

当Docker遇到IntelliJ IDEA，再次解放了生产力~

ImportNew 昨天

(给ImportNew加星标，提高Java技能)

转自：陶章好

链接：<https://juejin.im/post/5d026212f265da1b8608828b>

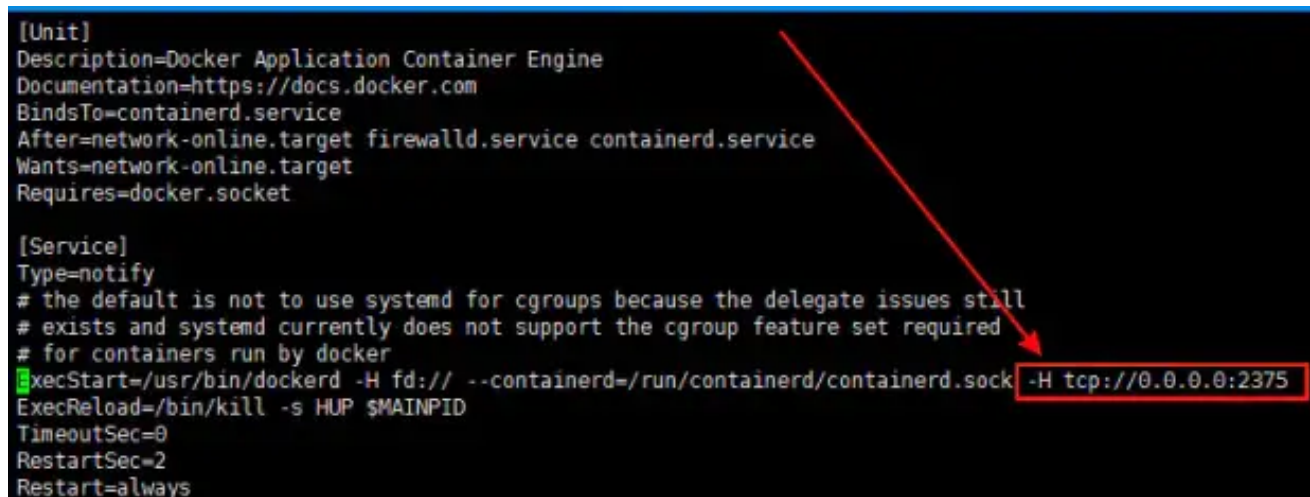
Idea是Java开发利器，SpringBoot是Java生态中最流行的微服务框架，docker是时下最火的容器技术，那么它们结合在一起会产生什么化学反应呢？

一、开发前准备

1. Docker的安装可以参考<https://docs.docker.com/install/>
2. 配置docker远程连接端口

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

找到 ExecStart，在最后面添加 -H tcp://0.0.0.0:2375，如下图所示



```
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
BindsTo=containerd.service
After=network-online.target firewalld.service containerd.service
Wants=network-online.target
Requires=docker.socket

[Service]
Type=notify
# the default is not to use systemd for cgroups because the delegate issues still
# exists and systemd currently does not support the cgroup feature set required
# for containers run by docker
ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock -H tcp://0.0.0.0:2375
ExecReload=/bin/kill -s HUP $MAINPID
TimeoutSec=0
RestartSec=2
Restart=always
```

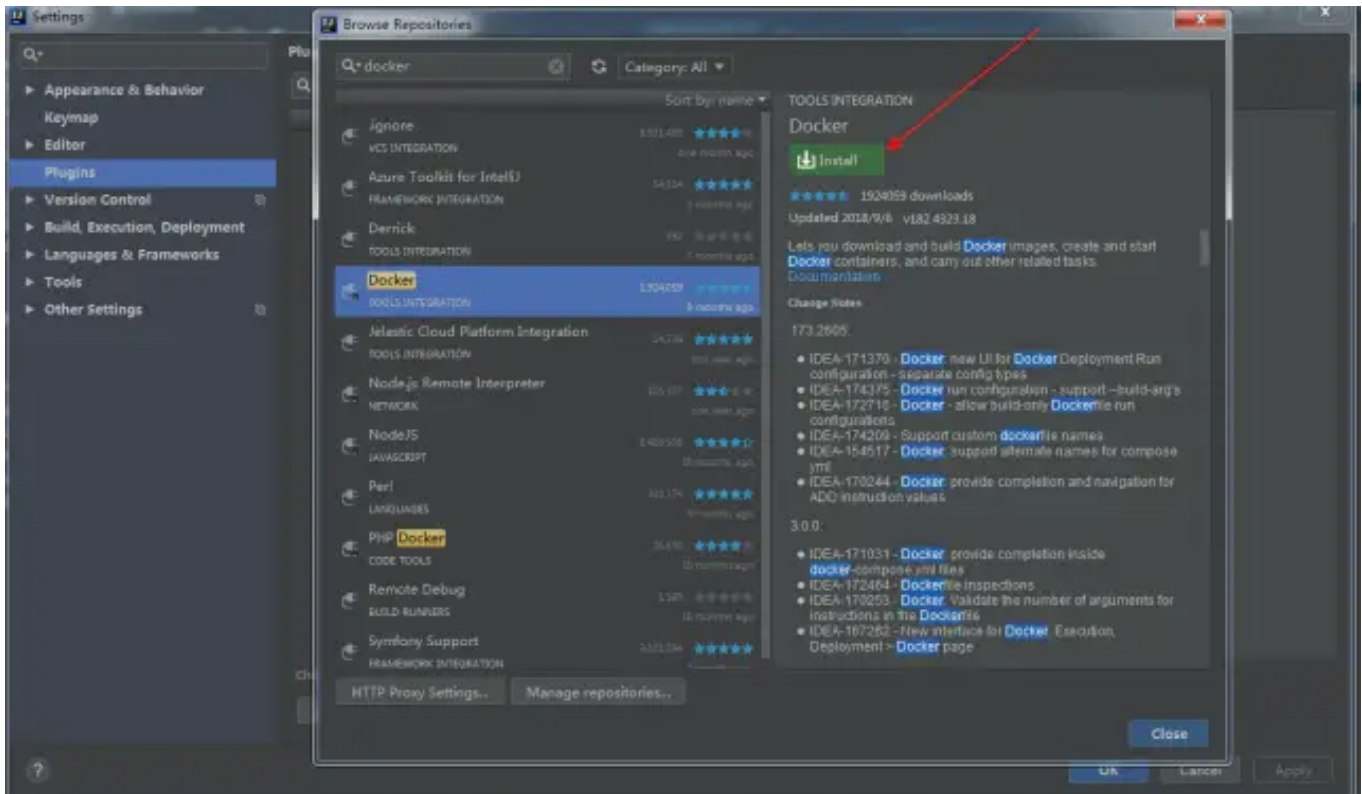
3. 重启docker

```
1 systemctl daemon-reload
2 systemctl start docker
```

