即在VPS上设置Shadowsocks。

**搭建Shadowsocks服务**

**安装组件：**

yum install m2crypto python-setuptools

easy\_install pip

(上述pip无法安装的情况下：

首先安装epel扩展源：

yum -y install epel-release

更新完成之后，就可安装pip：

yum -y install python-pip

安装完成之后清除cache：

yum clean all

这是在root用户时使用的命令，当前用户如果不具有root权限，加上sudo。)

pip install shadowsocks

（之后会显示shadowsocks安装完成，感觉pip安装后可以装很多库了，但那个scrapy虽然安装进行了，但出错，可能需要配合其他库的安装吧）

**安装完成后配置服务器参数：**

/etc目录下建立一个shadowsocks.json的文件

vi /etc/shadowsocks.json

随后写入如下配置：

{  
 "server":"0.0.0.0",   
 "server\_port":443,   
 "local\_address": "127.0.0.1",  
 "local\_port":1080,  
 "password":"LLK1026",  
 "timeout":300,  
 "method":"aes-256-cfb",  
 "fast\_open": false  
}

server是vps的地址

server\_port是vps开放的端口，后续要通过设置防火墙开放端口

Local\_address是本机地址

local\_port是本机使用的端口

password是对于local\_address来说的登陆ss的密码，并不是vps的登陆密码

method有几种，反正选了这一种

多端口的如下配置：（意思就是多人使用这个VPN吧）

{  
 "server":"0.0.0.0",  
 "local\_address": "127.0.0.1",  
 "local\_port":1080,  
 "port\_password": {  
 "443": "443",  
 "8888": "8888"  
 },  
 "timeout":300,  
 "method":"aes-256-cfb",  
 "fast\_open": false  
}

P.S.此处，若看到下面启动ss的方式的时候，也可以通过不加载配置文件的方式启动ss，故该配置文件也不是一定需要的。

**VPS上开放需要的端口**

许多文章上都是yum install firewalld

但没法安装，因为安装时压根找不到firewalld，也不知道是什么原因，

最后自己想办法直接修改iptables：

vi /etc/sysconfig/iptables

编辑iptables的时候增加一行开放443端口：

-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT

之后重新启动防火墙：

service iptables restart

但在vultr新加坡服务器上CentOS 7 x64系统，却能够yum install firewalld，奇怪了！

（因为之前都是CentOS6）

firewalld的基本使用

启动： systemctl start firewalld

查看状态： systemctl status firewalld

停止： systemctl disable firewalld

禁用： systemctl stop firewalld

那firewalld怎么开启一个端口呢

添加

firewall-cmd --zone=public --add-port=443/tcp --permanent    （--permanent永久生效，没有此参数重启后失效）

(

firewall-cmd --zone=public --add-port=439/tcp --permanent

firewall-cmd --zone=public --add-port=80/tcp --permanent

)

重新载入

firewall-cmd --reload

查看

firewall-cmd --zone=public --query-port=443/tcp

删除

firewall-cmd --zone= public --remove-port=443/tcp --permanent

但还会出现firewalld not running的情况

以及

[root@vultr etc]# systemctl start firewalld

Failed to start firewalld.service: Unit is masked.

这是因为

firewalld服务被锁定，不能添加对应端口

执行命令，即可实现取消服务的锁定

# systemctl unmask firewalld

下次需要锁定该服务时执行

# systemctl mask firewalld

**在centos 7下启用iptables：**

systemctl stop firewalld.service

systemctl disable firewalld.service

yum install iptables-services -y

systemctl enable iptables

systemctl start iptables

//编辑ipatbles的规则

vi /etc/sysconfig/iptables

//重启iptables服务

service iptables restart

**启动shadowsocks**

网上一些文章上写的是ssserver -c /etc/shadowsocks.json来启动shadowsocks,

但不能成功，试了下直接打ssserver命令，出现了其的相应用法，

[root@Giantslayer ~]# ssserver

ERROR: config not specified

usage: ssserver [OPTION]...

A fast tunnel proxy that helps you bypass firewalls.

You can supply configurations via either config file or command line arguments.

