



链滴

compose 搭建 B3log 开源博客

作者: [jockming112](#)

原文链接: <https://ld246.com/article/1650165969854>

来源网站: [链滴](#)

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

B3log开源博客搭建

docker 安装

```
yum install docker-ce-17.12.1.ce
```

docker-compose 安装

```
curl -L https://github.com/docker/compose/releases/download/1.24.0/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

```
# 设置 docker-compose 执行权限  
chmod +x /usr/local/bin/docker-compose
```

注意

- 事先购买好域名并配置好解析。
- 开放服务器80和443端口，云服务器在安全组中添加规则。
- 证书申请在SSL证书，我是用的阿里云服务器，证书可以免费使用一年，过期后需要重新申请。

项目结构

```
├── compose-nginx.yml  
├── compose-solo-mysql.yml # 这里我是自建的数据，之前有，就没有执行这个编排文件  
├── docker-compose.yml  
├── nginx  
│   ├── cert # https使用  
│   │   ├── 7620436_www.example.com.key  
│   │   └── 7620436_www.example.com.pem  
│   ├── conf  
│   │   └── app.conf  
│   └── logs  
│       ├── access.log  
│       └── error.log  
└── solo  
    └── skins # 博客皮肤，可以从官方仓库克隆项目，复制里面的skins
```

温馨提示：下面的配置请根据这个目录结构对号入座，本人主站也是使用该配置部署。有问题请评论联系我。

docker-compose.yml

```
version: "3"
```

```
services:
```

```
  solo:
```

```
    container_name: solo
```

```
    image: b3log/solo
```

```

restart: always
environment:
  RUNTIME_DB: "MYSQL"
  JDBC_USERNAME: "root"
  JDBC_PASSWORD: "123456"
  JDBC_DRIVER: "com.mysql.cj.jdbc.Driver"
  JDBC_URL: "jdbc:mysql://172.16.96.61:3306/solo?useUnicode=yes&characterEncoding=UTF-8&useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true"
command: --listen_port=8080 --server_scheme=https --server_host=www.example.com --lute_http=http://172.16.96.61:8249 --server_port= --static_server_host=cdn.jsdelivr.net --static_path=/gh/88250/solo/src/main/resources
volumes:
  # Pay attention to synchronization time
  # echo 'Asia/Shanghai' > /etc/timezone/timezone
  - /etc/timezone/timezone:/etc/timezone
  - /etc/localtime:/etc/localtime
  - ./solo/skins:/opt/solo/skins/:ro
ports:
  - 8080:8080
dns:
  - 114.114.114.114
  - 8.8.8.8
lute:
image: b3log/lute-http
container_name: lute
restart: always
volumes:
  # Pay attention to synchronization time
  # echo 'Asia/Shanghai' > /etc/timezone/timezone
  - /etc/timezone/timezone:/etc/timezone
  - /etc/localtime:/etc/localtime
ports:
  - 8249:8249

```

compose-solo-mysql.yml

```

version: "3"
services:
  blog-mysql:
    image: mysql:8.0.22
    restart: always
    container_name: blog-mysql
    privileged: false
    ports:
      - 3306:3306
    networks:
      - blog-extranet
    environment:
      - MYSQL_ROOT_PASSWORD=123456
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - ./mysql/conf:/etc/mysql/
      - ./mysql/data:/var/lib/mysql/
      - ./mysql/init:/docker-entrypoint-initdb.d/

```

```
- ./mysql/logs:/logs/
```

```
networks:  
  blog-extranet:  
    driver: bridge
```

compose-nginx.yml

```
version: "3"
```

```
services:  
  nginx:  
    image: nginx:1.14  
    restart: always  
    container_name: nginx  
    privileged: true  
    ports:  
      - 80:80  
      - 443:443  
    volumes:  
      - /etc/localtime:/etc/localtime:ro  
      - ./nginx/conf:/etc/nginx/conf.d/  
      - ./nginx/cert:/etc/nginx/cert/  
      - ./nginx/logs:/etc/nginx/log/  
    dns:  
      - 114.114.114.114  
      - 8.8.8.8  
    networks:  
      blog-extranet:  
        driver: bridge
```

app.conf (nginx配置)

```
upstream blog {  
  # server www.example.com:8080;  
  # Solo 监听端口，这里的ip最好是你的云服务器内网ip，可以通过ip addr 或者ifconfig查看  
  server 172.16.96.61:8080;  
}
```

```
server {  
  listen 80 default_server;  
  listen [::]:80 default_server;  
  server_name www.example.com;  
  # http 重定向到https 配置  
  #if ($http_x_forwarded_proto = "http") {  
  #  return 301 https://$server_name$request_uri;  
  #}  
  rewrite ^(.*)$ https://${server_name}$1 permanent;  
}
```

```
server {  
  listen 443 ssl;  
  server_name www.example.com;
```

```

charset utf-8;
access_log /etc/nginx/log/access.log;
error_log /etc/nginx/log/error.log;

ssl_certificate /etc/nginx/cert/7620436_www.example.com.pem;
ssl_certificate_key /etc/nginx/cert/7620436_www.example.com.key;
ssl_session_timeout 5m;
ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:
ADH:!RC4;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;

location / {
    proxy_pass http://blog$request_uri;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Host $server_name;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    client_max_body_size 10m;
}

# 防止爬虫抓取
if ($http_user_agent ~* "360Spider|JikeSpider|Spider|spider|bot|Bot|2345Explorer|curl|wget
webZIP|qihoobot|Baiduspider|Googlebot|Googlebot-Mobile|Googlebot-Image|Mediapartners
Google|Adsbot-Google|Feedfetcher-Google|Yahoo! Slurp|Yahoo! Slurp China|YoudaoBot|Sos
spider|Sogou spider|Sogou web spider|MSNBot|ia_archiver|Tomato Bot|NSPlayer|bingbot")
{
    return 403;
}

error_page 403 404 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html/error-page;
}
}

```