4. 开放端口

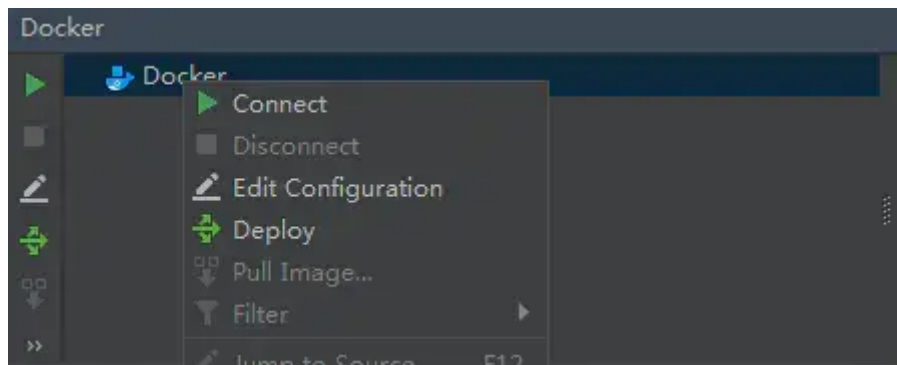
```
1 firewall-cmd --zone=public --add-port=2375/tcp --permanent
```

5. Idea安装插件,重启

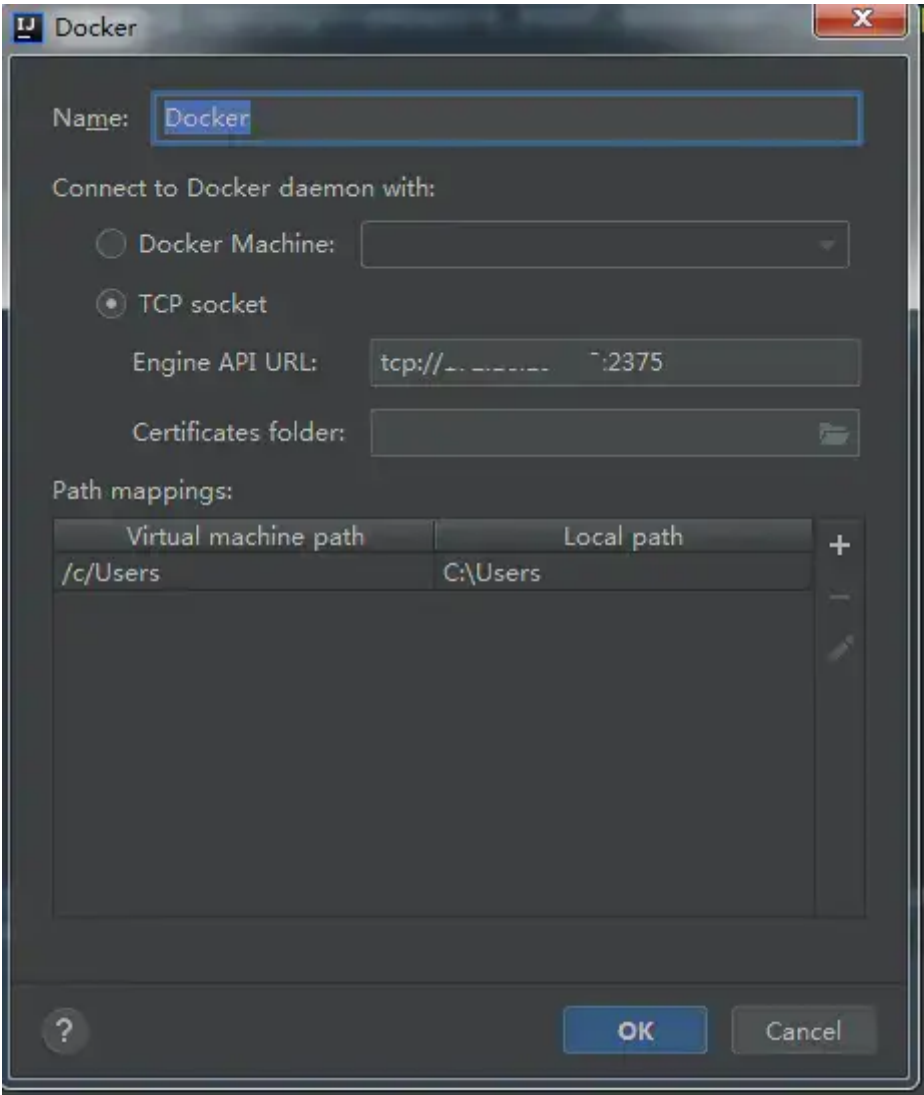


6. 连接远程docker

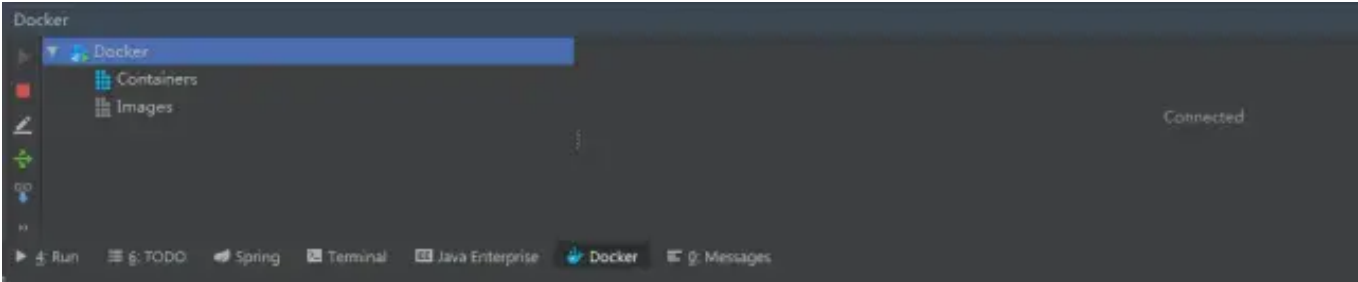
(1) 编辑配置



(2) 填远程docker地址



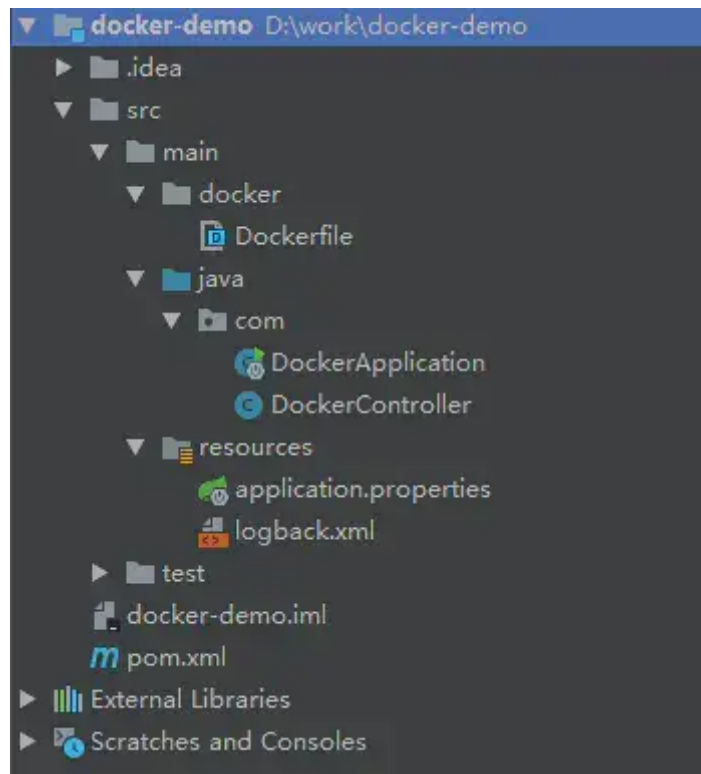
(3) 连接成功，会列出远程docker容器和镜像



二、新建项目

1. 创建springboot项目

项目结构图



(1) 配置pom文件

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>docker-demo</groupId>
8     <artifactId>com.demo</artifactId>
9     <version>1.0-SNAPSHOT</version>
10    <parent>
11        <groupId>org.springframework.boot</groupId>
12        <artifactId>spring-boot-starter-parent</artifactId>
13        <version>2.0.2.RELEASE</version>
14        <relativePath />
15    </parent>
16
17    <properties>
18        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
19        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
20        <docker.image.prefix>com.demo</docker.image.prefix>
21        <java.version>1.8</java.version>
22    </properties>
```

