

陆初

脑子不太好用的普通人。

HTTPS Available!

本站支持 HTTPS 啦！！！！！！

明明这件事情不算特别难，但我还是拖了那么久才完成，真是不好意思呜呜呜呜。

然后所以要在这里介绍一下实现过程啦！

其实踩了不少的坑，但是最后还是实现了，可喜可贺可喜可贺。希望可以在我在忘记这些坑之前赶紧记录下来！

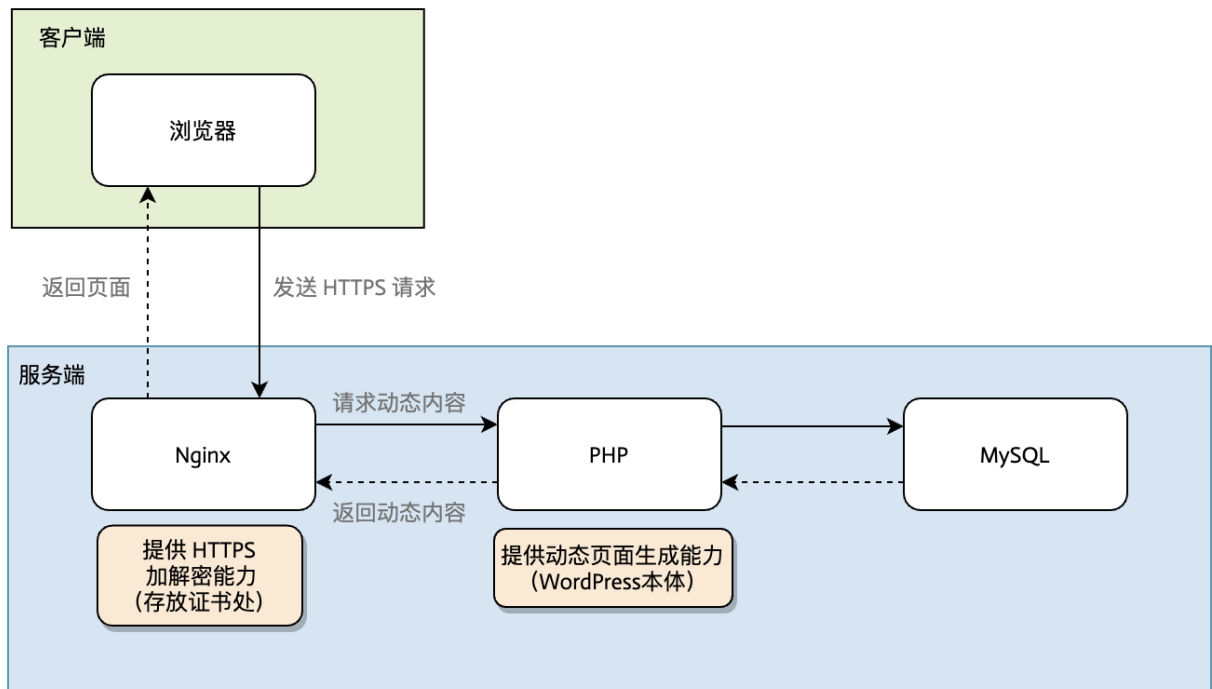
完成的内容

其实呢，我不仅是加上了 https 证书，还做了一些其他的改动：

1. 把官网教程对新人很他妈不友好文档也很他妈杂语法也他妈的完全没有好好介绍的傻逼 Apache 换成了新手教程很友好文档描述也很人性化的 Nginx。
2. 白嫖了阿里云的证书，至少可以撑到这个博客倒闭。
3. 把 DNS 服务迁移到了 Cloudflare，所以现在可以通过子域名 nblog.kazamihatsuroku.top 访问我的 Notion 了
4. 把 php 从远古的 5.6 版本升级到了 7.4

整体服务

用语言描述不太好描述，直接放个图在这里吧。



折腾过程

配置 HTTPS

首先，肯定得整个 SSL 证书。

然后我上了阿里云的证书区，直接给我弹这么一个东西：

云盾证书服务						不使用优惠
配置详情	付费方式	数量	时长	优惠金额	价格	
云盾证书服务类型：证书资源包 是否合并签发：否 域名类型：通配符 证书规格：DigiCert OV通配符 证书个数：1个 证书服务时长：1年	预付款	1	1 年	¥6,000.00	配置费用:¥34,000.00	

直接给孩子整傻了

可能是我农村人见识少，第一次见到那么贵的玩意，这肯定是没办法用的了。所以又得想办法弄一个免费的证书。或者不免费也行，便宜的也能接受。

然后在我找的时候，突然发现阿里云可以白嫖证书！



既然是免费的，那不用白不用，于是赶紧给自己填上域名的信息，很快就通过审核了。

一开始，我采用的是 傻逼Apache 作为我的服务器。所以自然而然地，要通过修改 Apache 的配置文件来完成 HTTPS 的配置。

然后下载了用于 Apache 的证书，一共有三个文件。

1. .key 文件，没猜错的话应该是私钥
2. xxxx_public.crt 文件，没猜错的话应该是公钥
3. xxxx_chain.crt 文件，这我实在不知道是啥，推测应该是对公钥的 hash，也就是证书本体。

