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？

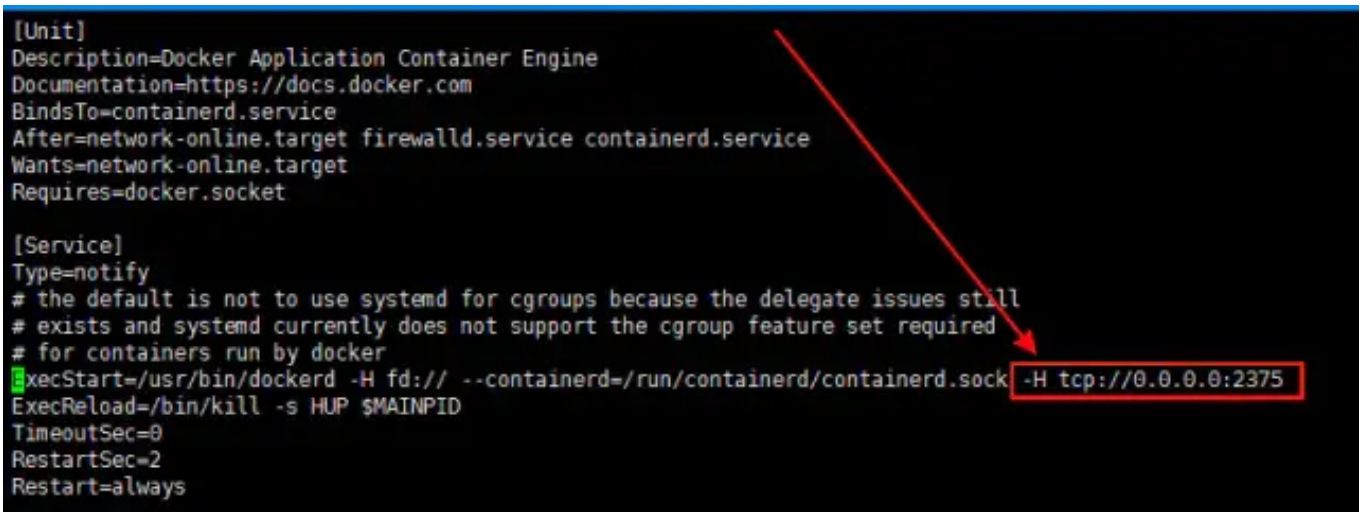
一、开发前准备

1. Docker的安装可以参考https://docs.docker.com/install/

2. 配置docker远程连接端口

```
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



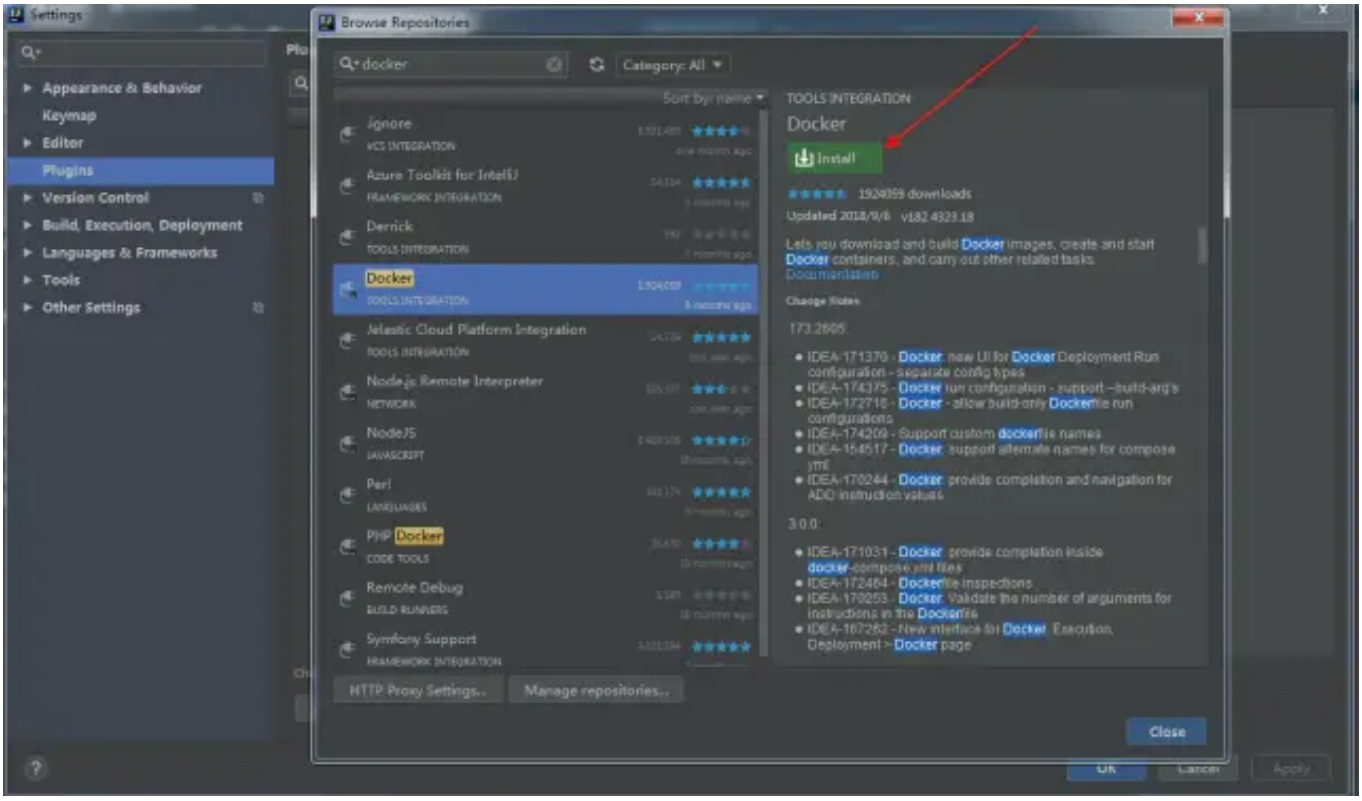
3. 重启docker