```
23     <build>
24         <plugins>
25             <plugin>
26                 <groupId>org.springframework.boot</groupId>
27                 <artifactId>spring-boot-maven-plugin</artifactId>
28             </plugin>
29             <plugin>
30                 <groupId>com.spotify</groupId>
31                 <artifactId>docker-maven-plugin</artifactId>
32                 <version>1.0.0</version>
33                 <configuration>
34                     <dockerDirectory>src/main/docker</dockerDirectory>
35                     <resources>
36                         <resource>
37                             <targetPath>/</targetPath>
38                             <directory>${project.build.directory}</directory>
39                             <include>${project.build.finalName}.jar</include>
40                         </resource>
41                     </resources>
42                 </configuration>
43             </plugin>
44             <plugin>
45                 <artifactId>maven-antrun-plugin</artifactId>
46                 <executions>
47                     <execution>
48                         <phase>package</phase>
49                         <configuration>
50                             <tasks>
51                                 <copy todir="src/main/docker" file="target/${proj
52                             </tasks>
53                         </configuration>
54                     <goals>
55                         <goal>run</goal>
56                     </goals>
57                 </execution>
58             </executions>
59         </plugin>
60     </plugins>
61 </build>
```

```

63 <dependencies>
64     <dependency>
65         <groupId>org.springframework.boot</groupId>
66         <artifactId>spring-boot-starter-web</artifactId>
67     </dependency>
68     <dependency>
69         <groupId>org.springframework.boot</groupId>
70         <artifactId>spring-boot-starter-test</artifactId>
71         <scope>test</scope>
72     </dependency>
73     <dependency>
74         <groupId>log4j</groupId>
75         <artifactId>log4j</artifactId>
76         <version>1.2.17</version>
77     </dependency>
78 </dependencies>
79 </project>

```

(2) 在src/main目录下创建docker目录, 并创建Dockerfile文件

```

1 FROM openjdk:8-jdk-alpine
2 ADD *.jar app.jar
3 ENTRYPOINT ["java", "-Djava.security.egd=file:/dev/./urandom", "-jar", "/app.jar"]

```

(3) 在resource目录下创建application.properties文件

```

1 logging.config=classpath:logback.xml
2 logging.path=/home/developer/app/logs/
3 server.port=8990

```

(4) 创建DockerApplication文件

```

1 @SpringBootApplication

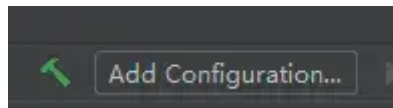
```

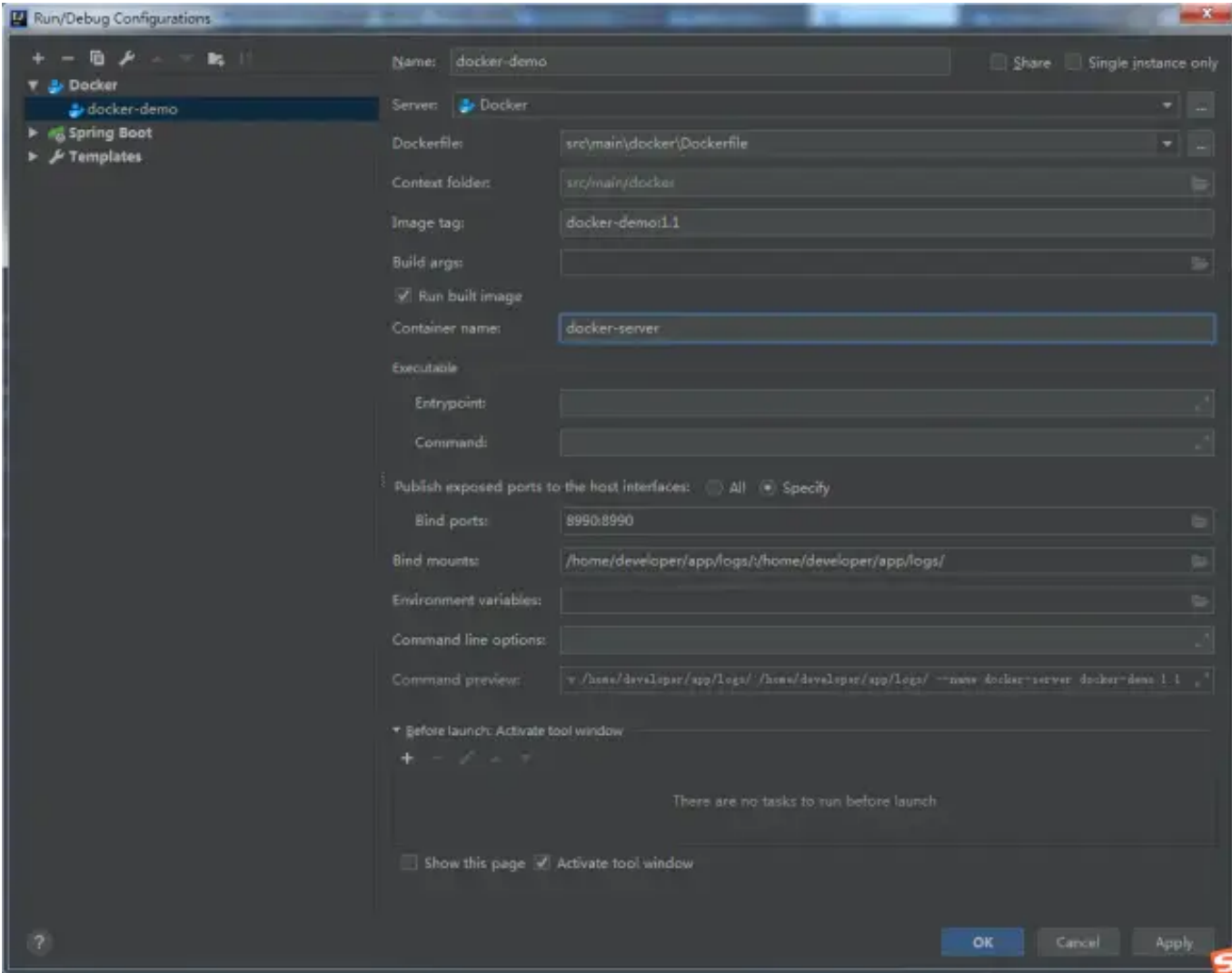
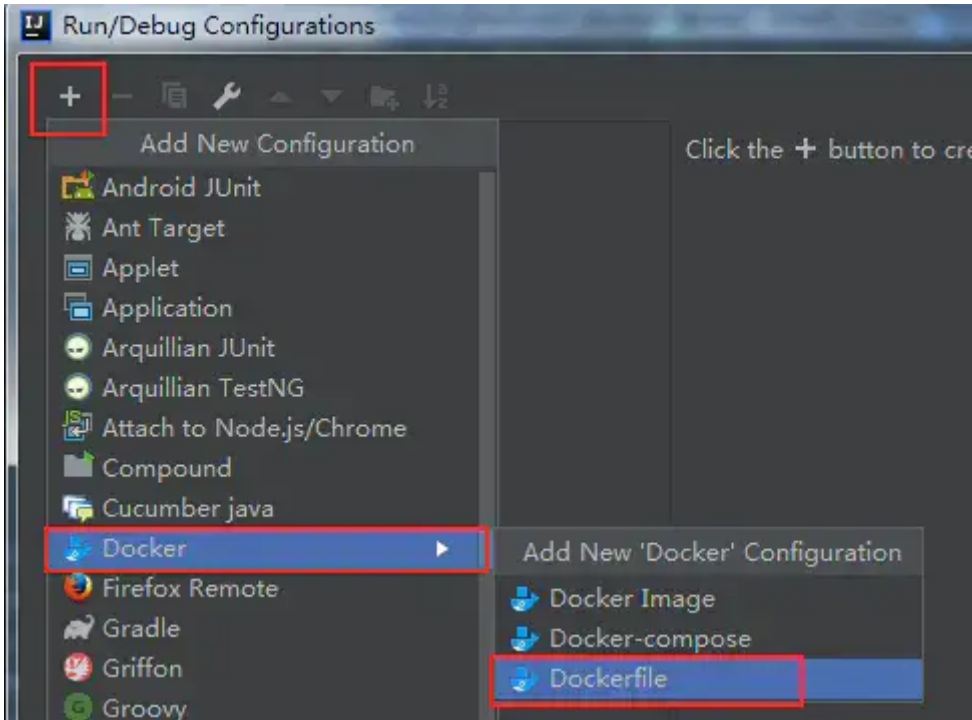
```
2 public class DockerApplication {  
3     public static void main(String[] args) {  
4         SpringApplication.run(DockerApplication.class, args);  
5     }  
6 }
```

(5) 创建DockerController文件

