



链滴

Docker-compose 安装 Solo

作者: [figo930](#)

原文链接: <https://ld246.com/article/1558334420005>

来源网站: [链滴](#)

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

Docker-compose安装Solo

一、环境

System: Ubuntu 18.04.2 LTS

Docker: Docker version 18.09.6, build 481bc77

Docker-composes: docker-compose version 1.24.0, build 0aa5906

APP: Solo

数据库: Mysql

Web服务器:nginx

二、安装环境

1、安装docker

```
root@root:/home#wget -qO- https://get.docker.com/ | sh
root@root:/home# docker -v
Docker version 18.09.6, build 481bc77
```

2、安装docker-composes

```
root@root:/home#apt install python-pip
root@root:/home#pip install docker-compose
root@root:/home#docker-compose -v
docker-compose version 1.24.0, build 0aa5906
```

三、安装ssl证书

1、安装acme.sh

```
curl https://get.acme.sh | sh
```

2、生成证书

现在我使用的是DNS验证方式，所以需要获取DNS域名账户的认证信息。我域名放在cloudflare上，装官方文档，现在需要cloudflare ID(ID就是邮箱)，和cloudflare key。(Key 可以在 My Profile找到)
[关文档](#)

```
export CF_Email="youmail@fly930.com" //这个是ID
export CF_Key="1234567890" //这是Key
/root/.acme.sh/acme.sh --issue --dns dns_cf -d fly930.com -d www.fly930.com
```

如果证书成功生成，文件会保存在/root/.acme.sh/fly930.com/ 文件夹下面。

```
root@root:~/acme.sh/fly930.com# ls
ca.cer fly930.com.cer fly930.com.conf fly930.com.csr fly930.com.csr.conf fly930.com.key fullch
in.cer
```

最后acme会帮你创建一个crontab定时任务,每天检测一次任务。如果证书过期会自动更新。

```
root@root:~/acme.sh/fly930.com# crontab -l
25 0 * * * "/root/.acme.sh"/acme.sh --cron --home "/root/.acme.sh" > /dev/null
```

四、配置相关配置文件

1、创建相关文件

```
root@root:~# mkdir -p /home/www/solo/conf /home/www/solo/data
```

```
root@root:~# cd /home/www/solo
```

```
root@root:/home/www/solo# touch docker-compose.yml
```

```
root@root:/home/www/solo# touch conf/nginx.conf
```

2、docker-compose.yml 内容

```
root@root:/home/www/solo#vi docker-compose.yml
version: "3"
services:
  mysql:
    container_name: mysql
    image: mysql:5.5.60
    restart: always
    networks:
      - backend
    volumes:
      - ./data:/var/lib/mysql
    ports:
      - "3306"
    environment:
      - MYSQL_ROOT_PASSWORD=root
  solo:
    container_name: solo
    image: b3log/solo
    restart: always
    networks:
      - frontend
      - backend
    ports:
      - "8080"
    environment:
      RUNTIME_DB: "MYSQL"
      JDBC_USERNAME: "root"
      JDBC_PASSWORD: "root"
      JDBC_DRIVER: "com.mysql.jdbc.Driver"
      JDBC_URL: "jdbc:mysql://mysql:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC"
    command: --listen_port=8080 --server_port=80 --server_scheme=https --server_host=www
```

```
fly930.com
nginx:
  container_name: nginx
  image: nginx:latest
  networks:
    - frontend
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - "./conf:/etc/nginx/conf.d"
    - "/root/.acme.sh/fly930.com:/etc/nginx/conf.d/fly930.com"
networks:
  frontend:
  backend:
```

3、nginx.conf内容

```
root@root:/home/www/solo#vi conf/nginx.conf
gzip on;
gzip_min_length 1k;
gzip_buffers 4 16k;
gzip_http_version 1.1;
gzip_comp_level 2;
gzip_types text/plain application/javascript application/x-javascript text/javascript text/css
application/xml application/xml+rss;
gzip_vary on;
gzip_proxied expired no-cache no-store private auth;
gzip_disable "MSIE [1-6]\.";

server_tokens off;
access_log off;

server {
  listen 80;
  server_name www.fly930.com fly930.com;
  return 301 https://$server_name$request_uri;
}
server {
  server_name www.fly930.com fly930.com;
  listen 443 ssl http2;
  ssl_certificate /etc/nginx/conf.d/fly930.com/fullchain.cer;
  ssl_certificate_key /etc/nginx/conf.d/fly930.com/fly930.com.key;
  ssl_session_timeout 5m;
  ssl_protocols TLSv1.1 TLSv1.2;
  ssl_prefer_server_ciphers on;
  ssl_ciphers "EECDH+CHACHA20:EECDH+CHACHA20-draft:EECDH+AES128:RSA+AES128:EE
DH+AES256:RSA+AES256:EECDH+3DES:RSA+3DES:!MD5";
  ssl_session_cache builtin:1000 shared:SSL:10m;
  location / {
    proxy_pass http://solo:8080;
  }
}
```

```
    access_log /etc/nginx/conf.d/www.fly930.com.log;
}
```

五、启动docker-compose

1、在solo目录下运行

```
root@root:/home/www/solo#docker-compose up -d
Creating network "solo1_backend" with the default driver
Creating network "solo1_frontend" with the default driver
Creating nginx ... done
Creating mysql ... done
Creating solo ... done
```

2、进入mysql容器创建solo数据库

////进入容器

```
root@root:/home/www/solo# docker exec -it mysql bash
```

///连接Mysql

```
root@f115521b4eca:/# mysql -u root -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 6

Server version: 5.5.60 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

////////创建solo

```
mysql>CREATE DATABASE IF NOT EXISTS solo DEFAULT CHARSET utf8mb4 COLLATE utf8mb
```

```
_general_ci;
```

```
Query OK, 1 row affected (0.02 sec)
```

```
////////退出数据库
```

```
mysql> \q
```

```
Bye
```

```
root@f115521b4eca:/# exit
```

```
exit
```

3、退出重启一下容器

```
root@root:/home/www/solo#docker-compose restart
```

```
Restarting mysql ... done
```

```
Restarting nginx ... done
```

```
Restarting solo ... done
```

4、最后访问你的域名。开始享受技术带来的欢乐吧。