```
systemctl daemon-reload
systemctl start docker
```

4. 开放端口

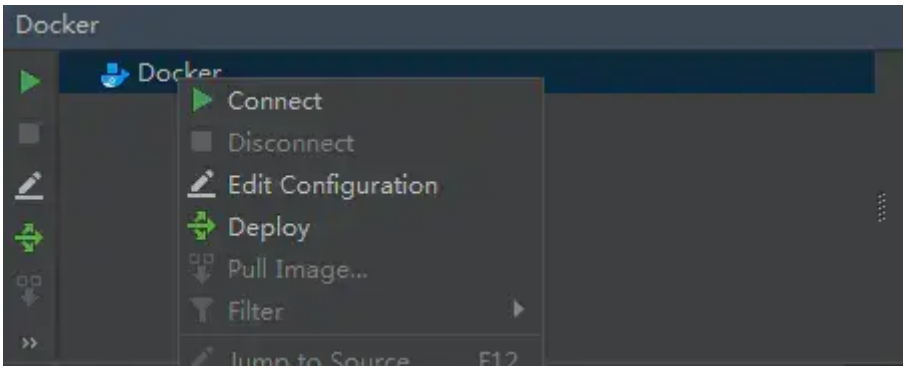
```
firewall-cmd --zone=public --add-port=2375/tcp --permanent
```

5. Idea安装插件,重启

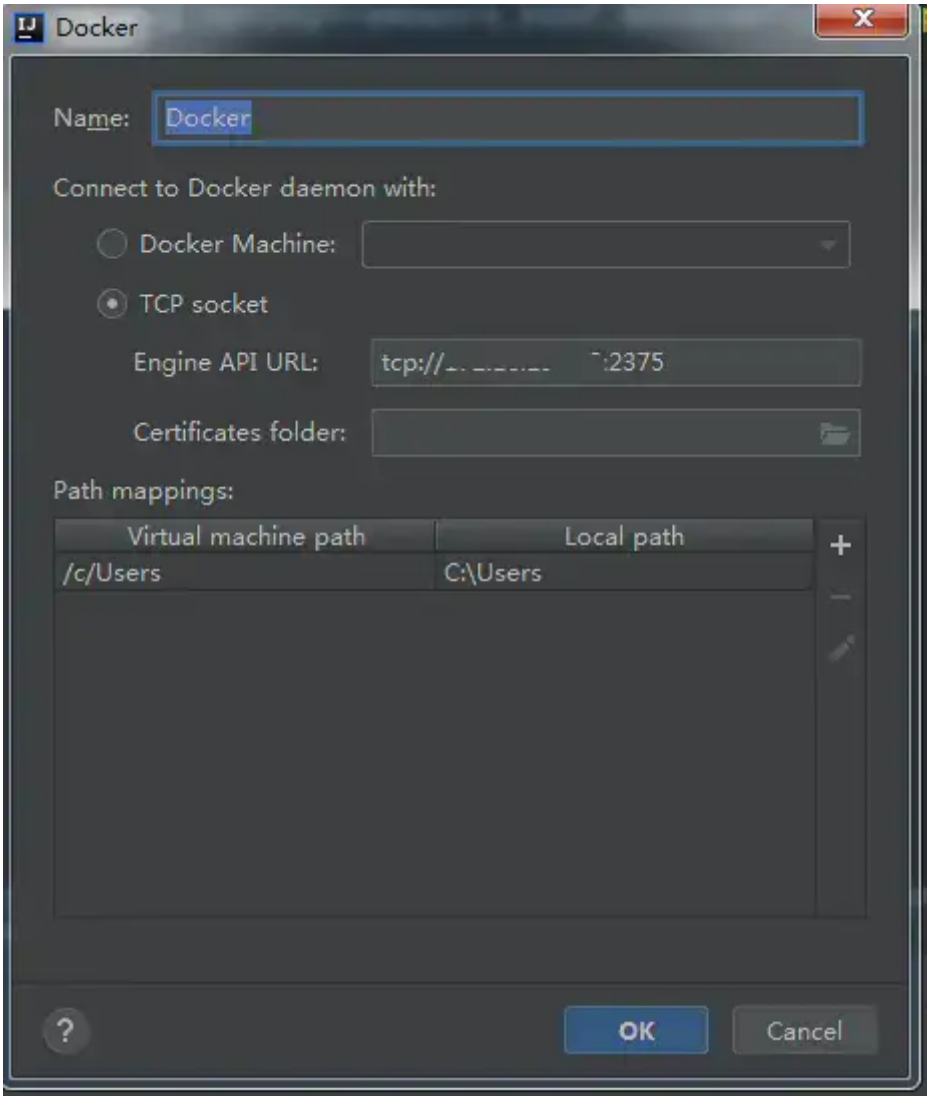


6. 连接远程docker

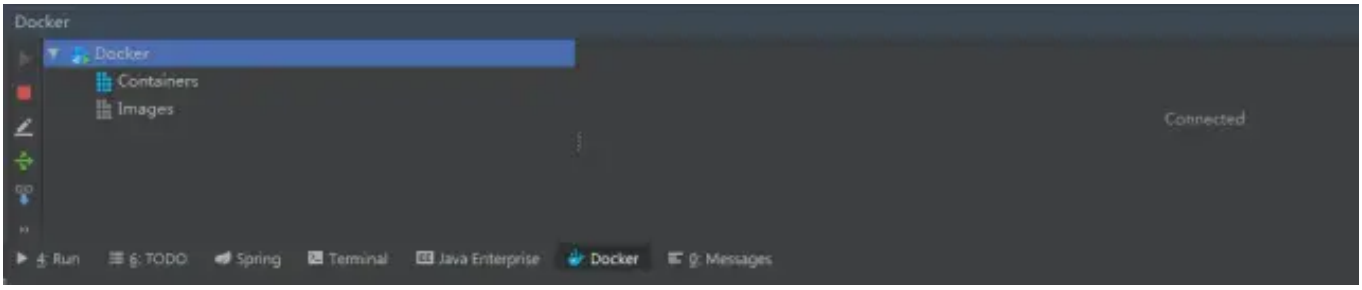
(1) 编辑配置



(2) 填远程docker地址



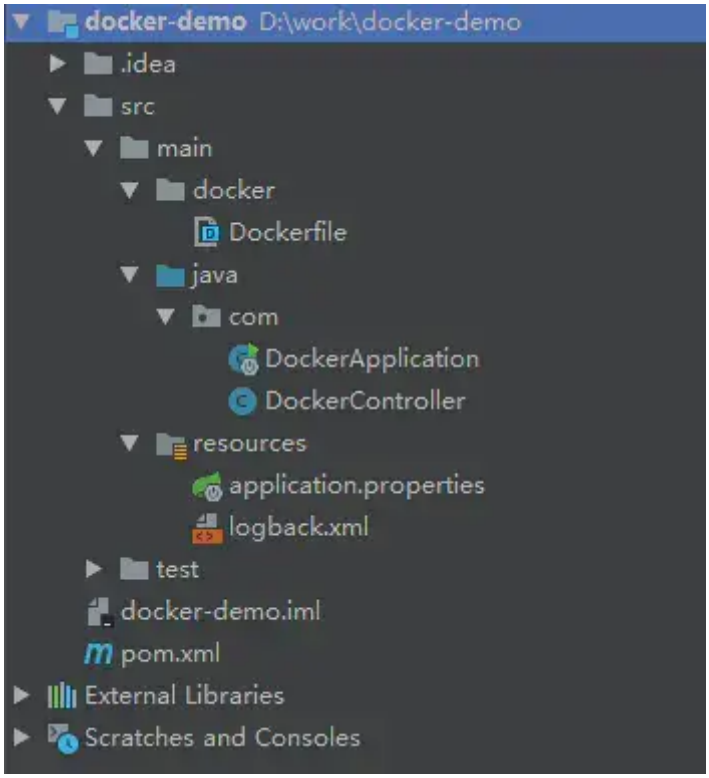
(3) 连接成功，会列出远程docker容器和镜像



二、新建项目

1. 创建springboot项目

项目结构图



(1) 配置pom文件

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.a
    <modelVersion>4.0.0</modelVersion>

    <groupId>docker-demo</groupId>
    <artifactId>com.demo</artifactId>
    <version>1.0-SNAPSHOT</version>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.0.2.RELEASE</version>
        <relativePath />
    </parent>

    <properties>
         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
         <project.reporting.outputEncoding>UTF-8</project.reporting.outputEnc
         <docker.image.prefix>com.demo</docker.image.prefix>
         <java.version>1.8</java.version>
    </properties>
```

```
    <build>
        <plugins>
          <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
          </plugin>
        <plugin>
           <groupId>com.spotify</groupId>
           <artifactId>docker-maven-plugin</artifactId>
           <version>1.0.0</version>
           <configuration>
              <dockerDirectory>src/main/docker</dockerDirectory>
              <resources>
                <resource>
                    <targetPath>/</targetPath>
                    <directory>${project.build.directory}</directory>
                    <include>${project.build.finalName}.jar</include>
                </resource>
              </resources>
           </configuration>
        </plugin>
        <plugin>
            <artifactId>maven-antrun-plugin</artifactId>
            <executions>
                 <execution>
                     <phase>package</phase>
                    <configuration>
                        <tasks>
                            <copy todir="src/main/docker" file="target/${pro
                        </tasks>
                     </configuration>
                    <goals>
                        <goal>run</goal>
                    </goals>
                    </execution>
            </executions>
        </plugin>

       </plugins>
    </build>
```

```
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
  <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
        <version>1.2.17</version>
    </dependency>
</dependencies>
</project>
```

