

使用Vultr的服务器搭建ShadowSocksR

1. 前言	1
2. 从易到难：从SHADOWSOCKSR客户端装起.....	2
2.1. 下载	2
2.2. 安装	2
2.3. 运行	3
2.3.1. Windows	3
2.3.1.1. 添加服务器	3
2.3.1.2. 关于代理模式	4
2.3.1.3. 检视windows系统代理设置.....	5
2.3.2. macOS	6
2.3.3. Android	8
2.3.4. iOS	8
3. 在服务器架设SSR服务端	9
3.1. 租服务器	9
3.1.1. 选择服务商	9
3.1.2. 建立服务器	9
3.2. 建立Mac到服务器的连接（不以windows PC举例）	11
3.3. 在服务器安装SSR服务端	14
3.3.1. 获取Python包	14
3.3.2. 获取源代码	15
3.3.3. 配置服务端	15
3.3.4. 创建用户	16
3.3.5. 修改、删除用户	16
3.3.6. 服务端运行与停止	17
4. 法律问题	18
5. 参考	18

1. 前言

ShadowSocksR (SSR) 是ShadowSocks (SS) 的派生项目。两个项目都开源，作者都是女生，不是男生。ShadowSocks的作者被喝过茶所以项目不再由她管理。ShadowSocksR的作者为了证明SSR比SS更能躲避侦测，开发了“能识别”旧版本SS的程序，与ShadowSocks的用户发生争执，最后被人身威胁，以此为导火线退出项目管理。因此两个项目都在由其他不知名但有名字的程序员维护。想了解ShadowSocks以及ShadowSocksR的运行原理，可以访问以下链接，先不看也行。

半官方指南：

[Shadowsocks 自动安装脚本](#)

[ShadowsocksR project 备份组织](#)

路边指南：

[ShadowSocks（影梭）不完全指南](#)

[ShadowsocksR Clients and Server](#)

2. 从易到难：从shadowsocksR客户端装起

2.1. 下载

Windows客户端发布页：[ShadowsocksR-win-4.8.0.rar](#)

macOS客户端发布页：[ShadowsocksX-NG-R8.dmg](#)

Android客户端发布页：[shadowsocksr-android-3.5.1.1.apk](#)

iOS客户端（全部中国下架）：Surge（贵），ShadowRocket（一般），Potatso2（值），PotatsoLite（便宜也值）

上述链接只提供GitHub链接，因为是我国民族企业剽窃作品的重要来源，暂时没封。

2.2. 安装

略。

2.3. 运行

2.3.1. Windows

详细: [ShadowsocksR Clients and Server](#)

解压后会有两个SSR, 运行哪个?

