

本人简略的分享下自己对 IPV6 VPS 的使用经验（流量转发）。欢迎大佬们多多指教。

该贴旨在分享使用技巧。想购买 IPV6 机的可以到我的帖子看下：

新商家浮云：<https://hostloc.com/thread-915735-1-1.html> 买鸡后可以 PY 老板开机，目前缺货状态。

2021 年 8 月份“老”商家柴犬云：<https://hostloc.com/thread-885059-1-1.html> 最近补了点货，老板经常不在线，一般周末处理问题。

废话不多说。

首先感谢两位大佬的分享！

KANIKIG 大佬 EasyGost 一键安装脚本项目地址：<https://github.com/KANIKIG/Multi-EasyGost>

XIU2 大佬的优选 CF IP 项目地址：<https://github.com/XIU2/CloudflareSpeedTest>

首先安装 Gost 脚本。

```
Wget --no-check-certificate -O gost.sh  
https://cdn.staticaly.com/gh/KANIKIG/Multi-EasyGost/master/gost.sh && chmod +x gost.sh  
&& ./gost.sh
```

```
Wget --no-check-certificate -O gost.sh https://cdn.jsdelivr.net/gh/  
KANIKIG/Multi-EasyGost/master/gost.sh && chmod +x gost.sh && ./gost.sh
```

1、按“1”先执行安装

```
[root@ ~]# ./gost.sh
脚本最新版本获取失败，请检查与github的连接！

脚本执行命令
gost 一键安装配置脚本 [1.1.0]
----- KANIKIG -----
特性：(1)本脚本采用systemd及gost配置文件对gost进行管理
      (2)能够在不借助其他工具(如screen)的情况下实现多条转发规则同时生效
      (3)机器reboot后转发不失效
功能：(1)tcp+udp不加密转发，(2)中转机加密转发，(3)落地机解密对接转发
帮助文档：https://github.com/KANIKIG/Multi-EasyGost

1. 安装 gost
2. 更新 gost
3. 卸载 gost
-----
4. 启动 gost
5. 停止 gost
6. 重启 gost
-----
7. 新增 gost转发配置
8. 查看现有gost配置
9. 删除一则gost配置
-----
10. gost定时重启配置
11. 自定义TLS证书配置
-----
请输入数字 [1-9]:1
```

2、安装成功后再次执行脚本./gost.sh

2、按 7 增加 gost 转发配置：

以下有三种使用方法

方法 一、(gost 加密隧道流量转发)

国内中转机（莞移 IPV6）操作

(1) 执行脚本./gost.sh → 按 7 增加 gost 转发配置 → 国内机按“2”加密隧道流量转发（落地机上按 3）

(2) 传输类型（WS 或是 WSS）（这里传输类型中转机和落地机必须相同）我选 3.

(3) 落地机是否开启了自定义tls 证书？看自己情况，我直接 N

(4) 请问你要将本机哪个端口接收到的流量进行转发？ 输入端口号 10086（这个就是本地 V2 客户端端和莞移 IPV6 连接通讯的端口）

服务器	
地址(address)	莞移IPV6地址2409:8a55
端口(port)	10086
用户ID(id)	di... 生成(G)
额外ID(alterId)	0
加密方式(security)	auto *随便选, 建议(auto)
别名(remarks)	*手填, 方便识别管理

