



链滴

Solo4.0 博客改变部署方式 docker-compose

作者: [yuanhenglizhen](#)

原文链接: <https://ld246.com/article/1585130421397>

来源网站: 链滴

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



01. 心血来潮

看到 D 大的 4.0 版本出，我基本上是每个版本都用过，每次的手动部署有点心累，代码里面的配置文改的脑瓜子疼。作为一个二流子运维，怎么能允许手动操作这种事情发生了？是的之前太懒了

02.h2 转 MySQL

转啥转，转的我脑瓜子疼。真心后悔第一次没有用 MySQL，查了下 h2 转 MySQL 挺麻烦的。一切麻烦皆是痛苦的根源于是放弃。

03. docker-compose 修改

基于社区同学的 solo-in-docker 修改，使其能合适的应用于自己的部署方式，取名叫 solo-easystart .

GitHub 地址: <https://github.com/mufengcoding/solo-easystart>

1. 前提准备

1.1 安装 docker

确保系统环境具有 docker + docker-compose

检查命令：

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

安装命令：

- CentOS:

```
yum install docker  
yum install docker-compose
```

- MacOS:

请参考这篇博客: [MacOS 安装 docker](#)

自行编译 caddy1 镜像

```
curl https://getcaddy.com | bash -s personal consul,dns,hook.service,http.authz,http.cache,http.cgi,http.cors,http.datadog,http.expires,http.filter,http.forwardproxy,http.geoip,http.git,http.gpkg,http.grpc,http.ipfilter,http.jwt,http.locale,http.login,http.mailout,http.minify,http.nobots,http.permission,http.prometheus,http.proxyprotocol,http.pubsub,http.ratelimit,http.realip,http.reuth,http.recaptcha,http.restic,http.s3browser,http.supervisor,http.torproxy,http.webdav,net,redis,supervisor,tls.dns.cloudflare
```

```
cp /usr/local/bin/caddy ./
```

```
docker build -t caddy .
```

1.2 获取证书

因为域名在 cloudflare 托管, 所以我用了 dns 插件。 如果不需要 cloudflare 在 Caddyfile 中直接把 t
s 那边修改为 `tls { protocols tls1.1 tls1.2 }`

2. 修改配置文件

在启动需要修改两个配置文件

- caddy/conf/Caddyfile
- docker-compose.yml

2.1 配置 Caddy

编辑 caddy/conf/Caddyfile 文件

将所有 solo.mufengs.com 替换为您自己的站点

有静态页面需要跑的, 直接 caddyfile 添加 `root /data/` 然后把页面放到 caddy/www

2.2 配置 docker-compose

编辑 docker-compose.yml 文件

按照"//"后面的备注进行修改

```
version: "2"
```

```
services:  
  solo:
```

```
container_name: solo
image: b3log/solo:latest
restart: always
ports:
  - "8080:8080"
volumes:
  - "/root/solo_h2:/opt/solo/h2/"
environment:
  RUNTIME_DB: "H2"
  JDBC_USERNAME: "root"
  JDBC_PASSWORD: "123456"
  JDBC_DRIVER: "org.h2.Driver"
  JDBC_URL: "jdbc:h2:/opt/solo/h2/db;MODE=MYSQL"
command: --listen_port=8080 --server_port=80 --server_scheme=https --server_host=solo
mufengs.com
caddy:
  container_name: caddy
  image: mufeng5619/caddy:latest //自建镜像修改这里
  restart: always
  environment: //不用cloudflare的这边可以去掉
    CLOUDFLARE_API_KEY: "be38b105926db0ad9"
    CLOUDFLARE_EMAIL: "mufen@gmail.com"

  ports:
    - "80:80"
    - "443:443"
  volumes:
    - "./caddy/conf/Caddyfile:/etc/caddy/Caddyfile" //配置文件挂载
    - "./caddy/www:/data" //其他页面
```

启动

- 启动命令

`docker-compose up -d`

- 停止命令

`docker-compose down`

- 查看日志的命令

`solo:`

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

效果展示

<https://solo.mufengs.com>

注意事项

- docker-compose 只是一个 docker 容器的编排工具，本质还是 docker 容器在运行
- 每一次命令 docker-compose 启动的时候，都会自动拉取最新 solo 的镜像，所以自动更新非常简单

接下来就是拿自己的博客做迁移测试，最近折腾 caddy2 有点蒙，看的出来对 dns 插件的支持不够人性化，~~放弃使用 caddy2 了，决定再等等~~

caddy1 测试迁移成功，目前博客运行在 docker 环境中，使用的是 h2 数据库

在论坛同学提示下，重新构建 caddy2，构建成功并测试使用

caddy2 的话比较麻烦，我决定另外开一个帖子来讲如何在 caddy2 中使用 cloudflare

以上所以的问题在非 cloudflare 用户中是不存在的，可以开箱即用