链滴

iView+SpringBoot 在 Docker 内构建工作流案例

作者：liumapp

原文链接：https://ld246.com/article/1528440091360

来源网站：链滴

前后端分离的情况下，使用目前非常流行的Vue作为前端框架，SpringBoot作为后端框架，再利用iView的Steps组件和Docker容器技术，构建一个基础的工作流案例。

前言

先上案例源代码: liumapp/file-workflow

前端项目在flow-ui下，后端项目在flow-service下。

利用Docker-compose了将后端项目和前端项目部署在Docker下，因为Vue单独运行需要nodejs环支持，所以在Docker下额外添加了Nginx进行前端项目的支持。

Nginx的配置文件和日志及www目录部署在nginx目录下。

环境配置

直接上docker-compose.yml代码:

```
version: '2'

services:
  flow-service:
    container_name: flow-service
    restart: always
    image: liumapp/flow-service:v1.0.0
    ports:
      - 2020:2020
    volumes:
      - ./flow-service/pic:/pic
    networks:
      - flow-net
```

liumapp/flow-service:v1.0.0需要从flow-service下，利用mvn编译后生成，不能直接从docker hub拉取，所以需要先运行./build-image.sh安装镜像。

并且在docker-compose down之后，mvn编译生成的docker image也不会自动删除，所以需要运行.rm-image.sh进行手动删除。

```
  nginx:
    container_name: nginx
    restart: always
    image: nginx:1.13
    ports:
      - 80:80
      - 443:443
    volumes:
      - ./nginx/conf/vhosts:/etc/nginx/conf.d
      - ./nginx/logs:/var/log/nginx
      - ./nginx/www/:/var/www/
    networks:
      - flow-net
```

配置Nginx，并将flow-ui下的vue项目在npm run build之后，将dist下的内容拷贝到./nginx/www/flwui下。

```
networks:
  flow-net:
    driver: bridge
```

配置docker容器内的网络。

后端

后端代码部署在/flow-service下，是一个标准的springboot web项目。

需要注意一点，前端上传的图像、文件信息是存放在/pic下，但是这个/pic目录，是表示docker容器的pic目录，这个目录利用了volumes与./flow-service/pic建立了映射关系。

所以前端上传的图片实际是存放在./flow-service/pic下。

前端

前端代码部署在/flow-ui下，是一个标准的vue2.0项目。

与后端交互的接口配置在/src/libs/util.js下。

需要注意一点，如果对flow-ui进行了改动，重新编译后，如果需要在docker下运行最新的效果，需将编译好的dist目录下的文件copy到/nginx/www/flowui下。

如果不希望让前端项目在docker下运行，直接在宿主机的nodejs环境下启动也是可以的，并不需要额外的改动，只需要在docker-compose.yml下，将nginx的相关配置注释掉即可（事实上不在意80口被占用的话，不注释也是可以的）。

结尾

项目运行的效果，及源码都非常简单，在项目的README下都有直观的体现。