

Vue + SpringBoot 项目的 Docker 部署

作者: owemshu

- 原文链接: https://ld246.com/article/1677935680944
- 来源网站:链滴
- 许可协议:署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)



一. 本地部署

1. 本地 Docker 配置

1) 配置 mirror

● 在 Docker Engine 内添加如下内容:

```
"registry-mirrors": [

"https://vlkh77ku.mirror.aliyuncs.com",

"https://registry.docker-cn.com/",

"http://hub-mirror.c.163.com/",

"https://docker.mirrors.ustc.edu.cn/",

"https://cr.console.aliyun.com/",

"https://mirror.ccs.tencentyun.com/"

]
```

2) 拉取镜像

docker pull openjdk:18.0.2.1 docker pull mysql:8.0.30 docker pull niginx

3) 启动容器

docker run -d --name mysql -p 33306:3306 mysql:8.0.30 -e MYSQL_ROOT_PASSWORD=root

1. 这里添加了环境变量 MYSQL_ROOT_PASSWORD, 设定了新建 mysql 默认 root 用户密码

4) 往 mysql 容器写入数据

主要配置文件 (有效)	常规 高级 数据库 SSL SSH HTTP
	es E
	Navicat数据库
	连接名: localhost-docker
	主机: localhost
	端口: 33306
	用户名: root 密码: •••••••
	✔ 保存密码
· 在17书 法 4 在 第1 题 + / / -	

2. AiSortTool 的 Docker 打包与部署

0). 修改本项目的 webpack 配置 (仅针对本项目)

仅针对本项目,如果存在不同情酌情判断是否需要修改

参考: https://github.com/bailicangdu/vue2-manage/issues/168

● npm 安装依赖

npm install --save--dev babel-preset-es2015@6.24.1

● 在 /build/webpack.base.conf.js 修改

// 将全部 include: [resolve('src'), resolve('test')] 修改为

include: [resolve('src'), resolve('test'), resolve('node_modules/time-formater')]

• 在 /config/index.js 修改

// 将全部 assetsPublicPath 修改为 assetsPublicPath: './'

1). 编写 nginx 配置

● 在根目录下新建目录 /nginx, 在该目录下新建文件 default.conf, 内容如下:

```
server {
  listen
           80;
  server name localhost;
  #charset koi8-r;
  access log /var/log/nginx/host.access.log main;
  error log /var/log/nginx/error.log error;
  location / {
    root /usr/share/nginx/html;
    index index.html index.htm;
  }
  #error page 404 /404.html;
  # redirect server error pages to the static page /50x.html
  #
  error_page 500 502 503 504 /50x.html;
  location = /50x.html {
    root /usr/share/nginx/html;
  }
}
```

2). 打包 vue 项目

• 在项目根路径下运行如下命令打包:

npm run build 或者 yarn build

Asset	Size	Chunks			Chunk Names
static/fonts/element-icons.b02bdc1.ttf	13.2 kB		[emitted]		
static/img/avatar.abbfb12.jpg	50.9 kB		[emitted]		
static/img/background.c488fa0.png	1.07 MB		[emitted]	[big]	
static/js/vendor.0dfb80c419b7c8ea0d5d.js	1.26 MB	Θ	[emitted]	[big]	vendor
static/js/app.f21547b57901610ba6d9.js	46.5 kB	1	[emitted]		app
static/js/manifest.3611a9af393001e7375b.js	1.45 kB	2	[emitted]		manifest
tatic/css/app.78dc56f0d08a6f1cc302027a9b6a2b8d.css	185 kB	1	[emitted]		арр
index.html	585 bytes		[emitted]		
static/favicon.ico	4.29 kB		[emitted]		

● 可以尝试运行静态网页内容进行测试, 这里是 /manage/index.html, 但通常是 /dict/index.html

3). 准备 Dockerfile

• 在根目录下创建 Dockerfile 内容如下 (具体文件名称需要修改)

FROM nginx COPY manage/ /usr/share/nginx/html/ COPY nginx/default.conf /etc/nginx/conf.d/default.conf

1. FROM: 指该镜像依赖项, 这里是本项目依赖的 nginx 版本默认为 latest

2. COPY:从上下文目录中复制文件或者目录到容器里指定路径,这里是将 manage/ 与 nginx/default conf 下内容复制到 nginx 配置目录下

4). 进行 Docker 构建

• 在同目录运行 (注意这里代码最后的点):

docker build -t aisorttool .

