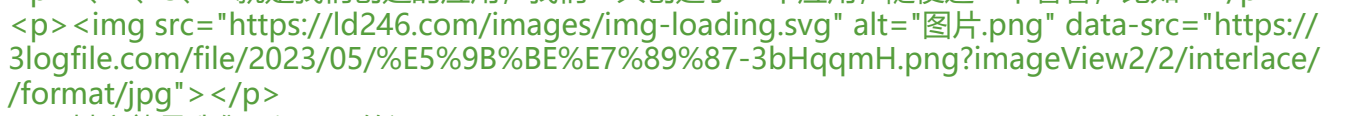
Spug 使用 Docker 发布 spring boot 项目

作者：luofeng0603
原文链接：https://ld246.com/article/1683621215907
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20190608.jpg?imageView2/1/w/768/h/432/interlace/1/q/100">
<h2 id="前言">前言</h2>
最近在使用 spug 发布项目，发布了一个 Spring Boot 项目，遇到各种坑，记录一下。
<h2 id="关于spug">关于 spug</h2>
spug 可以用来方便的管理项目的发布，我也是第一次用，官网地址：<a href="https://ld246.c
m/forward?goto=https%3A%2F%2Fspug.cc" target="_blank" rel="nofollow ugc">https://spu
.cc
我们使用的 spug 是自定的镜像，里面包含了 jdk17 和 node16，因为项目要用到。
使用如下命令创建的 spug：
<pre><code class="language-bash highlight-chroma"><span c
ass="highlight-cl">docker run -d \
<span cla
s="highlight-se">--restart=unless-stopped <sp
n class="highlight-se">\
<span cla
s="highlight-se">-e MYSQL_DATABASE<span c
ass="highlight-o">=spug \
<span cla
s="highlight-se">-e MYSQL_USER<span class=
highlight-o">=root \
<span cla
s="highlight-se">-e MYSQL_PASSWORD<span
class="highlight-o">=密码 \
<span cla
s="highlight-se">-e MYSQL_PORT<span class=
highlight-o">=3306 -e <span class="highlight-
v">MYSQL_HOST=127.0.0.1 <span class="highli
ht-se">\
<span cla
s="highlight-se">-v /var/lib/spug/data:/data \
<span cla
s="highlight-se">--name spug_jdk17 \
<span cla
s="highlight-se">-p 7766:80 \
<span cla
s="highlight-se">luofeng/spug:1.0.1
</code></pre>
看下 spug 容器里面的/data 目录：
<pre><code class="language-bash highlight-chroma"><span c
ass="highlight-cl">[root@f799cddfa826 data<span clas
="highlight-o">]# ls
mysql repos spu

<span class="high
ight-o">[root@f799cddfa826 data]<span class=
highlight-c1"># cd repos
<span class="high
ight-o">[root@f799cddfa826 repos]<span class
"highlight-c1"># ls
<span class="high
ight-m">1 2 3</s
an> 4 build
<span class="high

ight-o">[root@f799cddfa826 repos]<span class
"highlight-c1">#
</code></pre>
1、2、3、4 就是我们创建的应用，我们一共创建了 4 个应用，随便进一个看看，比如 2

其实就是我们 git 项目的源码。
<h2 id="设置Spring-Boot项目">设置 Spring Boot 项目</h2>
接下来进行 spring boot 项目的设置：
<h2 id="编写Dockerfile">编写 Dockerfile</h2>
<pre><code class="language-bash highlight-chroma"><span c
ass="highlight-cl"># 构建镜像
FROM eclipse-te
urin:17-jre
MAINTAINER luof
ng0603
WORKDIR /app
ADD target/*.jar a
p.jar
EXPOSE <span cla
s="highlight-m">8080

<span class="high
ight-c1"># 设置时区
RUN ln -sf /usr/sh
re/zoneinfo/Asia/Shanghai /etc/localtime
RUN <span class=
highlight-nb">echo 'Asia/Shanghai' >/etc/t
mezone

ENTRYPOINT <sp
n class="highlight-o">["sh",<span class="highl
ght-s2">"-c","java -jar app.jar --spring.profiles.active=</
pan>$SPRING_PROFILES_ACTIVE<span class="highlight
s2">"]
</code></pre>
使用 --spring.profiles.active=$SPRING_PROFILES_ACTIVE" 让 docker run 的时候可以接受
境变量 SPRING_PROFILES_ACTIVE 就可以自由切换环境配置了。
<h2 id="修改Pom-xml">修改 Pom.xml</h2>
pom 文件需要修改一下，新增 dockerfile-maven-plugin 插件
具体如下：
<pre><code class="language-xml highlight-chroma"><span cl
ss="highlight-cl"><plugin>
    <span class="h
ghlight-nt"><groupId>com.spotify</groupId&gt

