链滴

从零开始搭建 solo 个人博客系统

作者：vcjmhg
原文链接：https://ld246.com/article/1570344579042
来源网站：链滴

自己个人博客系统从搭建运行至今已然过去一月，整个搭建过程可以说是踩坑无数。因此将整个搭建程以及踩过的坑写到此处，希望可以给搭建个人博客系统的朋友提供一些帮助。整个过程教程包括服器选购、域名申请与备案、安装solo、nginx进行反向代理以及ssl证书的申请与使用，好了废话说少上开始。

1.博客系统的搭建流程

为了让大家更好的了解整个博客系统搭建的逻辑，因此我将其搭建流程做成一个流程图，希望能给大以直观的感受；大家在看流程图要注意有些环节是可选。

2.服务器选购

首先要有自己的服务器，如果有可以直接跳过这一步。如果没有的话，可以有好多服务器提供商可选国内的有阿里云、腾讯云、华为云、百度云等，国外的有AWS、vultr等，具体选型可以根据自己的务需求来定。我们这里以阿里云服务器的购买为例：

2.1阿里云学生主机

针对在校学生阿里云提供了学生优惠，可以通过云翼计划来购买可以节省一大笔费用。具体的购买流可以参考官方教程：阿里云学生机购买指南说明。对于系统的选择个人推荐centos。

2.2普通云主机

普通阿里云主机的购买可能涉及到选配置的问题，但是一般来说入门级的服务器就足够使用，当然如特殊需要另当他论。此处附上ECS云服务器的购买地址当然对于系统的选择个人推荐centos。

3.域名购买与备案（可选）

3.1域名购买

如果条件允许尽量买个域名，因为通过ip访问一方面不方便，另一方面可能会被封禁而且不能进行SE优化，而且现在非热门域名（热门域名包括.com、.cn、.net等）也并不是很贵如我自己购买的.top域三年才43块钱。具体购买方式跟购买服务器类似，一般云服务器提供商顺带都提供域名购买服务。这我们提供一个阿里云的域名购买地址：阿里云域名购买

3.2域名服务器备案

如果域名或者服务器是国外的一般不需要备案，但是如果说是国内的按照当前国内政策所有域名现在需备案。一般来说我们可以通过域名购买的服务商域名备案系统来进行备案，当然也可以通过工业与息化部的ICP/IP地址/域名信息备案管理系统来进行备案，这里建议通过前者，因为官方的备案系统进是很慢的远远没有第三方的备案系统进度快。这里我们以阿里云的域名备案系统为例，详细说明域名案的流程：

第一步：进入域名备案系统

第二步：填写信息和验信资料

第三步:等待阿里云初审

这一步大约会耗时一天左右，如果你第二步资料填写均符合要求，那么第二天阿里云客服会给你打来话，告诉你信息已经通过，将会提交至工信部。如果说资料不符合要求，那么客服会给你打电话并发件告诉你资料的不规范之处，更改完成之后重新提交，再次进行阿里云初审。这个过程一般耗时两到天。

第四步：管局审核

这一步是最耗时间的，首先在初审过了之后会有一个工信部短信验证，也就是确定备案手机号是否可使用。通过之后信息会正式提供给管局。从提供管局信息到得到反馈一般是7到20天不等（这个就看地管局心情了），如果管局信息通过不出意外，你会收到一个备案号和密码。此时域名备案便完成了。

3.3域名服务器解析

这一步是所有过程中最轻松的，登陆域名服务商的的域名管理平台，将域名的解析地址指向所购买服器的ip，大约两分钟之后设置便会生效。

当然我们可以通过ping命令来看解析是否生效：

4.solo安装

到这一步算是正式进入正题，根据官网提供的安装方式有两种分为**本地使用**、**Docker部署**。

4.1 本地试用

下载最新的 Solo 包解压，进入解压目录执行以下命令：

- Windows: java -cp "WEB-INF/lib/*;WEB-INF/classes" org.b3log.solo.Starter
- Linux: java -cp "WEB-INF/lib/*:WEB-INF/classes" org.b3log.solo.Starter

注意：官方是不太推荐通过war包发布或者源码构建部署，因为这样的部署方式不利于后续新版发布时的升级和更新。

4.2 Docker部署

第一步 获取最新镜像：

```
docker pull b3log/solo
```

运行结果：

第二步 安装MYSQL

我们既可以直接在服务器上直接安装也可以通过docker方式在docker容器中安装

方式一：docker安装

```
# 安装mysql:5.6,直接docker run 他会自动去官方镜想下载
# MYSQL_ROOT_PASSWORD=[的数据库密码，此处写的是123456
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 -d mysql:5.6
# docker安装的mysql默认允许远程连接，可以使用Navicat等软件连接数据库
# 进入容器mysql
docker exec -it mysql bash

# 进入数据库 p后面跟你的密码
mysql -uroot -p123456

# 创建数据库(数据库名:solo;字符集utf8mb4;排序规则utf8mb4_general_ci)
create database solo DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
# 出现Query OK, 1 row affected (0.00 sec)表示成功
#退出数据库
exit
#退出容器
exit
```