```
1 @RestController  
2 public class DockerController {  
3     static Log log = LoggerFactory.getLog(DockerController.class);  
4  
5     @RequestMapping("/")  
6     public String index() {  
7         log.info("Hello Docker!");  
8         return "Hello Docker!";  
9     }  
10 }
```

(6) 增加配置





命令解释

Image tag : 指定镜像名称和tag, 镜像名称为 docker-demo, tag为1.1

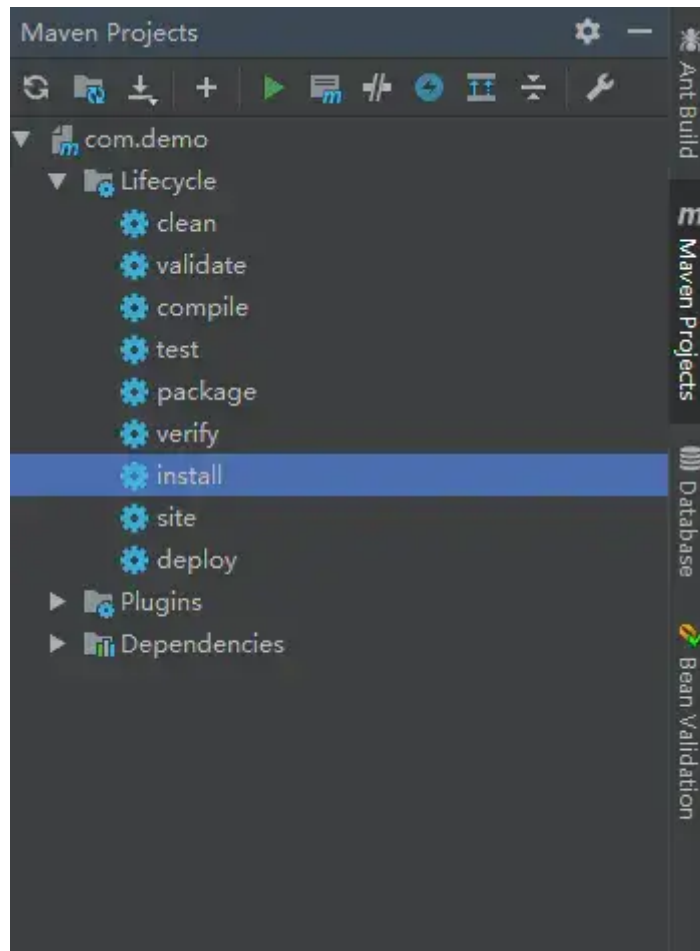
Bind ports : 绑定宿主机端口到容器内部端口。格式为[宿主机端口]:[容器内部端口]

