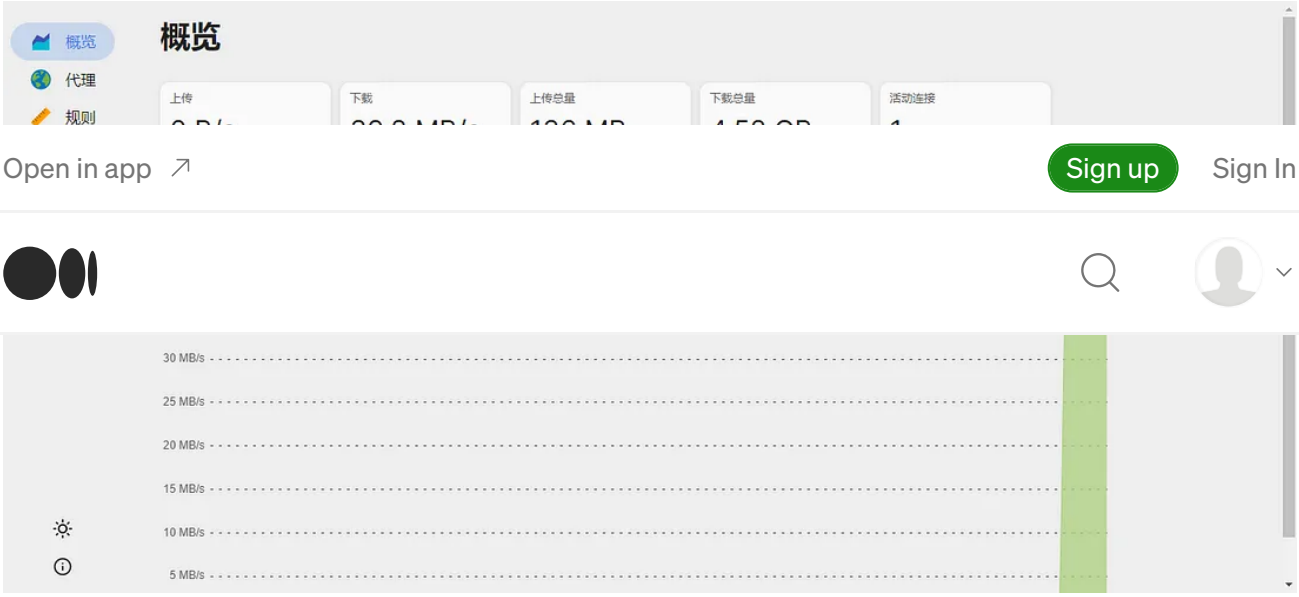
Terrance Siu Follow
Sep 9, 2022 · 11 min read



# 新玩法：在RouterOS跑Clash Premium




运行中的截图

RouterOS 7.x开始支持Docker，在7.4x版本中开始稳定，并支持tun特性，所以Clash Premium可以愉快的跑起来了。

**硬件要求：**

> *a. 运行在x86或者ARM平台的RouterOS 7.4以上的版本，目前的稳定版本是7.5。*
>
> *b. 因为Docker运行需要占用磁盘空间，所以需要插入U盘，或者额外加入一*

> *个16G的虚拟磁盘，并在Winbox格式化，最后提供给Docker存储，本文新增磁盘的挂载路径为/disk1*

1. 开启容器功能

参考下列链接，开启Docker，

> *https://help.mikrotik.com/docs/di tainer*

```
/system/device-mode/update container=yes
```

如果是在ESXi运行RouterOS，执行上面的命令后，应该直接在ESXi上点击“关闭电源”这样才能开启成功，在Winbox的shudown和reboot操作都会开启失败。

2. 建立网络接口并配置nat

```
/interface/veth/add name=veth1 address=172.17.0.2/24
gateway=172.17.0.1
/interface/bridge/add name=dockers
/ip/address/add address=172.17.0.1/24 interface=dockers
/interface/bridge/port add bridge=dockers interface=veth1
/ip/firewall/nat/add chain=srcnat action=masquerade src-
address=172.17.0.0/24
```

3. 设置镜像源

```
/container config
set registry-url=https://registry-1.docker.io
tmpdir=disk1/pull
```

4. 设置配置文件挂载点和环境变量

```
/container mounts
add dst=/.config/clash/ name=clash src=/disk1/config/
/container envs
add key=device name=clash value=/dev/net/tun
add key=net name=clash value=host
add key=cap-add name=clash value=NET_ADMIN
```

5. 通过SFTP创建配置文件目录并上传配置文件

5.1 这里通过Linux中的的sftp客户操作，Windows用户可以用winscp

```
#下载yacd webui文件到Linux中
cd /root
wget "https://github.com/haishanh/yacd/archive/gh-pages.zip"
#解压文件
unzip gh-pages.zip
#使用SFPT工具连接到RouterOS，在sftp的子shell中执行创建目录和上传文件的命令
sftp admin@172.31.34.254
mkdir /disk1/config/
lcd /root
put config.yaml /disk1/config/
mput yacd-gh-pages /disk1/config/
```