- (5) 你的落地 IP (可以是域名)
- (6) 落地机端口 10019 (这个是莞移 IPV6 机和落地机连接的端口)
- 完成国内中转机操作。

```
说明：只需在中转机设置
-----
请选择：2
请问您要设置的转发传输类型：
-----
[1] tls隧道
[2] ws隧道
[3] wss隧道
注意：同一则转发，中转与落地传输类型必须对应！本脚本默认开启tcp+udp
-----
请选择转发传输类型：3
注意：选择 是 将针对落地的自定义证书开启证书校验保证安全性，稍后落地机务必填写域名
落地机是否开启了自定义tls证书？[y/n]:n
-----
请问你要将本机哪个端口接收到的流量进行转发？
请输入：10086
-----
请问你要将本机从10086接收到的流量转发向哪个IP或域名？
注：IP既可以是[远程机器/当前机器]的公网IP，也可是以本机本地回环IP(即127.0.0.1)
具体IP地址的填写，取决于接收该流量的服务正在监听的IP(详见：https://github.com/KANIKIG/Multi-EasyGost)
请输入：域名或是落地IP
-----
请问你要将本机从10086接收到的流量转发向域名或是落地IP的哪个端口？
请输入：10019
配置已生效，当前配置如下
```

落地机操作

前面操作和中转机一样需要先安装 gost (这里省略，可回看前面的 gost 安装)

(1) 执行脚本./gost.sh → 按 7 增加 gost 转发配置 → 落地机上按“3” (解密由 gost 传输而来的流量并转发)

(2) 传输类型 (WS 或是 WSS) (这里传输类型和国内中转机必须相同) 我前面选择的是 WSS，相应的落地机这里也选择 3。

(3) 请问你要将本机哪个端口接收到的流量进行转发？ 输入端口号 10019 (这个就是莞移 IPV6 和本地连接的端口，对应的是前面中转机第 6 步操作的端口号 10019。)

特别提醒：落地机是 NAT 机器的需要在操作面板添加转发规则，如 NAT 的转发规则是随机分配的端口号那对应的落地机端口号就不一定是“10019”了，而是对应你添加规则后分配给你的公网 IP 端口，比如我下图中的，我落地转发端口实际为“20009”。

(有些 NAT 机器可以自己设置内网外网端口的，你就设置两个端口都为“10019”就好了。)

名称	转发IP: 端口	内部端口	协议	操作
gost	219.7...:20009	10019	tcp+udp	删除

- (5) 你的转发流量 IP (本机 IP 填 127.0.0.1 就对了)
- (6) 落地机上对应的“酸 S 酸 S 乳 R”端口 11119 (SSR 可以用 X-UI 创建一个酸酸乳连

接，端口号为 11119)

添加入站 ✕

备注:

启用: ☒

协议:

shadowsocks ▼

监听 IP [?]:

端口:

11119

 总流量(GB) [?]:

0

到期时间 [?]:

Select date

加密:

aes-256-gcm ▼

 密码:

eMjFONOX1P

网络:

tcp+udp ▼

传输:

tcp ▼

http 伪装: ☐

tls: ☐

sniffing [?]: ☒

```
-----
请选择: 3
请问您要设置的解密传输类型:
-----
[1] tls
[2] ws
[3] wss
注意: 同一则转发, 中转与落地传输类型必须对应! 本脚本默认开启tcp+udp
-----
请选择解密传输类型: 3
-----
请问你要将本机哪个端口接收到的流量进行转发?
请输入: 10019
-----
请问你要将本机从10019接收到的流量转发向哪个IP或域名?
注: IP既可以是[远程机器/当前机器]的公网IP, 也可以是本机本地回环IP(即127.0.0.1)
具体IP地址的填写, 取决于接收该流量的服务正在监听的IP(详见: https://github.com/KANIKIG/Multi-EasyGost)
请输入: 127.0.0.1
-----
请问你要将本机从10019接收到的流量转发向127.0.0.1的哪个端口?
请输入: 11119
配置已生效, 当前配置如下
-----
```

以下操作和用法“方法一 gost 隧道流量转发”类似，我就懒得截图发了。

方法二、(gost 无加密直接中转落地流量)

这个使用方法需要你的落地机使用加密处理后的流量（如 trojan，或是 v2+ws，亦或者 V2+WS+TLS 等等）

中转机操作

- (1) 执行脚本./gost.sh →按 7 增加 gost 转发配置 →中转机按“1”(tcp+udp 流量转发, 不加密)
- (2) 请问你要将本机哪个端口接收到的流量进行转发？ 输入端口号 2022（这个就是本地 V2 端和莞移 IPV6 连接的端口）

