DigitalOcean+Shadowsocks VPS搭建 | Mac端+iOS端全自动科学上网方案

2017-10-23

一直以来上谷歌搜索东西都异常麻烦，每次搜索都是一件十分心累的事情，Baidu实在搜不到东西，Bing效果也不怎么样。VPN各种不稳定，即使是自己买的VPN上网体验也十分差（可能是我买的是太便宜的VPN），然后无意间看到bestswifter的一篇全自动上网方案的文章，觉得这种全自动的方案真的很赞，于是就动手搭了一个VPS。

这里就将整个过程做个记录，也不完全是重复别人的文章吧，因为在搭建过程中也遇到不少问题，同时记录一下也方便自己后面查阅。

VPS 介绍

什么是VPS，下面是Wikipedia中对VPS的解释：

> **虚拟专用服务器**（英语：Virtual Private Server，缩写为 VPS），是将一台服务器分区成多个虚拟专享服务器的服务。实现VPS的技术分为容器技术和虚拟化技术 。在容器或虚拟机中，每个VPS都可分配独立公网IP地址、独立操作系统、实现不同VPS间磁盘空间、内存、CPU资源、进程和系统配置的隔离，为用户和应用程序模拟出“独占”使用计算资源的体验。VPS可以像独立服务器一样，重装操作系统，安装程序，单独重启服务器。VPS为用户提供了管理配置的自由，可用于企业虚拟化，也可以用于IDC资源租用。IDC资源租用，由VPS提供商提供。不同VPS提供商所使用的硬件VPS软件的差异，及销售策略的不同，VPS的使用体验也有较大差异。尤其是VPS提供商超卖，导致实体服务器超负荷时，VPS性能将受到极大影响。相对来说，容器技术比虚拟机技术硬件使用效率更高，更易于超卖，所以一般来说容器VPS的价格都高于虚拟机VPS的价格。 这些VPS主机以最大化的效率共享硬件、软件许可证以及管理资源。每个VPS主机都可分配独立公网IP地址、独立操作系统、独立超大空间、独立内存、独立CPU资源、独立执行程序和独立系统配置等. VPS主机用户除了可以分配多个虚拟主机及无限企业邮箱外， 更具有独立主机功能, 可自行安装程序，单独重启主机。

所以实质上VPS只是一个服务器，并不是科学上网的代名词，我们只是利用VPS搭建科学上网的服务。我们可以将VPS作为代理服务器（如shadowsocks）来进行科学上网，我们也可以在VPS上搭建VPN服务。

VPN与代理的区别

VPN的全称是virtual private network，中文翻译是虚拟专用网络，它利用开放的公共网络资源建立私有专用的传输通道。VPN建立后，两台计算机好像是通过专用网络直接连接在一起，并处在一个虚拟的局域网中，客户机的所有流量都会通过这个通道进行传输，VPN由于是加密的所以可以达到翻墙的目的。

代理通常有两种方式：HTTP代理和SOCKS代理，而shadowsocks使用的就是SOCKS协议，shadowsocks分为服务端和客户端，客户端程序和服务端的程序建立一个TCP连接，客户端程序将请求封装发送给服务器程序，服务器解析该请求并请求到相应资源再封装后返回给客户端程序，客户端程序解析后得到相应资源，这里服务器代替客户机进行请求，所以称为代理服务器。只有设置代理的程序的请求会通过代理服务器发起，所以是局部的效果，客户端和服务器之间的数据包是加密的，看起来是普通的TCP包，所以GFW放行，这是shadowsocks翻墙的原理。