    <span class="h
ghlight-nt"><artifactId>dockerfile-maven-plugin&l
;/artifactId>
    <span class="h
ghlight-nt"><version>1.4.13</version></spa
>
    <span class="h

ghlight-nt"><executions>
        <span class=
highlight-nt"><execution>
            <span clas
="highlight-nt"><id>default</id>
            <span clas
="highlight-nt"><goals>
                <span c
ass="highlight-nt"><goal>build</goal></sp
n>
            <span clas
="highlight-nt"></goals>
        <span class=
highlight-nt"></execution>
    <span class="h
ghlight-nt"></executions>
    <span class="h
ghlight-nt"><configuration>
        <span class=
highlight-nt"><repository>zgyx/${project.artifactId}<span class="highlight-nt"
</repository>
        <span class=
highlight-nt"><tag>latest</tag>
        <span class=
highlight-nt"><buildArgs>
            <span clas
="highlight-nt"><JAR_FILE>${project.build.finalName}.jar<span class="highligh
-nt"></JAR_FILE>
        <span class=
highlight-nt"></buildArgs>
    <span class="h
ghlight-nt"></configuration>
<span class="high
ight-nt"></plugin>
</code></pre>
我在使用阿里云仓库的时候发现 这个插件无法下载，换成中央仓库就好了~，下完再切换回阿里仓库。
注意此时 pom.xml 和 Dockerfile 在同一个目录。
<h2 id="设置spug">设置 spug</h2>
<h2 id="新建应用">新建应用</h2>
<img src="https://ld246.com/images/img-loading.svg" width="60%" data-src="https://b3logf
le.com/file/2023/05/图片-Tr7GUGY.png?imageView2/2/interlace/1/format/jpg">
这个比较简单，随便填写一下。
<h2 id="新建发布">新建发布</h2>
<img src="https://ld246.com/images/img-loading.svg" alt="图片.png" data-src="https://
3logfile.com/file/2023/05/%E5%9B%BE%E7%89%87-kkRYh3D.png?imageView2/2/interlace/1
format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="图片.png" data-src="https://
3logfile.com/file/2023/05/%E5%9B%BE%E7%89%87-r68ViFL.png?imageView2/2/interlace/1/f
rmat/jpg">
选择一下发布环境、目标主机（可以选多台）、Git 仓库地址，发布环境和目标主机如果没有的，先创建一下。
<img src="https://ld246.com/images/img-loading.svg" alt="图片.png" data-src="https://
3logfile.com/file/2023/05/%E5%9B%BE%E7%89%87-HTpzqSO.png?imageView2/2/interlace/1

format/jpg">
文件过滤是我们需要将哪些文件发布到目标主机上，有包含和排除两种模式，我上面选的是包含式，也就是说只有指定目录下的 jar 包和 Dockerfile 才是我需要的，其他文件对我来说用不着。
然后下面有两块区域分别是：代码检出钱执行和代码检出后执行，检出就是 git 中的 checkout 翻译的有点拗口。这里设置的命令是在 spug 容器执行的。那么上面的意思就是我希望 git 仓库代码下来之后，执行 mvn clean install -Dmaven.test.skip=true 命令，就是执行 maven 构建。因为我安装了 dockerfile-maven 插件，所以 maven 会帮我们执行 docker 镜像构建，DOCKER_HOST 指的是我们的目标服务器，意思就是在我们指定的目标服务器上执行 docker build 命令构建 docker 像。
<img src="https://ld246.com/images/img-loading.svg" alt="图片.png" data-src="https://3logfile.com/file/2023/05/%E5%9B%BE%E7%89%87-Kc58Wdj.png?imageView2/2/interlace/1/ormat/jpg">
部署路径指的是目标服务器的发布目录，这个发布目录里面只会包含我们在前面指定的满足过滤件的文件。
存储路径可以随便指定，里面放的是历史版本。
应用发布前和应用发布后两块内容，指的是在目标服务器上执行，我们在应用发布后，执行了：/p>
docker rm -f xxxx
docker run -d --name xxxx -p 8080:8080 -e SPRING_PROFILES_ACTIVE=test 镜像
就是先删除老的镜像，再启动新的镜像。
然后点击提交保存，基本上完成 99% 了。
<h2 id="测试发布">测试发布</h2>
接下来就是测试发布了，在发布申请中新建一个发布：
选择环境：


根据实际情况填写一下
<img src="https://ld246.com/images/img-loading.svg" alt="图片.png" data-src="https://3logfile.com/file/2023/05/%E5%9B%BE%E7%89%87-OVQ3bOq.png?imageView2/2/interlace//format/jpg">
点击发布：
<img src="https://ld246.com/images/img-loading.svg" alt="图片.png" data-src="https://3logfile.com/file/2023/05/%E5%9B%BE%E7%89%87-tw1OaW1.png?imageView2/2/interlace//format/jpg">
如果发生异常，仔细看一下日志，再做修复，修复后再测~
以上就是 Spug 使用 Docker 发布 Spring Boot 项目的大致流程~