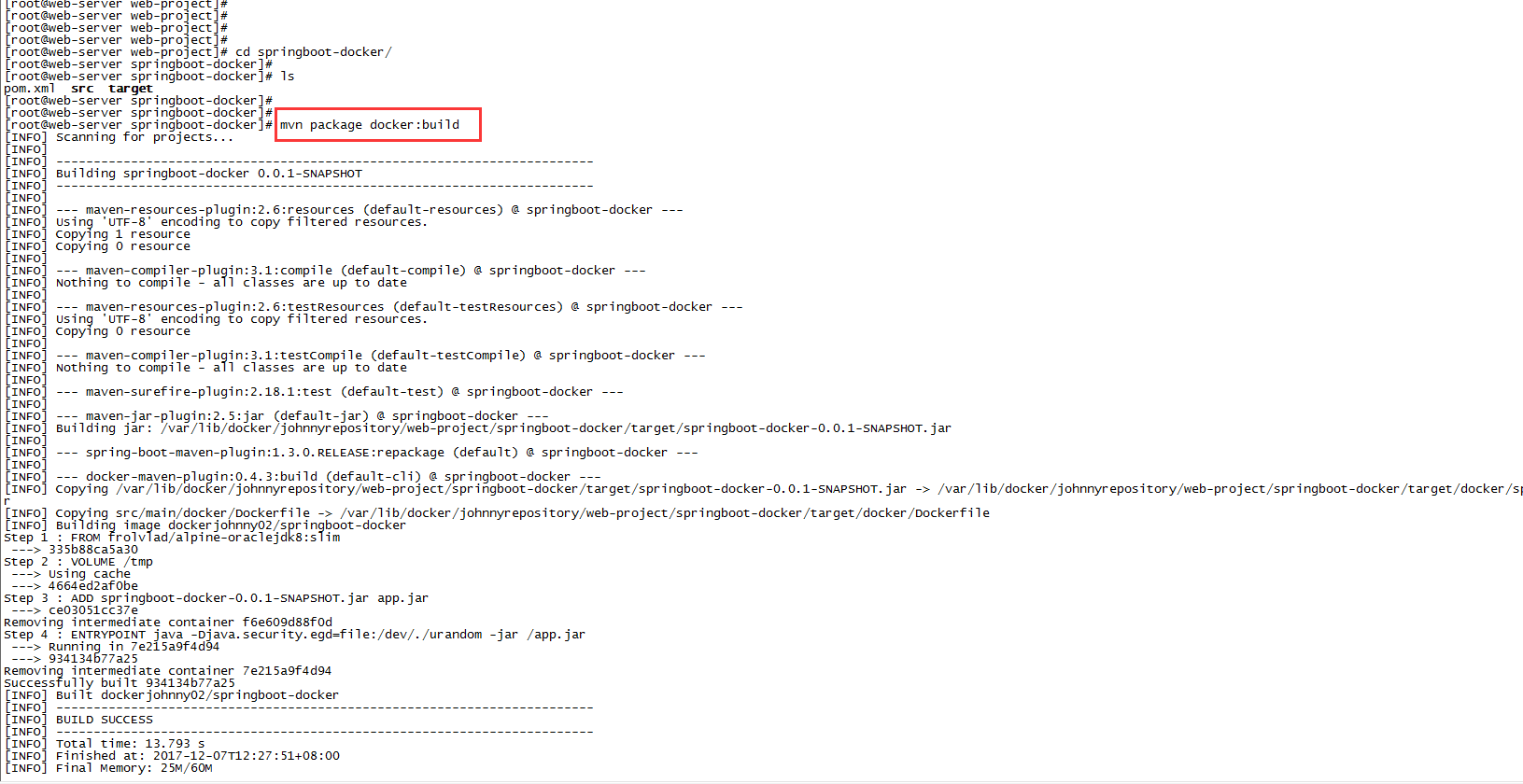
dockerDirectory指定 Dockerfile 的位置

resources是指那些需要和 Dockerfile 放在一起，在构建镜像时使用的文件，一般应用 jar 包需要纳入。本例，只需一个 jar 文件

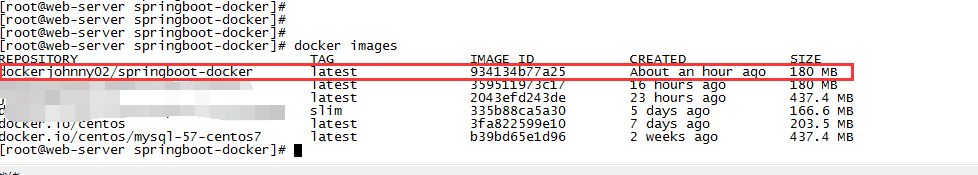
6.linux环境新建目录web\_project，将eclipse工程拉入该目录





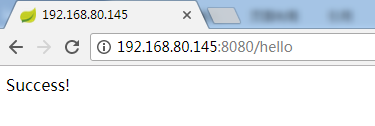


查看镜像库，可以看到自定义镜像已经躺在里面了。当然你还可以持久化（docker save [镜像ID] /var/../saveed.rar）该镜像,在其他地方加载（docker load /var/../saveed.rar）