然后去网上搜怎么配置 Apache 证书，好家伙，一找就找到了，还是[阿里云官方文档](#)。

然后我高高兴兴地按照文档说的来做，配好这玩意后：

```
120
121 <VirtualHost *:443>
122     ServerName kazamihatsurok
123     DocumentRoot /home/www/h
124     SSLEngine on
125     SSLProtocol all -SSLv2 -S
126     SSLCipherSuite HIGH:!RC4:
127     SSLHonorCipherOrder on
128     SSLCertificateFile cert/5
129     SSLCertificateKeyFile cer
130     SSLCertificateChainFile c
    链开头如果有#字符，请删除。
131 </VirtualHost>
```

基本全抄的阿里云教程

然后我想着那我应该可以开开心心地用 https 打开我的站点了吧！结果：

Testing 123..

This page is used to test the proper operation of the [Apache HTTP server](#) after it has been installed. If you can read this page it means that this site is working properly. This server is powered by [CentOS](#).

Just visiting?

The website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

Are you the Administrator?

You should add your website content to the directory `/var/www/html/`.

To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

Promoting Apache and CentOS

You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!



0

Boom!

直接一炸给我炸回默认配置。

我也不知道是为什么，直接给孩子吓傻了。赶紧去查 Apache 官方文档，是不是我配置写错了。

这[入门指南](#)一开始看着还可以，挺像人话，有模有样的。不错不错。

然后一到[介绍 SSL 配置](#)的部分就开始变得奇怪了。

我本身对这个又不是特别熟悉，看的还是这些密密麻麻的英文文档，就算翻译成中文，也没有办法快速找到自己想要的內容，当时心态是真的特别爆炸。

立即停止

信号：期限

```
apachectl -k stop
```

向父进程发送信号会导致它立即试图杀死其所有子进程。杀死子进程可能需要几秒钟的时间。

而且这个机翻真的有点吓人

所以思考过后，我决定抛弃 Apache，自己下一个 Nginx，改用 Nginx 作为我的 http 服务器。

相比之下，Nginx 的 [新手教程](#) 就舒服多了，对整个语法格式都有一定程度的介绍，不至于打开配置文件后一脸懵逼，还怕自己写错什么。

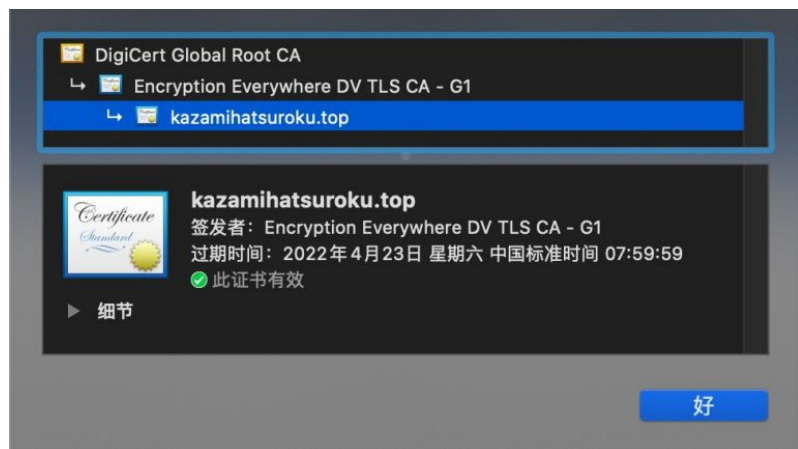
然后打开 [Nginx 的 HTTPS 配置教程](#)，可以说是一目了然，清清楚楚，读完上面的新手教程就能直接开始自己配了。