服务器端

购买服务器

首先需要一台国外的服务器，比较主流的服务器有三个：

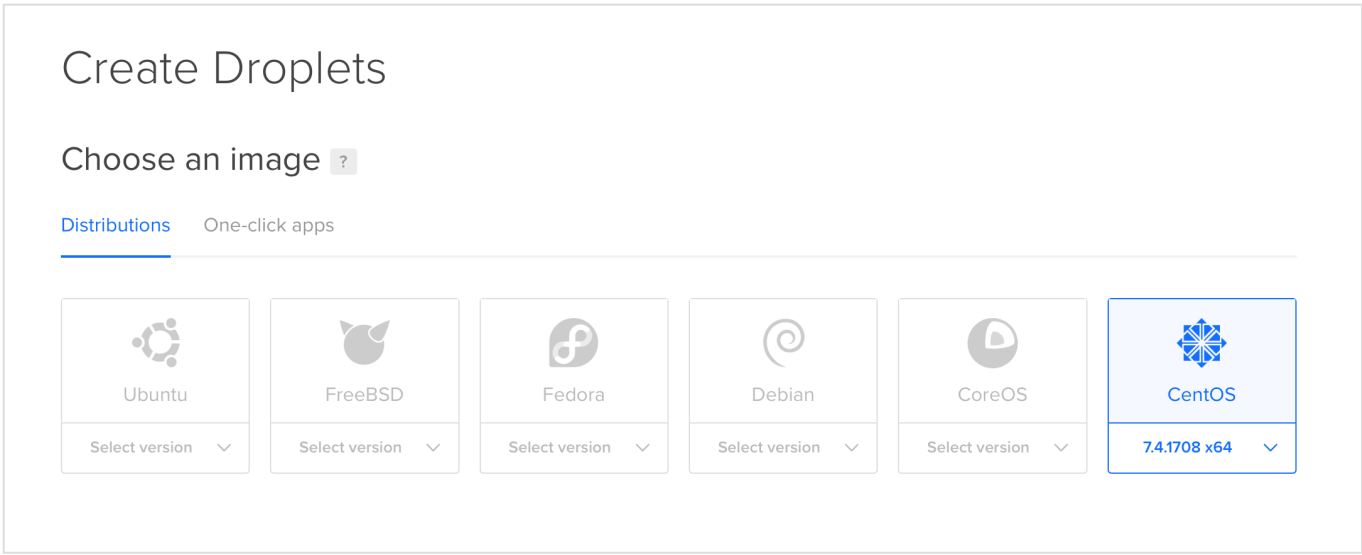
- DigitalOcean：每月5美元，1000GB流量，512MB的CPU，20G的SSD硬盘。
- 搬瓦工：每月2.99美元，500GB流量，512MB的内存，10G的SSD硬盘。
- Vultr：每月2.5美元，500GB流量，512MB的内存，20G的SSD硬盘。

三家的机器配置都差不多，Vultr的价格最低但据说网速不怎么样，我选择的是DigitalOcean，因为有GitHub学生优惠的50美元。如果点击上面DigitalOcean链接注册会有我送的$10，DigitalOcean虽然价格最贵但口碑很好，使用起来体验极好，如丝般柔滑，而且它的1000G流量也没用严格的控制，DigitalOcean没有提供查询流量的入口，据说即使超了流量也不会扣费，只要别超太多，而且如果个人当VPS使用的话不可能超流量的。

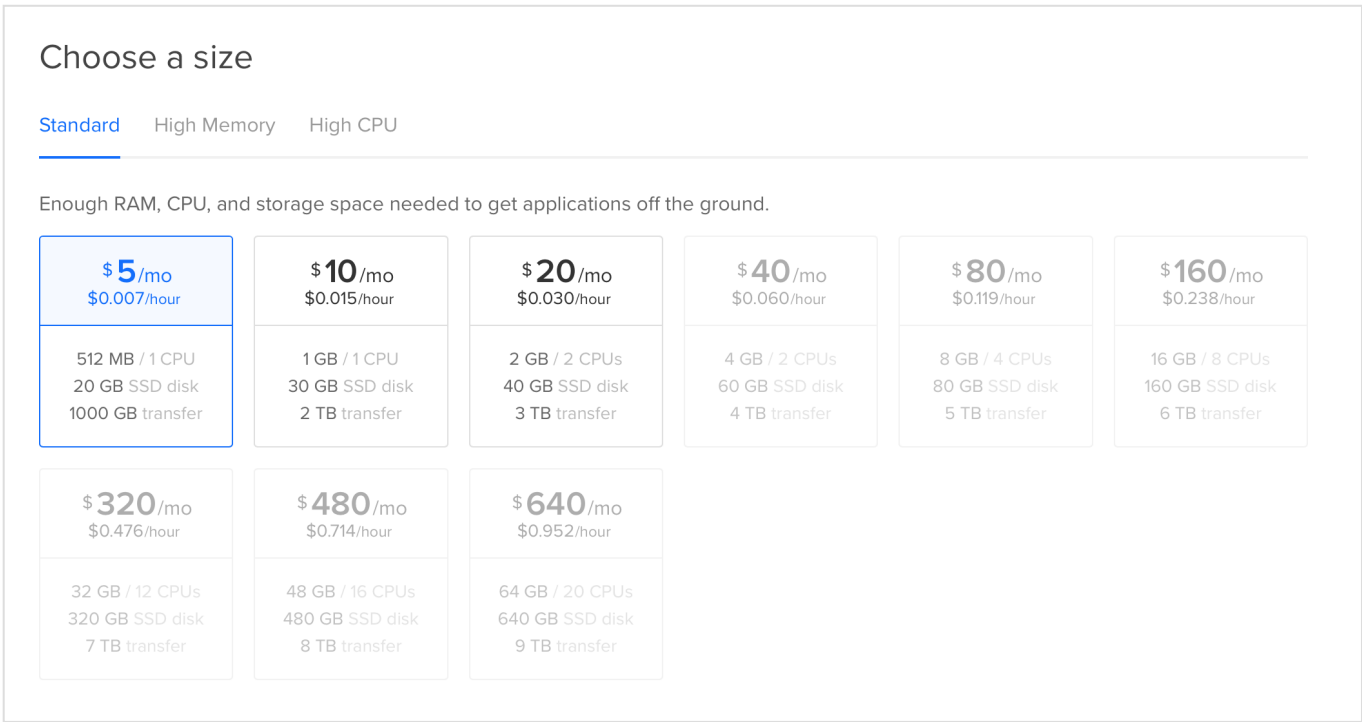
这里记录的是在DigitalOcean上搭建服务器的过程，如果你购买的是其他平台的服务器，可以移步其他教程，其实过程也大同小异，只是在网页上的一些点击操作。