1. 这里 -t 指定了该镜像的名称, 默认版本 latest

5). 运行容器

docker run -d --name AiSortTool -p 8002:80 aisorttool

6). 测试前端程序是否挂载成功

● 测试登录功能是否正常 (需要搭配后端)

× +						
st:8002/#/manage					Q 👌 🖈 📑 💽	🖬 🖈 🕹 🖬 🌍 🗄
調商						E
		数	据统计			
当日数据	新潮利户	新建订单	KUT	m		
总数据:	11880	过单	10.00.00			
2.9781 		-O- #288/ ·	O- NEUR -O- NEERA			128 ml 12 O
TH AN EN Rt Lighthouse						027 01 0 2
ERRTERE · S + +						
	. 78 28 WS Wasm ## ##					
1000 80 1200 80 1400 80 1800 1	40 1800 48 2000 40 2000 f	18 2400 88 2800 88 280	48 3000 48 3200 48	3400 89 3800 89	3800 89 400 89 400 89	A400 88 4600 88 4800 8
载用 根葉 根岳 目动器 NN Cookie						
: 10081, eessage: "#240", data: (usee (userType: 1, usermane: "123") pe: "#240" p: 10081	Type: 1, username: "123")}					
	* + #: 50002/#/manage #: #: #: #: #: #: #: #: #: #:	* + * 2002///manage R: <u>BERK 2002//manage</u> <u>BERK 2002//manage</u> <u>BER</u>	* ************************************	* ***	x + \$10002///marage \$2002///marage \$2002//marage <td>x + x x = x</td>	x + x x = x

3. AiSortToolBD 的 Docker 打包与部署

1). 修改配置文件 application.yml 中 database 部分

```
spring:
    datasource:
    username: root
    password: root
    url: jdbc:mysql://mysql:3306/AiSortToolLib?useUnicode=true&characterEncoder=utf-8&us
SSL=true&serverTimeZone=UTC
    driver-class-name: com.mysql.cj.jdbc.Driver
```

2). 调用 Maven 打包

● 调用 Maven 声明周期内的 clean 和 package

‴ 运行	
✔ AiSortToolBD [clean]: 773差秒	<pre>[INF0] maven-clean-plugin:3.2.0:clean (default-clean) @ AiSortToolBD [INF0] Deleting /Users/owemshu/Desktop/安恒/项目/打标工具/AiSortToolBD/target [INF0]</pre>
	进程已结束,退出代码8



● 将 target 目录下的 xxx.jar 包取出

3). 准备 Dockerfile

• 在 xxx.jar 包同目录下新建文件 Dockerfile, 并写入如下内容 (具体文件名称需要修改)

FROM openjdk:18.0.2.1 MAINTAINER owemshu VOLUME /tmp ADD AiSortTooIBD-0.0.1-SNAPSHOT.jar AiSortTooIBD.jar EXPOSE 8082 ENTRYPOINT ["java", "-jar", "/AiSortTooIBD.jar"]

1. FROM: 指该镜像依赖项, 这里是本项目依赖的 jdk 版本

2. MAINTAINER: 该镜像作者

3. VOLUME: 定义匿名数据卷, 在启动容器时忘记挂载数据卷, 会自动挂载到匿名卷, 这里使用了 sprin boot 默认挂载位置

4. ADD: 类似于 COPY 指令,从上下文目录中复制文件或者目录到容器里指定路径,这里进行了重命名

5. EXPOSE: 暴露端口

6. ENTRYPOINT: 类似于 CMD 指令,但其不会被 docker run 的命令行参数指定的指令所覆盖,这里执行了该 jar 包

4). 进行 Docker 构建

• 在同目录运行 (注意这里代码最后的点):

docker build -t aisorttoolbd .

