

---

**wall-guide**

**发布 2.2.0**

**2019 年 12 月 19 日**



---

## 目录:

---

1 引言	1
2 什么是 VPS	3
3 如何测速	5
4 如何加速	9
5 如何判断 ip 或端口被封	13
6 手动翻墙	19
7 Ubuntu 代理配置	25
8 SwitchyOmega 代理设置	37
9 Ubuntu 全局代理设置	43
10 ssr 配置	45
11 [brook] 服务端 + 客户端设置	49
12 v2ray 使用	53
13 [Google Helper] 辅助搜索工具	63
14 [Android]V2ray 使用	67
15 [todo]google cloud	69



为了更好的进行程序开发，常常要去 [github/google](#) 上查找资料，所以开始寻找如何能够科学上网的方法  
目前大致有 2 种模式

1. 购买网站套餐
2. 购买海外服务器自己搭建

### 1.1 购买网站套餐

最开始就是这样实现的，尝试过免费软件（比如[蓝灯](#)），后来就使用付费套餐（[MonoCloud](#)、[borderSS](#)、[影梭](#)）  
免费软件不稳定，付费套餐同样不稳定，刚开始尝试一两个月没有问题，等到付了 1 年的费用后，没几个月连网站都登不上了

尝试了几次后总觉得不靠谱，所以选择第二种方法

### 1.2 购买海外服务器自己搭建

购买海外 VPS 服务器

我使用的是[vultr](#)，类似的还有[linode](#)、[google cloud](#)、[BandwagonHOST](#)

无意间发现[比特熊](#)，没有使用过，不过价格很便宜

参考：

[Vultr 中文网](#)

[搬瓦工 VPS 教程](#)

## CHAPTER 2

---

### 什么是 VPS

---

参考: [VPS](#)

VPS, Virtual Private Server, 虚拟专用服务器, 其是将物理服务器通过虚拟化技术虚拟成为多个虚拟专用服务器

### 2.1 VPS 的优点

同一服务器上的多个 VPS 共享硬件、软件许可证以及管理资源, 以最大化效率运行

对于用户而言, 每个 VPS 平台的管理和运行与独立主机完全相同, 有独立的公网 IP 地址、操作系统、磁盘空间、内存、CPU 资源、进程和系统配置, 可以单独安装程序并且单独重启主机

### 2.2 VPS 的缺陷

同一台服务器上的某个 VPS 受到攻击或占用过多宽带资源时, 其余的 VPS 也会受到影响, 所以一台 VPS 被黑客入侵可能造成服务器瘫痪

同时硬件服务器的损坏也会影响到所有 VPS 主机

### 2.3 VPS 的实现

实现 VPS 的方式有两种, 一种是虚拟化技术, 比如 XEN, KVM, OpenVZ, 另一种是容器技术, 比如 Docker

## 2.4 VPS vs ECS

参考:

云服务器

云服务器 (ECS) 云主机云虚拟主机 VPS 等产品的区别?

阿里云服务器与 VPS 和虚拟主机有什么区别?

ECS(Elastic Computer Service) 指的是云服务器, 其实现是通过服务器集群来虚拟出独立服务器, 对服务器的资源扩展有更高的可操作性; 同时通过冗余的共享存储和智能备份, 单个服务器的故障不影响云服务器的使用, 大大提高了安全性和稳定性

## 2.5 什么是使用 VPS 以及 ECS

VPS 的价格一般比 ECS 低, 所以针对小规模并发访问, 资源扩展性不大的操作可以使用 VPS

VPS 厂商包括: vultr、linode、BandwagonHOST和google cloud

如果后期可能需要弹性增加资源, 同时对安全性和稳定性有更高要求的可以使用 ECS

ECS 厂商包括: 阿里云、腾讯云、亚马逊云



不同的线路对于网速差别极大，一般来说，距离越远，延时越大

### 3.1 在线测速

参考: [Vultr 测速方法](#)

[speedtest](#)

[Ping 检测](#)

[vultr 测速](#)

### 3.2 命令行测速

#### 3.2.1 ping

参考: [Calculate upload/download speed by ping](#)

```
$ ping 149.28.83.99
PING 149.28.83.99 (149.28.83.99) 56(84) bytes of data.
64 bytes from 149.28.83.99: icmp_seq=1 ttl=45 time=187 ms
64 bytes from 149.28.83.99: icmp_seq=2 ttl=45 time=187 ms
64 bytes from 149.28.83.99: icmp_seq=3 ttl=45 time=187 ms
```

(下页继续)

(续上页)

```
64 bytes from 149.28.83.99: icmp_seq=4 ttl=45 time=190 ms
64 bytes from 149.28.83.99: icmp_seq=5 ttl=45 time=186 ms
64 bytes from 149.28.83.99: icmp_seq=6 ttl=45 time=186 ms
64 bytes from 149.28.83.99: icmp_seq=7 ttl=45 time=187 ms
64 bytes from 149.28.83.99: icmp_seq=8 ttl=45 time=186 ms
64 bytes from 149.28.83.99: icmp_seq=9 ttl=45 time=188 ms
64 bytes from 149.28.83.99: icmp_seq=10 ttl=45 time=188 ms
64 bytes from 149.28.83.99: icmp_seq=11 ttl=45 time=187 ms
^C
--- 149.28.83.99 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10011ms
rtt min/avg/max/mdev = 186.880/187.645/190.295/1.125 ms
```

发送 64 字节平均需要 187.645 毫秒，延时非常严重

### 3.2.2 speedtest-cli

参考：

[speedtest-cli 命令行下测试服务器外网速度](#)

[如何在 Linux 的终端测试网速](#)

[github 有speedtest的命令行版本](#)

```
$ pip install speedtest-cli
```

测试网速

```
$ speedtest
Retrieving speedtest.net configuration...
Testing from UNICOM ZheJiang (101.68.71.72)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by Hangzhou , Zhejiang Unicom (Hangzhou) [4.91 km]: 2.893 ms
Testing download speed.....
↪.....
Download: 51.89 Mbit/s
Testing upload speed.....
↪.....
Upload: 3.94 Mbit/s
```

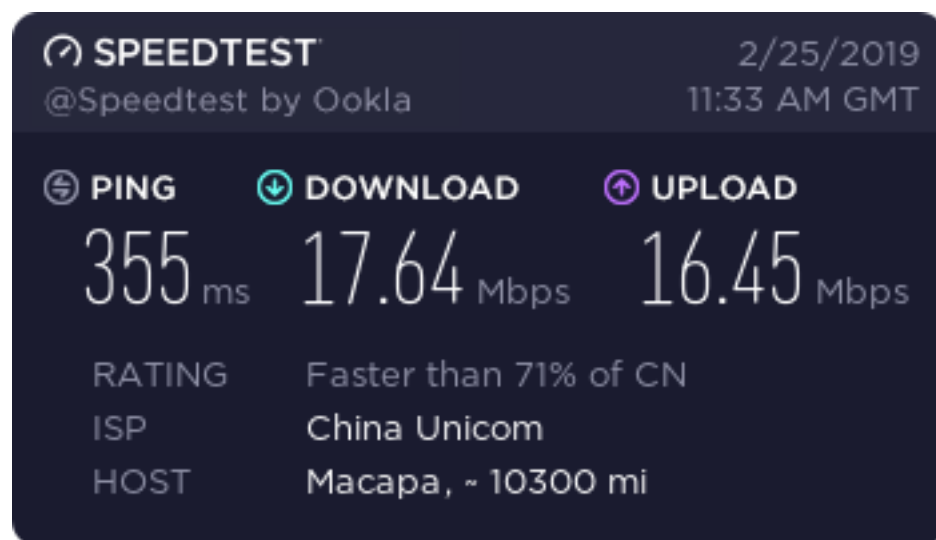
测试服务器 Zhejiang Unicom (Hangzhou), 距离是 4.91km, 下载速度为 51.89Mbit/s, 上传速度为 3.94Mbit/s

可以查找所有在美国的服务器

```
$ speedtest --list | grep -i america
8464) Bluesky Communications (Pago Pago, American Samoa) [8895.69 km]
20275) Blue Nap Americas (Willemstad, Curaçao) [15200.29 km]
16317) America Telecom (Macapa, Brazil) [16525.29 km]
3455) America Movil Peru SAC (Lima, Peru) [17321.89 km]
10275) Byteweb Telecom (Americana, Brazil) [18515.34 km]
18357) SIMNET (Americana, Brazil) [18515.34 km]
22460) NET Virtua (Americana, Brazil) [18515.34 km]
15091) SWISS-NET (Jardin America, Argentina) [19438.10 km]
...
...
```

指定服务器 id 进行测速, 参数--share 能够上传数据, 生成一个数据图片

```
$ speedtest --server 16317 --share
Retrieving speedtest.net configuration...
Testing from UNICOM ZheJiang (101.68.71.72)...
Retrieving speedtest.net server list...
Retrieving information for the selected server...
Hosted by America Telecom (Macapa) [16525.29 km]: 354.928 ms
Testing download speed.....
↔.....
Download: 17.64 Mbit/s
Testing upload speed.....81.8M.....
↔.....
Upload: 16.45 Mbit/s81.8M
Share results: http://www.speedtest.net/result/8068095904.png
```



常见的加速方法包括 BBR、锐速

#### 4.1 BBR 加速

参考：

TCP\_BBR

安装 Google BBR 加速 VPS 网络

BBR是 google 实现的一种 TCP 拥塞控制算法。BBR 已在新版本的 Linux 内核中集成，需要高于等于 4.9 的 Linux 内核才能开启

在 root 权限下执行

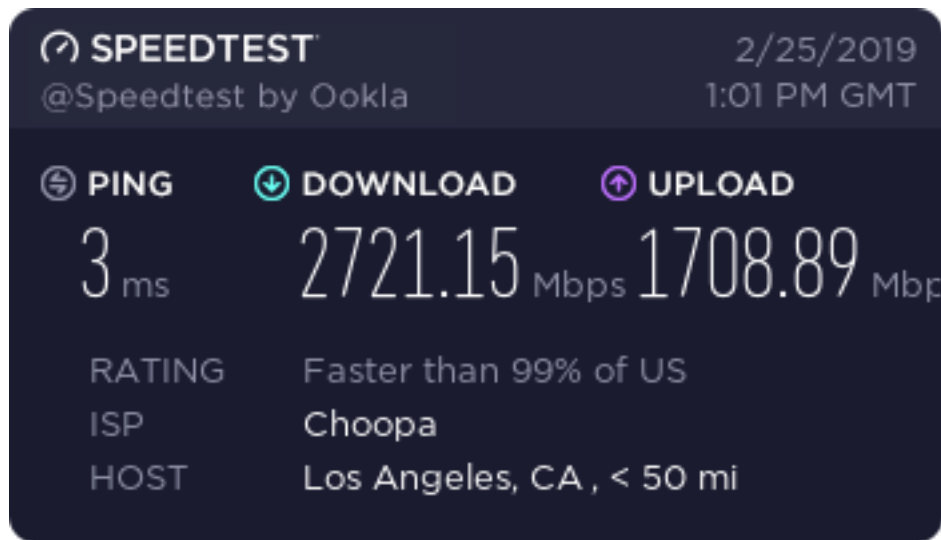
```
# 下载一键安装包
$ wget --no-check-certificate https://github.com/teddysun/across/raw/master/bbr.sh
# 赋予执行权限
$ chmod +x bbr.sh
# 执行脚本
$ ./bbr.sh
```

