



链滴

新年新气象，博客迁移，迎来 solo!

作者: [lizhongyue248](#)

原文链接: <https://ld246.com/article/1546352390764>

来源网站: [链滴](#)

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



好久没有更新博客啦！~一部分原因是期末了作业多事情多，还有一部分是 hexo-admin 真的让人头疼，访问速度感人。自己选了很多博客系统，选择的博客有那么点要求

1. 好看！必须的。
2. 我能够自定义！
3. 能够方便管理的（自从用过hexo后觉得这很重要，毕竟我不是部署在 github 上而是自己的服务器）
4. 能够支持自己的 七牛云 更好啦！
5. 必须支持 markdown
6. 上手简单

自己找了好多博客，wordpress、typecho 这些动态博客，还有 hexo 静态博客，都试过了，但是都有满足上面几个条件的，特别是由于只会java和前端的缘故（逃，最后今年九月发现了一款基于 java 的一款小而美的博客系统 solo，都要动手换了，但是发现没有太喜欢的皮肤，所以在[皮肤推荐](#)中推荐一款 [hexo-even](#)，当初就是因为这款主题用上了 hexo，宁愿受点苦都要用这个主题(可见一款好的皮真的会吸引人的哈哈)，原本打算等 V姐 出了后自己就立刻换的（自己前端真的渣），但是在这次 12 月发布的 2.9.7 版本上，加了一个新的皮肤 Jane！简直太棒了，在放假作业完成后，立刻花时间把他排到了我的服务器上，不论前台还是后台，速度都是很给力！不过中间也大大小小的遇到些问题，记下~~

服务器配置：centos7.2 + jdk8 + nginx

使用端口：8765

访问地址：<https://echocow.cn>

服务器部署

1. 使用 scp 复制 war 包到服务器

```
scp solo-2.9.7.war root@**.**.***.*/root/
```

2. 输入密码复制，然后创建 solo 文件夹并解压 war 包

```
mkdir solo
mv solo-2.9.7.war solo
cd solo/
jar -xvf solo-2.9.7.war
```

3. 修改配置文件

```
cd WEB-INF/classes/
vim latke.properties
```

```
#### Server ####
# Browser visit protocol
serverScheme=https
# Browser visit domain name 最终要访问的路径
serverHost=echocow.cn
# Browser visit port, 80 as usual, THIS IS NOT SERVER LISTEN PORT! 我使用 https , 所以指定 4
3
serverPort=443
```

```
vim local.properties
# 添加自己数据库的密码，我使用mysql数据库
```

4. 登录数据库创建 solo 库

```
>>> create database solo;
```

5. 开启后台服务

```
# 通过 -lp 选项指定部署在 8765 端口
nohup java -cp "WEB-INF/lib/*:WEB-INF/classes" org.b3log.solo.Starter -lp 8765 >/dev/null 2
&1 &
```

solo 也就算部署完毕啦

为什么不用 tomcat? 因为我自己的 tomcat 已经部署多个应用并且还要去修改配置文件，十分麻烦
所以选择这种方式运行，8080端口被tomcat使用，由nginx 转发，443、80 端口被 nginx 使用，所
随便选了一个 8765 端口进行部署。

hexo 迁移

不得不说最棒的就是提供了 hexo 的迁移，让我省去好多功夫！

直接复制 hexo 下的 source 文件夹到 solo/markdown 即可

```
# 我直接移动过去了==
mv source ../solo/
# 然后终于可以停止掉慢出翔的 hexo-admin 啦
[root@EchoLZY blog]# ps -ef | grep hexo
root      647 29099  0 22:34 pts/1    00:00:00 grep --color=auto hexo
root     15475 27799  0 2018 ?        00:50:17 node /root/blog/hexo_run.js
```

```
root 15481 15475 0 2018 ? 00:05:35 hexo
[root@EchoLZY blog]# kill -9 15475
[root@EchoLZY blog]# kill -9 15481
```

nginx 配置

要求，支持 https 访问。同时访问 <http://echocow.cn>、<http://www.echocow.cn>、<https://www.echocow.cn> 都直接转到 <https://echocow.cn> 上面去

1. 由于以前配置 hexo 的时候已经配置好了 ssl 证书，所以需要改动的只有 location 那里，对于 <http://echocow.cn>、<http://www.echocow.cn>、<https://www.echocow.cn> 这三个都是用重定向即可如下：

```
server {
    listen 80;
    server_name echocow.cn, www.echocow.cn;
    access_log logs/http_echocow.access.log main;
    location / {
        return 301 https://echocow.cn$request_uri;
    }
}

server {
    listen 443;
    server_name www.echocow.cn;
    ssl_certificate cert/www.echocow.cn.crt;
    ssl_certificate_key cert/www.echocow.cn.key;
    access_log logs/https_www_echocow.access.log main;
    location / {
        return 301 https://echocow.cn$request_uri;
    }
}
```

2. 配置 <https://echocow.cn>

```
server {
    listen 443 ssl;
    server_name echocow.cn;
    ssl_certificate cert/echocow.cn.crt;
    ssl_certificate_key cert/echocow.cn.key;
    access_log logs/https_echocow.access.log main;
    # 以前的 hexo 配置
    #location / {
    #    root /usr/local/nginx/html/public;
    #    index index.html index.htm;
    #}
    # 现在的 solo 配置
    location / {
        proxy_pass http://127.0.0.1:8765$request_uri;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;
        client_max_body_size 10m;
    }
}
```

3. 重启 nginx 即可

```
cd /usr/local/nginx/sbin  
./nginx -s reload
```

问题

从 hexo 迁移过来的数据时不时包 nginx 502 错误,

1. 查看 access 日志没有发现什么, 查看 nginx error 日志发现错误如下

```
tail /var/log/nginx/error.log  
2019/01/01 21:18:55 [error] 24232#0: *112566 upstream sent too big header while reading re  
ponse header from upstream, client: 1.48.57.4, server: echocow.cn, request: "GET /articles/201  
/02/12/1546344575473.html?ajax=true HTTP/1.1", upstream: "http://127.0.0.1:8765/articles/2  
18/02/12/1546344575473.html?ajax=true", host: "echocow.cn", referer: "https://echocow.cn/  
  
2019/01/01 21:18:55 [error] 24232#0: *112566 upstream sent too big header while reading re  
ponse header from upstream, client: 1.48.57.4, server: echocow.cn, request: "GET /articles/201  
/02/12/1546344575473.html HTTP/1.1", upstream: "http://127.0.0.1:8765/articles/2018/02/12  
1546344575473.html", host: "echocow.cn", referer: "https://echocow.cn/"
```

2. **sent too big header**说得很明确, 发送太大的头了, 所以配置下nginx.conf

```
vim conf/nginx.conf  
# 在http块内添加如下  
proxy_buffer_size 128k;  
proxy_buffers 32 32k;  
proxy_busy_buffers_size 128k;
```

页面渲染问题

使用默认的渲染引擎造成的结果就是, 中间含有代码块有序列表全部都是1所以不得已要使用 marked 进行渲染。

```
cd solo  
npm install marked --save  
nohup node js/marked/http.js >/dev/null 2>&1 &  
# 自定义的 solo 启动别名, 直接启动  
solo
```

效果就好多啦

总结

总的来说还算顺利, 部署加搭建前后一个小时不到吧, 感觉solo还是挺友好的, 后台方面也好, 分类好, 都挺不错, 话说他的陪图功能简直大爱! 唯一有个缺点就是搜索好丑啊哈哈。自己看看有没有力改再说吧。2019 来了, 再见了 2018.

再见,2018,有得有失。
你好,2019,与君共勉。