1. 这里 -t 指定了该镜像的名称, 默认版本 latest

5). 运行容器

参考: https://zhuanlan.zhihu.com/p/102802904

docker run -d --name AiSortToolBD --link mysql:mysql -p 8082:8082 aisorttoolbd

1. 这里 -- link 链接了 Docker 内的 mysql 容器



6). 测试后端程序是否挂载成功

POST login					+ •
●开发中 ~ login	HCRH +	BRS - GRE	◎克隆 回生成代码	12分享文档 医保存并归档	
POST ~ http://localhost login/					发送
Header Query Body 以這 预执行脚本 后执行脚本 一键压测	1				
● none ● form-data ● x-www-form-uriencoded ● raw Ison ✓					杰提取学校和描述 六美化 汪简化
1 { 2 3 4 } *password": "123"					
实时期应 请求头(7) 期应头(10) Cookies 成功示例 头放示例					@12:40:34 337.00ms 0.07kb 🚭
■代 用主 NE 用面与粉粉 可积化 utt8 / ① Q					师定的应结果到交!
1 { 2 "status": 10001, 3 "message": "\$?????? 4 "data": { 5 "userType": 1, 6 "username": "123" 7 } 8 }					



1. 服务器 Docker 配置

● 查看 docker 正在运行的容器是否包含 mysql

docker ps | grep mysql

[aisort@localhost aisorttool]\$ docker ps | grep mysql d360940b4f7b mysgl:5.7 "docker-entrypoint:s." 22 months ago Up 6 months 33868/tcp, 0.0.0.0:8:33306->3386/tcp [aisort@localhost aisorttool]\$

1. 注意到这里的服务器已经存在正在运行的 mysql 容器, 且向外暴露端口 33306, 版本 5.7

• 获取到数据库账号密码后, 在本地创建数据库表

3	行	数据长度	引擎	创建日期	修改日期
action_logging	0	16.00 KB	InnoDB	2023-03-02 08:33:20	
categories	0	16.00 KB	InnoDB	2023-03-02 08:33:21	
levels	0	16.00 KB	InnoDB	2023-03-02 08:33:21	
samples	0	16.00 KB	InnoDB	2023-03-02 08:33:21	
users	3	16.00 KB	InnoDB	2023-03-02 08:33:21	2023-03-02 08:3

2. AiSortTool 的 Docker 打包与部署

0) 修改项目请求 baseURL

● 将项目请求 baseURL 改为服务器地址和端口号

1) 上传相关文件



2) 打包与部署

●打包

[aisort@localhost aisorttool]\$ docker build -t aisorttool .
Sending build context to Docker daemon 2.643MB
Step 1/3 : FROM nginx
> 3f8a00f137a0
<pre>Step 2/3 : COPY manage/ /usr/share/nginx/html/</pre>
> 049829575d37
<pre>Step 3/3 : COPY nginx/default.conf /etc/nginx/conf.d/default.conf</pre>
> 695ab471e182
Successfully built 695ab471e182
Successfully tagged aisorttool:latest
[aisort@localhost aisorttool]\$

	latest	695ab471e182	5 minut	es ago	144MB		
aisort@localhost a	aisorttoolbd]\$ docker	run -dname AiSortTool	-p 5002:8	0 aisorttoo	n.		
9280c654a04a004c2	52ad8ea5dacf2911eff1cf	f9826462ef5ce438db695d05					
aisort@localhost a	aisorttoolbd]\$ docker	os grep AiSortTool					
19280c654a04	aisorttool	"/docker-entry	point"	18 second	is ago	Up 17 seconds	0.0.0.0:5002->80/tcp
aisort@localhost ;	aisorttoolbd]\$						