在注册完成DigitalOcean后需要充值5美元激活账号，如果没有国外的信用卡，可以选择使用PayPal，注册PayPal后绑定银联的银行卡就行，付费时直接从银行卡中扣费。

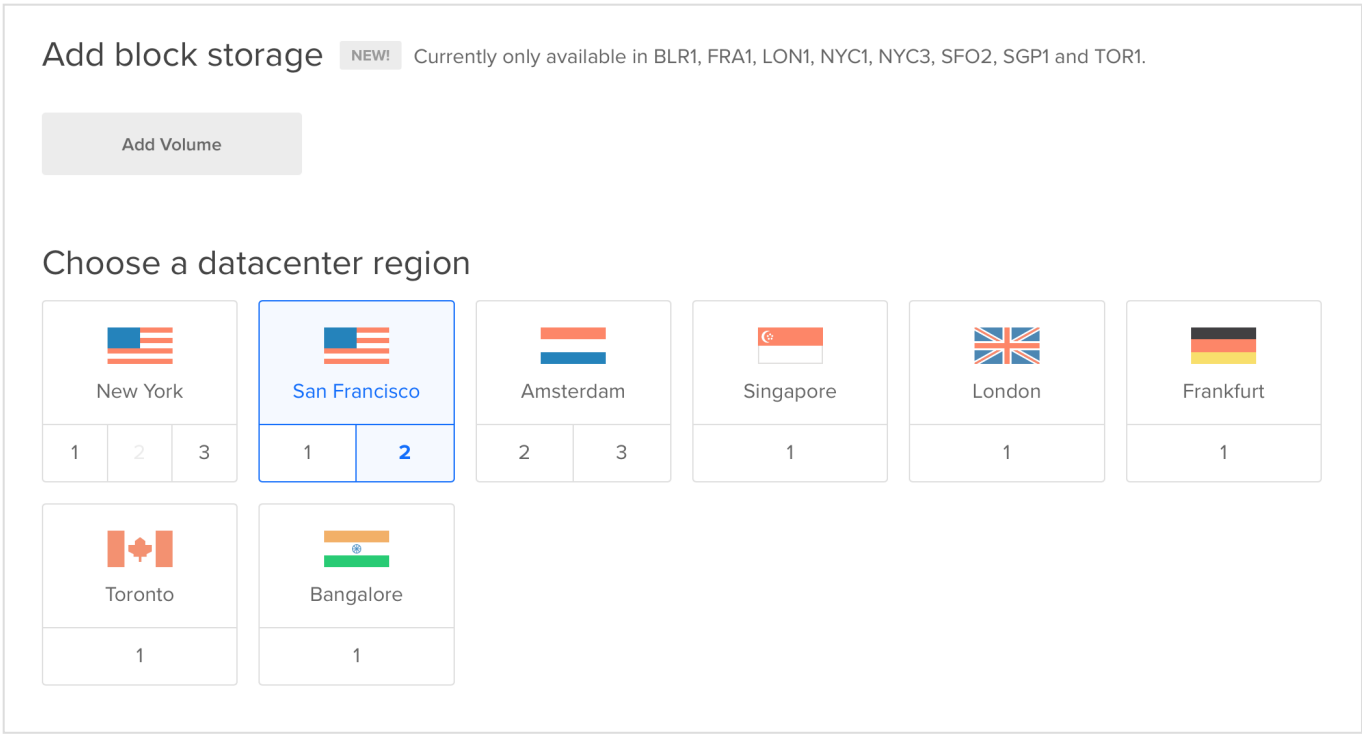
在DigitalOcean中服务器被称为Droplets，首先选择Create Droplets



这里我们选择Centos，版本就选择最新的就可以了，Centos应该可以算是服务器系统的首选了，Ubuntu比较适合个人使用，在稳定性上我感觉Ubuntu不如Centos。