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

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

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

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

(4) 创建DockerApplication文件

```
@SpringBootApplication
```

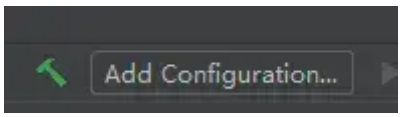
```
public class DockerApplication {
    public static void main(String[] args) {
        SpringApplication.run(DockerApplication.class, args);
    }
}
```

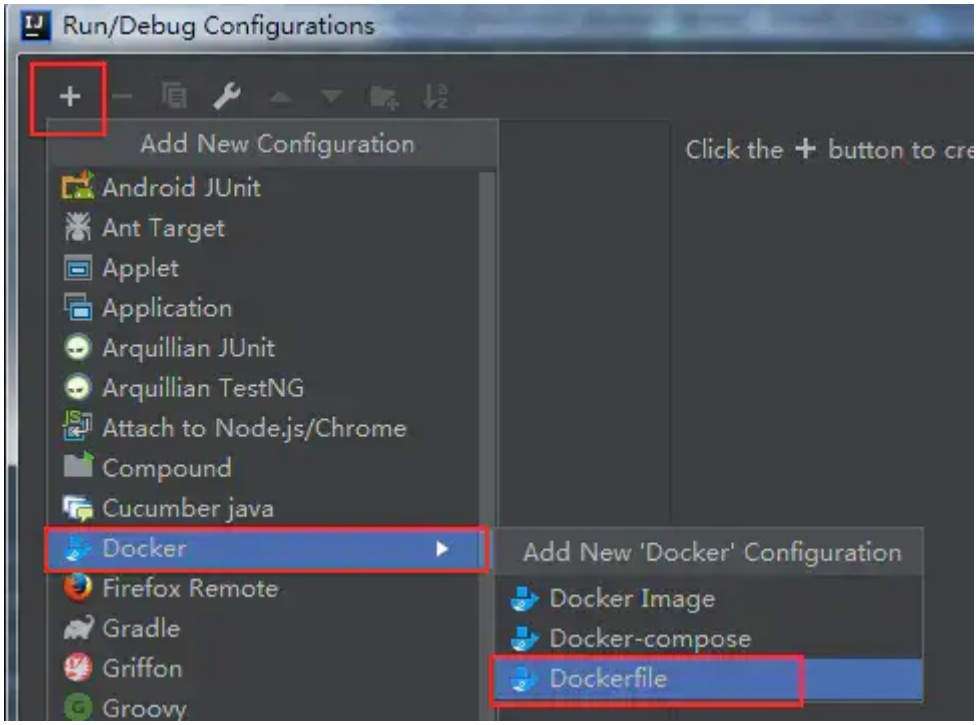
(5) 创建DockerController文件

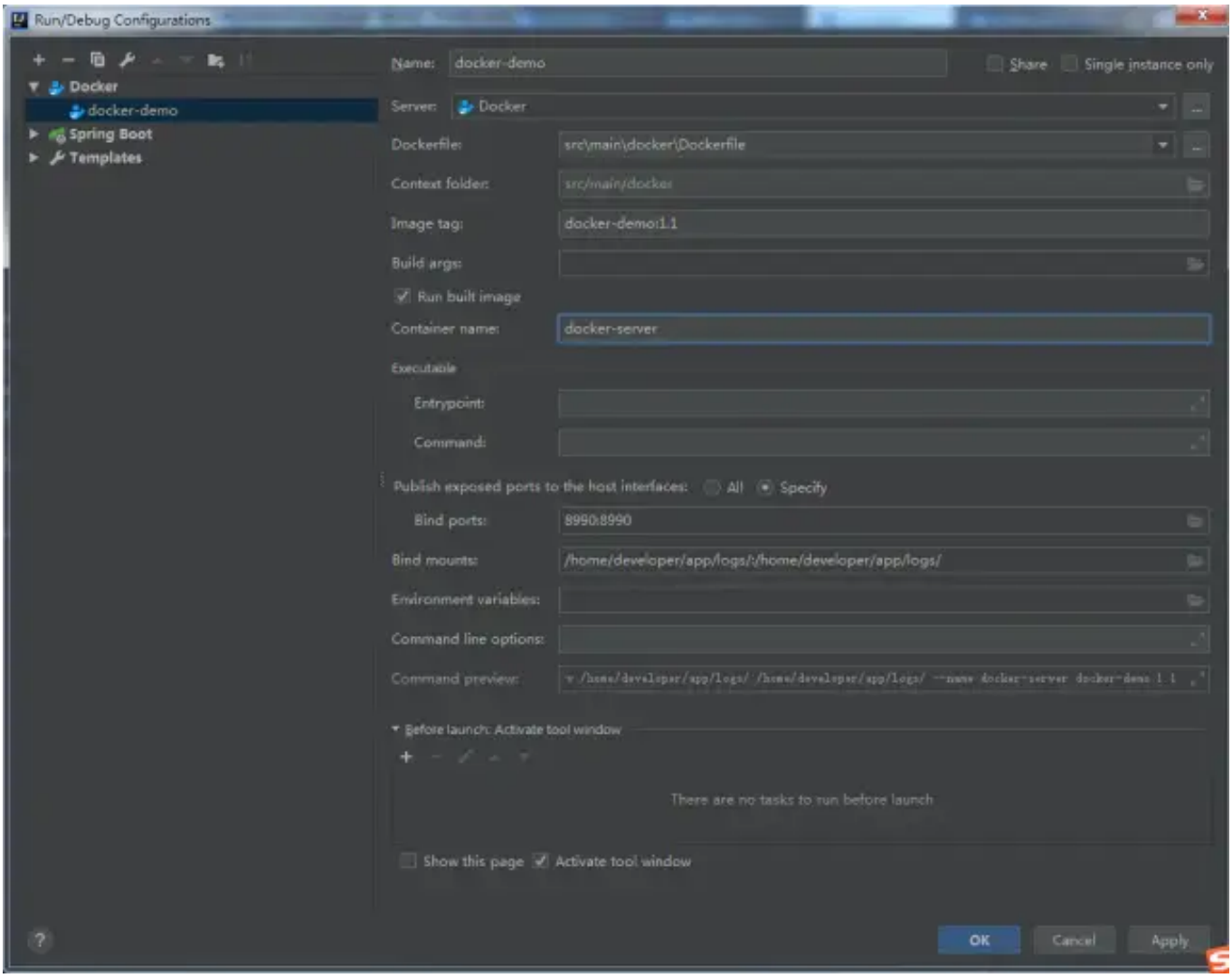
```
@RestController
public class DockerController {
    static Log log = LogFactory.getLog(DockerController.class);

    @RequestMapping("/")
    public String index() {
        log.info("Hello Docker!");
        return "Hello Docker!";
    }
}
```