Bind mounts : 将宿主机目录挂到容器内部目录中。格式为[宿主机目录]:[容器内部目录]。

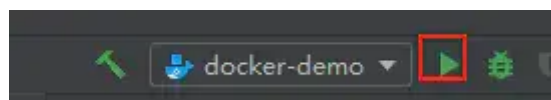
这个springboot项目会将日志打印在容器

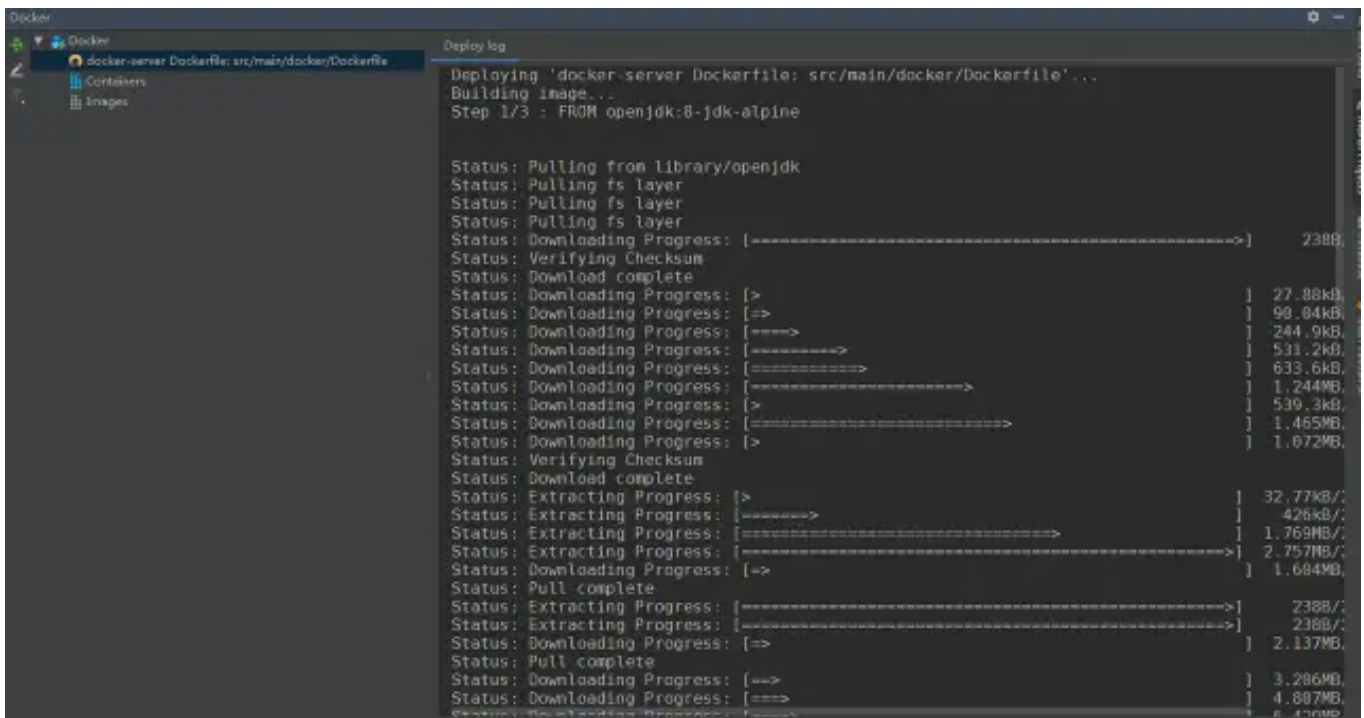
/home/developer/app/logs/ 目录下, 将宿主机目录挂载到容器内部目录后, 那么日志就会持久化容器外部的宿主机目录中。

(7) Maven打包

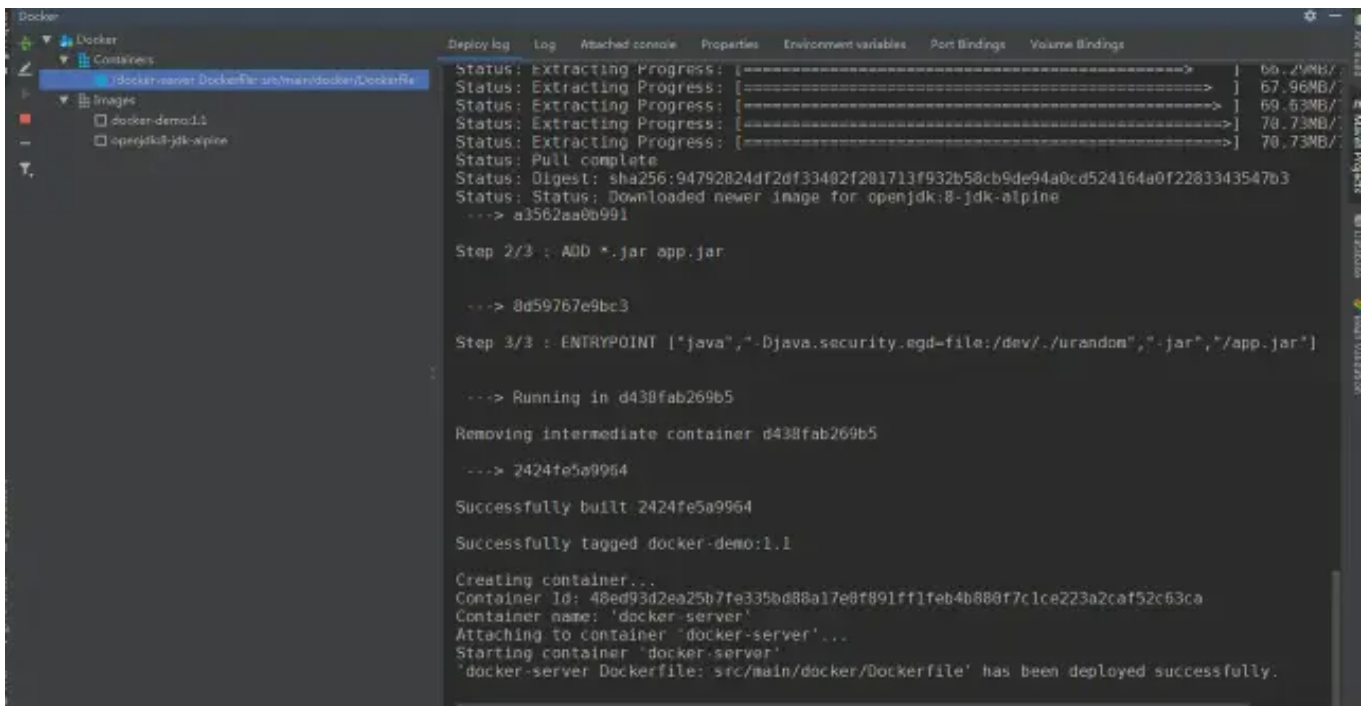


(8) 运行



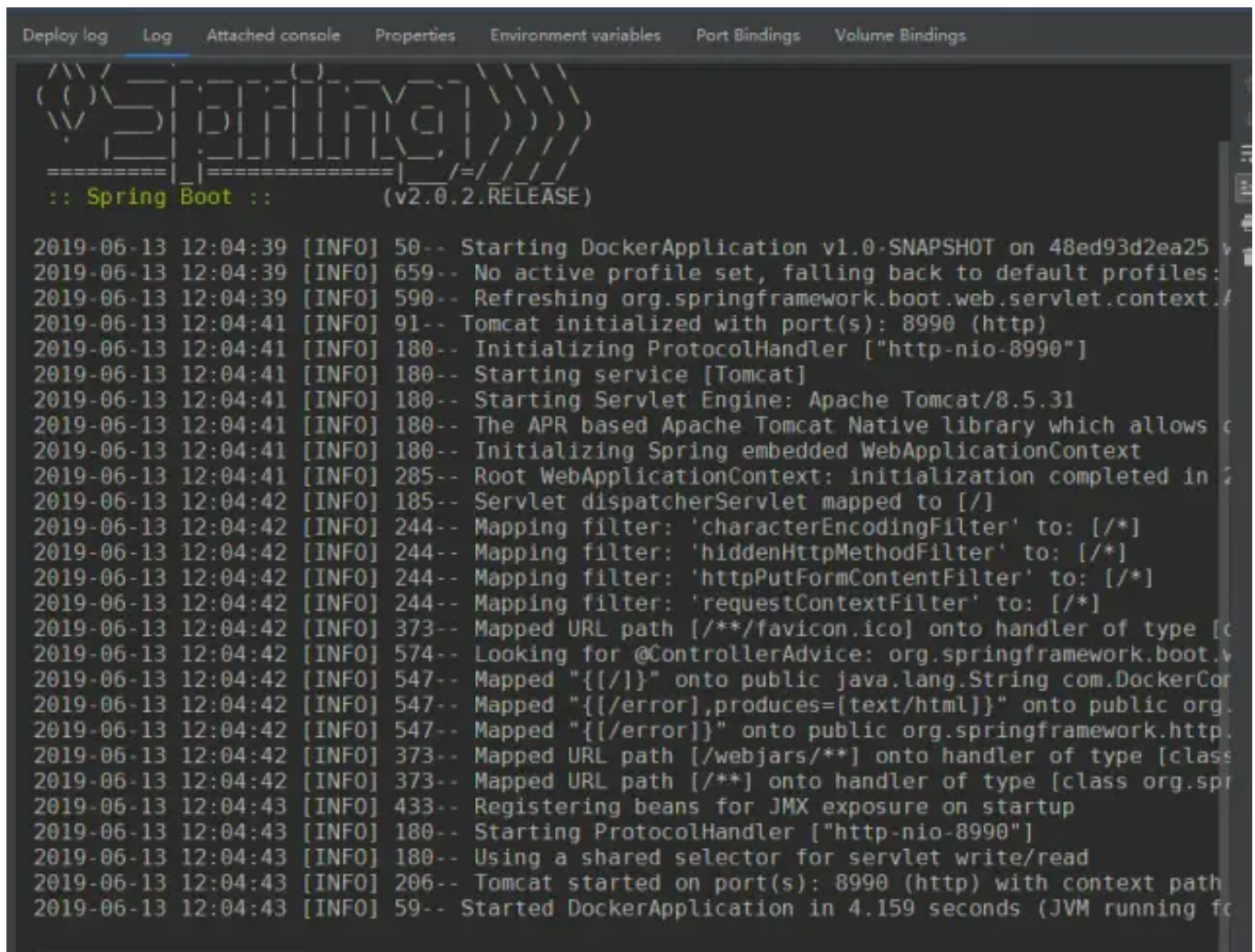


先pull基础镜像，然后再打包镜像，并将镜像部署到远程docker运行



这里我们可以看到镜像名称为docker-demo:1.1，docker容器为docker-server

(9) 运行成功