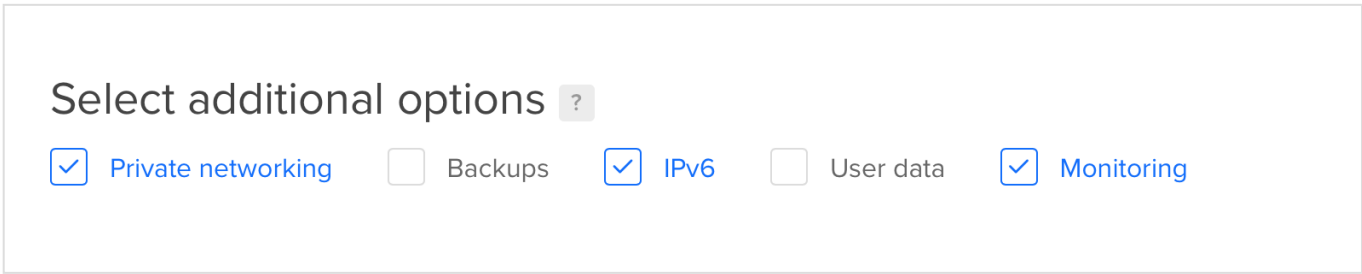


套餐直接选择最低配就可以了，完全够用了。

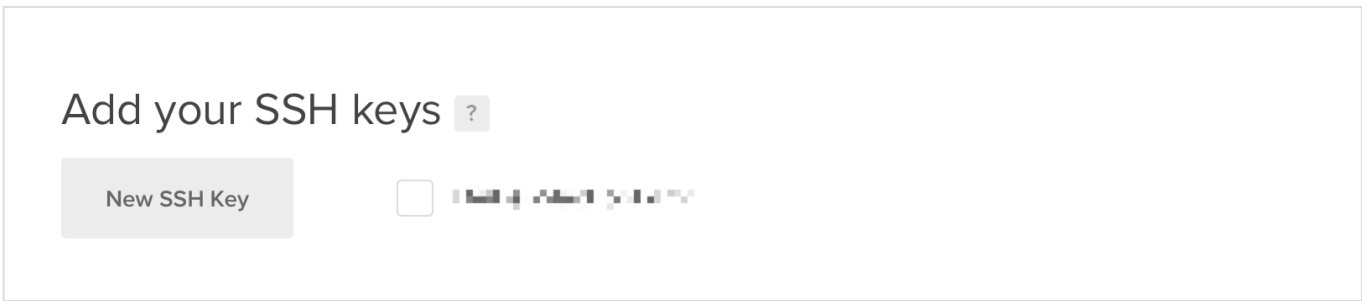


block storage对于我们来说不需要，这是要收费的。

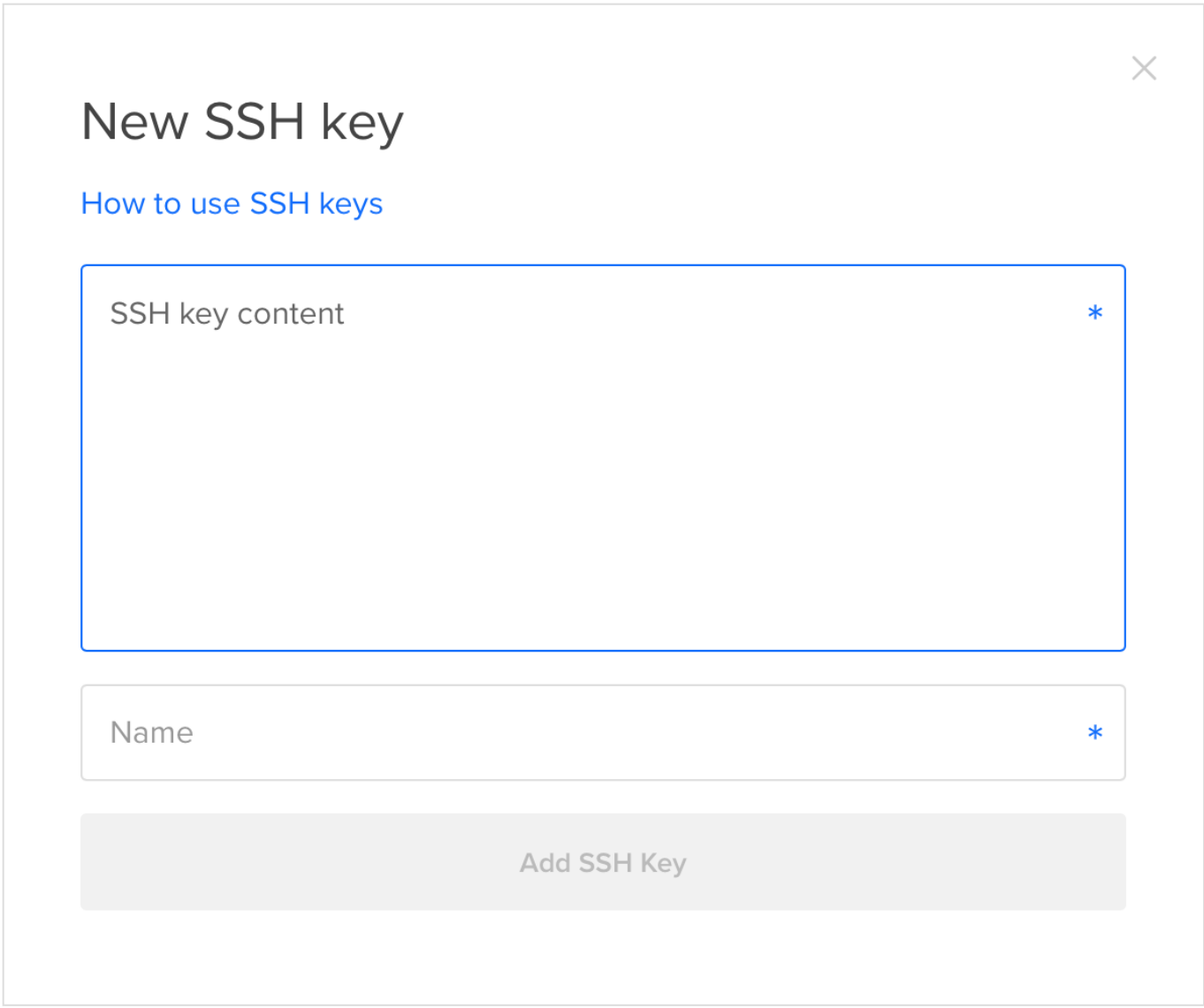
关于节点的选择，San Francisco到大陆的网速很快，选择该节点最优，也有人说New York的网速不错，我两个都试过，测速的结果是San Francisco是New York的好几倍。