(6) 增加配置



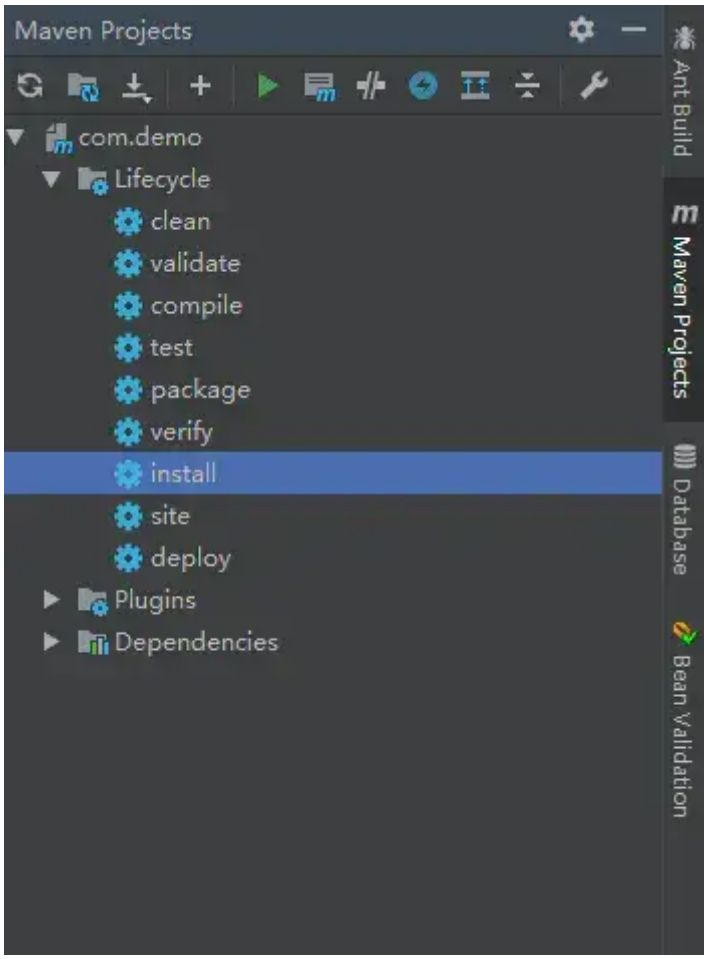

Run/Debug Configurations
Add New Configuration
Android JUnit
Ant Target
Applet
Application
Arquillian JUnit
Arquillian TestNG
Attach to Node.js/Chrome
Compound
Cucumber java
Docker
Firefox Remote
Gradle
Griffon
Groovy
Add New 'Docker' Configuration
Docker Image
Docker-compose
Dockerfile


Run/Debug Configurations
Docker
docker-demo
Spring Boot
Templates
Name: docker-demo
Share
Single instance only
Server: Docker
Dockerfile: src\main\docker\Dockerfile
Context folder: src/main/docker
Image tag: docker-demo:1.1
Build args:
Run built image
Container name: docker-server
Executable
Entrypoint:
Command:
