

1. 方法1

0.1 安装docker

0.2 打包项目

前端使用 `npm run build` 打包，最终打包好的文件夹在 `dist` 文件夹下。后端可以通过 `Maven` 打包，得到一个 `jar` 包。

0.3 上传至服务器

在根目录新建一个文件夹，上传打包后的文件。

再

0.3.1 安装MYSQL

```
1 docker pull mysql:8.0.32
2 docker run -d --name nj_test_mysql_13306 -p 13306:3306 -v
  /data/mysql:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=123456 mysql:8.0.32
```

参数解析：

- `-d`: 后台运行容器，并返回容器 ID
 - `-name myMysql`: 为容器指定一个名称
 - `-p`: 指定端口映射，格式为：主机(宿主)端口:容器端口
 - `-v`: 绑定一个卷，容器的 `/var/lib/mysql` 映射到 主机的目录 `/data/mysql`
 - `-e MYSQL_ROOT_PASSWORD=123456`: 设置环境变量，密码设置为 123456
- mysql:8.0.31：使用镜像 mysql:8.0.31

通过远程连接创建数据库和表

0.3.2 后端

Dockerfile

```
1 #添加Java启动的必要镜像
2 FROM williamyeh/java8
3 #将本地文件挂载到当前容器
4 VOLUME /tmp
5 #复制jar文件和配置文件所在的目录到容器里
6 ADD nj_test-0.0.1-SNAPSHOT.jar nj_test_spring.jar
7 ADD conf /conf
8 #声明需要暴露的端口
9 EXPOSE 8080
10 EXPOSE 8081
11 #配置容器启动后执行的命令,并指定使用项目外部的配置文件
12 ENTRYPOINT ["java", "-Xms256m", "-Xmx256m", "-Djava.security.egd=file:/dev/./urandom", "-jar", "/nj_test_spring.jar", "--spring.config.location=/conf/application.yaml"]
```

打包镜像 (-t 是指定镜像名称和版本号, nj_test_spring是镜像名, :后面的是版本号)

```
1 docker build -t nj_test_spring:0.0.1 .
```

运行容器

```
1 docker run -d -p 8080:8080 -p 8081:8081 --name nj_test_spring_8080  
nj_test_spring:0.0.1
```

0.3.3 打包前端并运行为容器

在前端文件夹 dist 的同一级文件夹下新建 Dockerfile 文件, 文件内容如下

```
1 FROM nginx:latest  
2 COPY ./dist /usr/share/nginx/html/  
3 EXPOSE 80
```

打包镜像

```
1 docker build -t nj_test_vue:0.0.1 .
```

运行容器

```
1 docker run -d -p 8888:80 --name nj_test_vue_8888 nj_test_vue:0.0.1
```