For >= Windows 8 or with .NET 4.0, use ShadowsocksR-dotnet4.0.exe

For <= Windows 7 or with .NET 2.0, use ShadowsocksR-dotnet2.0.exe

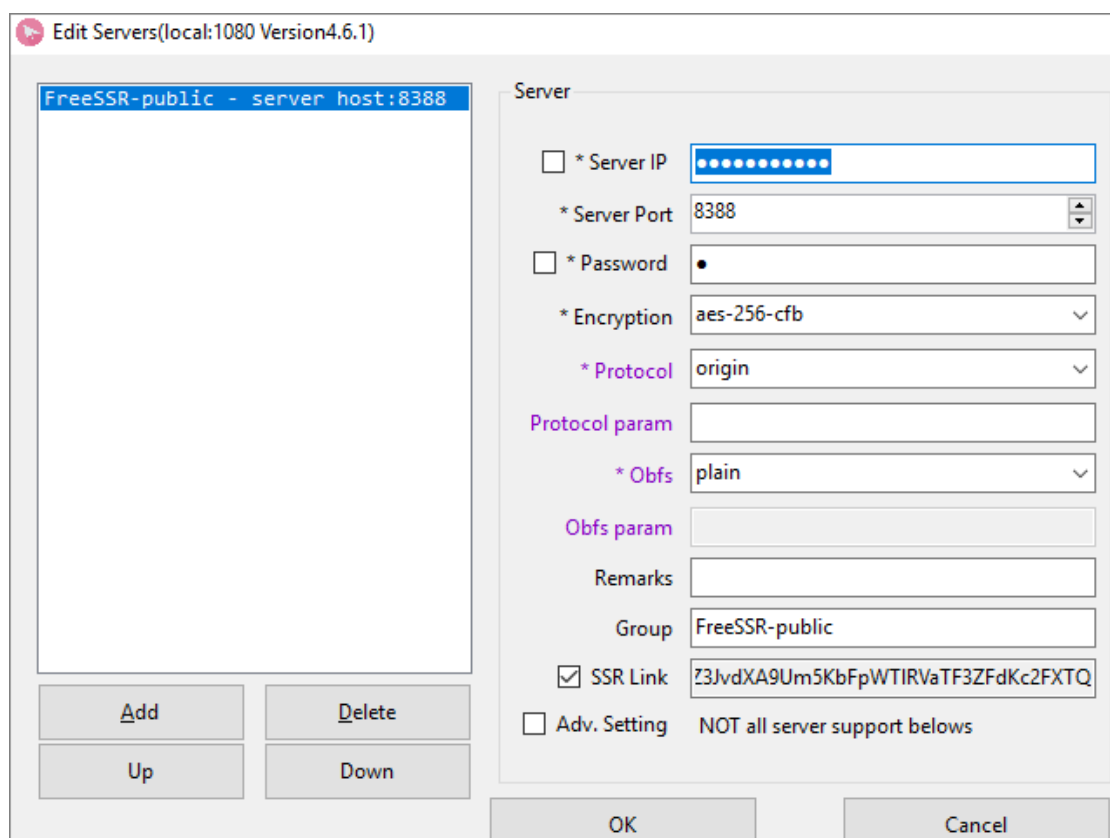
2.3.1.1. 添加服务器

当你从别人那里买了SSR服务或者自己在服务器搭建了SSR服务端, 你会得到三类信息:

1. 二维码 (QRCode) ;
2. 开头为ssr://的链接 (SSR Link) ;
3. 服务端用户的服务器地址 (server IP)、端口 (port)、密码、加密方式 (method)、协议插件 (protocol)、混淆协议 (obfs)

通过二维码添加选择**scan QRCode**。通过ssr link添加则选择**Import SSR links from clipboard**, 复制link后选择它就会自动添加。

通过具体参数添加



2.3.1.2. 关于代理模式

以windows10为例

Mode有个选项问你是否启用系统代理（Windows system-wide proxying），意思是，是否让系统所有（几乎）联网程序都依照SSR的设置进行联网。

举例，如果开启系统代理，而SSR设置使用日本服务器&全局模式，那么只要是默认使用系统网络的程序都会走日本服务器&全局模式。

如果不开系统代理，然后你只在Firefox浏览器的网络设置中手动设置代理（SOCKSV5，服务器IP 127.0.0.1， port 1080），那么只有这个浏览器的流量会走SSR，使用日本服务器&全局模式。除此之外的其他程序都会走正常网络。

