使用 acmesh 免费开启 https（详细概念介绍与操作步骤记录）

作者：JellyfishMIX

原文链接：https://ld246.com/article/1585995449808

来源网站：链滴

前言

记录一次使用acme.sh免费开启https的过程，前半部分列举一些用到的概念，后半部分记录具体操作骤。**文章已调整好线性阅读顺序，按顺序阅读即可。流程中涉及的概念会尽量进行讲解，以减少阅读篇文章时，额外检索产生的时间消耗。**

概念

acme.sh

acme.sh是github上的一个开源项目，实现了acme协议, 可以从letsencrypt生成免费的证书。

官方文档（官方文档的使用说明很详细，推荐阅读）：

英文：https://github.com/acmesh-official/acme.sh

中文：https://github.com/acmesh-official/acme.sh/wiki/说明

acme.sh 有以下特点（摘自官方文档）：

- 一个完全使用用Shell（Unix shell）语言编写的ACME协议的客户端
- 支持ACME v1和ACME v2协议
- 支持ACME v2通配符证书
- 简单，强大且非常易于使用。 您只需3分钟即可学习
- 和bash，dash, sh兼容
- Let's Encrypt免费证书客户端最简单的shell脚本
- 完全用Shell编写，不依赖python或官方的 Let's Encrypt客户端
- 只需一个脚本即可发布，续期和自动安装证书
- 不需要root/sudoer权限
- 对Docker友好的
- 支持IPv6
- 对证书续期和错误等有cron job通知

Let's Encrypt

Let's Encrypt是一个于2015年三季度推出的数字证书认证机构，旨在以自动化流程消除手动创建和装证书的复杂流程，并推广使万维网服务器的加密连接无所不在，为安全网站提供免费的SSL/TLS证。

-- 摘自 维基百科

网站开启https的时候需要证书，证书由CA机构（数字证书认证机构）签发，大部分传统CA机构签发书需要收费，这不利于https协议的推广。Let's Encrypt也是一个CA机构，但它是免费签发数字证书，通过它，我们可以免费开启https

ACME（自动证书管理环境）

ACME协议最初是由 Internet Security Research Group 为其公共 CA（公共证书颁发机构）——Let's Encrypt 开发的。ACME 协议通过在给定 Web 服务器上安装证书管理代理来运行。组织或域在一开就经过验证，代理协助域控制验证，一旦完成，代理可以请求，续订和撤销证书。

详情：ACME 协议：它是什么以及如何工作——asiaregister.com

ACME协议具体的工作流程这里就不细说了，感兴趣的朋友可以去详情即原出处查看。

全站https，通配符证书

通配符证书是一个可以被多个子域使用的公钥证书，主域名签发的通配符证书可以在所有子域名中使。在此之前，配置子域名也是需要每个子域名单独的申请证书的。2018年3月14日，Let’s Encrypt 外宣布ACME v2已正式支持通配符证书，这意外味着用户可以在 Let’s Encrypt 上免费申请支持通符的SSL证书。

具体操作

官方文档（官方文档的使用说明很详细，推荐阅读）：

英文：https://github.com/acmesh-official/acme.sh

中文：https://github.com/acmesh-official/acme.sh/wiki/说明

本文使用的操作系统（Linux各版本操作步骤基本一致）：CentOS 7.3

1. 安装acme.sh

输入

```
curl https://get.acme.sh | sh
```

或

```
wget -O -  https://get.acme.sh | sh
```

curl 命令

curl 命令是一个利用URL规则在命令行下工作的文件传输工具。它支持文件的上传和下载，所以是综传输工具。

详情：curl命令——linuxde.net

wget命令

wget 命令用来从指定的URL下载文件。wget非常稳定，它在带宽很窄的情况下和不稳定网络中有很的适应性，如果是由于网络的原因下载失败，wget会不断的尝试，直到整个文件下载完毕。如果是务器打断下载过程，它会再次联到服务器上从停止的地方继续下载。这对从那些限定了链接时间的服器上下载大文件非常有用。