```
Deploy log  Log  Attached console  Properties  Environment variables  Port Bindings  Volume Bindings

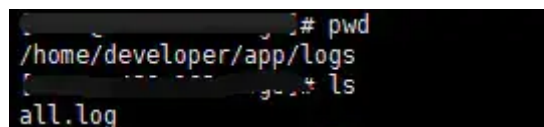
:: Spring Boot :: (v2.0.2.RELEASE)

2019-06-13 12:04:39 [INFO] 50-- Starting DockerApplication v1.0-SNAPSHOT on 48ed93d2ea25 v
2019-06-13 12:04:39 [INFO] 659-- No active profile set, falling back to default profiles:
2019-06-13 12:04:39 [INFO] 590-- Refreshing org.springframework.boot.web.servlet.context.
2019-06-13 12:04:41 [INFO] 91-- Tomcat initialized with port(s): 8990 (http)
2019-06-13 12:04:41 [INFO] 180-- Initializing ProtocolHandler ["http-nio-8990"]
2019-06-13 12:04:41 [INFO] 180-- Starting service [Tomcat]
2019-06-13 12:04:41 [INFO] 180-- Starting Servlet Engine: Apache Tomcat/8.5.31
2019-06-13 12:04:41 [INFO] 180-- The APR based Apache Tomcat Native library which allows c
2019-06-13 12:04:41 [INFO] 180-- Initializing Spring embedded WebApplicationContext
2019-06-13 12:04:41 [INFO] 285-- Root WebApplicationContext: initialization completed in 2
2019-06-13 12:04:42 [INFO] 185-- Servlet dispatcherServlet mapped to [/]
2019-06-13 12:04:42 [INFO] 244-- Mapping filter: 'characterEncodingFilter' to: [/]
2019-06-13 12:04:42 [INFO] 244-- Mapping filter: 'hiddenHttpMethodFilter' to: [/]
2019-06-13 12:04:42 [INFO] 244-- Mapping filter: 'httpPutFormContentFilter' to: [/]
2019-06-13 12:04:42 [INFO] 244-- Mapping filter: 'requestContextFilter' to: [/]
2019-06-13 12:04:42 [INFO] 373-- Mapped URL path [/**/favicon.ico] onto handler of type [c
2019-06-13 12:04:42 [INFO] 574-- Looking for @ControllerAdvice: org.springframework.boot.v
2019-06-13 12:04:42 [INFO] 547-- Mapped "[]" onto public java.lang.String com.DockerCor
2019-06-13 12:04:42 [INFO] 547-- Mapped "[/error],produces=[text/html]" onto public org.
2019-06-13 12:04:42 [INFO] 547-- Mapped "[/error]" onto public org.springframework.http.
2019-06-13 12:04:42 [INFO] 373-- Mapped URL path [/webjars/**] onto handler of type [class org.spr
2019-06-13 12:04:42 [INFO] 373-- Mapped URL path [/**] onto handler of type [class org.spr
2019-06-13 12:04:43 [INFO] 433-- Registering beans for JMX exposure on startup
2019-06-13 12:04:43 [INFO] 180-- Starting ProtocolHandler ["http-nio-8990"]
2019-06-13 12:04:43 [INFO] 180-- Using a shared selector for servlet write/read
2019-06-13 12:04:43 [INFO] 206-- Tomcat started on port(s): 8990 (http) with context path
2019-06-13 12:04:43 [INFO] 59-- Started DockerApplication in 4.159 seconds (JVM running fo
```

(10) 浏览器访问



(11) 日志查看



```
[root@localhost ~]# pwd
/home/developer/app/logs
[root@localhost ~]# ls
all.log
```

自此通过idea 部署springboot项目到docker成功! 难以想象, 部署一个Javaweb项目竟然如此简单方便!

IDEA 上位? 不! Eclipse Theia 1.0 发布!

MySQL 常用分库分表方案，都在这里了!

踩坑了，JDK8中HashMap依然会产生死循环问题!

看完本文有收获? 请转发分享给更多人

关注「ImportNew」，提升Java技能

ImportNew

分享 Java 相关技术干货 · 资讯 · 高薪职位 · 教程



微信号: ImportNew



长按识别二维码关注

伯乐在线 旗下微信公众号

商务合作QQ: 2302462408

好文章，我在看♡