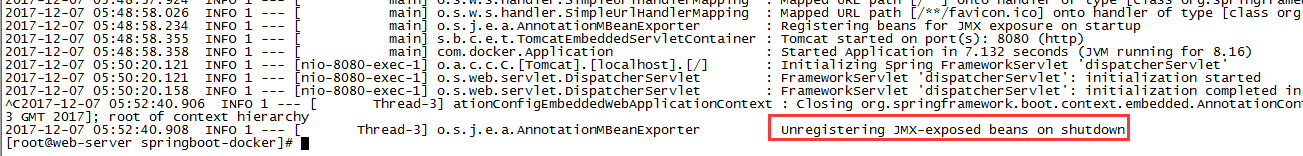


然后呢，在容器里面跑起来

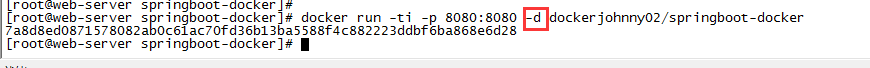




但是呢，有个问题，linux环境一旦退出，应用就会挂掉

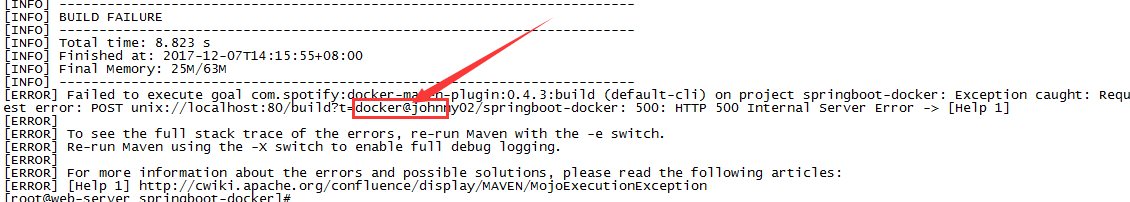


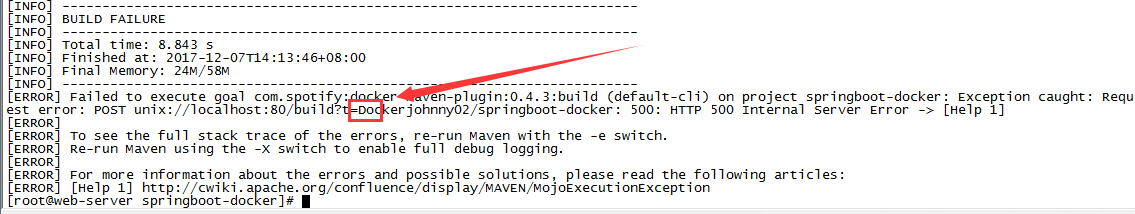
以守护进程挂在后台运行，这样你就可以浪起来了



坑点：

1镜像名<imageName>dockerjohnny02/springboot-docker</imageName>避免用特殊字符、大写什么的，否则会构建异常





参考：

<http://spring.io/guides/gs/spring-boot-docker/>

https://yq.aliyun.com/articles/47344