详情：wget命令——linuxde.net

acmesh安装内部流程

普通用户和 root 用户都可以安装使用. 安装过程进行了以下几步:

1. 把 acme.sh 安装到你的 home 目录（即~目录）下:

~/.acme.sh/

并创建一个 bash 的 alias, 方便你的使用: alias acme.sh=~/.acme.sh/acme.sh

(alias：中文释意"别名")

2. 自动为你创建 cronjob, 每天 0:00 点自动检测所有的证书, 如果快过期了, 需要更新, 则会自动新证书。

更高级的安装选项请参考: https://github.com/Neilpang/acme.sh/wiki/How-to-install

安装过程不会污染已有的系统任何功能和文件, 所有的修改都限制在安装目录中: ~/.acme.sh/

cron job

工具型软件cron是一款类Unix操作系统下的基于时间的任务管理系统。用户们可以通过cron在固定间、日期、间隔下，运行定期任务（可以是命令和脚本）。cron常用于运维和管理，但也可用于其他方，如：定期下载文件和邮件。cron该词来源于希腊语chronos（χρόνος），原意是时间。

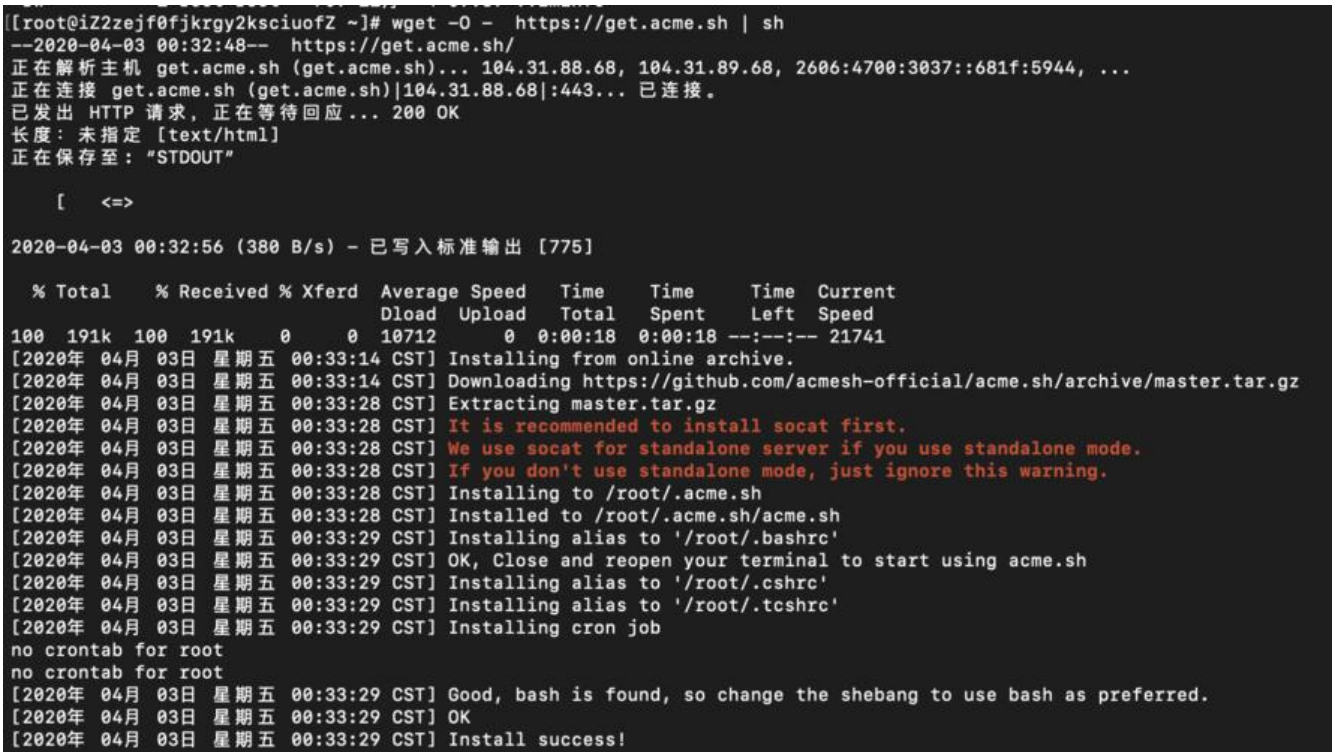
——摘自 维基百科

博主最先使用curl命令进行安装，但由于网络原因，失败了：

```
[root@iZ2zejf0fjkrgy2ksciuofZ ~]# curl https://get.acme.sh | sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   775    0   775    0     0    538      0 --:--:--  0:00:01 --:--:--   538
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0curl: (7) Failed connect to raw.githubusercontent.com:443; 拒绝连接
[root@iZ2zejf0fjkrgy2ksciuofZ ~]# ls -la
总用量 84
dr-xr-x---.  8 root root  4096 4月   3 00:31 .
dr-xr-xr-x. 18 root root  4096 11月 27 02:28 ..
-rw-------   1 root root 12303 4月   3 00:30 .bash_history
-rw-r--r--.  1 root root    18 12月 29 2013 .bash_logout
-rw-r--r--.  1 root root   176 12月 29 2013 .bash_profile
-rw-r--r--.  1 root root   176 12月 29 2013 .bashrc
drwx------   3 root root  4096 8月  18 2017 .cache