导入配置文件

服务器

地址(address)

莞移IPV6地址2409:8a5

端口(port)

2022

用户ID(id)

生成(g)

额外ID(alterId)

0

加密方式(security)

auto

*随便选, 建议(auto)

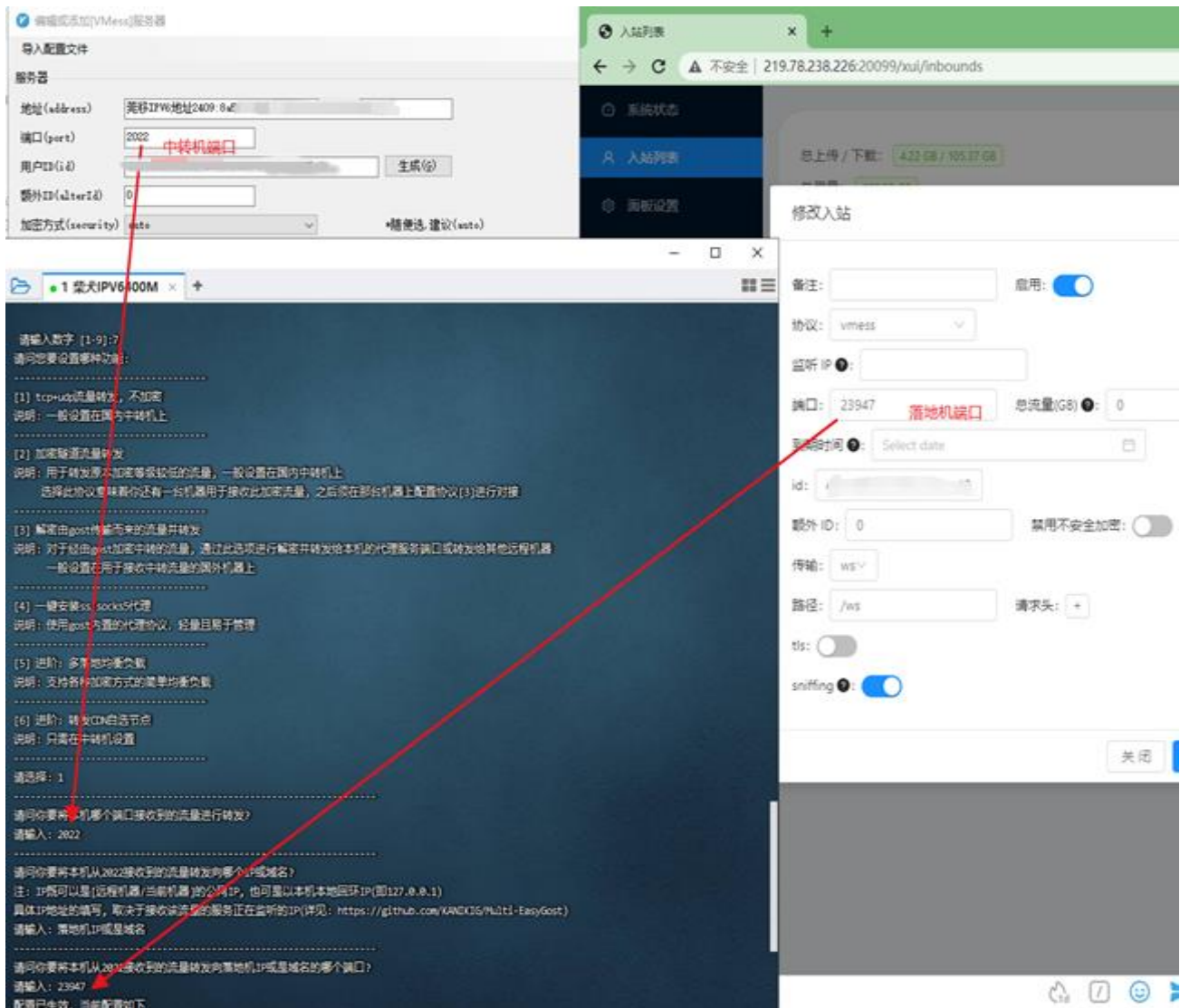
别名(remarks)