运行结果如下：

方式二：物理机上直接安装

由于配置稍显繁琐，可以参考下边这篇博客:

Centos安装mysql(YUM源方式)

第三步 安装solo

运行如下命令：

```
docker run --detach --name solo --network=host \
--env RUNTIME_DB="MYSQL" \
--env JDBC_USERNAME="root" \
--env JDBC_PASSWORD="123456" \
--env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \
--env JDBC_URL="jdbc:mysql://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=UT
-8&useSSL=false&serverTimezone=UTC" \
b3log/solo --listen_port=8080 --server_scheme=http --server_host=www.vcjmhg.top
```

- --detach: 这个选项告诉 Docker 在启动后将程序与控制台分离,使其进入“后台”运行。
- --name solo: solo是容器的名字，也可以改成自己喜欢的名字如mysolo，这个无所谓

- RUNTIME_DB="MYSQL"：指明我们此处使用的数据库为MYSQL,如果使用,H2 Database,将MYSQL成org.h2.Driver即可
- JDBC_USERNAME="root"： 指明MYSQL数据连接时使用的用户名，默认都是root
- JDBC_PASSWORD="123456"： 指明MYSQL数据库连接时用户密码，使用时注意将123456替换自己在上一步所设置的密码
- env JDBC_DRIVER="com.mysql.cj.jdbc.Driver"：数据库连接驱动包，如果使用,H2 Database,将m.mysql.cj.jdbc.Driver改成H2即可
- --server_host=www.vcjmhg.top： 个人域名，如果没有可设置为自己的服务器ip
- --env JDBC_URL=...:
- --listen_port=8080:指明solo监听的端口此处使用的是8080,如果不想配置nginx此处可以换成80

命令运行结果：

命令执行完成之后没有报错的话，通过docker ps查看当前当前容器列表中是否有名字叫solo的容器如果有证明启动成功了，此时可以通过个人域名/ip+:8080来进行访问,类似 http://192.168.217.132:080,如果不想配置nginx可以将8080换成80，可以直接通过域名/ip来直接进行访问，类似vcjmhg的客--https://vcjmhg.top。不出意外会出现如下界面（如果出现不能访问的情况考虑是否是防火墙配有问题，查看是否开发8080或者80端口）：

由于后边我们需要配置nginx进行反向代理以及配置ssl证书来实现https方式访问，因此在看到solo启正常之后，此处创建的solo镜像需要删除，等配置完nginx之后重新在创建一个。

删除solo容器直接执行下边命令

```
docker kill --name solo
docker rm --name solo
```

命令执行结果如下：

5.安装nginx（可选）

安装之前为了后续配置nginx方便，我们需要在本地创建几个文件，用来挂载nginx的配置文件

```
# 切换到服务器根目录
cd /
# 创建主目录
mkdir dockerData
# 创建文件
mkdir dockerData/nginx dockerData/nginx/conf dockerData/nginx/logs dockerData/nginx/w
w dockerData/nginx/ssl
```

上边的文件目录名称可以任意，此处我使用dockerDate

- dockerData/nginx 用于存放docker下nginx自定义文件
- dockerData/nginx/conf存放nginx配置文件
- dockerData/nginx/log存放nginx日志文件
- dockerData/nginx/www:存放nginx访问的资源文件
- dockerData/nginx/ssl存放ssl证书

命令执行结果如下：

启动nginx

```
docker run --name nginx -p 80:80 -d nginx
```

命令执行结果如下：

如果没有备案，80端口可能是禁止访问的，因此可以可以将上边的80:80换成8080:80。命令执行完之后，没有报错的话可以通过docker ps来看nginx是否正常运行，在运行的情况下访问的域名加上端号查看是否正常安装，如果使用的80端口默认可以省略。出现如下界面表示安装成功。

导出配置文件：

```
docker cp nginx:/etc/nginx/nginx.conf /dockerData/nginx/conf/nginx.conf  #导出配置文件ngin
.conf
docker cp nginx:/etc/nginx/conf.d /dockerData/nginx/conf/conf.d  #导出conf.d
cd /dockerData/nginx/
ls conf/   #查看配置文件是否导出成功
docker stop nginx  #删除刚才创建的nginx容器
docker rm nginx
```

命令执行结果：

重新创建一个nginx容器，挂载刚才本地导出的配置文件，便于后续更改nginx的配置信息

```
docker run -d -p 80:80 --name nginx \
-v /dockerData/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \
-v /dockerData/nginx/conf/conf.d:/etc/nginx/conf.d \
-v /dockerData/nginx/www:/usr/share/nginx/html \
-v /dockerData/nginx/logs:/var/log/nginx nginx
```