Publish exposed ports to the host interfaces: All Specify
Bind ports: 8990:8990
Bind mounts: /home/developer/app/logs/:/home/developer/app/logs/
Environment variables:
Command line options:
Command preview:
Before launch: Activate tool window
There are no tasks to run before launch
Show this page
Activate tool window
OK
Cancel
Apply

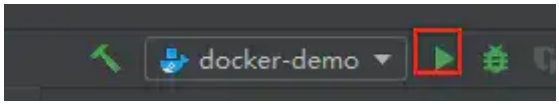
命令解释

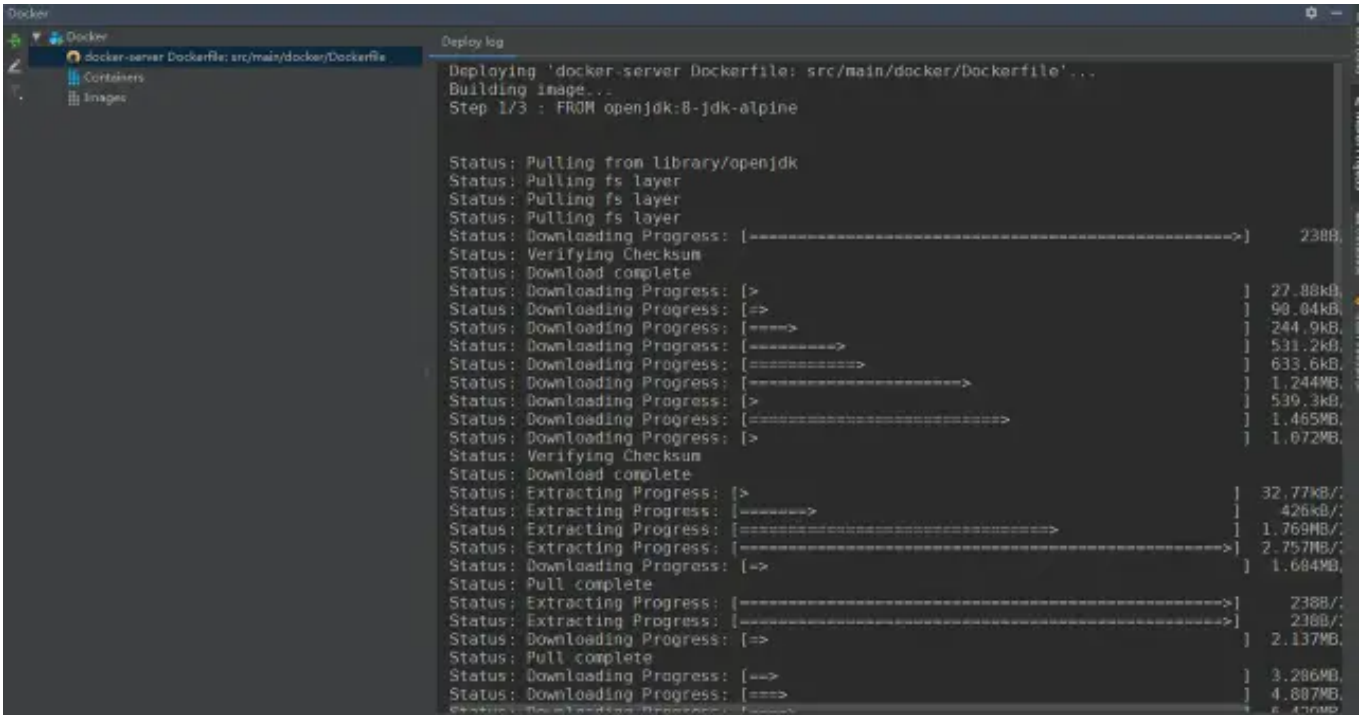
Image tag : 指定镜像名称和tag，镜像名称为 docker-demo，tag为1.1
Bind ports : 绑定宿主机端口到容器内部端口。格式为[宿主机端口]:[容器内部端口]
Bind mounts : 将宿主机目录挂到到容器内部目录中。格式为[宿主机目录]:[容器内部目录]。这个springboot项目会将日志打印在容器 /home/developer/app/logs/ 目录下，将宿主机目录挂载到容器内部目录后，那么日志就会持久化容器外部的宿主机目录中。