*手填, 方便识别管理

底层传输方式(transport)

- (3) 流量转发向哪个 IP 或域名： 你的落地 IP（或是域名）
- (4) 落地机端口“23947”（这个是莞移 IPV6 机和落地机连接的端口，就是你 VMESS 连接的端口号“23947”）

完成中转操作。（具体看下图）



方法 三、(gost 中转 CF+优选 IP) 直连转发优选 CF

这里操作基本和方法二的操作步骤一样，唯一不同的是需要给转发的域名修改 host 解析到自选 IP

中转机机操作

(1) 执行脚本 `./gost.sh` 按 **➡** 增加 gost 转发配置 中转机按 “1” (tcp+udp 流量转发, 不加密)

接下来的操作和方法二是一样的。

(2) 请问你要将本机哪个端口接收到的流量进行转发? 输入端口号 2022 (这个就是本地 V2 端和莞移 IPV6 连接的端口)

(3) 流量转发向哪个 IP 或域名: 你的域名 (套了 CF 的域名)

(4) 落地机端口 “80” 或是 “443” (这个端口就是你套 CF 后的连接端口, 看你落地机的配置)

```
请输入数字 [1-9]:7
请问您要设置哪种功能:
-----
[1] tcp+udp流量转发, 不加密
说明: 一般设置在国内中转机上
-----
[2] 加密隧道流量转发
说明: 用于转发原本加密等级较低的流量, 一般设置在国内中转机上
      选择此协议意味着你还有一台机器用于接收此加密流量, 之后须在那台机器上配置协议[3]进行对接
-----
[3] 解密由gost传输而来的流量并转发
说明: 对于经由gost加密中转的流量, 通过此选项进行解密并转发给本机的代理服务端口或转发给其他远程机器
      一般设置在用于接收中转流量的国外机器上
-----
[4] 一键安装ss/socks5代理
说明: 使用gost内置的代理协议, 轻量且易于管理
-----
[5] 进阶: 多落地均衡负载
说明: 支持各种加密方式的简单均衡负载
-----
[6] 进阶: 转发CDN自选节点
说明: 只需在中转机设置
-----
请选择: 1
-----
请问你要将本机哪个端口接收到的流量进行转发?
请输入: 2022
-----
请问你要将本机从2022接收到的流量转发向哪个IP或域名?
注: IP既可以是[远程机器/当前机器]的公网IP, 也可是以本机本地回环IP(即127.0.0.1)
      具体IP地址的填写, 取决于接收该流量的服务正在监听的IP(详见: https://github.com/KANIKIG/Multi-EasyGost)
请输入: cloudflare上自己的域名地址
-----
请问你要将本机从2022接收到的流量转发向cloudflare上自己的域名地址的哪个端口?
请输入: 80      80或443, 看你落地机的配置, WS就80,WS+TLS就443。
配置已生效, 当前配置如下
-----
```

(5) 修改 hosts 文件 让机器解析域名时指向优选 CF IP。

Vi /etc/hosts

添加优选 IP 地址+cloudflare 上自己的域名地址, 保存退出 “: q”。

```
2606:4700:f4:ce:fc49:b475:b44a:7fea cloudflare上自己的域名地址
      优选IP地址      IP和域名间 套CF域名
                      有个空格
```