安装完成后重启系统

```
# 检查 linux 内核
$ uname -r
4.20.12-042012-generic
# 开启 TCP BBR
$ sysctl net.ipv4.tcp_available_congestion_control
net.ipv4.tcp_available_congestion_control = reno cubic bbr
$ sysctl net.ipv4.tcp_congestion_control
net.ipv4.tcp_congestion_control = bbr
$ sysctl net.core.default_qdisc
net.core.default_qdisc = fq
# 查看 bbr 是否已启动
$ lsmod | grep bbr
tcp_bbr                20480  20
```

#### 测试网速

```
$ speedtest --share
Retrieving speedtest.net configuration...
Testing from Choopa, LLC (149.28.83.99)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by WebNX (Los Angeles, CA) [2.73 km]: 3.155 ms
Testing download speed.....
↔.....
Download: 2721.15 Mbit/s
Testing upload speed.....
↔.....
Upload: 1708.89 Mbit/s
Share results: http://www.speedtest.net/result/8068280714.png
```



#### 4.1.1 魔改 BBR

参考: 魔改 BBR 一键脚本 for Debian & CentOS

## 4.2 锐速

参考: ubuntu 一键更换内核安装锐速 - 16.04 14.04 可用





---

### 如何判断 ip 或端口被封

---

参考：

如何检查搬瓦工的 IP 是否被封

搬瓦工能 ping 通,ssh 不能连接,ss 打不开网页

解决 ss 突然无法联网

这一段时间经常无法连接外网，最常遇到的两个问题是

1. ping ip 不成功
2. ping ip 成功但是浏览器无法连接外网

在网上找了很久资料，终于对 IP 或端口被封有了一些理解

## 5.1 IP 检测

### 5.1.1 命令行方式

最简单的是在命令行窗口上测试

```
$ ping 136.244.112.3
PING 136.244.112.3 (136.244.112.3) 56(84) bytes of data.
64 bytes from 136.244.112.3: icmp_seq=1 ttl=46 time=240 ms
64 bytes from 136.244.112.3: icmp_seq=2 ttl=46 time=240 ms
```

(下页继续)

(续上页)

```
64 bytes from 136.244.112.3: icmp_seq=3 ttl=46 time=242 ms
64 bytes from 136.244.112.3: icmp_seq=4 ttl=46 time=239 ms
^C
--- 136.244.112.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 239.683/240.601/242.435/1.078 ms
```

### 5.1.2 在线方式

也可使用网站Ping 检测进行测试

**注意：**可单独测试海外连通性

除了给出响应数据之外，还会以图的形式显示

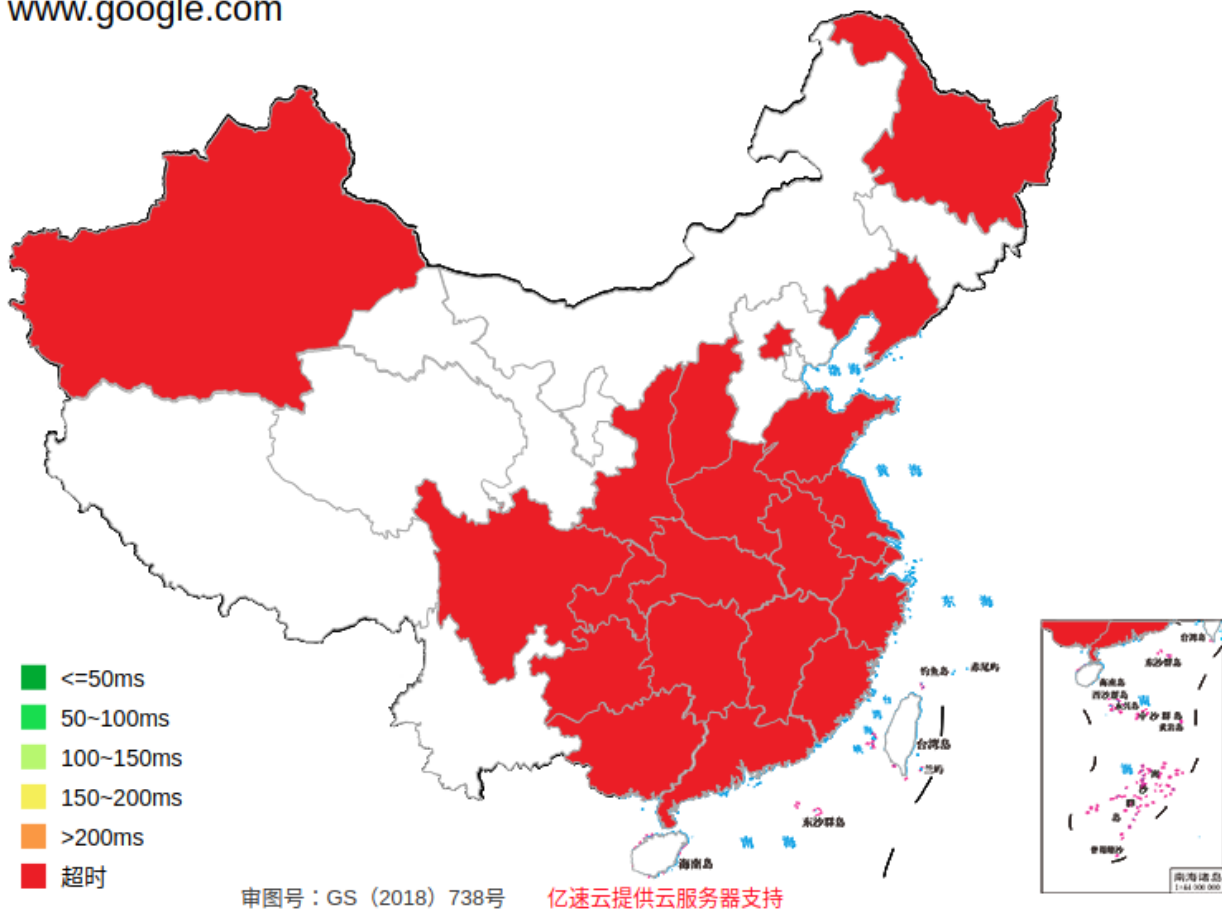
对于 `www.google.com`

141个探测点 (完成141个)

最快 山西太原[电信] &lt;1ms

最慢 韩国 213ms

www.google.com



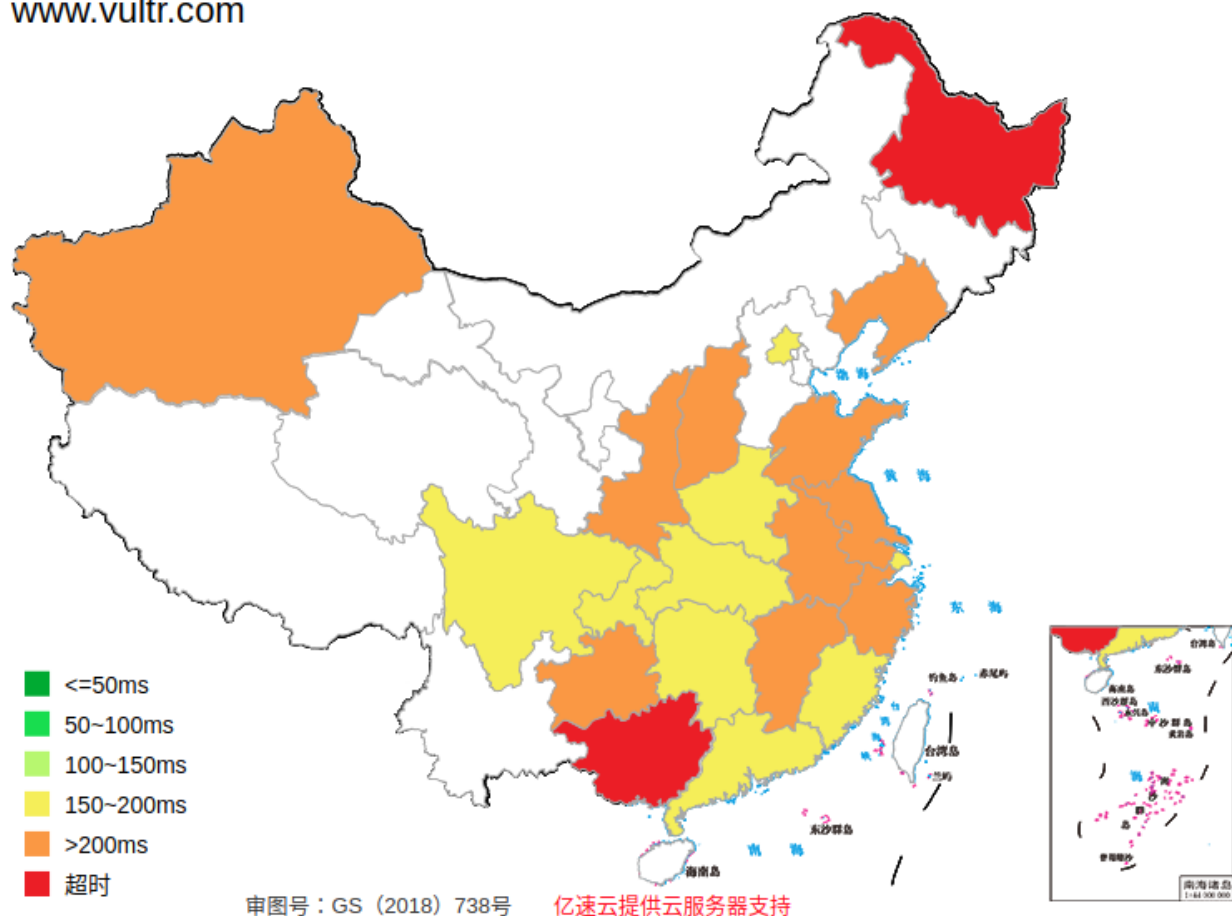
对于 www.vultr.com

141个探测点 (完成141个)

最快 美国 <1ms

最慢 江西新余[电信] 34

www.vultr.com



### 5.1.3 判断

- 如果国内和国外都 ping 不通，则是海外服务器的问题
- 如果国内 ping 不通，国外能 ping 通，那么 IP 被封了

## 5.2 端口检测

### 5.2.1 命令行方式

参考：如何 ping 指定 IP 的端口号

使用 telnet 命令测试端口：

```
$ telnet ip port
```

如果成功会建立一条 TCP 连接；如果失败（可能是 IP，也可能使端口）立即退出

```
(base) zj@zj-ThinkPad-T470p:~$ telnet 136.244.112.3 34521
Trying 136.244.112.3...
Connected to 136.244.112.3.
Escape character is '^['.
```

### 5.2.2 在线方式

使用国内网站测试 `ssr` 服务器端口是否连通：在线检测域名或者 ip 的端口是否开放

#### 在线检测域名或者ip的端口是否开放

多个以","隔开，支持范围书写，例如:1,2,3,10-20,6,100-105,33 一次最多不能超过50个

端口扫描

域名	端口	端口状态	ip
136.244.112.3	34521	开放	

使用国外网站测试 `ssr` 服务器端口是否连通：Port Forwarding Tester

#### open port finder

Remote Address  Port Number

☒ Use Current IP

🚩 Port 34521 is open on 136.244.112.3.