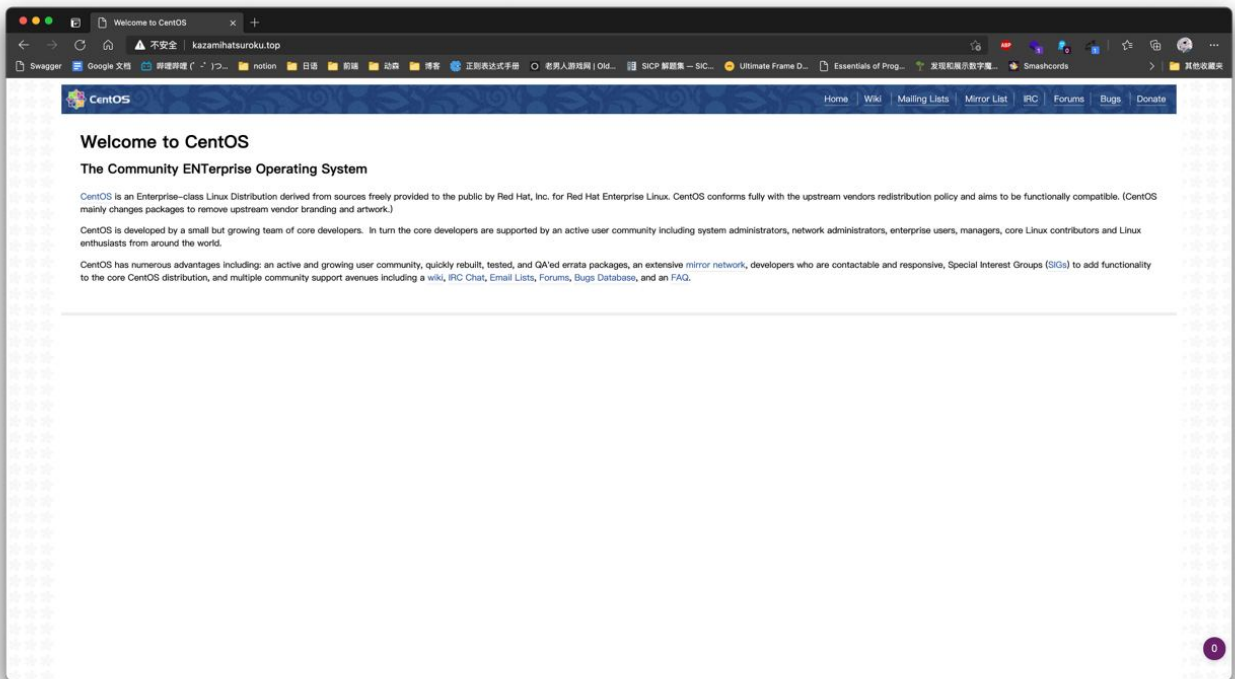
```
59 # Http redirect to https
60     server {
61         listen      80 default_server;
62         listen      [::]:80 default_server;
63         server_name _;
64         return      301 https://$host$request_uri;
65     }
66
67 # Settings for a TLS enabled server.
68     server {
69         listen      [::]:443 ssl http2 default_server;
70         listen      443 ssl http2 default_server;
71
72         keepalive_timeout 70;
73         ssl_certificate "/etc/pki/nginx/cert/5528258_kazamihatsuroku.top";
74         ssl_certificate_key "/etc/pki/nginx/cert/5528258_kazamihatsuroku.top.key";
75         ssl_session_cache shared:SSL:1m;
76         ssl_session_timeout 10m;
77         ssl_ciphers HIGH:!aNULL:!MD5;
78         ssl_prefer_server_ciphers on;
79
80         # Load configuration files for the default server block.
81         include /etc/nginx/wordpress.d/*.conf;
82     }
83
84 }
```

nginx.conf

```
1 # server {
2     ## Your website name goes here.
3     server_name kazamihatsuroku.top;
4     ## Your only path reference.
5     root /home/www/htdocs;
6     ## This should be in your http block and if it is, it's not needed here.
7     index index.php;
8 }
```