抽象化来看，SSR的客户端就像你电脑/手机中的服务器，有自己的IP（127.0.0.1），你要在自己电脑设置你的系统或个别程序是否连接这个服务器。

建议勾选，否则要在所有联网软件中手动配置代理。

主要有**PAC**自动识别模式和**全局**模式。

PAC模式对gfwlist中的站点以及用户自定义proxy rule的站点自动应用代理。

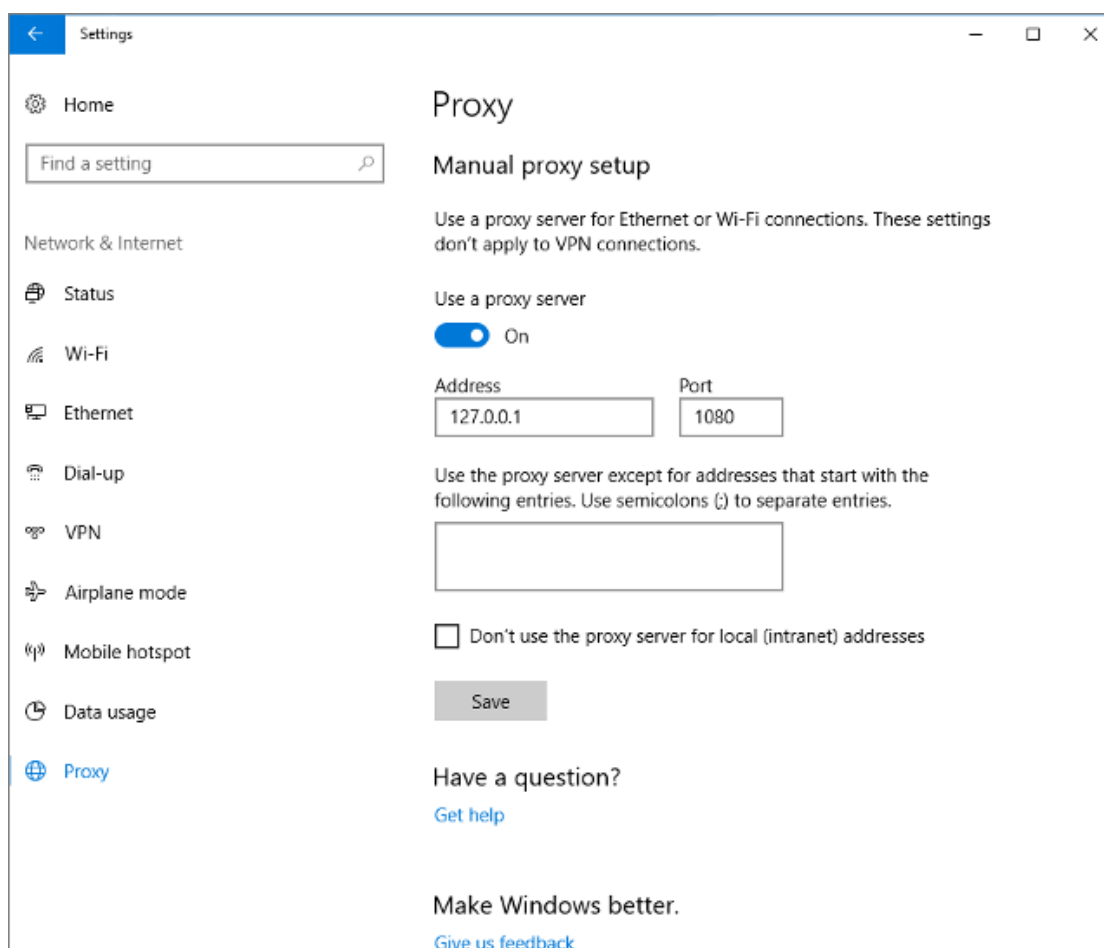
全局模式则将计算机（几乎，包括下载软件、网盘软件）所有连接都应用代理。

“如果选择了全局模式，则所有流量将无差别通过 ShadowsocksR 服务器进行访问，如果是访问国内的一些网络服务，速度会受到影响。一般建议选择 PAC 模式，这样系统会自动联网下载一个gfwlist文件（文件地址<https://raw.githubusercontent.com/gfwlist/gfwlist/master/gfwlist.txt>），简单理解，就是一个规则列表，若访问的网络服务满足列表中的网址规则，则会走 ShadowsocksR 流量，否则直接走本机网络访问。这样带来的好处是，会根据规则区分某个网站是否走代理，从而提升用户的网络体验。”

在不了解**PAC**运作的情况下建议先使用全局模式，通过google了解**PAC**模式后再用**PAC**模式。

2.3.1.3. 检视windows系统代理设置

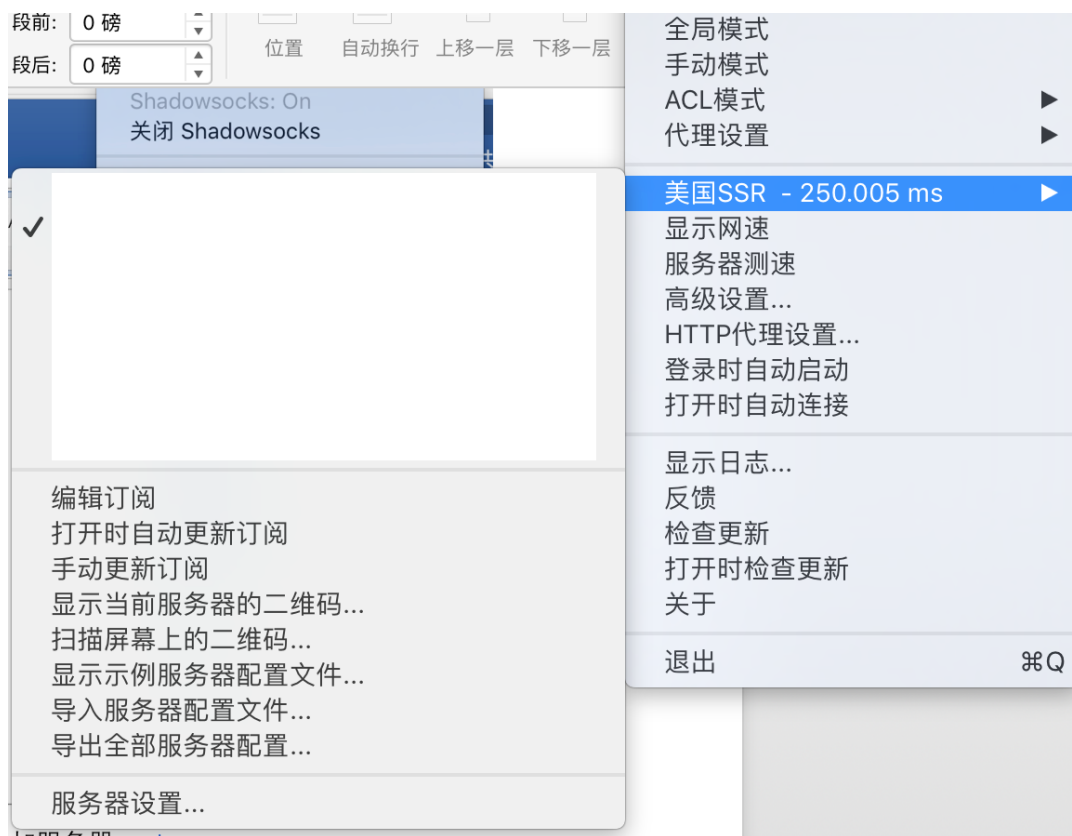
选择启用系统代理后，建议去**Settings > Network & Internet > Proxy**看看是不是已经配置了像这样的代理。



如无问题就可以用了。

2.3.2. macOS

跟Windows差不多方法添加服务器。



设置完服务器，到高级设置修改GFW LIST地址为：

<https://raw.githubusercontent.com/gfwlist/gfwlist/master/gfwlist.txt>



高级设置

如果你不明白这些选项是什么，请不要修改

本地 Socks5 监听地址: 127.0.0.1

本地 Socks5 监听端口: 1086

本地 PAC 监听地址: 127.0.0.1

本地 PAC 监听端口: 8,090

超时: 60 秒

☐ 启用 Udp Relay

☐ Enable Verbose Mode

GFW LIST URL:

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

ACL 白名单列表 URL:

`https://raw.githubusercontent.com/shadowsocksr-libev/master/acl/chn.acl`

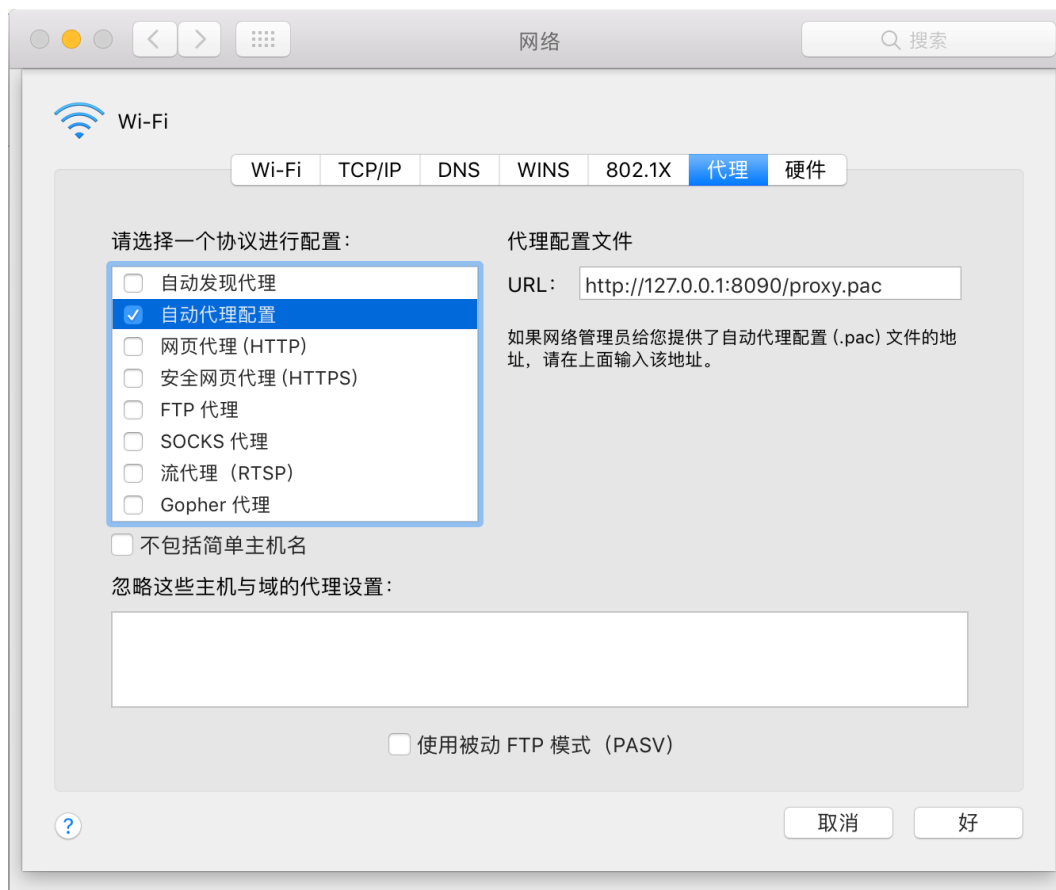
ACL GFW List 列表 URL:

`https://raw.githubusercontent.com/shadowsocksr-libev/master/acl/gfwlist.acl`

ACL Back China URL:

`https://raw.githubusercontent.com/shadowsocksr-libev/ShadowsocksX-NG/develop/ShadowsocksX-NG/backchn.acl`

然后到macOS的设置》网络》WIFI（或USB、Thunderbolt Ethernet）》代理，设定自动代理配置，如下图填URL（注意地址：端口要以上图高级设置里面PAC监听端口为准。



PAC相关设置在代理模式那里，用之前点一下更新PAC。用户自定义规则有时候不好用，不太懂。像这样把域名中间填进去。

.kmb.

.kknews.

.pixnet.