3. AiSortToolBD 的 Docker 打包与部署

0) 修改项目 application.yml 内 database.url

参考: https://blog.csdn.net/moshowgame/article/details/122018398

● 在最后加上 & useSSL=false

url: jdbc:mysql://192.168.30.240:33306/AiSortToolLib?useUnicode=true&characterEncoder=ut -8&useSSL=true&serverTimeZone=UTC&useSSL=false

1. 这里不进行修改会造成后端接收到请求后一直报错create connection SQLException

1) 上传相关文件

```
[aisort@localhost aisorttoolbd]$ pwd
/home/aisort/aisorttoolbd
[aisort@localhost aisorttoolbd]$ ll
总用量 61152
-rw-rw-r-- 1 aisort aisort 62611035 3月
-rw-rw-r-- 1 aisort aisort 168 3月
[aisort@localhost aisorttoolbd]$
```

-rw-rw-r-- 1 aisort aisort 62611035 3月 3 15:18 AiSortToolBD-0.0.1-SNAPSHOT.jar -rw-rw-r-- 1 aisort aisort 168 3月 3 15:18 Dockerfile

2) 打包与部署

● 打包

[aisort@localhost aisorttoolbd]\$ docker buildt aisorttoolbd
Sending build context to Docker daemon 62.61MB
Step 1/6 : FROM openjdk:18.0.2.1
> 71260f256d19
Step 2/6 : MAINTAINER owemshu
> Running in 87367909a8c7
Removing intermediate container 87367909a8c7
> 49430235ae71
Step 3/6 : VOLUME /tmp
> Running in e3bce31edf4a
Removing intermediate container e3bce31edf4a
> ab7763513ce5
<pre>Step 4/6 : ADD AiSortToolBD-0.0.1-SNAPSHOT.jar AiSortToolBD.jar</pre>
> 30898e9ca6f8
Step 5/6 : EXPOSE 8082
> Running in 475161eb37dd
Removing intermediate container 475161eb37dd
> bc7d0d0fccf3
<pre>Step 6/6 : ENTRYPOINT ["java", "-jar", "/AiSortToolBD.jar"]</pre>
> Running in 6899b2905046
Removing intermediate container 6899b2905046
> c26d01e16eb7
Successfully built c26d01e16eb7
Successfully tagged aisorttoolbd:latest
[aisort@localhost aisorttoolbd]\$

• 部署 (由于原先的端口被占用, 这里需要修改端口号)

[aisort@localhost aisorttoolbd]\$ docker images | grep aisorttoolbd aisorttoolbd latest e15e0320ded4 32 seconds ago 533MB [aisort@localhost aisorttoolbd]\$ docker run -d --name AiSortToolBD --link mysql5.7:mysql -p 8882:8082 aisorttoolbd 672d8af89709d65377938c79079eb835a95bb194e5d7281eec69578c129429fd [aisort@localhost aisorttoolbd]\$

三. Docker 动态部署

● 如果依然使用实时打包方式,在实际部署中更新文件会造成极大的麻烦,因此可以使用项目路径挂的方式,使容器只包含基本的配置文件同时动态加载新项目文件

1. 重写 Dockerfile

● 这里以 SpringBoot 文件为例子

FROM openjdk:18.0.2.1 MAINTAINER owemshu EXPOSE 8082 ENTRYPOINT ["java", "-jar", "/app/AiSortToolBD.jar", "--spring.config.location=/app/applicati n.yml"]

1. 这里执行 ENTRYPOINT 时候额外附带了 /app 目录,以及载入本地配置文件的命令

2. 打包及部署

docker build -t asbd .

docker run -d --name ASBD -p 8082:8082 -v /Users/owemshu/Desktop/app:/app asbd

1. 这里将本地目录 /Users/owemshu/Desktop/app 挂载到容器内 /app,其中本地的 jar 包文件和 置文件也被同步读入容器

3. 更新本地文件并重启容器