### 5.2.3 判断

- 如果国内和国外都不能连通，是服务器问题
- 如果国内不能连通，国外能连通，那么端口被封了

## 5.3 解决方式

- 对于 IP 被封，需要重新更换 IP
- 对于端口被封，需要重新设置 `ssr` 服务器端口号

重新设置 `ssr` 服务端端口号时，先删除已有进程，再重新设置

```
# 查询进程
$ ps aux | grep shadowsocks
# 删除
$ sudo kill xxx
# 重新启动 ssr 服务端
$ ss-fly/ss-fly.sh -ssr
```

重新更新客户端配置时，会出现 **1080 端口被占用**的提示，最好先杀死原先的客户端进程，再重新启动

```
# 查询占用 1080 端口的进程
$ sudo netstat -ltn | grep 1080
# 杀死进程
$ sudo kill xxx
# 重启 ssr 客户端
$ electron-ssr
```

---

### 手动翻墙

---

之前使用过很多个付费翻墙软件，最终都因为线路不稳定而放弃。最近一个是影梭，用了 4 个月还不错，今天让我重新续费，没想到再付了 3 个月的费用后无法使用新的线路

在网上找到手动翻墙的文章，尝试一下：

手把手教你搭建 shadowsocks 科学上网搭建 ss 翻墙

整个过程就是购买国外服务器，在上面搭建 Shadowsocks 服务器，然后在本地搭建 Shadowsocks 客户端以及浏览器配置

翻墙网站实现了前两步，所以在上面注册购买服务后只需在本地进行操作即可。

---

#### 目录

1. VPS 服务器部署
  2. Shadowsocks 服务器搭建
  3. Shadowsocks 客户端 & 浏览器配置
- 

### 6.1 VPS 服务器部署

国外的 VPS（虚拟专用服务器，Virtual Private Server）服务商有 linode, VULTR, DigitalOcean 等，我参考文章选择 VULTR

<https://www.vultr.com/>

VULTR 支持支付宝和微信支付，按小时计费

How am I billed for my instances?

All servers on your account are billed hourly up to the monthly rate cap. The hourly rate is determined by dividing the monthly rate by 672 hours (28 days). If your server is online for more than 672 hours in a calendar month, you will only be billed the monthly rate. Accumulated charges are invoiced to your account on the 1st of every month.

你账户上的所有服务器都按小时费率计费。每小时费率由每月 672 小时（28 天）除以。如果您的服务器在日历月上线超过 672 小时，您将只计入月度费率。累积费用在每月第一天的账单上开具。

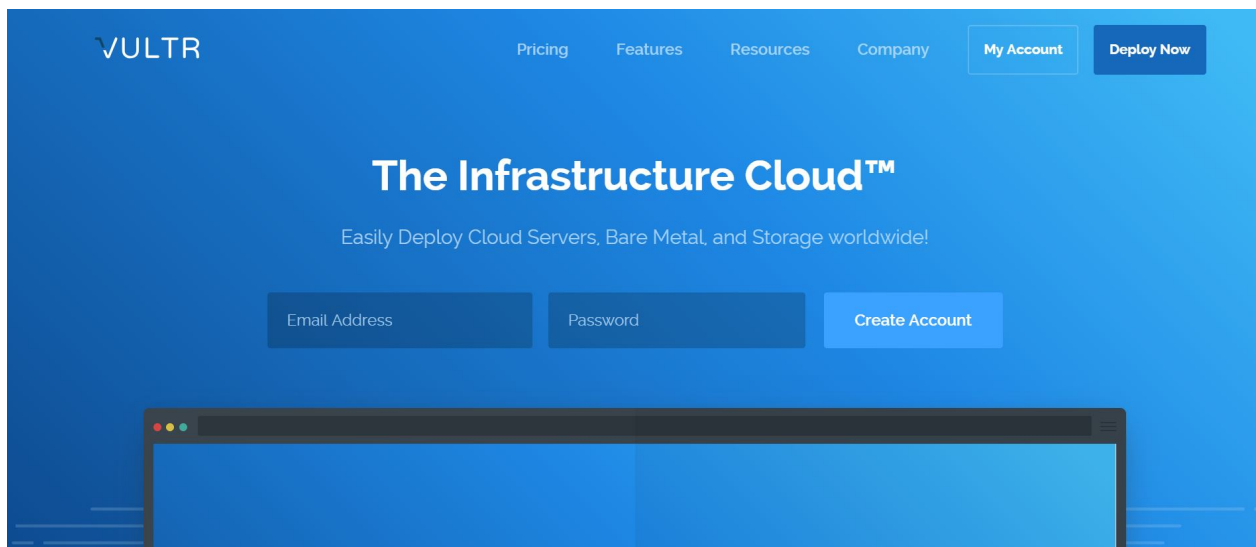
What payment methods do you accept?

We accept Visa, Mastercard, American Express, Discover, JCB, Bitcoin (BTC and BCH), Alipay, UnionPay, WeChat Pay, and PayPal. Depending on your geographic region, PayPal supports additional payment networks, including Giropay and bank transfers.

有一个中文相关网站：<https://www.vultryhw.com/>

### 6.1.1 注册账户

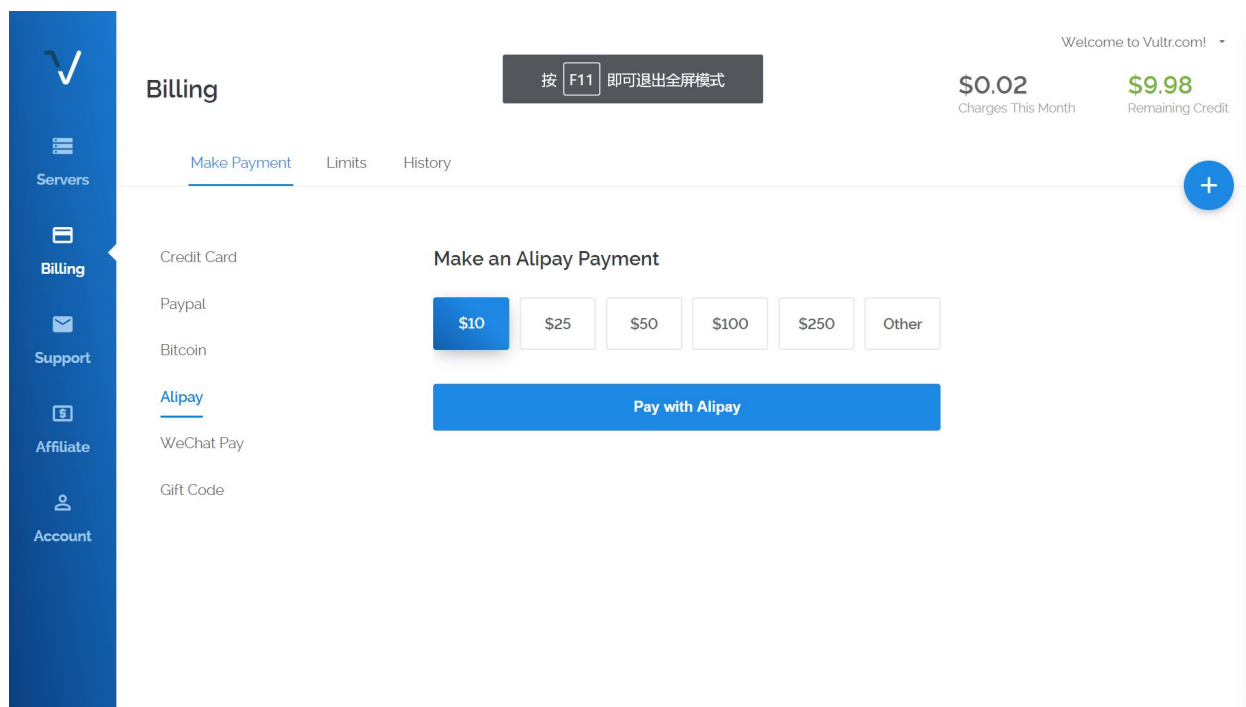
打开网站，输入邮箱和密码，点击 Create Account，然后在邮箱中确认即可



### 6.1.2 充值

注册完成进入个人页面后，点击左侧 Billing 选项，选择 Alipay，先充值最少的 \$10





### 6.1.3 部署服务器

点击右上角的加号图标



选择服务器位置 (Server Location), 服务器类型 (Server Type) 以及服务器配置 (Server Size)  
个人配置:

```
Server Location: Los Angeles(United States)
Server Type: Ubuntu 16.04 x64
Serve Size: 20GB SSD $3.50/mo $0.005/h
```

当时冲了 \$10 花了 70.7 元人民币,  $\$3.50 = \$24.745$

额外配置的话选择 Enable IPv6

4

## Additional Features

- ☒ Enable IPv6
- ☐ Enable Auto Backups \$0.70/mo
- ☐ Enable DDOS Protection ? \$10/mo
- ☐ Enable Private Networking ?

点击 Deploy Now, 一个服务器即配置完成并开始部署

点击该服务器, 即可查看相应的 ip, 用户名和密码

## 6.2 Shadowsocks 服务器搭建

上面完成了远程服务器的部署，本节实现远程服务器的连接以及在上面搭建 Shadowsocks 服务器

### 6.2.1 远程服务器连接

在本地打开命令行窗口进行 ssh 连接，Windows 系统可以使用 Cmder，XShell，SecureCRT 等

参考: Windows cmd

## 连接远程服务器

```
ssh root@ 服务器 ip
```

### 6.2.2 Shadowsocks 服务器搭建

## 使用一个脚本进行一键安装

```
# 原来的这个库被作者清空了
git clone https://github.com/Flyzy2005/ss-fly
# 在 github 上找了一个，自己也备份了一下
git clone https://github.com/zjZSTU/ss-fly.git
ss-fly/ss-fly.sh -i password 1024
```

将 password 替换成自定义的密码，1024 表示端口号（默认为 1024，可以更改）

### 搭建成功后会提示

```
INFO: loading config from /etc/shadowsocks.json
2018-11-01 07:06:58 INFO      loading libcrypto from libcrypto.so.1.0.0
started
[成功] 安装成功尽情冲浪!
你的服务器地址 (IP) : 149.28.83.99
你的密码             : 
你的端口             : 1024
你的加密方式         : aes-256-cfb
欢迎访问flyzy小站   : https://www.flyzy2005.com
你的ss链接: ss://YWVzLTl1NiIjZmI6emh1cm1ibG9yeS8wLWt1bGVyYXNpdDQ=
root@vultr:~# ss://YWVzLTl1NiIjZmI6emh1cm1ibG9yeS8wLWt1bGVyYXNpdDQ=
root@vultr:~#
```

将该命令加入到启动文件/etc/profile 实现开机自启动

## 6.3 Shadowsocks 客户端 & 浏览器配置

接下来的操作就和之前一样了，参考

[Shadowsocks - Clients](#)

[Proxy SwitchyOmega](#)

