

docker-compose 编排 solo+mysql 镜像

作者: [liumapp](#)

原文链接: <https://ld246.com/article/1554361976462>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一条命令在docker中启动solo，所有麻烦的配置全部用docker-compose编排解决

项目地址: [github/solo-in-docker](https://github.com/b3log/solo-in-docker)

如何使用

- 确保系统环境具有docker + docker-compose

检查命令:

```
docker -v
docker-compose -v
```

- 配置docker-compose.yml文件，需要注意的事项已经全部//备注好了

services:

mysql:

container_name: mysql

image: mysql:5.5.60

restart: always

volumes:

- ./mysql/data:/var/lib/mysql //mysql的数据文件存放地址

- ./mysql/conf/mysql.d.conf:/etc/mysql/mysql.conf.d/mysql.d.cnf //mysql的配置文件存放地

ports:

- "6603:3306" //6603代表宿主机端口，3306代编容器的端口

environment:

- MYSQL_ROOT_PASSWORD=adminadmin //mysql的root账号密码

solo:

container_name: solo

image: b3log/solo //直接使用最新版本的solo镜像

restart: always

ports:

- "8080:8080" //如果要部署到线上的话，请改为"80:80"

environment: //此处配置solo跟mysql的连接设置，来源为上面启动的mysql容器，如果要用自己mysql服务，那么请将mysql的service去掉

RUNTIME_DB: "MYSQL"

JDBC_USERNAME: "root"

JDBC_PASSWORD: "adminadmin"

JDBC_DRIVER: "com.mysql.jdbc.Driver"

JDBC_URL: "jdbc:mysql://mysql:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC" //此处，因为solo跟mysql同为docker容器，所以可以直接使容器名 + 容器端口来访问

command: --listen_port=8080 --server_scheme=http --server_host=www.liumapp.com //照solo官方要求，在solo启动之初，配置solo的域名、端口，如果是本地测试的话，将host改为localhost即可

- 启动命令

```
docker-compose up -d
```

- 停止命令

```
docker-compose down
```

- 查看solo日志的命令

```
docker logs -t -f --tail 100 solo
```

效果展示

<http://www.liumapp.com>