Proxy options:

-c CONFIG path to config file

-s SERVER\_ADDR server address, default: 0.0.0.0

-p SERVER\_PORT server port, default: 8388

-k PASSWORD password

-m METHOD encryption method, default: aes-256-cfb

-t TIMEOUT timeout in seconds, default: 300

--fast-open use TCP\_FASTOPEN, requires Linux 3.7+

--workers WORKERS number of workers, available on Unix/Linux

--forbidden-ip IPLIST comma seperated IP list forbidden to connect

--manager-address ADDR optional server manager UDP address, see wiki

General options:

-h, --help show this help message and exit

-d start/stop/restart daemon mode

--pid-file PID\_FILE pid file for daemon mode

--log-file LOG\_FILE log file for daemon mode

--user USER username to run as

-v, -vv verbose mode

-q, -qq quiet mode, only show warnings/errors

--version show version information

所以这么写：

通过加载配置文件的方式启动shadowsocks：

ssserver -c /etc/shadowsocks.json -d start

[root@Giantslayer ~]# ssserver -c /etc/shadowsocks.json -d start

INFO: loading config from /etc/shadowsocks.json

2017-07-11 16:16:35 INFO loading libcrypto from libcrypto.so.10

started

shadowsocks启动了，原来之前网上那些少了-d start|stop|restart啊

不通过加载配置文件的方式启动shadowsocks：

ssserver -s xx.xx.xx.xx -p server\_port -k ss-password -m aes-256-cfb -d start

测试：

ssserver -s 144.202.63.246 -p 438 -k kkkhhh -m aes-256-cfb -d restart

ssserver -s 149.28.120.47 -p 443 -k HLHT123 -m aes-256-cfb -d restart

**PC连接**

在本地电脑中下载一个shadowsocks的客户端，

启动该目录下的shadowsocks客户端程序，

服务器ip就是shadowsocks中设置的server，

服务器端口是server\_port，

密码是password，

加密就是method。

填写好后，就多了一个服务器。

随后，启用系统代理。

并选择 PAC模式，或是 全局模式。

PAC就是在本地ss客户端目录里有一个pac.txt文件，里面列明了大量被墙的网站，该pac也可以选择更新，或直接使用网上的pac文件。

至此，google.com可以上了。因为VPS设在日本，所以打开的是www.google.co.jp。

使用电信天翼的卡连接起来非常的慢，而小区的长城宽带就会快很多，

后者youtube的视频可以看，但还有些慢，需要后续再研究一些加速的问题了。

P.S.之前都是用ssserver -c /etc/shadowsocks.json -d start来启动shadowsocks的，密码用的是helloworld，其他都未更改，短时间上了半天，就马上出现服务器积极拒绝的情况，网页上显示的是：

500 Internal Privoxy Error

Privoxy encountered an error while processing your request:

**Could not load template file no-server-data or one of its included components.**

Please contact your proxy administrator.

If you are the proxy administrator, please put the required file(s)in the *(confdir)*/templates directory. The location of the *(confdir)*directory is specified in the main Privoxy config file. (It's typically the Privoxy install directory).

后来隔了两天，使用ssserver -s serve\_ip -p 443 -k qazwsxedcrfvtgb -m aes-256-cfb -d start就又能上了，

ssserver -s 45.76.199.207 -p 443 -k LLK1026 -m aes-256-cfb -d start

为什么呢？难道之前密码设置的太简单导致服务器直接拒绝，还是使用-c加载config的文件出错？奇怪了。

自己来解答，因为自己粗心，把设置文件shadowsocks.json中的server\_port错写成了server-port，后来设置多端口的时候，还将port\_password错写成了password！！！

网上看到的帖子，一键安装锐速破解版：

wget -N --no-check-certificate <https://github.com/91yun/serverspeeder/raw/master/serverspeeder.sh> && bash serverspeeder.sh

wget -N --no-check-certificate <https://raw.githubusercontent.com/91yun/serverspeeder/master/serverspeeder-all.sh> && bash serverspeeder-all.sh

第二个可以顺利安装，没觉得速度有提升啊，但也可能真的有提升只是自己没注意到。

手机端：

IOS使用的是wingy

Android使用的是shadowsocks

试了下，两者都可以正常使用。

**#更新yum库**

yum update -y

**CentOS7.4搭建shadowsocks，以及配置BBR**

#更新yum库

yum update -y

#安装 pip

curl "<https://bootstrap.pypa.io/get-pip.py>" -o "get-pip.py"

python get-pip.py

#安装配置 shadowsocks

pip install --upgrade pip

pip install shadowsocks

#安装vim

yum install vim -y

vim /etc/shadowsocks.json

{

"server": "0.0.0.0",

"local\_address": "127.0.0.1",

"local\_port": 1080,

"port\_password":

{ "8082": "HHH123",

"8083": "LLL456" },

"timeout": 600,

"method": "aes-256-cfb"