wordpress.d/wordpress.conf，当然下面还有一些有关 php 的设置，就不贴了





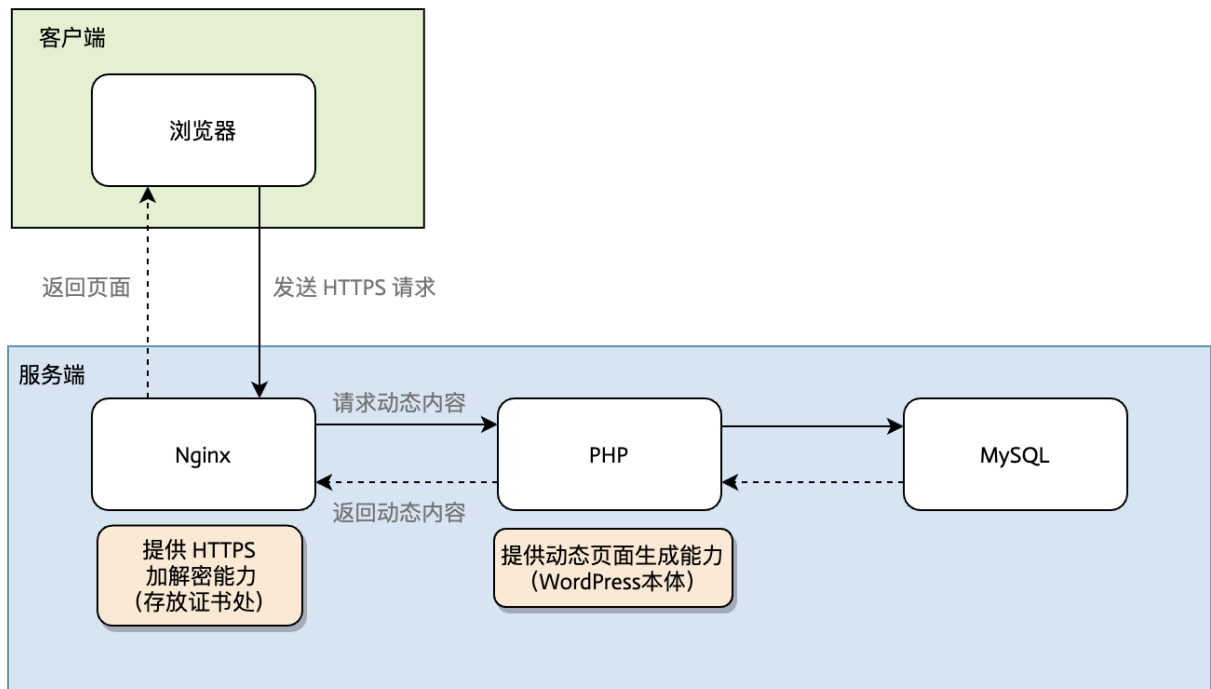
配完之后，证书直接生效，从此以后我也是有高贵的 https 的人了！

到这里，给 Nginx 添加 SSL 证书的部分就完成了。

配置 PHP

虽然 https 是配置完成了，但是我的博客是个 WordPress，不是静态页面啊，这样子肯定是不行的。既然 WordPress 是基于 php 的，我的想个办法让 Nginx 和 php 合作，动态生成页面。

后来找到了一个叫 fastcgi 的东西，[知乎上的这个回答](#) 写得挺不错。简单来说可以理解为这个图中：



Nginx 通过一个叫 fastcgi 的通用协议，调用 php 生成动态内容，php 在完成页面生成后，再把页面返回给 Nginx。而 php 中负责与 Nginx 交互的 fastcgi 程序叫做 php-fpm（这玩意特别重要）。所以我们需要配置好 Nginx 和 php-fpm 两端，才可以让它们之间能够交互。

由于 php 环境和 php-fpm 环境本来就有，所以 php 这边不需要怎么配置，主要配置好 Nginx 就好了。

因为这整个博客是阿里云自动生成的，里面所用到的 Apache、php 和 mysql 都不是用过包管理工具安装的，我猜是直接编译某一个版本的源码，就放在了一个地方。所以网上一些 php 配置文件的路径在我这里并不适用，我需要自己去寻找配置文件所在之处。

然后，我找死找不到。

Where Is The php.ini Configuration File In CentOS?

Posted by Jarrod on April 5, 2017

[Go to comments](#)

[Leave a comment \(0\)](#)

How can we find the default configuration files for the PHP packages in CentOS Linux?

By using the RPM command we can query a package to find all of its associated configuration files. Failing that we can use the find command to search the file system for these files, as we will see here.

Finally we'll cover how we can use a phpinfo file to show us which configuration file is in use.

RPM Query Configuration Files

The options we will use with the RPM command are -q to query a package, and -c which will list the configuration files that are part of the package. As shown below we have run RPM with the -qc options on the php-common package, which is a dependency when installing the php package and is where the php.ini file comes from in CentOS.

```
[root@centos7 ~]# rpm -qc php-common
/etc/php.d/curl.ini
/etc/php.d/fileinfo.ini
/etc/php.d/json.ini
/etc/php.d/phar.ini
/etc/php.d/zip.ini
/etc/php.ini
```

这是网上人们说的

```
[root@izwz93g2vcjbz8srmi1h4zz etc]# l /etc | grep php
[root@izwz93g2vcjbz8srmi1h4zz etc]#
```

我找到的

我??? 我找你妈??? 根本就没有啊???

后来我实在是受不了了，直接运行 `find / -name php`。结果还是没找到：

```
/usr/local/php56/lib/php
/usr/local/php56/php
/usr/local/php56/php/php
/usr/local/php56/include/php
/usr/local/php56/bin/php
```

不是??? 你配置文件呢!

后来我发现，是我自己的锅，`-name` 参数是匹配整个名字的，想要模糊搜索，那得用正则才行。

所以需要运行 `find / -regex ".*php.*"` 。

```
/usr/local/php56/etc
/usr/local/php56/etc/php.ini
/usr/local/php56/etc/php-fpm.conf.default
/usr/local/php56/etc/php-fpm.conf
/usr/local/php56/var
```

总算是给我找到了！

进入 `php-fpm.conf`，可以看到

```
6 [www]
7 listen = /home/www/logs/php-fpm.sock
8 listen.backlog = 128
9
```

咦？这里 `listen` 的怎么是一个文件，一般来讲不都是 `listen` 一个 `ip:port` 的吗？

这里其实会提到一个 `unix` 的概念，那就是万物皆文件。网络端口其实在 `unix` 也会被抽象成一个文件，`ip:port` 最终也需要绑定一个 `socket`，才可以完成通信。这里既然通信的双方都处于同一台机上，那就可以直接通过一个 `unix socket` 文件进行通信，而不需要通过网络进行通信。剩下的有关 `unix socket` 文件的概念可以在网上搜。我也是看到这个才第一次知道原来 `listen` 除了 `ip:port` 还有别的形式。

接下来我们打开 `nginx.conf`：

```
location ~ \.php$ {
    #NOTE: You should have "cgi.fix_pathinfo = 0;" in php.ini
    include fastcgi_params;
    fastcgi_intercept_errors on;
    # The following parameter can be also included in fastcgi_params file
    fastcgi_pass unix:/home/www/logs/php-fpm.sock;
    # fastcgi_pass 127.0.0.1:9000;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}
```