## CHAPTER 7

---

### Ubuntu 代理配置

---

目前用的 VPN 是影梭，发现它没有对应的 Linux 使用教程  
在网上找了一些资料，有多种方式可以实现翻墙，记录一下

---

主要内容：

1. Shadowsocks 安装和配置
  2. Chrome 代理配置
  3. 全局代理配置
  4. 相关
- 

### 7.1 Shadowsocks 安装和配置

Shadowsocks 可以通过命令行设置，也可以通过图形界面设置

仅仅通过设置 Shadowsocks 还不能连接外网，还需要在浏览器（以 Chrome 为例）上设置代理或者设置全局代理

官方 wiki: [shadowsocks/shadowsocks-qt5](https://shadowsocks.org/en/index.html)

### 7.1.1 命令行安装和配置

参考: Linux 下使用 shadowsocks (以 ubuntu16.04 为例, 非服务器)

github 上面还有好多不同版本的 shadowsocks, 比如 shadowsocks-go

安装:

```
sudo apt-get install shadowsocks
```

安装完成后可以使用命令 sslocal

```
zhujiang@zhujiang-virtual-machine:~$ sslocal -h
usage: sslocal [-h] -s SERVER_ADDR [-p SERVER_PORT]
              [-b LOCAL_ADDR] [-l LOCAL_PORT] -k PASSWORD [-m METHOD]
              [-t TIMEOUT] [-c CONFIG] [--fast-open] [-v] [-q]

optional arguments:
  -h, --help            show this help message and exit
  -s SERVER_ADDR        server address
  -p SERVER_PORT        server port, default: 8388
  -b LOCAL_ADDR         local binding address, default: 127.0.0.1
  -l LOCAL_PORT         local port, default: 1080
  -k PASSWORD           password
  -m METHOD              encryption method, default: aes-256-cfb
  -t TIMEOUT            timeout in seconds, default: 300
  -c CONFIG             path to config file
  --fast-open           use TCP_FASTOPEN, requires Linux 3.7+
  -v, -vv              verbose mode
  -q, -qq              quiet mode, only show warnings/errors

Online help: <https://github.com/clowwindy/shadowsocks>
```

可以通过命令行的方式配置, 也可以通过配置文件的方式进行

主要修改 server, server\_port 和 password

配置代理文件 shadowsocks.json

```
{
  "server": "服务器的 ip",
  "server_port": 服务器的端口,
  "local_address": "127.0.0.1",
  "local_port": 1080,
```

(下页继续)

(续上页)

```

    "password": "密码",
    "timeout": 300,
    "method": "aes-256-cfb",
    "fast_open": true
}

```

配置文件方式启动代理:

```
sslocal -c shadowsocks.json
```

```

zhujian@zhujian-virtual-machine:~/software/shadowsocks$ sslocal -c shadowsocks.j
son
INFO: loading config from shadowsocks.json
shadowsocks 2.1.0
2018-08-16 19:09:20 INFO      starting local at 127.0.0.1:1080
2018-08-16 19:09:20 INFO      starting remote at 127.0.0.1:1111

```

这

里写图片描述

## 7.1.2 客户端安装和配置

参考: [Ubuntu 下实现全局翻墙的配置过程](#)

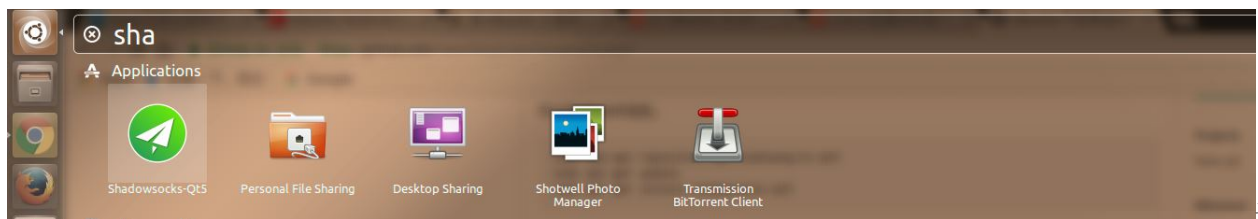
安装:

```

sudo add-apt-repository ppa:hzwhuang/ss-qt5
sudo apt-get update
sudo apt-get install shadowsocks-qt5

```

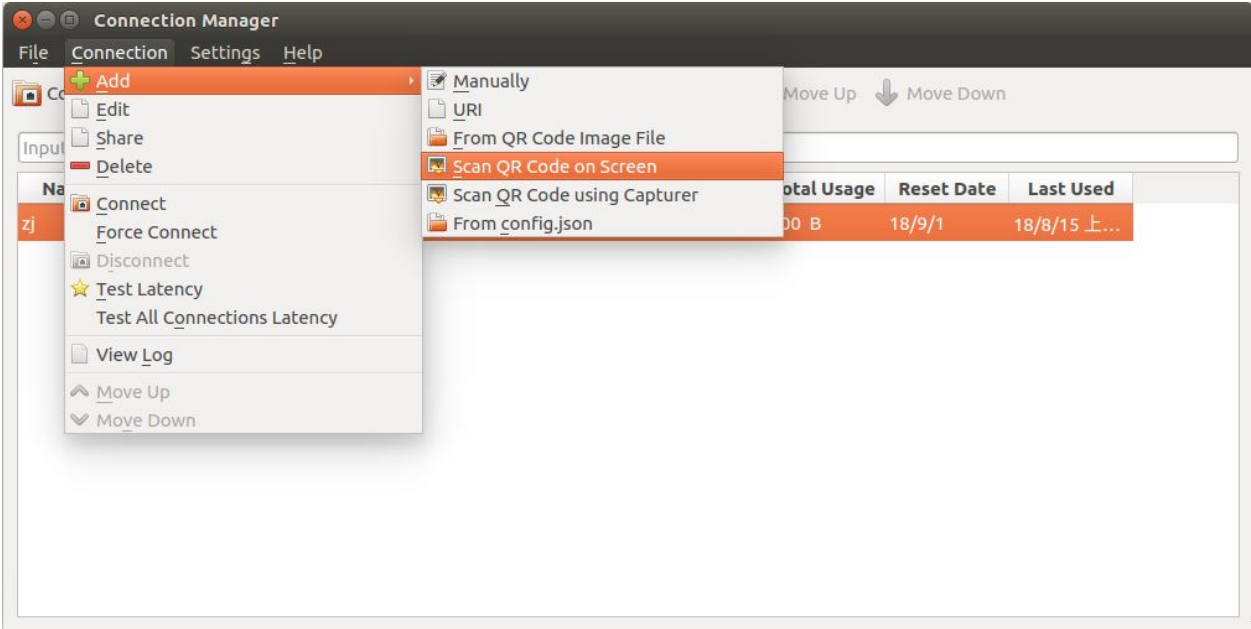
安装完成后可以在搜索栏中打开 Shadowsocks



这

里写图片描述

可以通过扫描二维码的方式获取 VPN 配置



这

里写图片描述

创建代理后点击连接即可

## 7.2 Chrome 代理配置

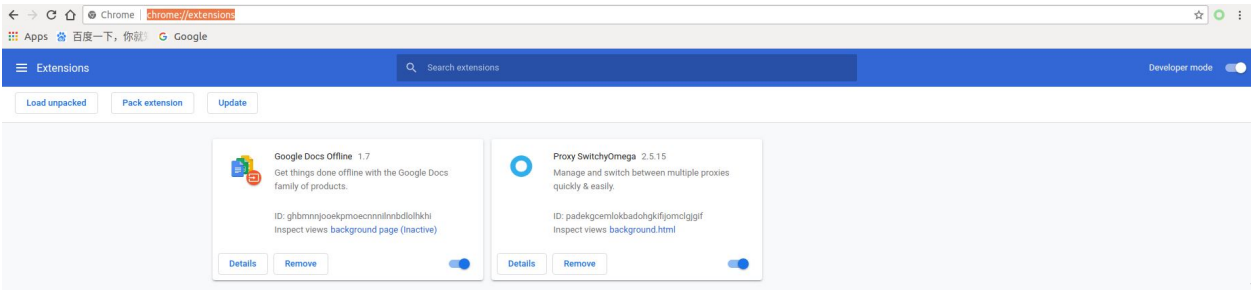
启动 Shadowsocks 代理后，还需要在浏览器（以 Chrome 为例）上进行代理设置，利用插件 SwitchyOmega 实现

### 7.2.1 SwitchyOmega 安装

参考：为 Chrome 配置 Shadowsocks 代理

国内下载网站：Proxy SwitchyOmega

下载插件后，打开 Chrome 浏览器，搜索栏中输入 `chrome://extensions/` 进入插件管理页面



这

里写图片描述

直接将插件拖拽到该页面即可自动安装



### 问题：插件无法安装

参考：

ubuntu 16.04 chrome 无法拖拽安装 crx 插件

linux 下怎么用命令行开启 chrome? 还有就是常见的安装软件 linux 下安装在哪儿?

如果上述方式无法安装插件，通过命令行启动 Chrome

```
google-chrome --enable-easy-off-store-extension-install
```

重复上述操作即可

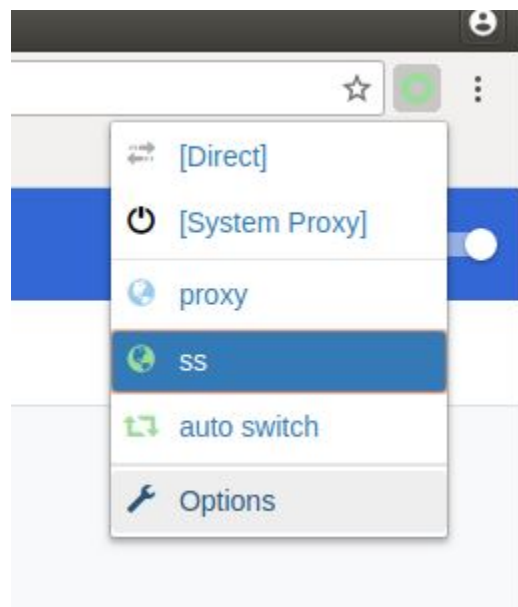
## 7.2.2 SwitchyOmega 代理

安装完成后，搜索栏旁边出现图标



里写图片描述

点击该图标 -> Options，打开设置页面



这里写图片描述

点击左侧 New profile 选项，新建配置文件

# SwitchyOmega



## SETTINGS

Interface

General

Import/Export

Pr

## PROFILES

proxy

ss

auto switch

New profile...