这个配置其实无所谓，记住不要勾选Backups选项，这个是要额外收费的



关于SSH Key不是强制的，但是推荐配置一下，不然每次远程登录时都需要输入密码（这个密码会在服务器搭建完成后收到的邮件中，如果配置的SSH key就不会收到密码），我们点击New SSH key，会看到如下界面：



我们需要将我们的ssh公钥复制到上面输入框中，在下面的输入框中为其起一个名字，名字怎么起随便你

下面看一下如何获得自己的公钥，首先打开终端，进入.ssh目录：

```
1  $ cd ~/.ssh
```

查看当前目录下的文件：

```
1  $ ls
```

如果你曾经生成过SSH Key，该目录下应该至少会有两个文件：id_rsa和id_rsa.pub，当然文件名可能不是id_rsa，但两个文件一定是成对存在的，两个文件名是相同的，使用cat命令查看xxx.pub文件的内容，复制该内容到上面的输入框即可

```
1  $ cat id_rsa.pub
```

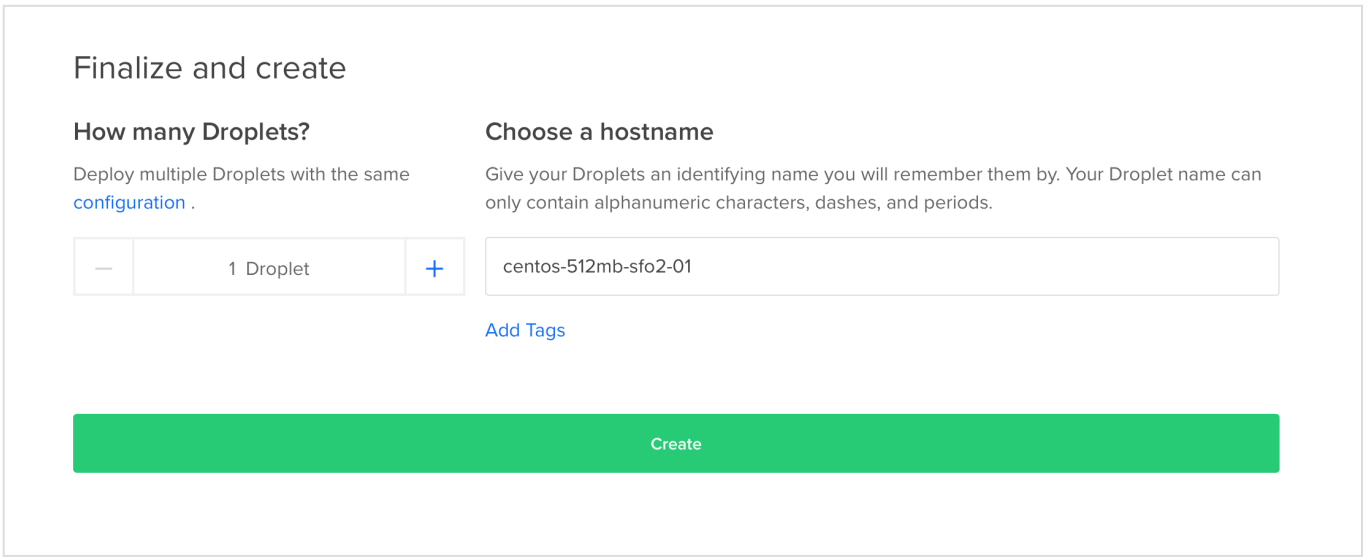
如果没有SSH key，可以使用ssh-keygen来生成一个：

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/schacon/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/schacon/.ssh/id_rsa.
Your public key has been saved in /Users/schacon/.ssh/id_rsa.pub.
The key fingerprint is:
43:c5:5b:5f:b1:f1:50:43:ad:20:a6:92:6a:1f:9a:3a schacon@agadorlaptop.local
```

你可以直接按三次回车来快速完成该过程，这也你的SSH Key的文件名是默认的id_rsa，然后在使用公钥时也没有密码

具体可以参考Github关于SSH key的教程

生成SSH key后再按照上面的步骤复制自己的公钥就可以添加SSH key啦



最后一步点击create就大工告成啦，当然如果想修改一下你服务器的名称也可以，推荐使用自己姓名加系统名。

最后等待服务器初始化完成，就可以开始配置了。

配置VPS

远程登录服务器

打开Mac的终端输入

```
$ ssh root@[your vps ip address]
```

如果之前配置了SSH key，这里就不用输入密码，直接登录成功，如果没有配置SSH key，这里就会提示输入密码，密码在DigitalOcean发给你的邮件中。

配置防火墙

其实在DigitalOcean的控制面板中提供了防火墙的配置，可以直接在线上配置，不过我通过线上没有配置成功，可能是因为生效会有延迟，所以这里直接自己安装防火墙。

1. 安装firewalld

```
$ yum install firewalld firewall-config
```

1. 启动firewalld

```
$ systemctl start firewalld
```

1. 修改SSH端口（可选）

```
$ vi /usr/lib/firewalld/services/ssh.xml
```

出现以下窗口：

```
<?xml version="1.0" encoding="utf-8"?>
  <service>