-rw-r--r--.  1 root root   100 12月 29 2013 .cshrc
drwxr-xr-x   2 root root  4096 2月  19 23:38 home
-rw-------   1 root root  1223 2月  19 23:46 .mysql_history
drwxr-xr-x   2 root root  4096 8月  18 2017 .pip
drwxr-----   3 root root  4096 4月   3 00:31 .pki
drwxr-xr-x   3 root root  4096 11月  5 23:04 Programming
-rw-r--r--   1 root root    64 8月  18 2017 .pydistutils.cfg
-rw-------   1 root root    20 12月  8 19:38 .rediscli_history
drwx------   2 root root  4096 11月  1 19:15 .ssh
-rw-r--r--   1 root root     0 12月 27 13:36 status
-rw-r--r--.  1 root root   129 12月 29 2013 .tcshrc
-rw-------   1 root root   709 11月  4 09:39 .viminfo
```

如果安装成功，~目录下，使用ls -la可以查看到有一个.acme.sh目录。

curl 命令安装失败后，改用wget命令，成功安装：



安装信息中有一段醒目的红色警告：

It is recommended to install socat first.We use socat for standalone server if you use standalone mode.If you don't use standalone mode, just ignore this warning.

即：

推荐先安装socat。如果你使用standalone mode，那么我们需要为了standalone server使用socat
如果你不使用standalone mode，那么请忽略这条警告

standalone mode 非开启https必须，在官方英文文档中被另一些配置用到（4. Use Standalone server to issue cert 5. Use Standalone ssl server to issue cert），如果感兴趣的话可以去看官方英文档。

简略介绍一下socat：

socat

socat 是一个多功能的网络工具，名字来由是“Socket CAT”，可以看作是 netcat 的加强版。它有些netcat所不具备却又很有需求的功能，例如ssl连接。socat是强大的，可以实现任意socket的转换而netcat被称为网络工具中的瑞士军刀，体积小巧，但功能强大。netcat可以在两台设备上面相互交，即侦听模式/传输模式。