By

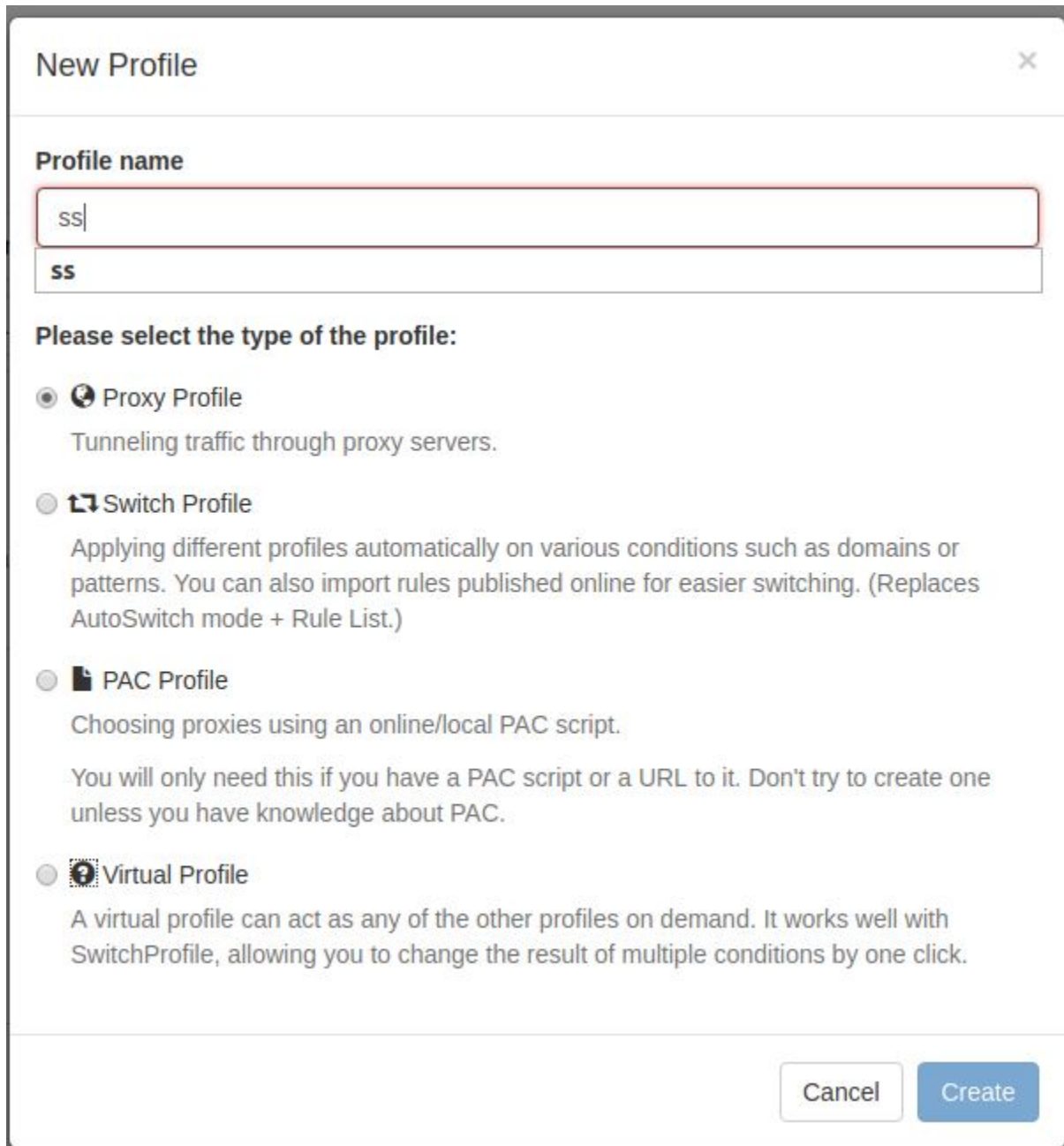
## ACTIONS

Apply changes

Discard changes

这里写图片描述

输入配置文件名（随意，比如 `ss`），默认选择 `Proxy Profile`，点击 `Create`



**New Profile**

**Profile name**

ss

ss

**Please select the type of the profile:**

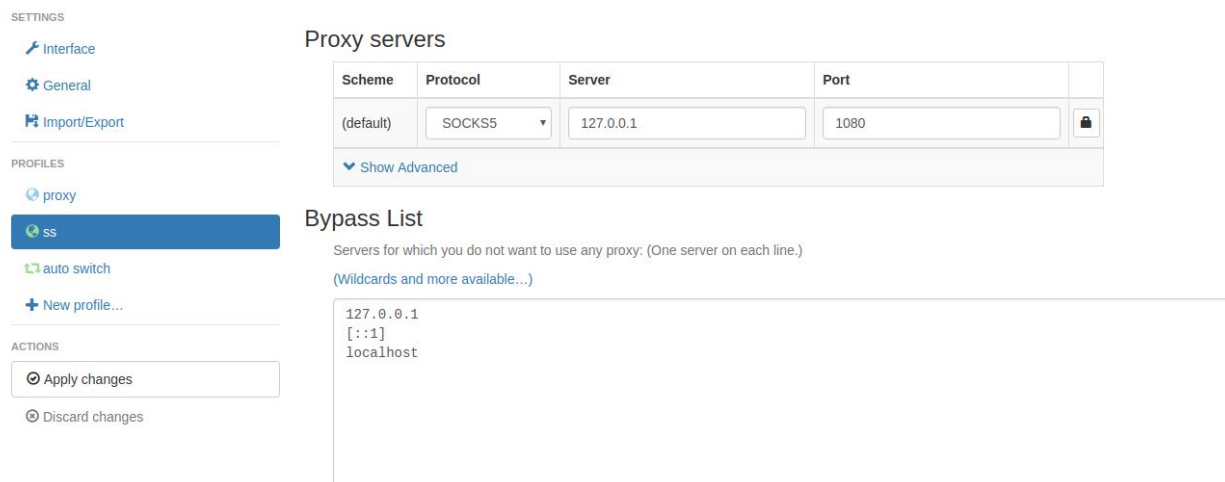
- ☒ **Proxy Profile**  
Tunneling traffic through proxy servers.
- ☐ **Switch Profile**  
Applying different profiles automatically on various conditions such as domains or patterns. You can also import rules published online for easier switching. (Replaces AutoSwitch mode + Rule List.)
- ☐ **PAC Profile**  
Choosing proxies using an online/local PAC script.  
You will only need this if you have a PAC script or a URL to it. Don't try to create one unless you have knowledge about PAC.
- ☐ **Virtual Profile**  
A virtual profile can act as any of the other profiles on demand. It works well with SwitchProfile, allowing you to change the result of multiple conditions by one click.

Cancel Create

这

里写图片描述

然后在新创建的配置文件中选择 Protocol -> SOCKS5, 输入 Server -> 127.0.0.1, 输入 Port -> 1080, 点击左下角的 Apply Changes



这

里写图片描述

完成上述操作后，点击 SwitchyOmega 图标，选择刚才创建的配置文件 ss，即可访问外网

## 7.3 全局代理配置

参考：

[ubuntu 基于 shadowsocks 的 PAC 全局代理：解决方案](#)

[ubuntu 下 shadowsocks（VPN）配置全局可用](#)

[JinnLynn/genpac](#)

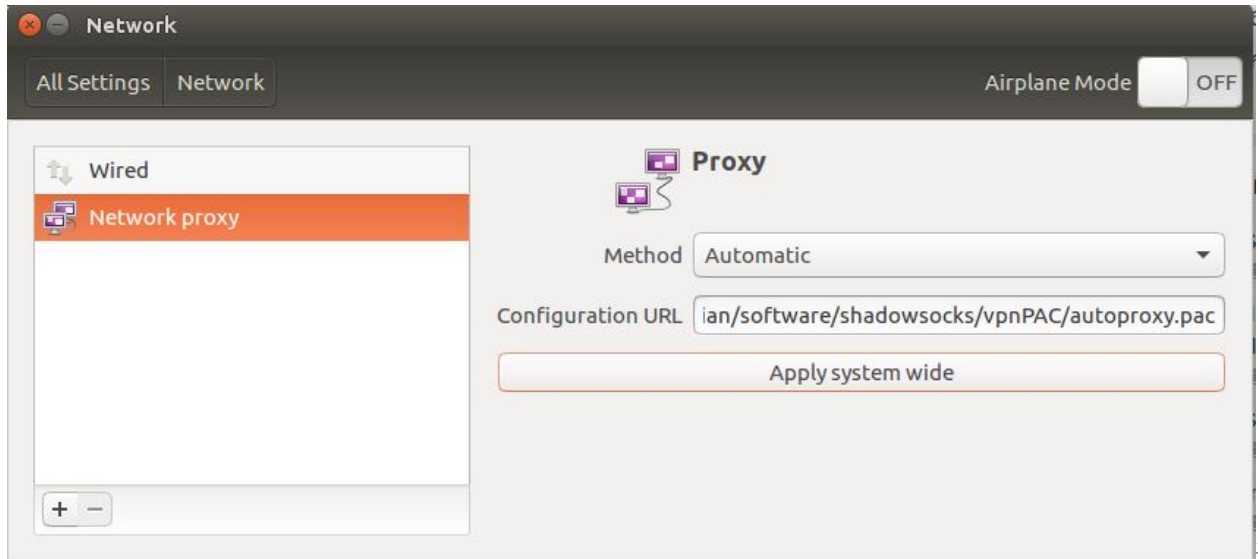
安装 GenPAC:

```
pip install genpac
```

生成本地 pac 文件

```
genpac --format=pac --pac-proxy="SOCKS5 127.0.0.1:1080" --output="autopproxy.pac"
```

打开系统设置 -> 网络 -> 网络代理



里写图片描述

Method 选项选择 Automatic, Configuration URL 输入本地文件

```
file:///you_path/autoproxy.pac
```

点击 Apply system wide 即可实现全局代理

### 7.3.1 ProxyChains

设置完成全局代理后, 在终端仍旧会出现无法连接的情况, 需要进一步设置代理, 可以使用工具 proxychains 或者 polipo

以 proxychains 为例

安装:

```
sudo apt-get install proxychains
```

修改配置文件 /etc/proxychains.conf

```
#socks4      127.0.0.1 9050
socks5 127.0.0.1 1080
```

重启应用

```
sudo /etc/init.d/proxychains restart
```

打开命令行窗口, 在输入命令的前面增加 proxychains 即可

```
proxychains curl www.google.com
```

问题: proxychains ping www.google.com 出错

```
ProxyChains-3.1 (http://proxychains.sf.net)
ERROR: ld.so: object 'libproxychains.so.3' from LD_PRELOAD cannot be preloaded (cannot_
↪open shared object file): ignored.
PING www.google.com (74.86.17.48) 56(84) bytes of data.
```

参考:

[proxychains-ld-preload-cannot-be-preloaded](#)

[Is ping not supposed to work via proxychains?](#)

这里面共有两个问题, 第一个问题: 没有预加载 libproxychains.so.3

查找:

```
locate libproxychains.so.3

/usr/lib/x86_64-linux-gnu/libproxychains.so.3
/usr/lib/x86_64-linux-gnu/libproxychains.so.3.0.0
```

可以修改预加载文件 /usr/bin/proxychains

```
cat /usr/bin/proxychains

#!/bin/sh
echo "ProxyChains-3.1 (http://proxychains.sf.net)"
if [ $# = 0 ] ; then
    echo "        usage:"
    echo "        proxychains <prog> [args]"
    exit
fi
export LD_PRELOAD=libproxychains.so.3
exec "$@"
```

或者自己添加到环境变量

```
export LD_PRELOAD=/usr/lib/x86_64-linux-gnu/libproxychains.so.3
```

第二个问题: proxychains 对 ping 没用

参考: [Is ping not supposed to work via proxychains?](#)

proxychains 使用的协议和 ping 使用的协议不同

---

## 7.4 相关

[ShadowsocksR](#)

[Linux 配置 SSR 客户端](#)

[在 Ubuntu 上使用 SSR 梯子](#)

[4 - Ubuntu 16.04 + SSR 翻墙](#)





---

### SwitchyOmega 代理设置

---

之前进行浏览器代理设置时只是简单的添加了代理服务器，并没有添加代理规则实现自动切换国内国外网站  
参考用 [SwitchyOmega](#) 管理代理设置进行自动切换代理设置