本身 `nginx` 官网就有 `fastcgi` 的教程，按照教程填就行了

到这里，`Nginx` 的部分应该就配置完了，这时候启动 `Nginx`，就应该能正常运行 `WordPress` 了！

然而并没有。

502 Bad Gateway

nginx/1.16.1

让人心肺停止的界面

这，这你吗到底是怎么回事啊！！！！

然后我就在网络上开始搜索「Nginx 报错 502」。（一般这个时候，应该直接去看 Nginx 的报错日志，而不是对着报错现象瞎 jb 乱搜。）

后来查了很久，才有一个回答说，可能是文件的权限访问问题。然后我看了一下我的 nginx.conf：

```
1 # For more information on configuration, see:
2 #   * Official English Documentation: http://nginx.org/en/docs/
3 #   * Official Russian Documentation: http://nginx.org/ru/docs/
4
5 user nginx;
```

再看看放 WordPress 文件夹的访问权限：

```
drwxr-x---  5 www  www  4.0K 7月   8 2018 .
drwxr-xr-x.  7 root  root  4.0K 4月  11 2018 ..
drwxr-x--x  6 www  www  4.0K 4月  26 21:42 htdocs
drwxr-xr-x  2 www  www  104K 4月  26 20:35 logs
drwxr----- 3 www  www  4.0K 7月   8 2018 .pki
```

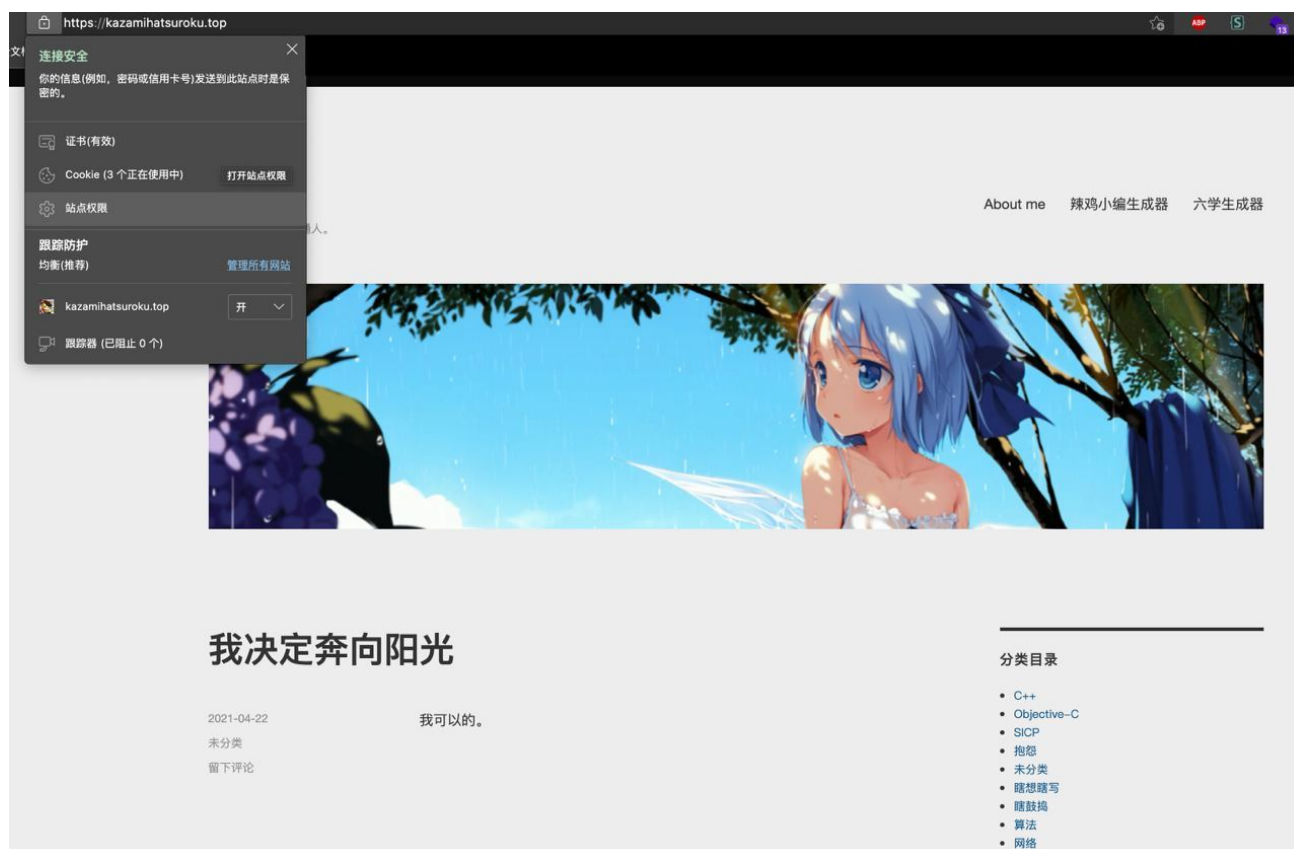
htdocs，其他用户没有读权限！！

也就是说，Nginx 根本没有权限去读取 htdocs 里面的 php 文件，所以很自然地返回了 502。

解决方法也简单，把 nginx.conf 里面的 user 从 nginx 改成 www 就行了，这样 nginx 就会以 www 的用户身份运行，自然也就有权限读 htdocs 了。