至此完成优选 CF 中转操作。

分享些 IPV6 地址。

IP 地址	已发送	已接收	丢包率	平均延迟	下载速度 (MB/s)
2606:4700:f4:abd4:41ec:2ba4:e3ac:1cee	4	4	0.00	71.24	29.43
2606:4700:f4:abd4:41ec:2ba4:9b3b:775f	4	4	0.00	70.78	28.57
2606:4700:f4:0:dc:b7fd:4e0c:96c	4	4	0.00	70.26	28.40
2606:4700:f4:abd4:369f:ee3a:4484:94bc	4	4	0.00	70.90	28.24
2606:4700:f4:abd4:41ec:f3ef:4767:8805	4	4	0.00	71.14	28.22
2606:4700:f4:abd4:41ec:f3ef:4788:febc	4	4	0.00	70.83	28.18
2606:4700:f4:0:b10d:6c56:33ea:8e3f	4	4	0.00	69.84	27.90

2606:4700:f4:abd4:41ec:2ba4:c322:a3c7	4	4	0.00	71.19	27.78
2606:4700:f4:ab36:5d01:db2d:1880:753d	4	4	0.00	71.15	27.43
2606:4700:f4:abd4:366a:4b2:dc20:2409	4	4	0.00	71.10	27.14

具体的优选 CF 操作如下

优选 CF IP (ipv4)

<https://github.com/XIU2/CloudflareSpeedTest/issues/42>

这里就套用下大佬的教程（网上都有）：

如果是第一次使用，则建议创建新文件夹（后续更新请跳过该步骤）

```
mkdir CloudflareST
```

进入文件夹（后续更新，只需要从这里重复下面的下载、解压命令即可）

```
cd CloudflareST
```

下载 CloudflareST 压缩包（自行根据需求替换 URL 中版本号和文件名）

```
wget -N https://github.com/XIU2/CloudflareSpeedTest/releases/download/v1.5.1/CloudflareST_linux_amd64.tar.gz
```

解压（不需要删除旧文件，会直接覆盖，自行根据需求替换 文件名）

```
tar -zxvf CloudflareST_linux_amd64.tar.gz
```

赋予执行权限

```
chmod +x CloudflareST
```

运行

```
./CloudflareST
```

执行后会出现测试结果。如果线路很理想的话就往下看


```
[root@ test]# ./CloudflareST
# XIU2/CloudflareSpeedTest v1.5.1

开始延迟测速 (模式: TCP IPv4, 端口: 443, 平均延迟上限: 9999.00 ms, 平均延迟下限: 0.00 ms) :
55200 / 55200 [-----] 100.00%
开始下载测速 (下载速度下限: 0.00 MB/s, 下载测速数量: 20, 下载测速队列: 20) :
20 / 20 [-----] 100.00%
IP 地址      已发送  已接收  丢包率  平均延迟  下载速度 (MB/s)
104.19.231.115 4       4       0.00    31.19     209.78
104.17.108.97  4       4       0.00    57.73     89.52
141.101.115.62 4       4       0.00    59.66     87.66
198.41.192.201 4       4       0.00    59.74     74.77
104.17.111.127 4       4       0.00    57.31     66.49
104.19.40.200  4       4       0.00    60.10     63.96
104.19.160.217 4       4       0.00    56.48     63.31
198.41.193.37  4       4       0.00    59.75     60.41
198.41.197.91  4       4       0.00    59.24     57.36
104.19.157.182 4       4       0.00    62.84     56.52
104.16.251.65  4       4       0.00    46.68     56.02
104.17.18.108  4       4       0.00    51.71     55.90
190.93.247.219 4       4       0.00    58.86     55.67
198.41.209.130 4       4       0.00    58.79     54.50
104.19.162.46  4       4       0.00    56.21     54.11
104.17.144.226 4       4       0.00    44.54     53.94
198.41.199.41  4       4       0.00    59.60     53.75
104.19.173.136 4       4       0.00    46.13     53.08
104.19.164.74  4       4       0.00    49.90     52.90
104.19.27.148  4       4       0.00    51.92     52.55

↑ 2.6 KB/s
↓ 16.3 KB/s
```