    <short>SSH</short>
    <description>Secure Shell (SSH) is a protocol for logging into and executing
    <port protocol="tcp" port="22"/>
  </service>
```

如果想要修改，将 `port="22"` 改为 `port="xxx"` （建议不要修改）

修改后重载firewalld

```
$ firewall-cmd --permanent --add-service=ssh
$ firewall-cmd --reload
```

执行上述命令如果显示FirewallD is not running，使用 `systemctl start firewalld` 启动防火墙。

注意如果在搭建完成后无法科学上网，很有可能是防火墙导致的，快速定位问题可以使用以下命令关闭防火墙测试：

```
$ systemctl stop firewalld
```

如果关闭防火墙后可以正常科学上网，说明是端口没有正确添加到防火墙中。

安装shadowsocks

1. 安装pip

```
$ yum install m2crypto python-setuptools
$ easy_install pip
```

1. 使用pip安装shadowsocks

```
$ pip install shadowsocks
```

1. 启动shadowsocks

直接在前台启动shadowsocks

```
$ ssserver -p 443 -k password -m aes-256-cfb
```

在后台启动shadowsocks

```
$ sudo ssserver -p 443 -k password -m aes-256-cfb --user nobody -d start
```

停止后台的shadowsocks

```
$ sudo ssserver -d stop
```

推荐在通过配置文件启动ss，以下命令创建了shadowsocks.json文件，并用vi打开进行编辑

```
$ vi /etc/shadowsocks.json
```

如果对vim不熟，记住几个规则：在任意模式下 esc 返回 normal 模式，在 normal 模式下按 i 进入 insert 模式，在 normal 模式下输入 :wq 保存并退出，在 insert 模式下可以通过上下左右移动光标到你要编辑的地方，编辑完

成后返回 normal 模式，并保存退出。

将该文件配置成下面这个样子，注意其中需要修改的地方，第一项 server 修改成你 vps 的IP地址，第二项 server_port 可以不修改，但最好修改一下，修改后要记住这个端口，我们后面配置防火墙的时候要用到，第五项 password 修改成你的密码，其他东西不用修改。

```
{
    "server":"my_server_ip",
    "server_port":8388,
    "local_address": "127.0.0.1",
    "local_port":1080,
    "password":"mypassword",
    "timeout":300,
    "method":"aes-256-cfb",
    "fast_open": false
}
```

详情可以参考shadowsocks官方文档

然后通过配置文件启动ss