}

#配置自启动

vim /etc/systemd/system/shadowsocks.service

[Unit]

Description=Shadowsocks

[Service]

TimeoutStartSec=0

ExecStart=/usr/bin/ssserver -c /etc/shadowsocks.json

[Install]

WantedBy=multi-user.target

#执行以下命令启动 shadowsocks 服务

systemctl enable shadowsocks

systemctl start shadowsocks

#检查是否启动成功

systemctl status shadowsocks -l

#确认服务启动成功后，配置防火墙规则

#开放你配置的端口，不然客户端是无法连接的

firewall-cmd --zone=public --add-port=8082/tcp --permanent

firewall-cmd --zone=public --add-port=8083/tcp --permanent

firewall-cmd --reload

#配置BBR加速

rpm --import <https://www.elrepo.org/RPM-GPG-KEY-elrepo.org>

rpm -Uvh <http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm>

yum --enablerepo=elrepo-kernel install kernel-ml -y

#更新grub文件并重启系统

egrep ^menuentry /etc/grub2.cfg | cut -f 2 -d \'

grub2-set-default 0

reboot

#重启完成后查看内核是否已更换为4.14版本

#实际操作下来没有更换为4.14，但也可行

uname -r

#开启bbr

vim /etc/sysctl.conf

net.core.default\_qdisc = fq

net.ipv4.tcp\_congestion\_control = bbr

#加载系统参数

sysctl -p

#如果输出了我们添加的那两行配置代表正常

#确定bbr已经成功开启

[root@vultr ~]# sysctl net.ipv4.tcp\_available\_congestion\_control net.ipv4.tcp\_available\_congestion\_control = bbr cubic reno

[root@vultr ~]# lsmod | grep bbr

tcp\_bbr 20480 2

[root@vultr ~]#

#输出内容如上，则表示bbr已经成功开启

**鹤给的Utunbu的教程：**

使用方法：

使用root用户登录，运行以下命令：

wget --no-check-certificate <https://raw.githubusercontent.com/teddysun/shadowsocks_install/master/shadowsocks-libev-debian.sh>

chmod +x shadowsocks-libev-debian.sh

./shadowsocks-libev-debian.sh 2>&1 | tee shadowsocks-libev-debian.log

安装完成后，脚本提示如下：

Congratulations, shadowsocks-libev install completed!

Your Server IP:your\_server\_ip

Your Server Port:your\_server\_port

Your Password:your\_password -- Test=123

Your Local IP:127.0.0.1

Your Local Port:1080 --8388

Your Encryption Method:aes-256-cfb --rc4-md5

Welcome to visit:http://teddysun.com/358.html

Enjoy it!

卸载方法：

使用 root 用户登录，运行以下命令：

./shadowsocks-libev-debian.sh uninstall

其他事项：

本脚本安装完成后，已将 shadowsocks-libev 加入开机自启动。

使用命令：

启动：/etc/init.d/shadowsocks start

停止：/etc/init.d/shadowsocks stop

重启：/etc/init.d/shadowsocks restart

状态：/etc/init.d/shadowsocks status

更新（2015年08月01日）：

1、新增自定义服务器端口功能（如不设定，默认端口为 8989）；

更新（2015年04月30日）：

1、本脚本会始终安装最新版的 Shadowsocks；

2、修改配置文件 /etc/shadowsocks-libev/config.json 同时启用 IPv4 与 IPv6 支持：

{

"server":["[::0]","0.0.0.0"],

"server\_port":your\_server\_port,

"local\_address":"127.0.0.1",

"local\_port":1080,

"password":"your\_password",

"timeout":600,

"method":"aes-256-cfb"

}

3、Shadowsocks libev 版不能通过修改配置文件来多端口（只能开启多进程），如果你需要多端口请安装 Python 或 Go 版；

更新（2015年03月09日）：

1、新增启动脚本/etc/init.d/shadowsocks，使用更简单。

2、优化安装过程。

更新（2014年10月31日）：

1、修复写入自启动命令到 /etc/rc.local 中的错误。

特别说明：

已安装旧版本的 shadowsocks 需要升级的话，需下载本脚本的最新版，运行卸载命令

./shadowsocks-libev-debian.sh uninstall

然后，再次执行本脚本即可安装最新版。

参考链接：

<https://github.com/madeye/shadowsocks-libev>

JP--

Congratulations, Shadowsocks-libev server install completed!

Your Server IP : 207.148.90.125

Your Server Port : 17923

Your Password : sshuuu1980

Your Encryption Method: aes-256-cfb