其他的都参考windows的设置。

2.3.3. Android

没用过，都差不多的。可以参考[这里](#)。

2.3.4. iOS

每个软件都差不多意思，不是很想解释。参数不要填错，二维码不要扫错就是了。

不知道怎么注册、购买其他区appleid的可以看[这里](#)。点卡去淘宝买。

iOS这几个软件都可以添加主页widget，很方便。

3. 在服务器架设SSR服务端

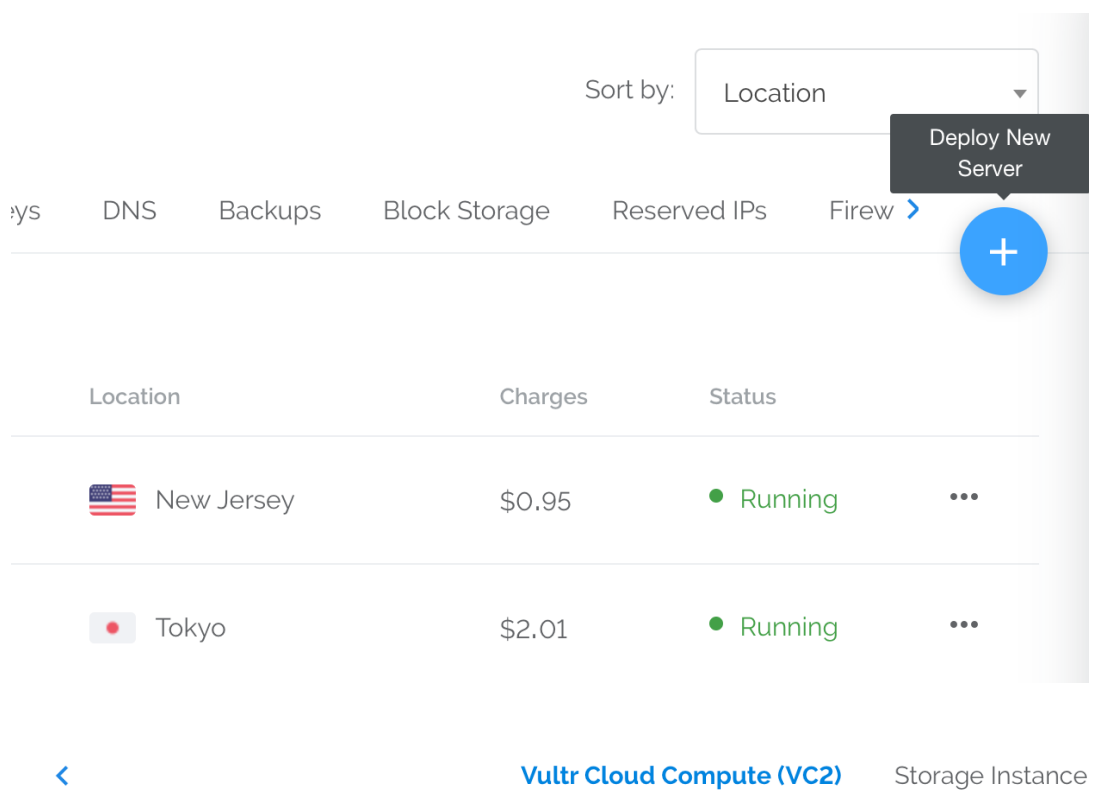
3.1. 租服务器

3.1.1. 选择服务商

暂时推荐Vultr的日本Tokyo和美国New York (NJ) 服务器。具体可以在[这里](#)测试下载速度。Tokyo每月5刀起，New York每月2.5刀起。



付款方式：信用卡、Alipay、PayPal、BTC

3.1.2. 建立服务器



Sort by: Location

Deploy New Server

Location	Charges	Status
 New Jersey	\$0.95	● Running
 Tokyo	\$2.01	● Running

< Vultr Cloud Compute (VC2) Storage Instance

1 Server Location

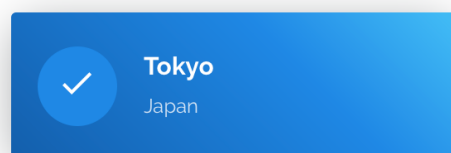
< All Locations

America

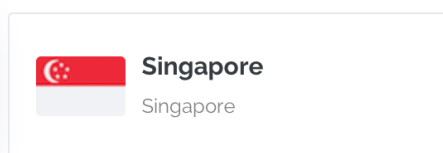
Europe


Australia

Asia



✓ Tokyo
Japan











 Singapore
Singapore

服务器系统选择Debian8 x64

2 Server Type

< **64 bit OS** 32 bit OS Application Upload ISO ISO Library Backup Snapshot

 CentOS SELECT VERSION	 CoreOS Stable x64	 Debian SELECT VERSION 9 x64 8 x64 7 x64
 Fedora SELECT VERSION	 FreeBSD SELECT VERSION	 OpenBSD 6 x64
 Ubuntu	 Windows	