```
$ ssserver -c /etc/shadowsocks.json
```

最好在后台启动ss，这样我们可以继续进行后续配置，下面是后台启动和停止的命令

```
$ ssserver -c /etc/shadowsocks.json -d start
$ ssserver -c /etc/shadowsocks.json -d stop
```

1. 防火墙添加shadowsocks端口

```
$ firewall-cmd --add-port=8388/tcp --permanent
```

这里要注意的是端口号8388要修改成你上面配置文件中的端口号

让shadowsocks开机自启

1. 安装supervisor

```
$ easy_install supervisor
```

1. 创建配置文件

```
$ echo_supervisord_conf > /etc/supervisord.conf
```

1. 修改配置文件

```
$ vi /etc/supervisord.conf
```

在文件末尾加上：

```
[program:ssserver]
command = ssserver -c /etc/shadowsocks.json
autostart=true
autorestart=true
```

1. 设置supervisor开机自启

```
$ vi /etc/rc.local
```

在文件末尾另起一行，添加： `supervisord`

1. CentOS7 还需要为 rc.local 添加执行权限

```
$ chmod 777 /etc/rc.local
```

这样ss开启自动后台运行就配置好了，下面重启服务器就好了，可以在digitalocean控制面板中重新启动服务器。

客户端

Mac端

shadowsocks客户端

ShadowsocksX

Shadowsocks 作者被请喝茶了，所有 shadowsocks 的 GitHub 仓库都只剩下一个 README.md：

```
Removed according to regulations.
```

不过Mac端的 ShadowsocksX 可以在这里下载。

它的配置非常简单，打开服务器设定，点击加号，输入你 VPS 的 IP 地址还有你前面 shadowsocks 配置文件中的 server_port 和 password 字段中的值，加密方式不用修改，备注也可以不用管，点击确定服务器就配置好了。

点击打开 Shadowsocks，通过浏览器测试是否能上谷歌。

如果打不开谷歌，检查你客户端服务器地址、端口号、密码是否正确配置，如果都没有问题，那么回到之前防火墙配置的章节，关闭防火墙后进行测试。

ShadowsocksX-NG

Next Generation of ShadowsocksX

该版本 ShadowsocksX 相比老版本添加了 http 代理，同时监听4个端口，分别是socks5代理端口、PAC模式代理端口、http代理端口、Kcptun加速的代理端口，并且都可以直接在客户端修改端口号。

服务器的配置参考老版本的步骤，需要注意的是加密方式，新版本的SS默认加密方式是aes-128-gcm，如果你之前服务器的加密方式没有修改，那么默认就是aes-256-cfb，这里注意要改成对应的加密方式。

配置完成后选择全局模式并打开 shadowsocks，同样通过浏览器测试是否成功。

你可以在这里下载NG版 ShadowsocksX

GoAgentX

GoAgentX 是 Mac 下使用代理服务的图形界面控制软件，它支持的代理服务类型非常多，有goagent、west-chamber-season-3、SSH、stunnel 及 shadowsocks 等，我们要使用的就是它的 shadowsocks 服务。

配置方式和 ShadowsocksX 很类似，点击显示主窗口，点击加号，选择 Shadowsocks 服务，配置服务器地址、端口号、加密方式、密码、还有本地端口，完成后选择全局模式，并打开该服务器连接，通过浏览器进行测试。

GoAgentX 的官方 GitHub 仓库也被移除了，你可以在这里下载。

Chrome科学上网

其实我们科学上网最主要的一个途径就是通过浏览器，我们这里选择配置Chrome，因为它有一个非常好用的插件 SwitchyOmega。

SwitchyOmega

SwitchyOmega 是一个代理插件，他可以将 http 流量转化为 socks 流量，通过我们的 shadowsocks 客户端的 socks 代理，只需要上面三个客户端的任意一个即可。

首先要调整上面三个客户端的模式：老版本的 ShadowsocksX 可以直接点击关闭，NG 版本的 ShadowsocksX 选择手动模式，GoAgentX 选择独立模式，这个时候三个客户端只是监听自己对应的端口，没有监听全局的流量，我们通过 SwitchyOmega 将流量导到他们对应的端口上。

在我们安装完 SwitchyOmega 后，会默认有一个 auto switch 模式，该模式会根据条件自动选择使用代理还是直接连接。

我们创建一个代理的模式，点击新建情景模式，输入情景模式名称：ss（名称随便），选择类型为代理服务器，代理协议选择SOCKS5，代理服务器为：127.0.0.1

关于端口对应三个客户端不一样

老版本ShadowsocksX：1080

NG版ShadowsocksX：点击偏好设置 -> 高级，输入本地Socks5监听端口中的值

GoAgentX：端口号为你设置服务器时设置的本地端口，可以在其主窗口中查看

完成后点击应用选项。

回到 auto switch 模式中将默认情景模式改为ss，应用选项后打开谷歌测试是否生效。

这时有个偷懒的做法就是将默认情景改为直接连接，然后当你在上网时遇到打不开的网页时，点击右上角 SwitchyOmega 图标，可以直接添加规则，使用代理模式打开。

配置PAC

如果想要有更好的体验，可以选择使用 PAC，PAC 分为黑名单和白名单，黑名单方式就是在黑名单中的网站都通过代理，其他都直接连接，白名单的方式是白名单中的网站都直接连接，其他网站都走代理，如果黑名单更新不够及时，那么会经常遇到网站打不开的情况，而白名单就不会出现，白名单的缺点是更浪费流量，更多的流量会通过 VPS，不过一般 VPS 的流量都是用不完的，所以这些流量不算什么。

有人在 Github 维护了一个白名单的仓库，这里我们这个白名单 PAC 进行配置。

首先打开 SwitchyOmega 控制面板，新建情景模式，输入名称：white_list，类型选择 PAC 情景模式，在 PAC 网址中输入