当然直接 `sudo chmod -R o+r ./htdocs`，给其他用户增加读权限也行的，两种解决思路。

做完这一步之后，我们终于可以正常运行博客首页了。



可喜可贺！可喜可贺！到这里就可以结束了！

博客无法加载出某些页面

如果真的到这里就可以结束就好了，我也不用折腾那么久了。

我开开心心打开博客准备些这篇总结文章的时候，一些不太对劲的事情发生了。

主页是可以正常打开的，但是其他页面，特别是设置页面，就不可以了，点击新建文章更是一片空白，什么都没有。

于是我 F12 打开控制台准备看看 network，结果发现整个 php 文件加载都失败了，报了一个 `net::ERR_HTTP2_PROTOCOL_ERROR 200` 的错误。

然后我就搜啊搜，搜了差不多一个小时，还以为是什么 http2 协议的问题，吓得我赶紧把 `nginx.conf` 里面有关 http2 的内容全部删掉了，然而并没有什么乱用，还是报错了。

在我实在失去头绪的时候，我才突然想起来，一般这种软件都是有 log 功能的，nginx 肯定在什么地方存了一个报错的 log，比起查这种表面上的错误，不如直接看 log 报了什么错。

打开 `/var/log/nginx/error.log`，最后的 log 写着这么一句话：

```
628 open() "/var/lib/nginx/tmp/fastcgi/1/04/0000000041" failed
(13: Permission denied)
```

这？？这又是咋回事？？怎么又是权限问题？？

然后进入 `/var/lib/nginx/tmp` 一看，好家伙，所有者是 nginx，除了所有者和所有者所在的用户组之外的其他人没有读权限。因为我改了配置，nginx 是以 www 的身份启动的，所以自然没有读这个文件夹的权限。这波啊，这波是什么，这就是伏笔啊观众们！

别的不说，直接 `sudo chmod -R a+r /var/lib/nginx`，给所有人加个读权限。

这样子，设置页面和新文章页面也能正常打开了，可喜可贺，可喜可贺，这下博客总算处于正常能用的状态了。

把 DNS 解析转移到 cloudflare，加上对 Notion 博客的重定向

其实在没有配置好之前，我有好长一段时间没用过这个博客了，一是因为慢（……），二是看到 http 就不是很想点进去。所以在这段时间里都是使用 [初陆 \(notion.so\)](https://notion.so) 作为博客的。

既然现在的博客变得可用了，那自然的，也想要把 notion 博客也给套上个二级域名，这样就不用通过一大串连接来访问我的 notion 博客了。

具体怎么做在这里（[利用 URL 转发给你的主页一个自己的网址 \(notion.so\)](#)）已经讲得很清楚了，我就不再赘述了。总之，现在我的 DNS 已经交给 cloudflare 托管，所以现在你可以看到我的 https 证书不再是上面截图的那个了，而是 cloudflare 的证书。但这并不代表我之前的证书就没用了。

因为实际上 cloudflare 相当于一个反向代理。你输入 `kazamihatsuroku.top` 之后，会首先访问 cloudflare，cloudflare 再把你带到我的服务器。所以，虽然没有配置 https 的页面在接入 cloudflare 之后会有 cloudflare 的证书，但是在 cloudflare 和 真正的服务器之间的通信就是不安全（没有加密）的了。

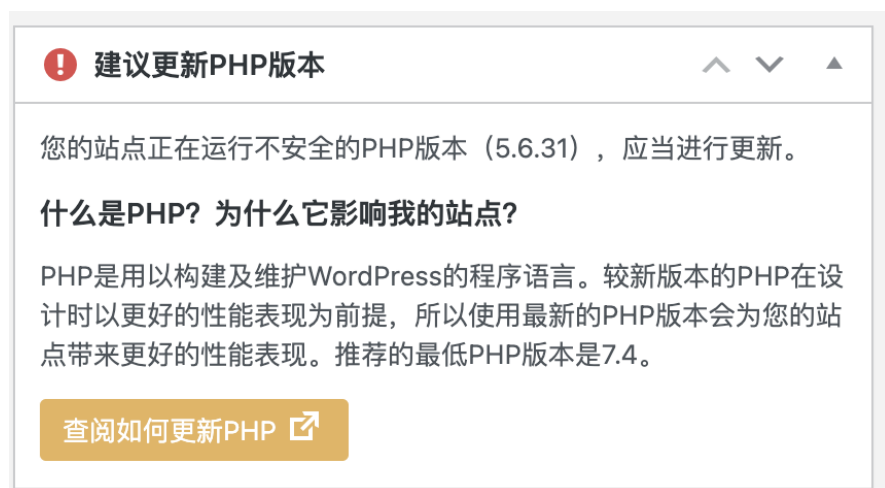
所以，现在 nblog.kazamihatsuroku.top 就会指向我的 notion 博客啦！

升级 php 版本

看上去一切都完成了，是吗？

我也是这么觉得的。

但。是。



我千算万算没算到，居然是 WordPress 官方来嫌我的 php 版本太低

那咋办啊！

没办法，我本以为这样就折腾完了，那现在看来只能把 php 给升级了！

然后我执行 `yum install php`！然后直接 `php -v`！

```
PHP 5.6.31 (cli) (built: Aug  7 2017 22:59:26)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
```

——嗯????????