6. 配置文件参考,这里不需要过多的规则，因为我们的规则定义在RouterOS的DNS转发上，所以默认全局代理。

```
mixed-port: 25570
redir-port: 3128
bind-address: "*"
allow-lan: true
mode: Rule
log-level: silent
external-ui: yacd-gh-pages
external-controller: '0.0.0.0:25571'
secret: ''
tun:
   enable: true
   stack: system
dns:
```

```
    enable: true
    ipv6: false
    listen: ':53'
    enhanced-mode: fake-ip
    default-nameserver:
      - 114.114.114.114
    fake-ip-filter:
      - '*.lan'
    nameserver:
      - 1.0.0.1
      - 8.8.8.8
      - 9.9.9.9

proxies:
  -
    name: "AWS Tokyo 1"
    type: ss
    server: aws-jp-1.abc.com
    port: 8388
    cipher: aes-256-gcm
    password: "password"
    udp: true
  -
    name: "AWS Singapore 1"
    type: ss
    server: aws-sg-1.abc.com
    port: 8388
    cipher: aes-256-gcm
    password: "password"
    udp: true

proxy-groups:
-
  name: Proxy
  type: select
  proxies:
    - DIRECT
    - 'AWS Tokyo 1'
    - 'AWS Singapore 1'

rules:

# Local

- 'DOMAIN-SUFFIX,local,DIRECT'
  - 'IP-CIDR,127.0.0.0/8,DIRECT'
  - 'IP-CIDR,172.16.0.0/12,DIRECT'
  - 'IP-CIDR,192.168.0.0/16,DIRECT'
  - 'IP-CIDR,10.0.0.0/8,DIRECT'
  - 'IP-CIDR,17.0.0.0/8,DIRECT'
  - 'IP-CIDR,100.64.0.0/10,DIRECT'
```

```
  - 'MATCH,Proxy'
```

7. 运行容器

7.1 根据你的网络到docker快慢，可能这里需要执行非常久。

```
/container
add remote-image=dreamacro/clash-premium:latest
interface=veth1 logging=yes mounts=clash root-
dir=disk1/dreamacro/clash-premium
```

8. 通过RouterOS的日志查看容器运行日志

```
/log print
 10:20:45 container,info,debug 02:20:45 INF [Config] initial
compatible provider name=Proxy
```

9. 添加路由，让数据包可以通过Clash的tun接口转发

9.1 目前Clash采用fakeip方式，对应的Fakeip地址是198.18.0.0/16，所以我们需要添加一条静态路由

```
/ip route
add disabled=no distance=1 dst-address=198.18.0.0/16
gateway=172.17.0.2 pref-src=0.0.0.0 routing-table=main
scope=30 suppress-hw-offload=no target-scope=10
```

10. 测试

10.1 如果198.18.0.1能ping通，证明路由可达到TUN接口，如果198.18.0.2不能ping通，说明需要进入容器的Linux系统中开启内核的IPv4转发。

```
```

```
ping 198.18.0.1
ping 198.18.0.2
```

10. 2 开启IPv4转发，通过RouterOS进入容器的Shell环境，开启内核转发，然后再次测试ping 198.18.0.2

```
/container/shell 0
echo "1" > /proc/sys/net/ipv4/ip_forward
```

11. 端口映射，以便在外部访问yacd 的webui

```
/ip firewall nat
add action=dst-nat chain=dstnat dst-port=25571 protocol=tcp
to-addresses=172.17.0.2 to-ports=25571
add action=dst-nat chain=dstnat dst-port=25570 protocol=tcp
to-addresses=172.17.0.2 to-ports=25570
add action=dst-nat chain=dstnat dst-port=3128 protocol=tcp
to-addresses=172.17.0.2 to-ports=3128
```

12. 导入DNS分流规则

```
/ip/dns/cache/flush;
/ip/dns/static/add name=raw.githubusercontent.com forward-
to=198.18.0.1
/tool/fetch mode=https
url="https://raw.githubusercontent.com/terrancesiu/routeros/m
ain/dns.rsc"
/import file-name=dns.rsc
```

12.1 设定一个一个计划任务，每周更新

```
/system scheduleradd
comment="https://github.com/terrancesiu/routeros" interval=1w
```

```
name=schedule2000 on-event="/ip/dns/cache/flush;\r\
\n:delay 5s;\r\    \n/ip/dns/static/add
name=raw.githubusercontent.com forward-to=198.18.0.1\r\
\n:local URL
\"https://raw.githubusercontent.com/terrancesiu/routeros/main
/dns.rsc\"\r\    \n:local GET [/tool/fetch mode=https
url=\$URL]\r\    \n:delay 5s;\r\    \n:local NAME [/file/get
dns.rsc name]\r\    \n:if (\$NAME = \"dns.rsc\") do={\r\
\n/ip/dns/static/remove [find forward-to=198.18.0.1];\r\
\n/import file-name=\$NAME;\r\    \n:log/info \"fetch: file
\\\"\$NAME\\\" importd\";\r\    \n/file/remove dns.rsc;\r\
\n:log/info \"fetch: file \\\"\$NAME\\\" removed\";\r\
\n}"
policy=ftp,reboot,read,write,policy,test,password,sniff,sensi
tive,romon start-date=nov/19/2020 start-time=00:00:00
```

13. 测试

```
curl -vv http://www.youtube.com
* About to connect() to www.youtube.com port 80 (#0)
*   Trying 198.18.0.12...
* Connected to www.youtube.com (198.18.0.12) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: www.youtube.com
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Content-Type: application/binary
< X-Content-Type-Options: nosniff
< Cache-Control: no-cache, no-store, max-age=0, must-
revalidate
< Pragma: no-cache
< Expires: Mon, 01 Jan 1990 00:00:00 GMT
< Date: Fri, 09 Sep 2022 03:27:07 GMT
< Location: https://www.youtube.com/
< Server: ESF
< Content-Length: 0
< X-XSS-Protection: 0
< X-Frame-Options: SAMEORIGIN
<
* Connection #0 to host www.youtube.com left intact
```