想要安装socat的话，可以使用yum安装socat：

yum install socat

2. 生成证书

acme.sh 实现了 acme 协议支持的所有验证协议. 一般有两种方式验证: http 和 dns 验证。

1.http 方式（推荐）

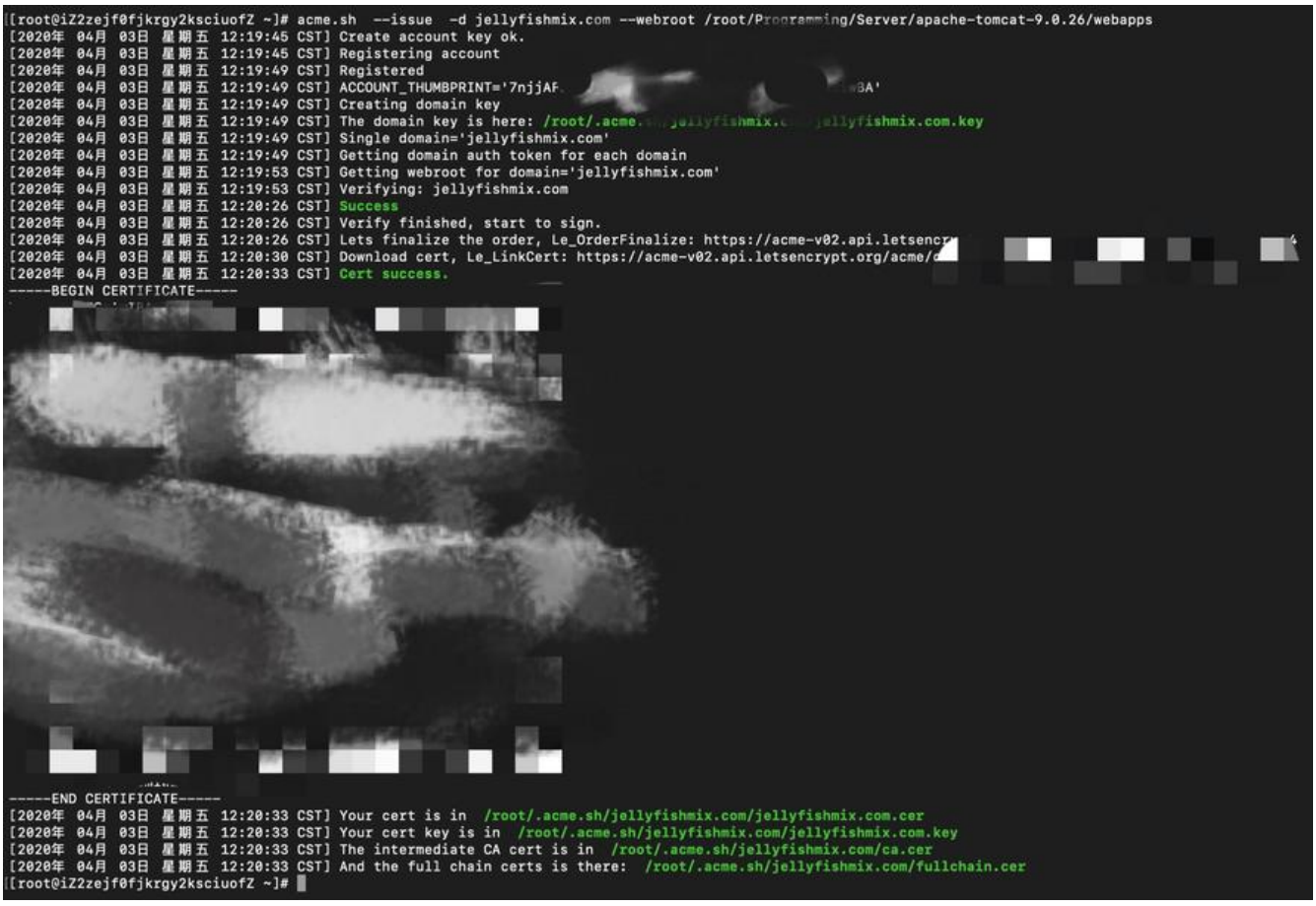
http 方式需要在你的网站根目录下放置一个文件, 来验证你的域名所有权。完成验证， 然后就可以生证书了。

acme.sh --issue -d mydomain.com -d www.mydomain.com --webroot /home/wwwroot/mdomain.com/