搜了一下才发现，我的这个 CentOS 7 所带的 yum 提供的 php 版本最高只到 5.6。想要安装更高版本的 php，需要另想法子。

我当时的参考方法已经忘了，上网现搜了一个：[快如闪电的安装php7.4套件（centos 7） | PHP 技术论坛 \(learnku.com\)](#)，应该也是一样的。

噢对了，我在自己安装的时候，发现没有带上 php-fpm，我还以为是 yum 提供的版本不带 php-fpm，得自己下载 php 源码编译，但其实是因为我在安装的时候命令行没带上，只要带上了就好了。我看了一下上面的链接，是有带上的，所以应该没有问题。

安装完之后，我现在的服务器就有两个 php 版本共存了，一个是 php5.6，一个是 php7.4。但实际上，系统开机自启的究竟是哪个版本的 php 呢？

而我又恰好对 linux 的了解浅得要死，也就只知道 ls cd 之类的命令。我根本就想不到要怎么去查看对应的是哪个版本。

然后我花了大概一整个下午去找 systemctl 和 service 相关的内容，把 [17.3 systemctl 针对 service 类型的配置文件 | 鸟哥的 Linux 私房菜：基础学习篇 第四版 \(gitbooks.io\)](#) 和 [systemd.service \(www.freedesktop.org\)](#) 都给大概看了一遍，才知道怎么确定运行的 service 实际上是在哪个。

然后我找到 `/usr/lib/systemd/system/php-fpm.service`，查看里面的内容：

```
# It's not recommended to modify this file in-place, because it
# will be overwritten during upgrades.  If you want to customize,
# the best way is to use the "systemctl edit" command.

[Unit]
Description=The PHP FastCGI Process Manager
After=syslog.target network.target

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/php-fpm
ExecStart=/usr/sbin/php-fpm --nodaemonize
ExecReload=/bin/kill -USR2 $MAINPID
PrivateTmp=true
RuntimeDirectory=php-fpm
RuntimeDirectoryMode=0755

[Install]
WantedBy=multi-user.target
```

可以看到，运行的是 `/usr/sbin/php-fpm`，我自己看了一下，发现这个就是 7.4 版本的 php-fpm，真棒！

那么接下来只需要看 php-fpm 的配置，然后和刚才一样和 nginx 对应配置一下就行了！

由于是通过 yum 安装的，配置文件所在的位置也十分规范，在 `/etc/php-fpm.d` 里。

进去之后，发现里面有一个 `www.conf` 文件，这应该就是我们想要找的配置文件。

```
; Note: This value is mandatory.  
listen = 127.0.0.1:9000
```

可以看到，这次不是 unix socket 文件了，而是 ip:port。那我们要把 nginx 的 cgi_pass 地址也改成这个地址才行。

改完之后，应该就大功告成了！让我们打开博客吧！

No such file or directory

建立数据库连接时出错

这意味着您在 wp-config.php 文件中指定的用户名和密码信息不正确，或我们未能在 localhost 联系到数据库服务器。这可能意味着您主机的数据库服务器未在运行。

- 您确定用户名和密码正确吗？
- 您确定输入的主机名正确吗？
- 您确定数据库服务器在运行吗？

如果您不明白这些意味着什么，您应该联系您的主机提供商。如果您仍需要帮助，请访问[WordPress支持论坛](#)。

好的，不要慌，出现这个至少说明我们的 WordPress 是可以运行的，只是运行起来有点问题。

——等等！究竟是什么问题啊！我的 php-fpm 没有配错才对的啊？我没有改过任何的数据库文件啊？？

然后我切回了旧版本的 php-fpm，一切正常，没有任何问题。

然后又切回了 7.4 版本的 php-fpm，依旧报错。

用的是同一份数据库文件，同一个 nginx。

这就很神奇了！

然后我搜了好多相关的内容，大部分说的都是 wp-config.php 上的数据库配置有问题，但我试着用里面的配置去连接数据库，发现没有任何的问题。

所以，怎么回事啊！

然后我突然发现，wp-config.php 上面有这么一行：

```
69 /**
70  * 开发者专用：WordPress调试模式。
71  *
72  * 将这个值改为true，WordPress将显示所有用于开发的提示。
73  * 强烈建议插件开发者在开发环境中启用WP_DEBUG。
74  *
75  * 要获取其他能用于调试的信息，请访问Codex。
76  *
77  * @link https://codex.wordpress.org/Debugging_in_WordPress
78  */
79 define('WP_DEBUG', true);
"./wp-config.php" [dos] 96L, 2910C
```

