链滴

docker-compose 编排技术搭建 solo 个人博客

作者：mysolo1130

原文链接：https://ld246.com/article/1638266858830

来源网站：链滴

1. 安装docker

- CentOS

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

2. 获取证书

- 证书申请请移步： 安装SSL证书

3. 修改配置文件

3.1 配置Nginx

- 编辑 nginx/conf/nginx.conf文件
- 将所有 dduan.site替换为您自己的站点
- 然后修改下述两块证书文件名称(路径不用改，直接改证书名即可)

```
 ssl_certificate /etc/nginx/conf.d/dduan.site/1_dduan.site_bundle.crt;
 ssl_certificate_key /etc/nginx/conf.d/dduan.site/2_dduan.site.key;
```

3.2 配置docker-compose

- 编辑 docker-compose.yml文件，按照**#** 后面的备注进行修改

```
[root@prometheus docker-compose-solo]# cat docker-compose.yml
version: "2"

services:
  mysql:
    container_name: mysql
    image: mysql:5.7
    restart: always
    volumes:
      # MySQL数据存放地址
      - ./mysql/data:/var/lib/mysql
    ports:
      # 6603代表宿主机端口，3306代编容器的端口
      - "6603:3306"
    environment:
      # mysql的root账号密码
      MYSQL_ROOT_PASSWORD: "adminadmin"
    # 在这里配置mysql的全局参数
    command: --max_allowed_packet=32505856
  solo:
    # 直接使用最新版本的solo镜像
    container_name: solo-docker-compose
    image: b3log/solo:latest
```

```
    restart: always
    ports:
      # https部署的方式不需要修改此处
      - "8090:8090"
    environment:
      RUNTIME_DB: "MYSQL"
      JDBC_USERNAME: "root"
      JDBC_PASSWORD: "adminadmin"
      JDBC_DRIVER: "com.mysql.cj.jdbc.Driver"
      # 此处，因为solo跟mysql同为docker容器，所以可以直接使用容器名 + 容器端口来访问
      JDBC_URL: "jdbc:mysql://mysql:3306/solo?useUnicode=yes&characterEncoding=UTF-8&u
eSSL=false&serverTimezone=UTC"
    # 按照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
localhost即可
    command: --listen_port=8090 --server_port= --server_scheme=http --server_host=localhos

  nginx:
    container_name: nginx
    image: nginx:latest
    restart: always
    ports:
      - "90:80"
      - "443:443"
#     volumes:
      # 映射nginx目录到docker容器中
#       - "./nginx/conf:/etc/nginx/conf.d"
      # 映射服务器的证书目录到docker容器中
#       - "/docker/solo/https/cert/ws.site:/etc/nginx/conf.d/ws.site"
      # 映射www目录到docker容器中，后期您可以在这里部署自己的静态站点或者php站点
#       - "./nginx/www/:/var/www/"
```

4. 启动

- 启动命令

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

- 停止命令

```
docker-compose down
```

- 查看solo日志，查看Nginx日志，只需要solo改为Nginx

```
docker logs -f solo
```

5. 建数据库

- 进入MySQL容器

```
docker exec -it mysql bash
```

- 登录MySQL

```
mysql -uroot -pXXX
```

- 创建solo数据库

```
create database solo DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

- 退出数据库和容器，执行 exit 命令

7. 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
- 执行docker-compose 命令需要在docker-compose.yml所在目录执行
- 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
- 数据备份问题，docker 容器死亡的时候，容器内数据会自动清除，除非我们使用 volumes 构建映关系，这里我只将最重要的 mysql 数据库文件映射在 mysql/data 目录下

8. 完整配置文件

GitHub：https://github.com/dadeity/solo-docker-compose.git