## 8.1 安装

github 下载地址：[FelisCatus/SwitchyOmega](#)

csdn 下载地址：[switchyomega](#)

安装步骤参考[SwitchyOmega 安装](#)

## 8.2 配置

### 8.2.1 删除已配置模式

安装完成后默认已设置好两个情景模式：`auto switch` 和 `proxy`，可以删除它们，删除顺序为 `auto switch`→`proxy`

### 8.2.2 添加代理服务器

新建一个情景模式 `New Profile`，选择类型为 `Proxy Profile`(代理服务器)，命名为 `ss`

在 Proxy Servers 中选择 Protocol 为 SOCKS5, Server 为 127.0.0.1, Port 为 1080, 最后点击左下角的 Apply Changes

## Proxy servers

Scheme	Protocol	Server	Port	
(default)	SOCKS5	127.0.0.1	1080	
<a href="#">Show Advanced</a>				

## 8.2.3 添加自动切换模式

再次新建一个 New Profile, 选择类型为 Switch Profile, 命名为 ss\_auto\_switch

### 规则列表设置

首先配置 Rule List Config, 选择 Rule List Format 为 AutoProxy, 在 Rule List URL 中填入规则列表

```
https://raw.githubusercontent.com/gfwlist/gfwlist/master/gfwlist.txt
```

点击 Download Profile Now, 里面包含了许多国外网址

其次配置 Switch Rules, 对于类型 Rule list rules, 选择 Profile 为 ss (就是刚才配置的代理服务器), 选择类型 Default 为 Direct (直连, 不经过代理服务器)

	<input checked="" type="checkbox"/> Rule list rules	(Any request matching the rule list below)	ss	
	Default		[Direct]	

其操作规则就是符合下面规则列表中的网址都经过代理服务器

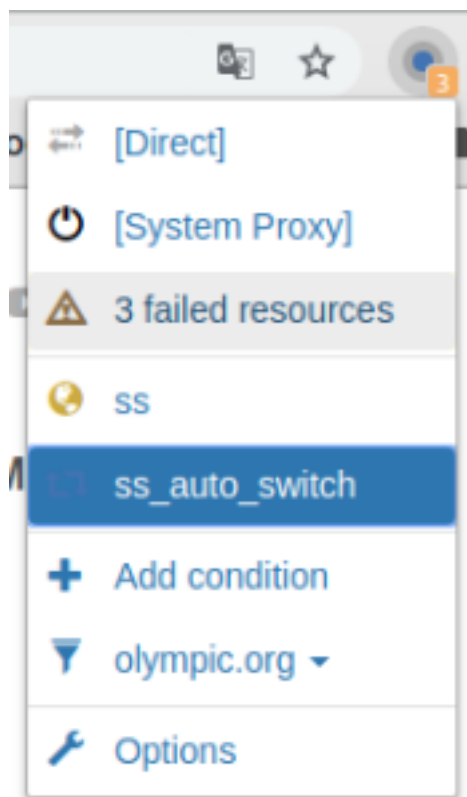
### 情景模式设置

规则列表中没有包含全部的国外网站, 还可以单独设置情景模式

在 Switch rules 中点击 Add condition 添加一行, 在 Condition Details 中添加想要切换的网址, 比如 \*.github.com, 在 Profile 中选择 ss, 这样符合条件的网址会通过代理服务器进行连接

Sort	Condition Type	Condition Details	Profile	Actions
↑↓	Host wildcard	<input type="text" value="*.github.com"/>	ss	

也可以在实际操作中添加切换情景, 点击 SwitchyOmega 图标, 选择 failed resources->add condition, 即可将这些网址添加到情景模式中







### Add condition to ss\_auto\_switch


A few resources failed to load due to issues with your network, proxy server or the webpage.

SwitchyOmega is just the reporter of these issues, not the cause of them.

















You can review the following domains and use proxy for them when appropriate.

- ☒  \*.googletagservices.com
- ☒  \*.googletagmanager.com
- ☒  \*.satismeter.com
- ☒  \*.olympicchannel.com

Use this profile for all selected domains

 SS

Cancel
 Add condition

↑↓	Host wildcard	*.olympicchannel.com	 SS	  
↑↓	Host wildcard	*.satismeter.com	 SS	  
↑↓	Host wildcard	*.googletagmanager.com	 SS	  
↑↓	Host wildcard	*.googletagservices.com	 SS	  

### 8.3 ip 查询

使用<https://www.geoiptool.com/zh/>或者<http://ip111.cn/>查询我的 ip 地址

当我直连时，使用的是杭州的 ip；当我使用代理服务器时，使用的是美国的 ip

Host/IP

主機名稱: 39.186.245.107

IP 地址: 39.186.245.107

国家:  China

国家代码: CN (CHN)

地区: Zhejiang

城市: Hangzhou

当地时间: 20 Apr 10:36 (CST+0800)

邮政编码: None

纬度: 30.2936

经度: 120.1614

Host/IP

Q

主機名稱: 140.82.7.93.vultr.com

IP 地址: 140.82.7.93

国家:  United States

国家代码: US (USA)

地区: California

城市: San Jose

当地时间: 19 Apr 19:33 (PDT-0700)

邮政编码: 95131

纬度: 37.3874

经度: -121.9024

---

### Ubuntu 全局代理设置

---

配置SwitchyOmega 代理设置仅仅能够实现浏览器代理，还需要设置全局代理才能在命令行访问国外资源

#### 9.1 生成 pac 文件

参考Ubuntu 代理配置安装 `genpac`

参考ubuntu 使用 `shadowsocks` 设置全局代理生成配置文件 `autoproxy.pac`

```
$ genpac --proxy="SOCKS5 127.0.0.1:1080" --gfwlist-proxy="SOCKS5 127.0.0.1:1080" -o
↪ autoproxy.pac --gfwlist-url="https://raw.githubusercontent.com/gfwlist/gfwlist/master/
↪ gfwlist.txt"
```

#### 9.2 系统配置

打开系统设置->Network->Network proxy，选择 Method 为 Automatic，在 Configuration URL 中填入生成的文件路径

```
file://文件路径
# 我的
file:///home/zj/software/vpn/autoproxy.pac
```

点击 Apply system wide 即可生效

## 9.3 代理工具设置

代理操作使用 SOCKS5 协议，大多命令行操作使用 HTTP/HTTPS 协议，所以需要使用代理工具进行转换

参考[Ubuntu 代理配置安装 ProxyChains](#)

测试命令

```
$ proxychains curl www.google.com
# 或
$ proxychains wget www.google.com
```



最近这几天经常出现网络断线的问题，重新找了 **ssr**(酸酸乳) 的服务端设置方法

### 10.1 ip 查找

参考：怎样检查搬瓦工 IP 被墙

使用网址Ping 检测查看外网服务器是否可以连接

使用网址全方位查询您的 IP 地址查看自己的 ip 信息

### 10.2 SSR 服务端

文章正确的翻墙姿势-SS&SSR比较详细的讲述了 **SS/SSR** 的发展历程，并给出了一些配置的建议

文章记录酸酸乳（SSR）服务器搭建历程，我只是想查个资料比较详细的讲解了利用 **vultr** 进行 **ssr** 服务端搭建的过程，其仓库地址已失效，使用如下地址：

```
# 原来的这个库被作者清空了
git clone https://github.com/Flyzy2005/ss-fly
# 在 github 上找了一个，自己也备份了一下
git clone https://github.com/zjZSTU/ss-fly.git
```

当前操作

1. 执行 `ssr`

```
$ ss-fly/ss-fly.sh -ssr
```

1. 输入密码

```
Please enter password for ShadowsocksR:  
(Default password: teddysun.com):
```

1. 输入端口号 (默认)

```
Please enter a port for ShadowsocksR [1-65535]  
(Default port: 10644):
```

1. 选择流密码 (默认)

```
Which cipher you'd select(Default: aes-256-cfb):
```

1. 选择协议: `auth_chain_a`

```
Which protocol you'd select(Default: origin):7
```

1. 选择混淆 (默认)

```
Which obfs you'd select(Default: plain):
```

完成服务端配置:

```
Congratulations, ShadowsocksR server install completed!  
Your Server IP      :  xx.xx.xx.xx  
Your Server Port    :  xxxxxx  
Your Password       :  xxxxxx  
Your Protocol       :  auth_chain_a  
Your obfs           :  plain  
Your Encryption Method: aes-256-cfb  
  
Welcome to visit:https://shadowsocks.be/9.html  
Enjoy it!
```

注意: 配置完成后如果失效, 尝试删除已有的进程, 再重新配置一遍

```
$ ps -aux | grep shadowsock
```

## 10.3 SSR 客户端

之前使用 SS 搭梯子的时候使用的是 Shadowsocks-Qt5，里面没有加密协议的选项，所以参考在 [Ubuntu 上使用 SSR 梯子安装electron-ssr](#)

参考：Debian 系列——Ubuntu18.04 为例

安装 electron-ssr 完成后打开，输入服务器端配置，设置为 PAC 代理即可

关于浏览器配置和系统设置，参考：

[SwitchyOmega 代理设置](#)

[Ubuntu 全局代理设置](#)



---

## [brook] 服务端 + 客户端设置

---

最近用 `ssr` 搭建的梯子经常会被封掉，打算换一种代理工具 - `brook`

### 11.1 介绍

`brook` 是一个跨平台的代理/VPN 软件，支持 Windows/Linux/Android/iOS 环境

其主要是通过命令行进行客户端和服务器的配置，网上说用的人还不多

### 11.2 下载

下载Linux `brook`

```
$ wget https://github.com/txtthinking/brook/releases/download/v20190601/brook
```

赋值可执行权限：

```
$ chmod +x brook
```

其他版本参考[Download](#)

## 11.3 服务器设置

启动服务器：

```
# ./brook server -l :port -p passwd
```

自定义端口号 `port` 和密码 `passwd`

## 11.4 客户端设置

```
$ ./brook client -l 127.0.0.1:1080 -i 127.0.0.1 -s server_ip:port -p passwd
```

1080 是 `sock5` 协议默认端口号，依次修改服务器 IP (`server_ip`)、端口号 (`port`) 和密码 (`passwd`)  
在浏览器和全局设置参考之前文章即可

## 11.5 优化

可以将命令行设置为后台运行或启动服务运行

### 11.5.1 后台运行

参考：

[ubuntu 后台运行的几种方法！](#)

[Shell 传递参数](#)

格式如下：

```
$ nohup 命令 &
```

可以新建服务端脚本 `brook-server.sh`：

```
#!/bin/bash
nohup ./brook server -l :$1 -p $2 &
```

执行脚本命令：

```
$ sh brook-server.sh port passwd
```

依次输入端口号和密码即可

同样可以新建客户端脚本 `brook-client.sh`：

```
#!/bin/bash
nphup ./brook client -l 127.0.0.1:1080 -i 127.0.0.1 -s server-ip:port -p passwd &
```

执行脚本命令：

```
$ sh brook-client.sh server-ip port passwd
```

依次输入服务器 ip，端口号和密码即可

## 11.5.2 启动服务运行

参考：Brook——一款优秀的小众代理软件（酸酸乳和酸酸的完美替代品）

通过启动服务来运行服务端 brook。新建文件 brook.service

```
[Unit]
Description=brook service
After=network.target syslog.target
Wants=network.target

[Service]
Type=simple
ExecStart=/path/to/brook server -l :port -p passwd # 使用绝对路径

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

设置 root 权限：