```
https://raw.githubusercontent.com/breakwa11/gfw_whitelist/master/whitelist.pac
```

然后点击立即更新，更新完成后点击网址右边的差号，将网址清除。

然后修改 PAC 脚本的第一行：

```
1   var wall_proxy = "SOCKS5 127.0.0.1:1080; SOCKS 127.0.0.1:1080;";
```

将两个1080改成对应客户端监听的端口，比如我的端口是1086，那么修改后就是：

```
1   var wall_proxy = "SOCKS5 127.0.0.1:1086; SOCKS 127.0.0.1:1086;";
```

然后点击 auto switch，将默认模式改为 white_list，然后测试是否可以科学上网。

这个时候如果再有无法打开的网页再单独去添加规则要么通过代理要么直接连接，按说是不会有打不开的网页了。

特定应用科学上网

如果有某个应用（比如Telegram）必须要通过代理，那么可以使用 Proxifier 将其流量导向 shadowsocks 客户端。

Proxifier

1. 进入该软件的主界面后，点击Proxies -> Add，Address 输入：127.0.0.1，Port 输入对应 shadowsocks 客户端监听的端口，协议选择 SOCKS Version 5，然后点击 OK。
2. 点击Rules -> Add，点击 Application 下面的加号，选择特定的App，Action 选择第一步中创建的代理，点击 OK。

到此你选择的应用就可以科学上网了。

终端科学上网

终端科学上网是一件比较麻烦的事情，终端需要使用 http 代理，但是对于 NG 版本的 ShadowsocksX 就太方便了，在该软件的下拉框中有一个选项叫 Copy HTTP Proxy Shell Export Line，然后打开终端粘贴后运行，终端就可以科学上网了。

这里我写了一个脚本，可以一个命令切换终端的代理和直连模式：

```
1   function cw() {
```

```
    if [ "$http_proxy" = "" ]
    then
            export http_proxy=http://localhost:1087
            export https_proxy=http://localhost:1087
    export ftp_proxy=http://localhost:1087
    else
            export http_proxy=""
            export https_proxy=""
            export ftp_proxy=""
    fi
    curl ip.cn &
}
```

将这个复制到你shell的配置文件中，比如.zshrc中

然后执行： `source .zshrc` ，或者重新打开一个终端窗口，然后执行cw（cross wall的缩写），你就能看到你的 IP 地址改变了

注意其中1087的端口号要改成自己 NG 版 ShadowsocksX 的 http 端口号。

iOS端

A.BIG.T

关于 iOS 端，我用的软件叫做 A.BIG.T，这个软件十分强大，其中已经帮你配置好了规则，你只要将自己的服务器 IP 、端口号、密码、加密方式配置好后，国内的流量会走直连的方式，国外的流量会走代理的方式。

不过这是一个收费软件，在我使用之后觉得及时收费也是值得的因为实在太强大了（不过我是限免的时候免费下载的😄）

Surge

这是一个超级强大的软件，不仅仅能用来科学上网，而且其主要功能不是用于科学上网，而是给开发者使用的开发工具，至于其强大之处我也没用过，也不知道，大家可以自行谷歌。

Waterdrop

这是我在苹果应用商店里发现的，一个很简单的 shadowsocks 客户端，没有别的多的功能。

可惜的是这三个应用都需要使用国外的账号才能下载，国内 App Store 已经下架了。

参考链接

全自动科学上网方案分享

Digital Ocean上搭建VPS小记

Virtual private network

关于shadowsocks的一些思考

VPN和代理之间的区别是什么?

vpn原理详解

‹ Masonry源码分析

浅谈我对ES2017中异步函数的理解 ›