



链滴

使用 Solo 搭建个人博客的心得 (docker 部署 + https 配置)

作者: [ReturnYG](#)

原文链接: <https://ld246.com/article/1598671417081>

来源网站: [链滴](#)

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

```
<p></p>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
</script>
(adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p>在自己捣鼓了大概两天之后，自己的博客算是基本搭建完成了 :smile:， 这第一篇文章就讲讲这期间遇到的一些小问题以及如何解决的。:heart:</p>
<h2 id="为何选择Solo搭建博客">为何选择 Solo 搭建博客</h2>
<p>百度和谷歌了很久，搜到了很多种开源博客系统，包括基于 java, php, node.js 等等很多个平的。但其实搜的越多就越知道到底想要什么风格的博客。我其实比较倾向于界面好看，有较稳的社区，使用人群较多的平台。好看不用多说了，稳定的社区是为了在自己遇到问题时可以更快的解决问题。从咱们的社区来看，做的相当优秀。所以还是选择了 Solo。</p>
<h2 id="基础搭建部分">基础搭建部分</h2>
<p>我从零到有的时间花费的并不多，因为在搭建这个博客之前，我有使用 Halo, Hexo, Wordpress 等其他博客系统，所以对我来说搭一个能访问的站点并不难。我也是推荐看官方给出的教程视频来搭就好了。</p>
<p>在搭建过程中遇到的问题是 docker 长长的启动代码，在这里，如果读者使用的是云服务器，我常不建议使用云服务器商提供的远程连接。虽说阿里云现在的远程连接页也做得不错，但还是使用第三方软件输入命令不会出现较多的意外情况，如何使用第三方 ssh 工具网上的教程还是很多的。</p>
<p>话说回来，启动代码我建议复制到文档文件里，编辑好之后，再复制进 ssh 工具运行就好了，按视频来应该是不会出错的。</p>
<h2 id="基础配置">基础配置</h2>
<p>在搭建完成之后，在偏好设定--&gt; 信息配置中，除了一些基本信息之外，统计代码应该放在 h ad 部分，ICP 和公安联网备案号应该放在页脚部分，两部分均可使用 html 代码。</p>
<p>在参数设置的 Favicon 这个地方，是我遇到的最大问题，因为目前的 Solo 的图片均不在本地存和使用。但是我在社区中并没有找到哪里可以上传图片，然后在博客端使用，故我想在自己的服务器存放一些少量我自用的图标或图片。</p>
<p>起初并未找到解决方案，后来在社区的一篇文章中得到灵感，使用 nginx 反向代理即可。步骤如：</p>
<ul>
<li>在你的服务器上创建好你存放图片的文件夹，例如我的文件夹路径为 /var/solo-upload/，在这文件夹中还有两个子文件夹，用于存放不同类型的图片，一个是 myicon，另一个是 myimages</li>
<li>配置 nginx 反向代理</li>
</ul>
<pre><code class="highlight-chroma">upstream backend {
    server localhost:8080;#此处换成你的Solo运行的端口
}
server {
    listen    80;
    server_name www.example.com;#此处换成你的域名
    access_log off;
    location / {
        proxy_pass http://backend$request_uri;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        client_max_body_size 10m;
    }
    location /upload/ { #你想在域名后通过哪个标识访问本地资源
```

```
    alias /var/solo-upload/; #本地资源的存放路径, 注意: 要填写绝对路径给
}
}
```

</code></pre>

<p>在这个配置中, 也遇到了一个小坑, 就是其中的 alias 和 root 用哪个, 从原理上讲, 两个都可实现这个功能, 但是 location 中指定的/upload/必须是在 root 指定的/var/solo-upload/目录中真存在的。所以掌握了这一点, 使用 root 还是 alias 都是可以的。</p>

配置完成后使用 nginx -t 检测配置文件语法是否有问题, 没有问题后就可以 nginx -s reload 用新的配置文件了。

这样在设置中设置 Favicon 路径时, 例如我的文件放在本地/var/solo-upload/myicon/1.ico, 可以这样输入 http://yourdomain.com/upload/myicon/1.ico

<p>ok, 这样本地的图片也可以通过链接的方式引用了, 其实这也是早期 Solo 版本图片存放本地, 后升级为不再使用本地图片时, 解决已发布文章的正常显示方法。</p>

<h2 id="配置ssl证书">配置 ssl 证书</h2>

<p>完成上面的步骤之后, 网站是已经可以运行的了, 但是没有 ssl 证书, 不是 https 访问, 总是会得少了点什么。所以不如一次到位, 全都整好。</p>

<p>我使用的是免费的 ssl 证书, 每个证书服务商提供的免费 ssl 证书都大同小异, 申请完成后你会到 example.crt 和 example.key 两个文件</p>

<p>你需要把这两个文件上传至服务器中, 路径随你喜好但是你要记得, 此处我举例路径为/etc/ssl/mycert/</p>

<p>配置 nginx</p>

```
<pre><code class="highlight-chroma">upstream backend {
    server localhost:8080;
}
server {
```

```
listen    80;
```

```
server_name www.example.com;
```

```
access_log off;
```

```
return 301 https://$server_name$request_uri;#http请求全部转发至https访问
```

```
}
```

```
server {
```

```
listen    443; # 修改监听接口
```

```
server_name example.com www.example.com;
```

```
charset utf8; # 修改默认字符
```

```
ssl on; # 开启ssl
```

```
# 设定你的ssl证书
```

```
ssl_certificate /etc/ssl/mycert/example.crt;#刚才让你上传的example.crt文件路径
```

```
ssl_certificate_key /etc/ssl/mycert/example.key;#刚才让你上传的example.key文件路径
```

```
# 原来http访问路径不需要改动
```

```
location / {
```

```
    proxy_pass http://backend$request_uri;
```

```
    proxy_set_header Host $host:$server_port;
```

```
    proxy_set_header X-Real-IP $remote_addr;
    client_max_body_size 10m;
}
location /upload/ {
    alias /var/solo-upload/;
}
}
```

</code></pre>

<p>这样配置完成后，和之前配置 nginx 访问本地资源一样，nginx -t 检查是否语法有问题，无问之后 nginx -s reload 即可。</p>

<p>最后要重启 docker，并切记的在启动代码中修改--server_scheme 的参数为 https，其他不需修改。</p>

<p>再访问网站你可能会发现还是显示不安全，此处我出现这个问题的原因是，我配置 https 访问之，设置的 Favicon 的路径是 http，修改为 https 就解决了。</p>

<h2 id="总结">总结</h2>

<p>说起来都不是什么很大的问题，之所以还是会花这么多时间，其实就是自己的技术还是不到位，有很大的学习空间，希望各位读者也可以继续努力，有能力自己解决小问题。如有问题，可评论提问日常在线。</p>