```
$ sudo chmod 777 brook.service
```

放置在/etc/systemd/system 文件夹内，使用工具 systemctl 启动服务

```
systemctl start brook          # 启动 brook
systemctl stop brook           # 停止 brook
systemctl status brook         # 查看 brook 服务状态
systemctl restart brook        # 重启 brook 服务
systemctl enable brook         # 将 brook 加入开机启动项
```

## 11.6 相关

除了 brook 外，还找到一个代理工具V2Ray

youtube 上有关翻墙软件的测评：哪种翻墙软件更隐蔽?! 联通后台数据实测 SS/SSR/V2Ray/Brook 翻墙 APP! 请耐心等待看完本教程，能让你避免被请喝茶!



参考：

如果使用 SS 或 SSR 频繁被墙 IP 墙端口那么你需要使用 [v2ray 科学上网](#)

[V2Ray 搭建详细图文教程](#)

今天翻墙又双🇬🇪不行了，这次打算使用[v2ray](#)，不过其官网一直打不开，所以主要参考[233boy/v2ray](#)

## 12.1 什么是 v2ray

v2ray 是一个代理软件，相比于 `shadowsocks`，它实现了更多的代理协议和功能（包括 `shadowsocks`）

## 12.2 安装

有 2 种方式：

1. 官方脚本
2. 第三方脚本（推荐）

### 12.2.1 官方脚本

v2ray 提供了一个安装脚本

```

$ sudo su

# bash <(curl -L -s https://install.direct/go.sh)
Installing V2Ray v4.20.0 on x86_64
Downloading V2Ray: https://github.com/v2ray/v2ray-core/releases/download/v4.20.0/v2ray-
↳linux-64.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  608      0  608    0     0   527      0  --:--:--  0:00:01  --:--:--   528
100 11.3M  100 11.3M    0     0 1442k      0  0:00:08  0:00:08  --:--:-- 2501k
Extracting V2Ray package to /tmp/v2ray.
Archive:  /tmp/v2ray/v2ray.zip
  inflating: /tmp/v2ray/config.json
  creating: /tmp/v2ray/doc/
  inflating: /tmp/v2ray/doc/readme.md
  inflating: /tmp/v2ray/geoip.dat
  inflating: /tmp/v2ray/geosite.dat
  creating: /tmp/v2ray/systemd/
  inflating: /tmp/v2ray/systemd/v2ray.service
  creating: /tmp/v2ray/systemv/
  inflating: /tmp/v2ray/systemv/v2ray
  inflating: /tmp/v2ray/v2ctl
  extracting: /tmp/v2ray/v2ctl.sig
  inflating: /tmp/v2ray/v2ray
  extracting: /tmp/v2ray/v2ray.sig
  inflating: /tmp/v2ray/vpoint_socks_vmess.json
  inflating: /tmp/v2ray/vpoint_vmess_freedom.json
PORT:19088
UUID:98405c82-8056-4694-9676-d166da29f0c8
Created symlink from /etc/systemd/system/multi-user.target.wants/v2ray.service to /etc/
↳systemd/system/v2ray.service.
V2Ray v4.20.0 is installed.

```

其仅实现了安装功能，其他操作（管理/卸载等）都很麻烦，不推荐使用

### 12.2.2 第三方脚本

233boy 提供了一个一键安装和管理脚本，参考[V2Ray 一键安装脚本](#)，里面不仅集成了安装/卸载功能，还具备配置选项

下载并执行脚本 `v2ray.sh`，当前选择默认配置

```
$ bash <(curl -s -L https://git.io/v2ray.sh)
..... V2Ray 一键安装脚本 & 管理脚本 by 233v2.com .....
```

帮助说明: <https://233v2.com/post/1/>

搭建教程: <https://233v2.com/post/2/>

1. 安装

2. 卸载

请选择 [1-2]:1

请选择 V2Ray 传输协议 [1-32]

1. TCP
2. TCP\_HTTP
3. WebSocket
4. WebSocket + TLS
5. HTTP/2
6. mKCP
7. mKCP\_utp
8. mKCP\_srtp
9. mKCP\_wechat-video
10. mKCP\_dtls
11. mKCP\_wireguard
12. QUIC
13. QUIC\_utp
14. QUIC\_srtp
15. QUIC\_wechat-video
16. QUIC\_dtls
17. QUIC\_wireguard
18. TCP\_dynamicPort
19. TCP\_HTTP\_dynamicPort
20. WebSocket\_dynamicPort
21. mKCP\_dynamicPort
22. mKCP\_utp\_dynamicPort
23. mKCP\_srtp\_dynamicPort
24. mKCP\_wechat-video\_dynamicPort
25. mKCP\_dtls\_dynamicPort

(下页继续)

(续上页)

```

26. mKCP_wireguard_dynamicPort
27. QUIC_dynamicPort
28. QUIC_utp_dynamicPort
29. QUIC_srtp_dynamicPort
30. QUIC_wechat-video_dynamicPort
31. QUIC_dtls_dynamicPort
32. QUIC_wireguard_dynamicPort

```

备注 1: 含有 [dynamicPort] 的即启用动态端口..

备注 2: [utp | srtp | wechat-video | dtls | wireguard] 分别伪装成 [BT 下载 | 视频通话 | 微信视频通话 | DTLS 1.2 数据包 | WireGuard 数据包]

(默认协议: TCP):

V2Ray 传输协议 = TCP

---

请输入 V2Ray 端口 [1-65535]

(默认端口: 60663):

V2Ray 端口 = 60663

---

是否开启广告拦截 (会影响性能) [Y/N]

(默认 [N]):

广告拦截 = 关闭

---

是否配置 Shadowsocks [Y/N]

(默认 [N]):

.... 准备安装了咯.. 看看有毛有配置正确了...

----- 安装信息 -----

V2Ray 传输协议 = TCP

(下页继续)

(续上页)

```
V2Ray 端口 = 60663
```

```
是否配置 Shadowsocks = 未配置
```

```
----- END -----
```

按 Enter 回车键 继续.... 或按 Ctrl + C 取消.

```
extracting: /tmp/v2ray/v2ctl.sig
inflating: /tmp/v2ray/v2ray
extracting: /tmp/v2ray/v2ray.sig
inflating: /tmp/v2ray/vpoint_socks_vmess.json
inflating: /tmp/v2ray/vpoint_vmess_freedom.json
```

.. 由于你的 VPS 内核支持开启 BBR ... 已经为你启用 BBR 优化....

```
----- V2Ray 配置信息 -----
```

```
地址 (Address) = 207.xx.xx.22
```

```
端口 (Port) = 6xx63
```

```
用户 ID (User ID / UUID) = 7ca73c1f-9f46-4bf7-9435-75b26724xxxa2
```

```
额外 ID (Alter Id) = 233
```

```
传输协议 (Network) = tcp
```

```
伪装类型 (header type) = none
```

```
----- END -----
```

V2Ray 客户端使用教程: <https://233v2.com/post/4/>

提示: 输入 v2ray url 可生成 vmess URL 链接 / 输入 v2ray qr 可生成二维码链接

完成之后如果想要重新配置, 直接在命令行输入 v2ray 即可

```
# v2ray
```

(下页继续)

```
..... V2Ray 管理脚本 v3.14 by 233v2.com .....
```

## V2Ray 版本: v4.20.0 / V2Ray 状态: 正在运行 ##

帮助说明: <https://233v2.com/post/1/>

反馈问题: <https://github.com/233boy/v2ray/issues>

TG 群组: <https://t.me/blog233>

捐赠脚本作者: <https://233v2.com/donate/>

捐助 V2Ray: [https://www.v2ray.com/chapter\\_00/02\\_donate.html](https://www.v2ray.com/chapter_00/02_donate.html)

1. 查看 V2Ray 配置
2. 修改 V2Ray 配置
3. 下载 V2Ray 配置 / 生成配置信息链接 / 生成二维码链接
4. 查看 Shadowsocks 配置 / 生成二维码链接
5. 修改 Shadowsocks 配置
6. 查看 MTPROTO 配置 / 修改 MTPROTO 配置
7. 查看 Socks5 配置 / 修改 Socks5 配置
8. 启动 / 停止 / 重启 / 查看日志
9. 更新 V2Ray / 更新 V2Ray 管理脚本
10. 卸载 V2Ray
11. 其他

温馨提示...如果你不想执行选项...按 Ctrl + C 即可退出

请选择菜单 [1-11]:

## 12.3 配置文件

参考: [Ubuntu 中 v2ray 客户端配置实例](#)

v2ray 安装完成后通过 `/etc/v2ray/config.json` 文件进行配置, 首先查询服务器配置信息:

```
----- V2Ray 配置信息 -----

地址 (Address) = 149.xx.8x.30

端口 (Port) = xxxxx

用户 ID (User ID / UUID) = 9d514xxx0-069e-4979-9b6e-c66085a915e1

额外 ID (Alter Id) = 233

传输协议 (Network) = tcp

伪装类型 (header type) = none
```

在客户端安装 v2ray, 修改配置文件如下:

```
$ cat /etc/v2ray/config.json

{
  "inbounds": [{
    "port": 1080,
    "listen": "127.0.0.1",
    "protocol": "socks",
    "settings": {
      "udp": true
    }
  }],
  "outbounds": [{
    "protocol": "vmess",
    "settings": {
      "vnext": [{
        "address": "149.28.84.30", // 服务器 IP
        "port": 39755,           // 服务器端口号
        "users": [{ "id": "9d514420-069e-4979-9b6e-c66085a915e1" }] // 用户 ID
      }]
    }
  ]
}
```

(下页继续)

(续上页)

```
},{
  "protocol": "freedom",
  "tag": "direct",
  "settings": {}
}],
"routing": {
  "domainStrategy": "IPOnDemand",
  "rules": [{
    "type": "field",
    "ip": ["geoip:private"],
    "outboundTag": "direct"
  }]
}
}
```

## 12.4 启动

v2ray 以服务形式管理应用

```
# 启动 v2ray
$ service v2ray start
# 停止 v2ray
$ service v2ray stop
# 查看 v2ray 状态
$ service v2ray status
```

## 12.5 SS

v2ray 允许开启 Shadowsocks 功能，在命令行输入 v2ray，选择 5. 修改 Shadowsocks 配置即可