只需要指定域名, 并指定域名所在的网站根目录。 acme.sh 会全自动的生成验证文件, 并放到网站的目录, 然后自动完成验证。最后会聪明的删除验证文件，整个过程没有任何副作用。

第二个参数"example.com" 是您要为其颁发证书的主要域。这里至少要填写一个域名。

博主的网站根目录填写的是tomcat服务器的webapps目录,文章后面有不需要你不需要指定网站根目的办法。nginx服务器在80端口做转发，转发到8080端口的tomcat服务器。



如果你用的 apache服务器, acme.sh 还可以智能的从 apache的配置中自动完成验证, 你不需要指定站根目录:

acme.sh --issue -d mydomain.com --apache

如果你用的 nginx服务器, 或者反代, acme.sh 还可以智能的从 nginx的配置中自动完成验证, 你不需指定网站根目录:

acme.sh --issue -d mydomain.com --nginx

原文链接：使用 acmesh 免费开启 https（详细概念介绍与操作步骤记录）

证书每60天自动更新一次。

请注意, **无论是 apache 还是 nginx 模式, acme.sh在完成验证之后, 都会将服务器配置文件恢复到前的状态, 不会私自更改你本身的配置。** 好处是你不用担心配置被搞坏, 也有一个缺点, **你需要自己配置 ssl 的配置。acme.sh只能成功生成证书, 需要手动配置ssl，才能访问https。**这样做虽然麻烦，但是了配置的安全, 你还是自己手动改配置吧。

这里"你需要自己配置 ssl 的配置"的意思是：

为服务器安装ssl模块（Apache默认没有预装，需要自行安装。nginx默认预装了ssl模块，无需再次装）。然后在服务器的配置文件中，写证书位置（在第3.步中我们将完成此项操作）。

2.手动 dns 方式（如使用第一种，可忽略第二种方法）

手动在域名上添加一条 txt 解析记录, 验证域名所有权。

这种方式的好处是, 你不需要任何服务器, 不需要任何公网 ip, 只需要 dns 的解析记录即可完成验证。坏处是，如果不同时配置 Automatic DNS API，使用这种方式 acme.sh 将无法自动更新证书，每次需要手动重新解析验证域名所有权。

第二种使用方式请见官方文档（文章开头链接），博主使用的http方式，手动dns方式这里不细说了。

3. copy/安装 证书

前面证书生成以后, 接下来需要把证书 copy 到真正需要用它的地方。

请注意，默认生成的证书都放在安装目录下: ~/.acme.sh/, 请不要直接使用此目录下的文件，例如: 不直接让 nginx/apache 的配置文件使用这下面的文件。这里面的文件都是acmesh工具内部使用, 目录构在将来可能会变化，进而导致服务器配置文件中填写的证书路径错误的情况。

正确的使用方法是使用 --installcert 命令,并指定目标位置, 然后证书文件会被copy到相应的位置, 例如:

Apache example

```
acme.sh --installcert -d example.com \
--cert-file      /path/to/certfile/in/apache/cert.pem  \
--key-file       /path/to/keyfile/in/apache/key.pem  \
--fullchain-file /path/to/fullchain/certfile/apache/fullchain.pem \
--reloadcmd     "service apache2 force-reload"
```

Nginx example

```
acme.sh --installcert -d example.com \
--key-file       /path/to/keyfile/in/nginx/key.pem  \
--fullchain-file /path/to/fullchain/nginx/cert.pem \
--reloadcmd     "service nginx force-reload"
```

(一个小提醒, 这里用的是 service nginx force-reload, 不是 service nginx reload, 据测试, reload 并会重新加载证书, 所以用的 force-reload)

Nginx 的配置 ssl_certificate 使用 /etc/nginx/ssl/fullchain.cer ， 而非 /etc/nginx/ssl/<domain>.cr ， 否则 SSL Labs 的测试会报 Chain issues Incomplete 错误。

