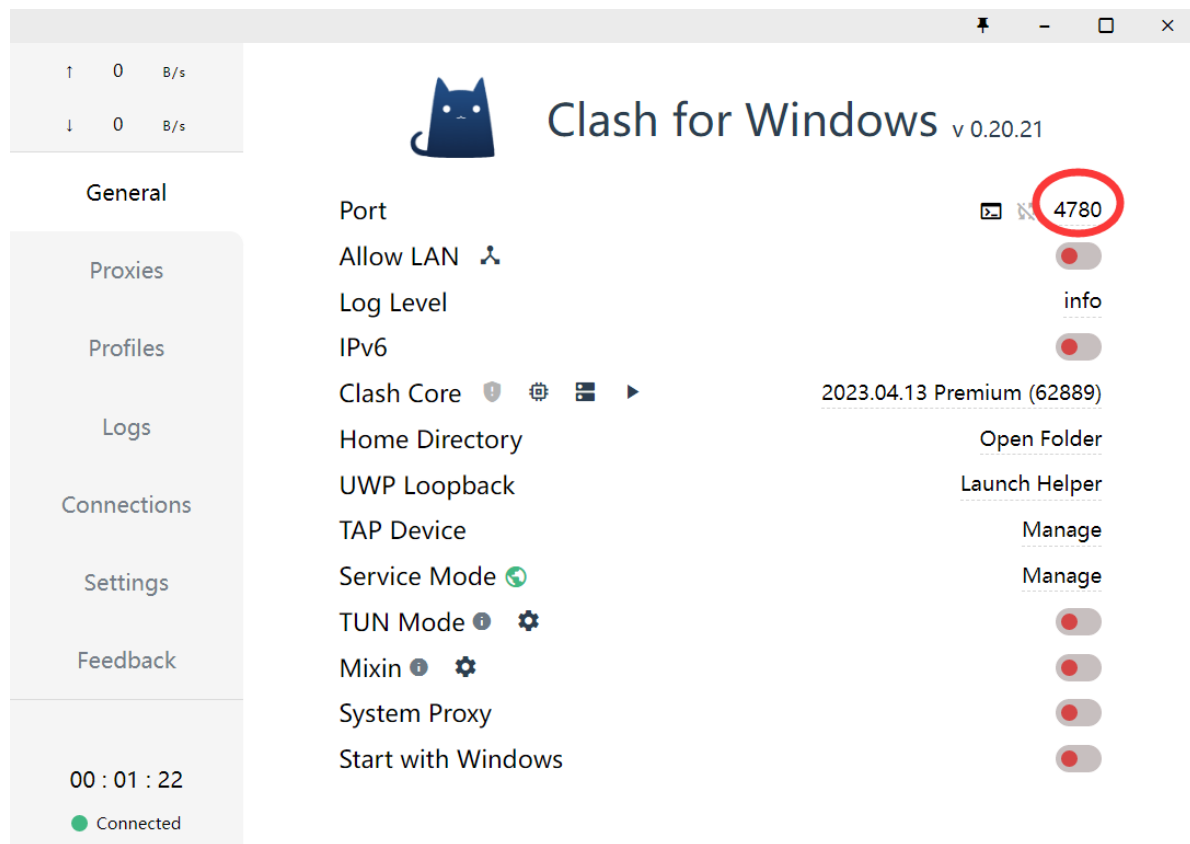


问题1 如何用本地代理实现服务器代理

假设服务器端的代理要放在2333端口上，查看本地clash端口为4780



使用管理员权限开启**本地**终端 其中2333和4780的位置根据你的自身情况适配

```
ssh -R 2333:127.0.0.1:4780 username@ip
```

```
Microsoft Windows [版本 10.0.19044.3086]
(c) Microsoft Corporation. 保留所有权利。

C:\WINDOWS\system32>ssh -R 2333:127.0.0.1:4780 username@ip
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-131-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

219 个可升级软件包。
146 个安全更新。

Last login: Thu Oct 12 11:55:05 2023 from 192.168.4.254
(base) username@ubuntu: $
```

现在 所有服务器上的2333端口的流量都会被转发到本地的4780端口上

接下来设置服务器上的流量走服务器2333端口

服务器端输入

```
export http_proxy=http://127.0.0.1:2333
export https_proxy=http://127.0.0.1:2333
```

完成设置 并使用

```
curl cip.cc
```

检查ip, 如果未成功, 请检查以下几点:

- 本地代理是否开了全局模型
- 端口号有无写错
- 是否打了多余的空格

请注意以下几点

- 最初使用 `ssh -R` 连接的那个终端不能关闭 否则流量不能转发 你在使用 `ssh -R` 连接之后使用另外的终端 (如 vscode新建的终端)
- 每次设置的export只对当前终端生效, 开启一个新终端需要重新设置
- 跑python代码的时候可以使用

```
os.environ['http_proxy'] = 'http://127.0.0.1:2333'
os.environ['https_proxy'] = 'http://127.0.0.1:2333'
```