- -v /dockerData/nginx/conf/nginx.conf:/etc/nginx/nginx.conf ： 挂载配置文件nginx.conf
- -v /dockerData/nginx/conf/conf.d:/etc/nginx/conf.d： 挂载配置文件default.conf
- -v /dockerData/nginx/www:/usr/share/nginx/html： 挂载项目文件
- -v /dockerData/nginx/logs:/var/log/nginx ： 挂载配置文件

命令执行结果如下：

注意： 我自己在搭建过程中发现执行docker ps命令后发现nginx并没有被启动,使用命令docker logs ginx发现挂载文件时权限不足，果断在创建nginx容器是加上--privileged=true参数（如果没有出现问题可不加上述参数）。

即执行如下命令

```
docker run -d -p 80:80 --name nginx --privileged true \
-v /dockerData/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \·
-v /dockerData/nginx/conf/conf.d:/etc/nginx/conf.d \
-v /dockerData/nginx/www:/usr/share/nginx/html \
-v /dockerData/nginx/logs:/var/log/nginx nginx
```

执行docker ps -a 此时容器运行正常

容器创建完成之后重新访问在浏览器访问可能会出现如下界面：

此时你可以在www目录下创建一个html文件，也可以先不用管，等后续配置完成之后自然会消失。

6. 配置ssl证书（可选）

从http升级到https只需要在nginx中配置一个证书即可，一般性的ssl证书是可以免费申请的

6.1证书选购

阿里云或者腾讯云都提供证书申请服务，这里以阿里云的证书申请为例：

第一步: 进入阿里云的证书购买地址https://www.aliyun.com/product/cas

第二步： 进入购买页面，选择免费型DV SSL

第三步: 支付

第四步： 进入控制台，进行证书下载

6.2配置nginx配置文件

下载之后会得到一个名字类似于2793667_www.vcjmhg.top_nginx.zip的文件，将其上传到服务器/dckerData/nginx/ssl目录中,解压后（通过unzip命令）会得到如下两个文件

大家可以参考我的配置文件进行配置，配置自己的default.conf文件。

```
server {
    listen       443;
    server_name  localhost;
    ssl on;
    ssl_certificate /ssl/2793667_www.vcjmhg.top.pem;  # ssl 证书目录
    ssl_certificate_key /ssl/2793667_www.vcjmhg.top.key;
    ssl_session_timeout 5m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;

    #charset koi8-r;
    #access_log  /var/log/nginx/host.access.log  main;

    location / {
      #root   /usr/share/nginx/html;
      # index  index.html index.htm;
      # 官方博客上此处用的是域名，但配置时发现不好使，所以我用的是服务器ip
      proxy_pass http://39.105.61.192:8080;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
server{
  listen 80;
  server_name www.vcjmhg.top;
  rewrite ^(.*) https://$host$1 permanent;
```

```
}
```

注意:上边的配置文件只是参考，要根据自己的服务器做出相应更改。

由于我们现在用的nginx容器并未监听443端口，所以需要删除现在的容器，重新启动一个新的nginx器

```
#先删除原来的nginx容器
docker stop nginx;
docker rm nginx;
#创建新的nginx容器
docker run -d -p 80:80 -p 443:443 --name nginxs \
-v /dockerData/nginx/conf/nginx.conf:/etc/nginx/nginx.conf \
-v /dockerData/nginx/conf/conf.d:/etc/nginx/conf.d \
-v /dockerData/nginx/ssl:/ssl/ \
-v /dockerData/nginx/www:/usr/share/nginx/html \
-v /dockerData/nginx/logs:/var/log/nginx nginx

#创建新的solo容器并映射到8080端口，用上边的nginx进行反向代理
docker run --detach --name solo --network=host \
--env RUNTIME_DB="MYSQL" \
--env JDBC_USERNAME="root" \
--env JDBC_PASSWORD="123456" \
--env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \
--env JDBC_URL="jdbc:mysql://192.168.217.132:3306/solo?useUnicode=yes&characterEncodi
g=UTF-8&useSSL=false&serverTimezone=UTC" \
b3log/solo --listen_port=8080 --server_scheme=https --server_host=www.vcjmhg.top --serve
_port=
```

- --server_scheme=http换成--server_scheme=https即可
- --server_port：最终访问端口，使用浏览器默认的80或者443的话值留空即可
- 如果出现权限不足问题，启动时加上 -privileged true参数即可

重启nginx，docker restart nginx,然后用浏览器访问https://域名类似于https://www.vcjmhg.top,陆github账户后出现如下界面

后记

首先官方已经给了一些安装的教程，一位叫墨殇的博主也已经给了特别详细的安装教程(地址，点这里)但是我自己在配置过程中出现了好多问题，因此我在这里写下来这篇博客，希望能给他人提供一些帮。当然可能个人水平有限，中间难免会出现一些错误，如若发现恳请指出，不胜赐教。如果按照本教在配置过程中遇到什么问题，欢迎在博客下边留言，我若看到的话一定第一时间回复，谢谢！