注：这里提醒下作者的 IP 库不全，需要自己加 IP 库。（自行网上找吧，这里不作分享。怕大佬 da 我。）

接下来是替换 hosts

执行命令 `cd CloudflareST && bash cfst_hosts.sh`

```
[root@ test]# bash cfst_hosts.sh
开始测速...
# XIU2/CloudflareSpeedTest v1.5.1

开始延迟测速 (模式: TCP IPv4, 端口: 80, 平均延迟上限: 100.00 ms, 平均延迟下限: 0.00 ms) :
55200 / 55200 [-----] 100.00%
开始下载测速 (下载速度下限: 0.00 MB/s, 下载测速数量: 5, 下载测速队列: 5) :
5 / 5 [-----] 100.00%
IP 地址      已发送  已接收  丢包率  平均延迟  下载速度 (MB/s)
104.16.102.3  4       4       0.00    34.72     57.94
104.17.254.6  4       4       0.00    68.90     57.65
104.17.211.109 4       4       0.00    74.71     57.53
104.17.58.53  4       4       0.00    72.13     57.51
162.159.201.25 4       4       0.00    29.24     0.00

↑ 2.9 KB/s
↓ 43.1 KB/s

完整测速结果已写入 result.csv 文件, 请使用记事本/表格软件查看。

旧 IP 为 104.16.114.100
新 IP 为 104.16.102.3 记录这个IP

开始备份 Hosts 文件 (hosts_backup) ...
开始替换...
完成...
```

这里记录得出的最优 IP。（首次运行是没有旧 IP 的，记录新 IP。）

1、添加 hosts 解析 `vi /etc/hosts` 加入自己域名和最优 IP。举个栗子：104.19.19.19 bilibili.com（这里的 IP 地址就是刚刚记录的 CF 优选 IP 地址，哔哩哔哩换成自己的 CF 域名）

接下来的操作是保存退出 VI

按 Esc（键盘）：wq

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
104.16.102.3 .workers.dev
104.16.102.3
```

还是不会，如何用 VI 工具自己百度（g u g e）度吧。都这么过来。
这个给的小提示：可以添加多个 CF 域名。可以实现本地解析 CF 域名全部解析到优选 IP。

2、定时任务（借用大佬的教程）

安装 Cron

一般各 Linux 系统都自带了 Cron，执行 `crontab -l` 命令，如果提示命令不存在，说明没有安装，反之则跳过该步骤。

确认你是哪个系统，然后选择对应的命令安装 Cron 程序。

CentOS 系统：

`yum install vixie-cron`

`yum install crontabs`

Debian/Ubuntu 系统：

`apt-get install cron`

配置 Cron

开始编辑定时任务，

`crontab -e`

首次使用可能会提示你如下内容：

这就是让你选一个文本编辑器，新手建议用默认的 nano，直接回车即可。

Select an editor. To change later, run 'select-editor'.

1. `/bin/nano` <---- easiest

2. `/usr/bin/vim.basic`

3. `/usr/bin/vim.tiny`

4. `/bin/ed`

然后就会看到一大片的文本，不用管，那些只是注释（井号 # 开头的都是注释），写的是使用方法。

如果你选的是 nano 编辑器，那么可以直接开始编辑了。

如果你选的是 vim 编辑器，则需要按下 I 键 进入编辑模式。

定时任务格式其实很简单：

```
* * * * * cd /xxx && ./cfst_hosts.sh
- - - - -
| | | | |
| | | | +---- 星期中星期几 (0 - 7) (星期天 为 0)
| | | +----- 月份 (1 - 12)
| | +----- 一个月中的第几天 (1 - 31)
| +----- 小时 (0 - 23)
```

+----- 分钟 (0 - 59)

还是看不懂？没关系，我直接给你几个示例（脚本下载）：

假设脚本位于 `/root/CloudflareST` 目录下（其他位置自己改下面示例），那么：

每天凌晨 5 点 0 分，执行一次脚本

