

Ubuntu

Ubuntu

- [Ubuntu](#) ▢▢
- [nginx](#) ▢▢
 - [Nginx Conf](#) ▢▢▢▢
- [▢▢▢▢](#)
 - [MTProxy GO](#)
 - [v2ray](#)
- [ServerStats](#)
- [acme.sh SSL](#) ▢▢

Ubuntu 18.04

Ubuntu 18.04

```
#18.04
sudo apt-get update

#18.04.1
sudo apt-get upgrade
```

18.04 Root 18.04

```
sudo passwd root
```

18.04 Root

```
sudo -i
```

18.04 BBR

18.04.1

```
echo "net.core.default_qdisc=fq" >> /etc/sysctl.conf
echo "net.ipv4.tcp_congestion_control=bbr" >> /etc/sysctl.conf
```

18.04

```
sysctl -p
```

18.04

```
reboot
```

18.04.1


```
sysctl net.ipv4.tcp_available_congestion_control
```



```
net.ipv4.tcp_available_congestion_control = reno cubic bbr
```



```
lsmod | grep bbr
```



```
tcp_bbr 20480 29
```


nginx 

nginx 

nginx

Nginx Conf

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    listen 80;
    server_name ;

    ssl_certificate SSL/cert.pem;
    ssl_certificate_key SSL/key.pem;

    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-XSS-Protection "1; mode=block" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header Referrer-Policy "no-referrer-when-downgrade" always;
    add_header Content-Security-Policy "default-src * data: 'unsafe-eval' 'unsafe-inline' "
always;
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
always;

    location / {
        proxy_pass ;
        proxy_connect_timeout 300s;
        proxy_read_timeout 300s;
        proxy_send_timeout 300s;
        proxy_set_header    Host                $host:$proxy_port;
        proxy_set_header     X-Real-IP           $remote_addr;
        proxy_set_header     X-Forwarded-For     $proxy_add_x_forwarded_for;
        proxy_set_header     Via                 "nginx";
    }
}
```



```
vim /etc/nginx/sites-enabled/nginx
```

```
|||||
```

```
|||||
```

```
systemctl reload nginx
```






MTProxy GO

Root

```
sudo -i
```



```
wget -N --no-check-certificate
https://github.com/whunt1/onekeymakemtg/raw/master/mtproxy_go.sh && chmod +x mtproxy_go.sh &&
bash mtproxy_go.sh
```



```
MTProxy-Go [v2.0.0]
---- Toyo && July | doubibackup.com/es5fj9se.html ----

0. [ ]

1. [ ] MTProxy
2. [ ] MTProxy
3. [ ] MTProxy

4. [ ] MTProxy
5. [ ] MTProxy
6. [ ] MTProxy

7. [ ] [ ]
8. [ ] [ ]
9. [ ] [ ]
10. [ ] [ ]
```


{}

[0-10]:1



[] MTProxy[] !

[] [] ..

[] MTProxy [] [1-65535]

({} 443):

=====

[] : Port

=====

[] MTProxy []32[] [0-9][a-z][A-Z] []

([]TLS[]):

[]TLS[] Y/n]

([]Y []):

[]TLS[]

([]itunes.apple.com):

=====

[] : key

[]TLS[] : YES

=====

[] MTProxy [] TAG[]TAG[]32[]TAG[] @MTProxybot []

([]):

[]NAT[]IP[] 10. xx. xx. xx [] IPv4[]

([] IPv4 []):

=====

[]NAT-IPv4 : IP

=====

====NAT==== IPv6

(IPv6):

=====

NAT-IPv6 :

=====

====[Y/n]

[] () (dd)

(Y):

=====

: YES

=====

[] ..

[] iptables ..

[] iptables ..

[] iptables ..

[] ..

[] MTProxy ..

[] MTProxy !



Mtproto Proxy

[] IP

[] Port

[] Key

[] tg: //proxy?server=IP&port=port&secret=key

[] https: //t. me/proxy?server=ip&port=port&secret=key

TLS[] YES

[] dd TLS

TAG TAG

■■■■

v2ray

■■ root