--installcert命令可以携带很多参数, 来指定目标文件。 并且可以指定 reloadcmd, 当证书更新以后, reoadcmd会被自动调用,让服务器生效。

–reloadcmd “service nginx force-reload” 是为了在让acmesh 自动更新时候能够重启nginx使得书生效。

值得注意的是, 这里指定的所有参数都会被自动记录下来, 并在将来证书自动更新以后, 被再次自动调。

首先，我们需要新建一个路径用于存放拷贝的证书（路径可自定义）。习惯是放在/etc/nginx/ssl/目下。博主新建的路径：

/etc/nginx/ssl/jellyfishmix

执行（nginx方式）

```
acme.sh --installcert -d jellyfishmix.com \
--key-file       /etc/nginx/ssl/jellyfishmix/key.pem \
--fullchain-file /etc/nginx/ssl/jellyfishmix/cert.pem \
--reloadcmd     "service nginx force-reload"
```

--key-file 参数填写：你的自定义路径 + key.pem

--fullchain-file 参数填写：你的自定义路径 + cert.pem

这里的key.pem和cert.pem并不表示一个已经存在的文件，而是表示拷贝粘贴后的文件将被命名的名。

```
[[root@iZ2zejf0fjkrgy2ksciuofZ ssl]# cd jellyfishmix
[[root@iZ2zejf0fjkrgy2ksciuofZ jellyfishmix]# pwd
/etc/nginx/ssl/jellyfishmix
[[root@iZ2zejf0fjkrgy2ksciuofZ jellyfishmix]#
[root@iZ2zejf0fjkrgy2ksciuofZ jellyfishmix]# acme.sh --installcert -d jellyfishmix.com \
> --key-file       /etc/nginx/ssl/jellyfishmix/key.pem \
> --fullchain-file /etc/nginx/ssl/jellyfishmix/cert.pem \
[> --reloadcmd     "service nginx force-reload"
[2020年 04月 04日 星期六 02:34:57 CST] Installing key to:/etc/nginx/ssl/jellyfishmix/key.pem
[2020年 04月 04日 星期六 02:34:57 CST] Installing full chain to:/etc/nginx/ssl/jellyfishmix/cert.pem
[2020年 04月 04日 星期六 02:34:57 CST] Run reload cmd: service nginx force-reload
Redirecting to /bin/systemctl force-reload  nginx.service
[2020年 04月 04日 星期六 02:34:57 CST] Reload success
[root@iZ2zejf0fjkrgy2ksciuofZ jellyfishmix]#
```

nginx相关

nginx的安装与使用：CentOS 7 下 yum 安装和配置 Nginx

nginx的端口转发：nginx反向代理——将80端口请求转发到8080

nginx的配置文件路径查看：nginx快速查看配置文件的方法

nginx.conf配置ssl

出处：linux nginx配置https

想要https就要监听443端口，nginx.conf已经预留出了server，只要我们放开权限，修改即可。

监听443端口

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

```
        ssl_certificate /etc/nginx/ssl/jellyfishmix/cert.pem;
        ssl_certificate_key /etc/nginx/ssl/jellyfishmix/key.pem;
        ssl_session_timeout  5m;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;    #指定SSL服务器端支持的协议版本
        ssl_ciphers  HIGH:!aNULL:!MD5;
        #ssl_ciphers  ALL：!ADH：!EXPORT56：RC4+RSA：+HIGH：+MEDIUM：+LOW：+SSLv
：+EXP;   #指定加密算法
        ssl_prefer_server_ciphers   on;   #在使用SSLv3和TLS协议时指定服务器的加密算法要优先于
户端的加密算法
}
```