(7) Maven打包

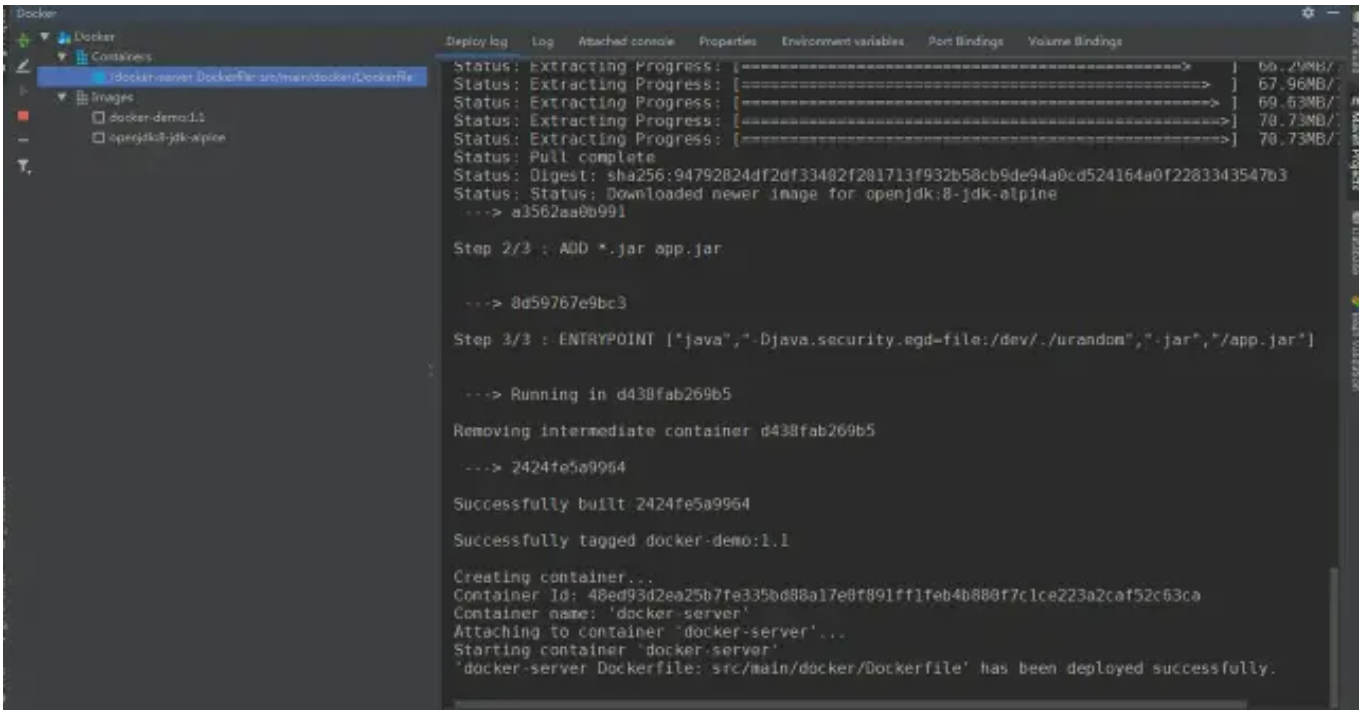


(8) 运行



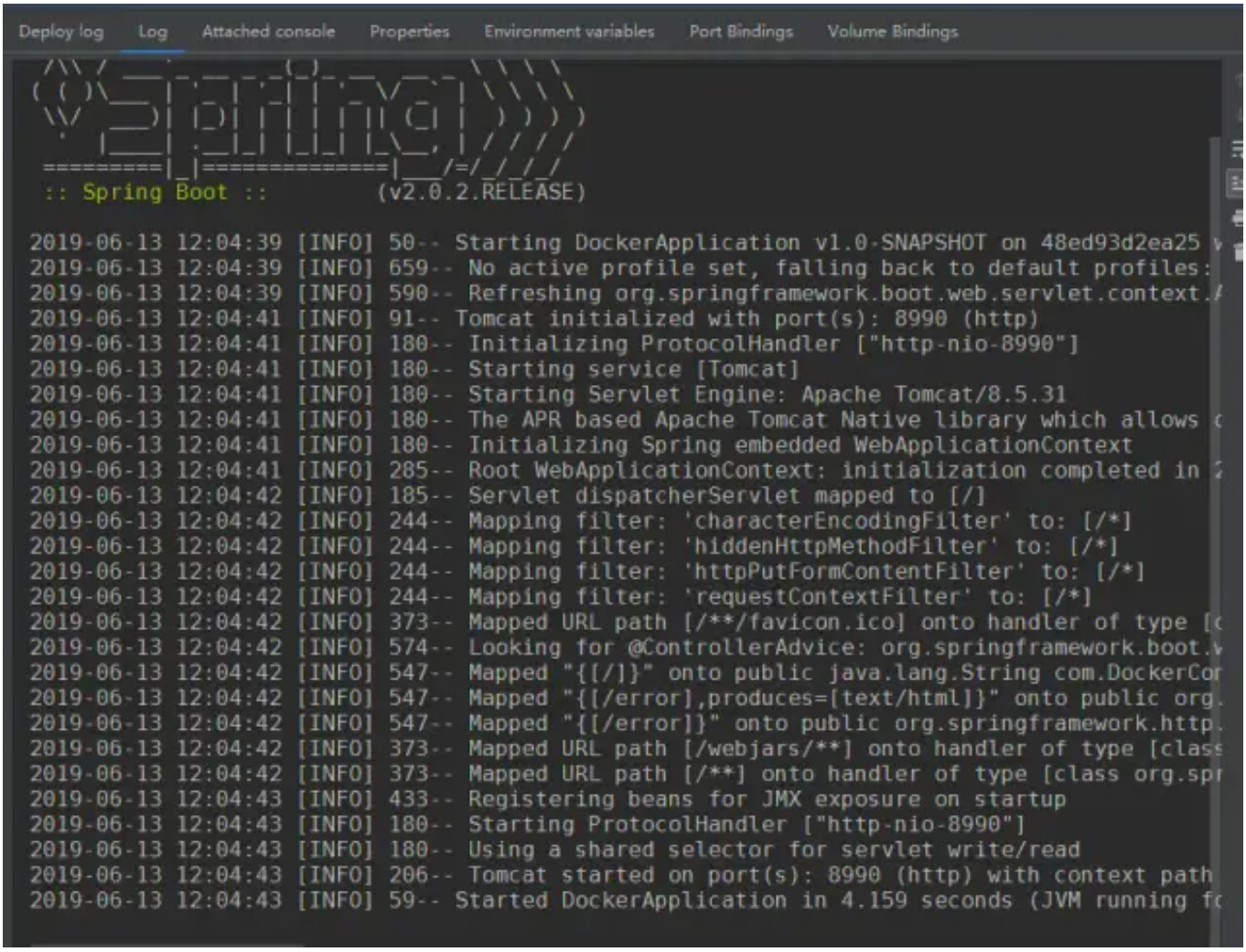


先pull基础镜像，然后再打包镜像，并将镜像部署到远程docker运行



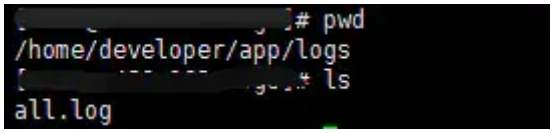
这里我们可以看到镜像名称为docker-demo:1.1，docker容器为docker-server

(9) 运行成功



(10) 浏览器访问

(11) 日志查看



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！

推荐阅读 点击标题可跳转

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♡