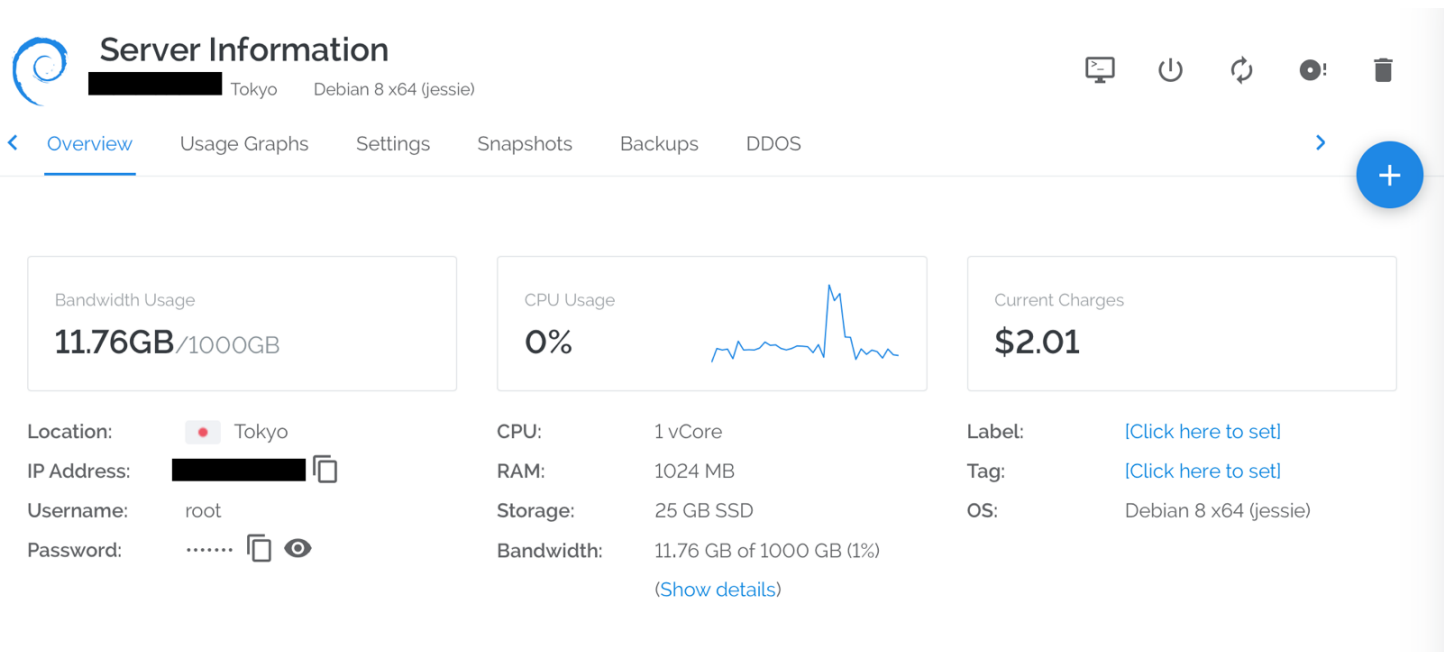
3 Server Size

Temporarily Sold Out 20 GB SSD \$2.50/mo \$0.004/h 1 CPU 512MB Memory 500GB Bandwidth	25 GB SSD \$5/mo \$0.007/h 1 CPU 1024MB Memory 1000GB Bandwidth	40 GB SSD \$10/mo \$0.015/h 1 CPU 2048MB Memory 2000GB Bandwidth
60 GB SSD \$20/mo \$0.03/h	100 GB SSD \$40/mo \$0.06/h	200 GB SSD \$80/mo \$0.119/h

如果是New York，有每个月2.5刀的。

其他不用管了，直接点Deploy Now.

3.2. 建立Mac到服务器的连接（不以windows PC举例）



管理页面是<https://my.vultr.com>，点进服务器你会看到具体信息如IP地址和密码。右上角那五个依次是控制台、关机、重启、重装系统（别重装，要就重新建一个服务器，因为重装会有机器验证的问题，很麻烦）以及删除服务器。

Window如果安装了Git客户端，可以直接命令使用ssh登入

ssh 用户@服务器IP

ssh -p 端口 用户@服务器IP

ssh root@192.168.68.1

或者下载putty客户端

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

作者：小耳朵图图是我

链接：<http://www.jianshu.com/p/6eaa86c1e7b8>

来源：简书

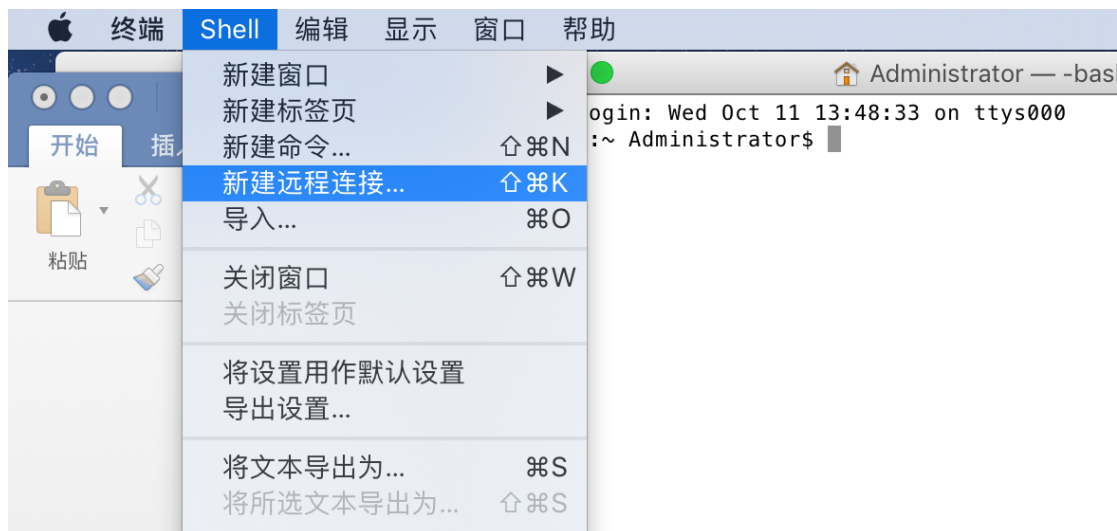
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