卧槽，这不就是我需要的吗！

打开之后，页面变成了这样：

Warning: mysqli_real_connect(): (HY000/2002): No such file or directory in
/home/www/htdocs/wp-includes/wp-db.php on line 1653

No such file or directory

建立数据库连接时出错

这意味着您在wp-config.php文件中指定的用户名和密码信息不正确，或我们未能在localhost联系到数据库服务器。这可能意味着您主机的数据库服务器未在运行。

- 您确定用户名和密码正确吗？
- 您确定输入的主机名正确吗？
- 您确定数据库服务器在运行吗？

如果您不明白这些意味着什么，您应该联系您的主机提供商。如果您仍需要帮助，请访问[WordPress支持论坛](#)。

然后我进去看了一下那一行，是用来连接数据库的语句：

```
1651
1652         if ( WP_DEBUG ) {
1653             mysqli_real_connect( $this->dbh, $host, $this->d
1654                 buser, $this->dbpassword, null, $port, $socket, $client_flags );
```

也就是说，确实是连接数据库的时候出了问题。

那到底是什么问题呢！！！！！！

然后这个问题卡了我大概三个小时左右，怎么找也找不到答案。

然后我看到了一个答案说：

「尝试把你的 localhost 改成 127.0.0.1」

我抱着**死马当作活马医**的想法，改了一下。

然后，成功了。

成功了。

成功了。

成功了？

成功了！

为啥啊！！！！！！！！

为啥啊！！！！！！！！！！

不是，这也太弱智了！！ 怎么回事啊！！！！ 为什么 localhost 不行 127.0.0.1 就行啊！！！！

然后我去查阅了相关资料，mysql 是这么说的：

B.3.2.2 Can't connect to [local] MySQL server

A MySQL client on Unix can connect to the **mysqld** server in two different ways: By using a Unix socket file to connect through a file in the file system (default `/tmp/mysql.sock`), or by using TCP/IP, which connects through a port number. A Unix socket file connection is faster than TCP/IP, but can be used only when connecting to a server on the same computer. A Unix socket file is used if you do not specify a host name or if you specify the special host name `localhost`.

[MySQL :: MySQL 8.0 Reference Manual :: B.3.2.2 Can't connect to \[local\] MySQL server](#)

也就是说，如果在连接的时候没指定域名，或者使用了 localhost，则会采用 unix socket 文件的方式进行连接而不是把 localhost 解析成 127.0.0.1 再进行网络连接。

改成 127.0.0.1 之后，就会走网络连接而不是 unix socket 文件连接了。

其实到这里就算是全部解决了，但是我还是想配置一下使用 unix socket 文件进行连接。

于是我打开 `/etc/php.ini`，并且把所有有关 mysql 连接的 `default_socket` 都改成了 `/etc/my.cnf` 里的指定的 socket（一般是 `/tmp/mysql.sock`）。然后我再次尝试使用 `localhost` 进行连接——

不行。

我还以为是我的 `default_socket` 设置失败了，于是又重新检查了一次，又用 `php -r 'echo phpinfo();' | grep default_socket` 检查了一遍，确定是在新版本的 php 里配置好的。

但不知道为什么，就是没有生效，还是没有办法连接到数据库。尽管网上很多教程都说保持这几个配置的 `.sock` 文件一致就行了，但是我就算配成了一致，也还是没有办法连接，重启服务，重启机器也是一样的结果。

算了，反正用 127.0.0.1 也能跑起来，暂时就到这里吧。这就当做一个遗留问题，等到以后再解决了。

到此，折腾总算是结束了。

折腾完了，然后呢？

其实这次折腾还是给我带来了不少好处的。

首先是对整个博客的运作方式有了更深的认识，本来我就只是想当做一个无代码配置的博客使用的（就像是 notion 那样），但越写越发现……不配不行。虽然这个折腾的过程很艰辛，也踩了不少坑，但至少也学到了不少东西，所以收获还算是非常大的。

接下来打算折腾什么？

大家访问的时候应该也感觉出来了，我的博客页面加载 **非常慢**。我看了一下，大概还是因为服务器配置太低的原因。单核，2G 内存，1 Mbps 的网络，摆明就是学生试用机。

所以接下来我会想办法把这个博客给备份下来，然后看看跑在自己电脑上的 WSL 上会怎么样。如果可行的话，说不定以后我就不用买服务器了，想办法搞个配置好点的瘦机放在出租屋里就当服务器用（笑）

结语

那么就先到这里啦！有疑问的都可以留言或者联系我！拜拜！



作者：陆初

脑子不太好用的普通人。顺带一提性格也有点古怪。在老妈子和厌世肥宅中来回切换。

[查看陆初的所有文章](#)



陆初 / 2021-04-26 / 瞎鼓捣 /

此站点使用Akismet来减少垃圾评论。[了解我们如何处理您的评论数据。](#)

陆初 / 自豪地采用WordPress