```
$ v2ray
...
...
请选择菜单 [1-11]:5
```

大佬... 你没有配置 Shadowsocks ... 不过现在想要配置的话也是可以的 ^\_^

(下页继续)



(续上页)

是否配置 Shadowsocks [Y/N]

(默认 [N]): Y

请输入 Shadowsocks 端口 [1-65535], 不能和 V2ray 端口相同

(默认端口: 47133):

Shadowsocks 端口 = 47133

---

请输入 Shadowsocks 密码

(默认密码: 233blog.com): zhujian

Shadowsocks 密码 = zhujian

---

请选择 Shadowsocks 加密协议 [1-7]

1. aes-128-cfb

2. aes-256-cfb

3. chacha20

4. chacha20-ietf

5. aes-128-gcm

6. aes-256-gcm

7. chacha20-ietf-poly1305

(默认加密协议: chacha20-ietf-poly1305): 2

Shadowsocks 加密协议 = aes-256-cfb

---

(下页继续)

(续上页)

按 Enter 回车键 继续.... 或按 Ctrl + C 取消.

----- Shadowsocks 配置信息 -----

服务器地址 = 207.xx.xx.22

服务器端口 = 4xx33

密码 = zhujian

加密协议 = aes-256-cfb

SS 链接 = ss://YWVzLTl1Ni1jZxxxemh1amlhbKAYMDcuMTQ4LjQuMjI6NDcxMzM=#233v2.com\_ss\_207.  
↪148.4.22

备注: Shadowsocks Win 4.0.6 客户端可能无法识别该 SS 链接

提示: 输入 v2ray ssqr 可生成 Shadowsocks 二维码链接

## 12.6 额外

[Just My Socks 详细图文购买教程](#)

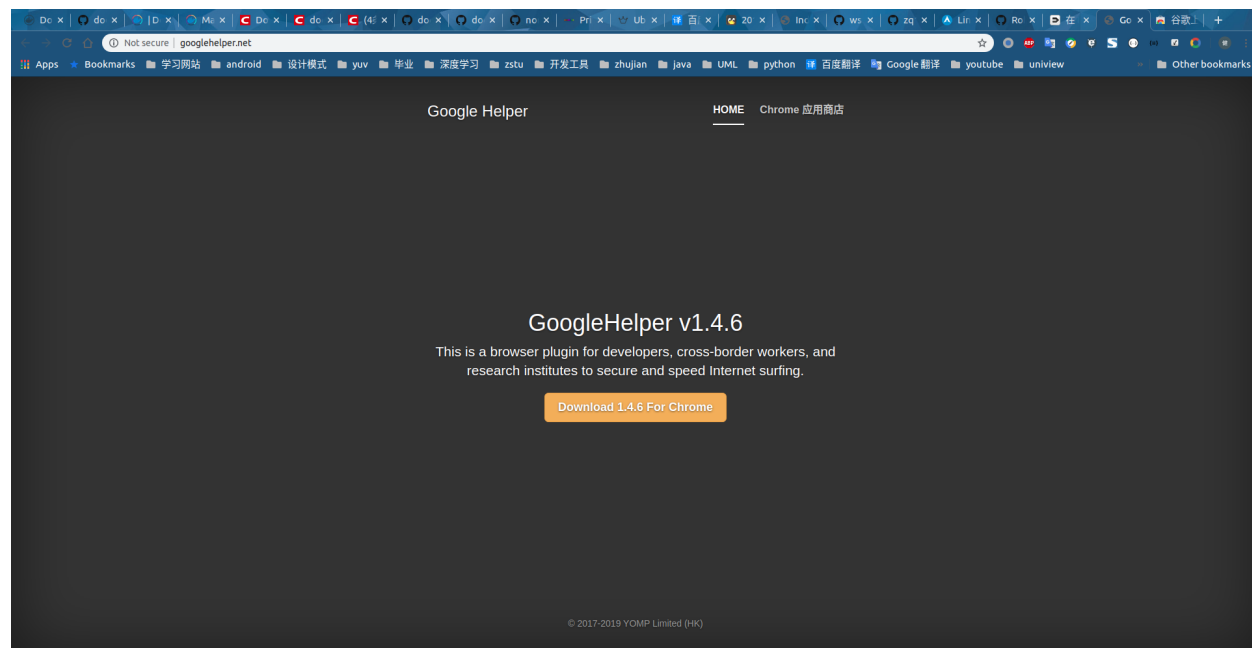
[\[科普\] 为什么你用的 VPN 网速那么慢?](#)

## [Google Helper] 辅助搜索工具

最近一段时间梯子限速很严重，听朋友介绍了一个 Chrome 插件 Google Helper，能够在没有梯子的情况下进行 Google 搜索

### 13.1 概述

官网地址：[Google Helper](http://googlehelper.net)



其介绍如下：

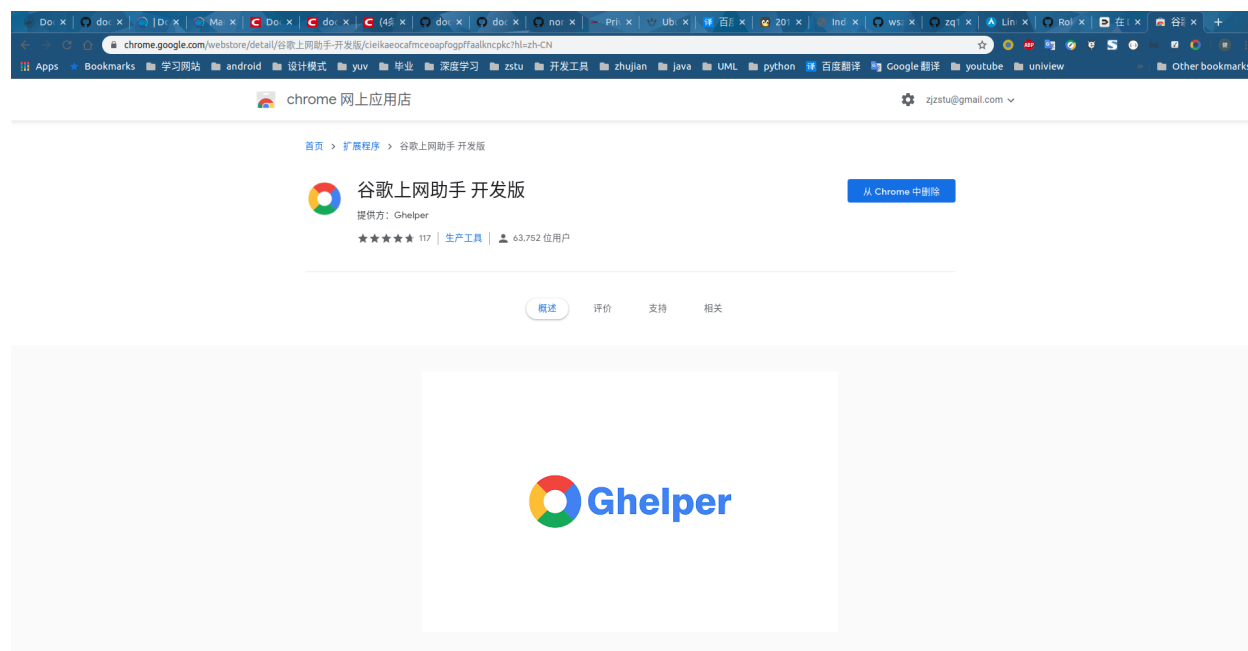
专门为科研、外贸、跨境电商、海淘人员、开发人员服务的上网加速工具，chrome 内核浏览器专用！可以解决 chrome 扩展无法自动更新的问题，同时可以访问谷歌 google 搜索，gmail 邮箱，google+ 等谷歌产品

## 13.2 安装

有两种方式进行安装：

1. 下载安装包
2. Google 应用商店（推荐）

官方推荐第二种，这样能够保证版本正常升级，如果没有外网的话选择第一种方式

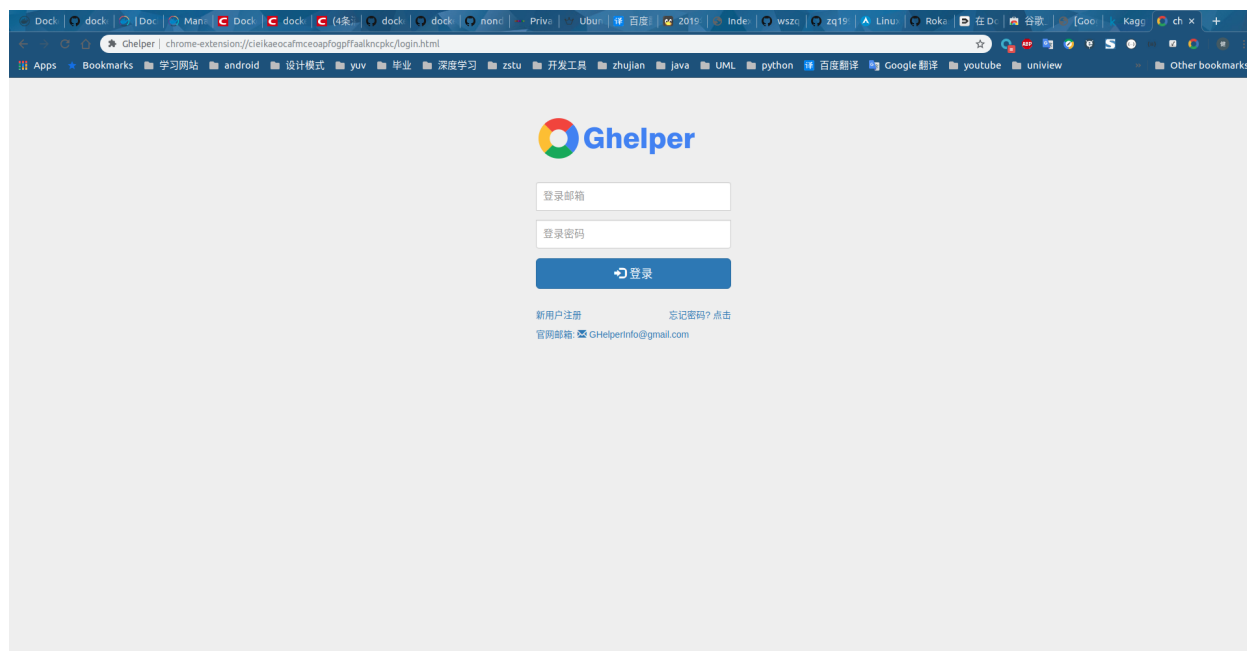


## 13.3 体验

安装完成后在插件栏会出现以下图标



需要点击登录账户



首次登录会有 3 天的 VIP 体验，这时候就可以使用 Google 搜索了，除了一些指定网站不能打开（比如 YOUTUBE/FACEBOOK 等），其他一些国外的学习网站（比如 KAGGLE）都可以打开

之后的使用可以点击左侧加速白名单，可以选择设置代理（之前有的话）或者购买 VIP 会员（就是官方的代理）

使用 *Chrome Helper* 会关闭其他代理软件，比如 *SwitchyOmega*



参考：V2Ray 安卓手机客户端下载、安装及使用教程

### 14.1 移动客户端

下载2dust/v2rayNG的最新版本

### 14.2 配置

进入服务器，输入命令 `v2ray`，选择选项 3

3. 下载 V2Ray 配置 / 生成配置信息链接 / 生成二维码链接

再选择选项 4，即可生成二维码链接

请选择菜单 [1-11]:3

1. 直接下载 V2Ray 客户端配置文件（仅支持 Xshell）
2. 生成 V2Ray 客户端配置文件下载链接
3. 生成 V2Ray 配置信息链接

(下页继续)

(续上页)

#### 4. 生成 V2Ray 配置二维码链接

使用手机客户端扫描浏览器生成的二维码即可完成配置



## CHAPTER 15

---

[todo]google cloud

---

参考：

谷歌云免费服务器申请方法（附一键搭建 SSR 和 BBR 教程）

用 Google Cloud Platform 搭建 Shadowsocks 服务教程

google 云服务地址：google cloud

### 15.1 信用卡

需要申请 visa 信用卡或者 master 信用卡