```
sudo -i
```

■■ v2Ray

```
bash <(curl -s -L https://git.io/v2ray.sh)
```

■■■■ enter ■■■■

■■■■■■■■■■

```
----- V2Ray ■■■■ -----
```

```
■■ (Address) = IP
```

```
■■ (Port) = Port
```

```
■■ID (User ID / UUID) = ID
```

```
■■ID (Alter Id) = ID
```

```
■■■■ (Network) = tcp
```

```
■■■■ (header type) = none
```

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

■■ Telegram Proxy ■■■■

```
v2ray tg
```


■■■■■■■

```
❏ .. ■■■ Telegram MTPROTO ... ■■■■■■■■■■■ ^_^

■■■ Telegram MTPROTO [ Y/N]
( ■■ [N]): y

■■■ Telegram MTPROTO ■■ [1-65535] ■■■ V2Ray ■■■
( ■■■: 22693):

Telegram MTPROTO ■■ = 22693
```

■■■■■■■■■■

■■■ enter ■■■

```
----- Telegram MTPROTO ■■■ -----

■■■ (Hostname) = IP

■■■ (Port) = 22693

■■■ (Secret) = key

Telegram ■■■■ = ■■■■■
```

■■■■■■■■■■■■■■ Proxy ■~~~

■■■

v2Ray ■■■■

```
v2ray info ■■ V2Ray ■■■
v2ray config ■■ V2Ray ■■
v2ray link ■■ V2Ray ■■■■
v2ray infolink ■■ V2Ray ■■■■
v2ray qr ■■ V2Ray ■■■■
```



```
v2ray ss -s Shadowsocks -s
v2ray ssinfo -s Shadowsocks -s
v2ray ssqr -s Shadowsocks -s
v2ray status -s V2Ray -s
v2ray start -s V2Ray
v2ray stop -s V2Ray
v2ray restart -s V2Ray
v2ray log -s V2Ray -s
v2ray update -s V2Ray
v2ray update.sh -s V2Ray -s
v2ray uninstall -s V2Ray
```

v2Ray -s

```
V2Ray -s/etc/v2ray/config.json
Caddy -s/etc/caddy/Caddyfile
-s: /etc/v2ray/233blog_v2ray_backup.conf
```


ServerStats

Ubuntu 安装 ServerStatus 脚本

安装

1. 安装 ServerStatus

```
wget https://raw.githubusercontent.com/CokeMine/ServerStatus-Hotaru/master/status.sh && chmod +x status.sh
```

使用

```
# 安装
bash status.sh s

# 运行
bash status.sh c
```

2. 运行

```
bash status.sh c
```

运行结果

```
ServerStats [ 0-10]: 1
[ 0] ServerStats [ 0-10] ..
[ 0] ServerStats [ 0-10] IP/[ 0] server]
( [ 0] 127. 0. 0. 1): server

=====
[IP/[ 0] server]:  server
=====

ServerStats [ 0-10] [ 1- 65535] [ 0-10]
( [ 0] 35601): 35601
```



```

00 ServerStatus 00000000[ username] 00/00000000

```

(☐ ☐) : vps

```
[[[ username]: vps
```

--	--

```
ServerStatus [password] /
```

```
(☐ doub.io): password
```

```
[[[ password]: password
```

[ServerStatus !

```
[root@localhost ~]# iptables -F
```

```
[root@localhost ~]# iptables -F
```

```
[root@localhost ~]# iptables -F
```

```
[ ] ServerStatus Client [ ] !
```

[illegible]

acme.sh SSL

acme.sh SSL

SSL

```
acme.sh --list
```

```
acme.sh --info -d example.com
```

```
crontab -l

56 * * * * "/root/.acme.sh"/acme.sh --cron --home "/root/.acme.sh" > /dev/null
```

acme.sh

```
acme.sh
acme.sh --upgrade

acme.sh
acme.sh --upgrade --auto-upgrade

acme.sh
acme.sh --upgrade --auto-upgrade 0
```