建立Mac到服务器的连接，通过macOS的“终端”实现。

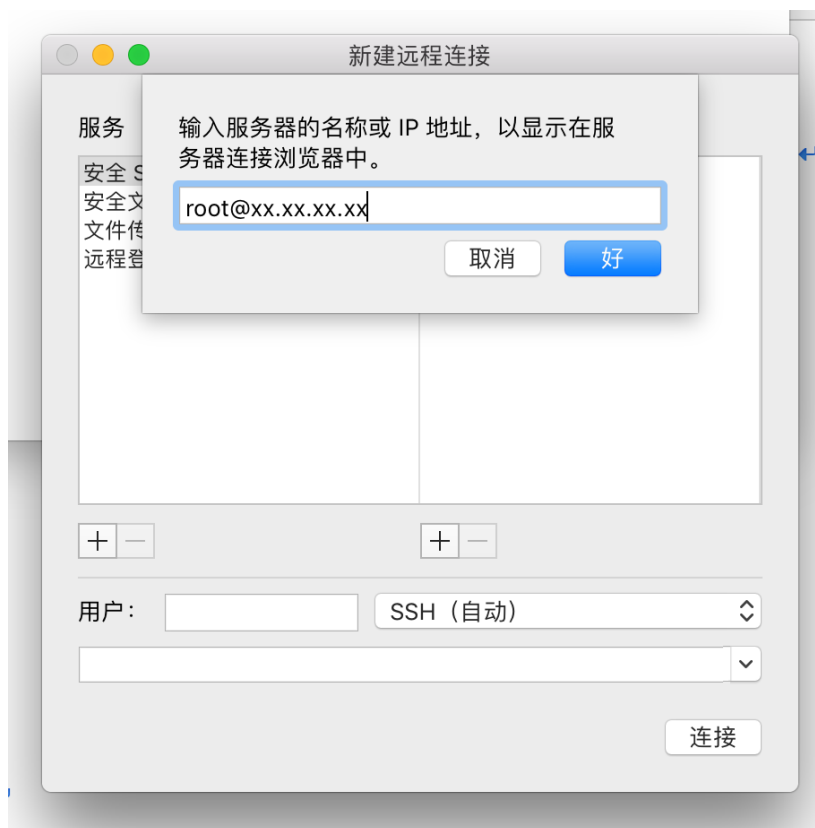


Windows上的ssh工具忘记叫什么了，搜索“windows连接ssh”。操作方法几乎一样的，只是工具不同。

macOS下如图操作：



在 服务 中选择 安全SSH，然后点“+”，按这个格式输IP地址，即IP地址加上root@前缀。



选中新建的服务器后点连接，第一次连接服务器会出现问题问yes or no，键入yes，回车。

```
The authenticity of host '45[REDACTED]' can't be established.  
ECDSA key fingerprint is SHA256:fRTUf[REDACTED]  
Are you sure you want to continue connecting (yes/no)? yes
```

这个问题的意思很简单，就是问你是不是是一名优秀的程序员，我们当然要回答yes了。

```
root@45[REDACTED]'s password: [REDACTED]
```

然后会叫你输入密码，复制粘贴进去就可以。

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
root@vultr:~#
```

完成初始化。

3.3. 在服务器安装SSR服务端

官方教程在[这里](#)，里面包含了不常用的命令，第一次做完以后可以去全面了解一下。这个教程有一个问题，就是`cd ~/shadowsocksr`这个命令用不了，得用我下文的`cd shadowsocksr`

可以忽略有框的内容

要开始输命令了，不要分心，我不会每个返回结果都截图出来，说什么照做就好。

依次键入，输完一句就回车。

注意命令中包含的空格，复制、修改的时候不要漏了，很多地方有空格

3.3.1. 获取Python包

```
apt-get install python-pip
```

返回结果

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
[root@vultr:~#  
[root@vultr:~# apt-get install python-pip  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  python-chardet python-colorama python-distlib python-html5lib  
  python-pkg-resources python-requests python-setuptools python-six  
  python-urllib3  
Suggested packages:  
  python-genshi python-lxml python-distribute python-distribute-doc  
Recommended packages:  
  build-essential python-dev-all python-wheel python-ndg-httpsclient  
  python-openssl python-pyasn1  
The following NEW packages will be installed:  
  python-chardet python-colorama python-distlib python-html5lib python-pip  
  python-pkg-resources python-requests python-setuptools python-six  
  python-urllib3  
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.  
Need to get 1006 kB of archives.  
After this operation, 3938 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

问你问题，作肯定回答就行y 回车继续

然后

```
apt-get install git
```

3.3.2. 获取源代码

```
git clone -b manyuser https://github.com/shadowsocksr-backup/shadowsocksr.git
```

执行完毕后此目录会新建一个shadowsocks目录，其中根目录的是多用户版（即数据库版），子目录中的是单用户版。

根目录即 ./shadowsocksr

子目录即 ./shadowsocksr/shadowsocks

3.3.3. 配置服务端

先指定shadowsockr目录，指定后光标会提示root@vultr:~/shadowsocksr#