注：ssl_certificate 和 ssl_certificate_key 的路径就是我们ssl证书申请的路径

ssl_certificate 证书其实是个公钥，它会被发送到连接服务器的每个客户端，ssl_certificate_key私钥用来解密的，所以它的权限要得到保护但nginx的主进程能够读取。当然私钥和证书可以放在一个证文件中，这种方式也只有公钥证书才发送到client。

ssl_session_timeout 客户端可以重用会话缓存中ssl参数的过期时间，内网系统默认5分钟太短了，以设成30m即30分钟甚至4h。

ssl_protocols 指令用于启动特定的加密协议，nginx在1.1.13和1.0.12版本后默认是ssl_protocols SSv3 TLSv1 TLSv1.1 TLSv1.2，TLSv1.1与TLSv1.2要确保OpenSSL >= 1.0.1 ，SSLv3 现在还有很多地在用但有不少被攻击的漏洞。

ssl_ciphers 选择加密套件，不同的浏览器所支持的套件（和顺序）可能会不同。这里指定的是OpenSL库能够识别的写法，你可以通过 openssl -v cipher ‘RC4:HIGH:!aNULL:!MD5’ （后面是你所指的套件加密算法） 来看所支持算法。

ssl_prefer_server_ciphers on 设置协商加密算法时，优先使用我们服务端的加密套件，而不是客户浏览器的加密套件。

监听80端口

```
server {
        listen 80;
        server_name www.example.com;
        rewrite ^(.*) https://$server_name$1 permanent;
}
```

因为http是默认端口，监听80端口可以让http重定向到https端口上。

博主的nginx.conf（部分，如要复制，请把所有"jellyfishmix.com"字样替换成自己的域名，证书路替换为自己的路径）：

```
server {
        # listen       80 default_server;
        # listen       [::]:80 default_server;
        # server_name  _;

        listen       443 ssl;
        server_name     www.jellyfishmix.com;

        ssl_certificate /etc/nginx/ssl/jellyfishmix/cert.pem;
        ssl_certificate_key     /etc/nginx/ssl/jellyfishmix/key.pem;
```

```
        ssl_session_timeout     5m;
        # 指定SSL服务器端支持的协议版本
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        # ssl_ciphers  ALL：!ADH：!EXPORT56：RC4+RSA：+HIGH：+MEDIUM：+LOW：+SSLv
：+EXP;       指定加密算法
        ssl_ciphers  HIGH:!aNULL:!MD5;
        # 在使用SSLv3和TLS协议时指定服务器的加密算法要优先于客户端的加密算法
        ssl_prefer_server_ciphers   on;
        root         /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        location / {
            proxy_pass http://39.97.254.25:8080;
        }

        error_page 404 /404.html;
            location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }
    }

    server {
        listen          80;
        server_name     www.jellyfishmix.com;
        rewrite ^(.*)   https://$server_name$1  permanent;
    }
```

然后重新加载nginx：

```
systemctl reload nginx
```

此时请访问自己的域名：example.com或www.example.com 正常情况下，此时可以正常访问域名并且连接所使用的协议为https。

4. 续期证书

目前证书在 60 天以后会自动续期, 你无需任何操作。今后有可能会缩短这个时间, 不过都是自动的, 不用关心。

当然了，你可以强制手动续期：

```
acme.sh --renew -d example.com --force
```

5. 如何停止证书续期

如果要停止证书续期，你可以执行以下命令将证书从续期列表中移除：

```
acme.sh --remove -d example.com
```

cert/key 文件不会从硬盘中被移除。

你可以自行移除隐藏目录（例如：：~/.acme.sh/example.com）

6. 更新 acme.sh

目前由于 acme 协议和 letsencrypt CA 都在频繁的更新, 因此 acme.sh 也经常更新以保持同步。

升级 acme.sh 到最新版：

```
acme.sh --upgrade
```

如果你不想手动升级, 可以开启自动升级：

```
acme.sh  --upgrade  --auto-upgrade
```

之后, acme.sh 就会自动保持更新了。

你也可以随时关闭自动更新：

```
acme.sh --upgrade  --auto-upgrade  0
```

-- END