来起到一样的效果

```
C:\WINDOWS\system32>ssh -R 2333:127.0.0.1:4780 [redacted]
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-131-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

219 个可升级软件包。
146 个安全更新。

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Oct 21 00:10:32 2023 from 192.168.4.253
(base) [redacted]@ubuntu:~$ export http_proxy=http://127.0.0.1:2333
(base) [redacted]@ubuntu:~$ export https_proxy=http://127.0.0.1:2333
(base) [redacted]@ubuntu:~$ curl cip.cc
IP       : 194.147.16.120
地址     : 瑞士 瑞士
数据二   : 瑞士
数据三   : 西班牙
URL      : http://www.cip.cc/194.147.16.120
(base) [redacted]@ubuntu:~$
```

另一个来自群友的可能有用的方法 <https://github.com/mzz2017/gg>

问题2 如何本地终端配置代理

一个很常见的使用场景是 `git bash` 假设本机代理端口在4780

对于类 linux 系统（包括 windows 的 `git bash`）在终端输入

```
export http_proxy=http://127.0.0.1:4780
export https_proxy=http://127.0.0.1:4780
```

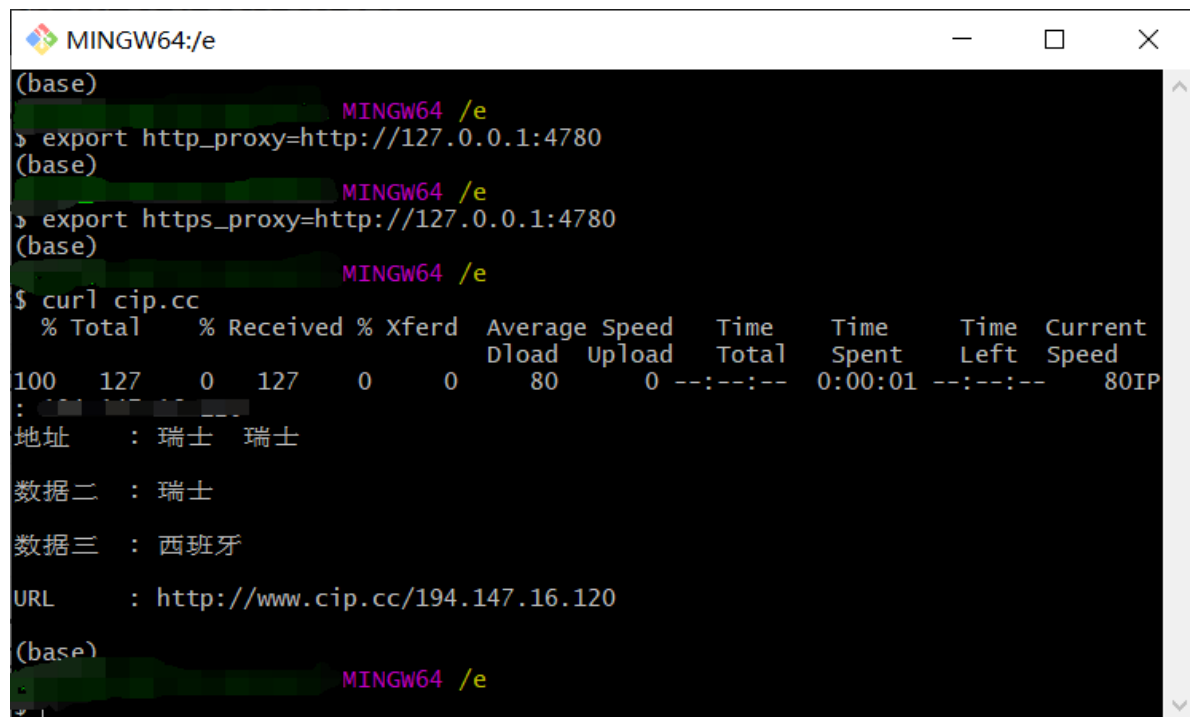
对于 windows 系统 在终端输入

```
set http_proxy=http://127.0.0.1:4780
set https_proxy=http://127.0.0.1:4780
```

后使用

```
curl cip.cc
```

查看代理是否成功



The screenshot shows a terminal window titled "MINGW64:/e". The user enters the following commands:

```
(base) MINGW64 /e
$ export http_proxy=http://127.0.0.1:4780
(base) MINGW64 /e
$ export https_proxy=http://127.0.0.1:4780
(base) MINGW64 /e
$ curl cip.cc
```

The output of the `curl` command shows a successful connection to `cip.cc`. The output includes a table with columns: % Total, % Received, % Xferd, Average Speed, Time, Time, Time, Current, and a final line showing the IP address: 194.147.16.120.

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Speed
100	127	0	127	0	0	80	0
--:--:-- 0:00:01 --:--:-- 80IP							

Address: 瑞士 瑞士
Data 2: 瑞士
Data 3: 西班牙
URL: http://www.cip.cc/194.147.16.120

(base) MINGW64 /e