```
cd shadowsocksr
```

然后执行以下针对该目录下shadowsocksr的命令

```
bash initcfg.sh
```

用nano命令修改userapiconfig.py

```
nano userapiconfig.py
```

对以下内容进行相应修改：

```
API_INTERFACE = 'mudbjson' #修改接口类型
```

```
SERVER_PUB_ADDR = '127.0.0.1' #修改生成链接对应的公网IP
```

把引号内的值改成上面的值，默认值不是这样，我不记得默认是什么了。

注意不要删了其他内容。

修改完，按Control+X退出编辑，再按Y，回车确认保存。（Mac键盘）

接着，通过使用脚本mujson_mgr.py添加端口及相应的加密、协议、混淆等配置，具体方法通过执行以下命令查看该脚本的说明及提示：

```
python mujson_mgr.py -h
```

3.3.4. 创建用户

先指定shadowsocksr目录，指定后光标会提示root@vultr:~/shadowsocksr#

```
cd shadowsocksr
```

(端口、密码是必要参数，更多参考帮助)

```
python mujson_mgr.py -a -u 用户名（可不填） -p 端口（五位数以内） -k 密码 -m 加密方式 -O 协议插件 -o 混淆协议
```

举例（如果不玩游戏不在乎延迟的话，本人推荐这个配置，对网速影响不大）

```
python mujson_mgr.py -a -u user1 -p 7777 -k abcdefg -m none -O auth_chain_a -o tls1.2_ticket_auth
```

然后就会返回用户参数，可以添加服务器了，马上保存该信息，以后再查查不到那么细的，只能通过修改用户来重设信息。

```
user : user
port : 7777
method : none
passwd : hcw2936700
protocol : auth_chain_a
obfs : tls1.2_ticket_auth
(自动生成) ssr://.....
(二维码要在电脑端导出服务器二维码才能生成)
```

建议在第一次使用前重启一遍服务，后文有介绍。

3.3.5. 修改、删除用户

先指定shadowsocksr目录，指定后光标会提示root@vultr:~/shadowsocksr#

```
cd shadowsocksr
```

修改用户

```
python mujson_mgr.py -e
```

删除用户

```
python mujson_mgr.py -d
```

更多参考帮助python mujson_mgr.py -h

关于加密方式、协议插件、混淆协议选择，具体看[这里](#)。

- 协议推荐：协议用auth_chain_a最佳，此时推荐不使用加密（设置为none），混淆随意
- 加密选择：若协议用auth_chain_a，那加密用none（但不代表密码可以不写或两边不匹配），若协议不是auth_aes128_md5或auth_aes128_sha1，那么不能使用rc4加密（可用rc4-md5）。这时加密可以在rc4-md5、salsa20、chacha20-ietf三个里面选择（rc4-md5可换为aes系列，salsa20可换为chacha20或bf-cfb），如果使用SSR还可特别选择rc4-md5-6。

- 混淆推荐：如果QoS在你的地区明显，混淆建议在http_simple与tls1.2_ticket_auth中选择，具体选择可以通过自己的试验得出。如果选择混淆后反而变慢，那么混淆请选择plain。如果你不在乎QoS，但担心你的个人vps能不能持久使用，那么混淆选择plain或tls1.2_ticket_auth，协议选择auth_chain_a或auth_aes128_md5或auth_aes128_sha1
- 如果你用于玩游戏，或对连接延迟有要求的情况下，建议不要使用tls1.2_ticket_auth混淆，用其它混淆或plain
- 服务端里，http_simple与http_post是相互兼容的，没有使用上的区别
- 如果你在公司，或学校，或某些环境下，发现原版SS协议不可用，建议你启用http_simple、http_post或tls1.2_ticket_auth混淆，同时端口相应使用80或443，通常能解决问题。同时能躲避你所在环境下的网络封锁（如禁止访问网盘禁止上传等等）
- 如果使用tls1.2_ticket_auth混淆或不开启混淆，那么协议最好不要使用origin或verify_sha1
- 如果使用二重代理，一般你只需要考虑越过防火墙的那一段使用混淆或加强协议，除非为了匿名

3.3.6. 服务端运行与停止

先指定shadowsockr目录，指定后光标会提示root@vultr:~/shadowsocksr#

```
cd shadowsocks
```

然后执行以下针对该目录下shadowsocksr的命令

后台运行（无log，ssh窗口关闭后也继续运行）

```
./run.sh
```

后台运行（输出log，ssh窗口关闭后也继续运行）

```
./logrun.sh
```

后台运行时查看运行情况

```
./tail.sh
```

停止运行

```
./stop.sh
```

注：通过脚本运行默认日志会保存在根目录的ssserver.log，可手动查看。

如要重启，先运行./stop.sh再运行./run.sh

很多问题都是重启试试就好。

4. 法律问题

使用代理网络进行大量BT下载可能造成服务器被服务商封停。建议使用SSR代理下载种子后再用非代理网络下载文件。

使用SSR进行经营活动，依照《中华人民共和国刑法》第二百八十五条第三款，可能触犯提供侵入、非法控制计算机信息系统的程序、工具罪。详见[邓杰威提供侵入、非法控制计算机信息系统程序、工具一审刑事判决书](#)。

5. 参考

半官方指南：

[Shadowsocks 自动安装脚本](#)

[ShadowsocksR project 备份组织](#)

路边指南：

[ShadowSocks（影梭）不完全指南](#)

[ShadowsocksR Clients and Server](#)

[搭建shadowsocksR服务器](#)