```
0 5 * * * cd /root/CloudflareST && ./cfst_hosts.sh
```

每天凌晨 5 点 30 分，执行一次脚本

```
30 5 * * * cd /root/CloudflareST && ./cfst_hosts.sh
```

每 6 个小时（0 分时），执行一次脚本

```
0 */6 * * * cd /root/CloudflareST && ./cfst_hosts.sh
```

每小时 0 分，执行一次脚本

```
0 * * * * cd /root/CloudflareST && ./cfst_hosts.sh
```

写入后，保存定时任务：

nano：按下 `Ctrl+X` 键、按下 `Y` 键、按下回车键，即可保存。

vim：按下 `Esc` 键退出编辑模式，直接输入 `:wq` 并回车（英文模式下），即可保存。

这时候再去查看定时任务，看看是否保存成功：

`crontab -l`

备注：在 `cfst_hosts.sh` 脚本内可以根据自己的需要修改脚本执行命令

```
./CloudflareST -dn 5 -dt 2 -tl 100 -tp 80 -n 1000 -o result.csv
```

参数：

`-n 200`

测速线程数量；越多测速越快，性能弱的设备（如路由器）请适当调低；（默认 200 最多 1000）

`-t 4`

延迟测速次数；单个 IP 延迟测速次数，为 1 时将过滤丢包的 IP，TCP 协议；（默认 4）

`-tp 443`

延迟测速端口；延迟测速 TCP 协议的端口；（默认 443）

`-dn 20`

下载测速数量；延迟测速并排序后，从最低延迟起下载测速的数量；（默认 20）

`-dt 10`

下载测速时间；单个 IP 下载测速最长时间，单位：秒；（默认 10）

`-url https://cf.xiu2.xyz/Github/CloudflareSpeedTest.png`

下载测速地址；用来下载测速的 Cloudflare CDN 文件地址，如地址含有空格请加上引号；

-tl 200
平均延迟上限；只输出低于指定平均延迟的 IP，可与其他上限/下限搭配；(默认 9999 ms)

-tll 40
平均延迟下限；只输出高于指定平均延迟的 IP，可与其他上限/下限搭配，过滤被假蓄的 IP；(默认 0 ms)

-sl 5
下载速度下限；只输出高于指定下载速度的 IP，凑够指定数量 [-dn] 才会停止测速；(默认 0.00 MB/s)

-p 20
显示结果数量；测速后直接显示指定数量的结果，为 0 时不显示结果直接退出；(默认 20)

-f ip.txt
IP 段数据文件；如路径含有空格请加上引号；支持其他 CDN IP 段；(默认 ip.txt)

-o result.csv
写入结果文件；如路径含有空格请加上引号；值为空时不写入文件 [-o ""]；(默认 result.csv)

-dd
禁用下载测速；禁用后测速结果会按延迟排序 (默认按下载速度排序)；(默认 启用)

-ipv6
IPv6 测速模式；确保 IP 段数据文件内只包含 IPv6 IP 段，软件不支持同时测速 IPv4+IPv6；(默认 IPv4)

-allip
测速全部的 IP；对 IP 段中的每个 IP (仅支持 IPv4) 进行测速；(默认 每个 IP 段随机测速一个 IP)

-v
打印程序版本+检查版本更新

-h
打印帮助说明

补充：

1、 关于下载测速不可用 0.00 MB/s 的情况说明 及 解决方法

<https://github.com/XIU2/CloudflareSpeedTest/issues/168>

大佬都写出来教程了，我就不多此一举。

2、 HK CF IP4 被拔线后无法直连 HK IPV4，现在还能用 IPV6 救下。

IPV6 基本很稳不用变，随便一个 IPV6 地址都可以跑。

<https://hostloc.com/thread-968662-1-1.html>