



链滴

安装 wsl2 后一些必要的配置

作者: [zhaobingchun](#)

原文链接: <https://ld246.com/article/1605342250324>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



在wsl中使用主机代理

首先我安装的wsl发行版是ubuntu，其他系统差不多。

我们知道在Linux中使用代理的方式：

```
export http_proxy=sock5://ip:port
export https_proxy=sock5://ip:port
```

那么我们主机的IP在局域网中总会变，但是我们发现有一个IP是在/etc/resolv.conf中那么我们可以用个命令取出来，我们直接打开家目录下的.bashrc，vim ~/.bashrc输入

```
export hostip=$(cat /etc/resolv.conf | grep -oP '(?<=nameserver\ ).*')
alias proxy='export http_proxy="sock5://${hostip}:1080"; export https_proxy="http://${hostip}
1080"; echo "open proxy"'
alias noproxy='export http_proxy= ;export https_proxy= ; echo "close proxy"'
```

这样我们就可以输入proxy打开代理，输入noproxy关闭代理

但是，我们发现这样是连不上主机的代理的，因为从wsl发出的网络包被windows的防火墙拦截了

解决方法是，按win+x键以管理员方式打开powershell添加以下防火墙规则

```
New-NetFirewallRule -DisplayName "WSL" -Direction Inbound -InterfaceAlias "vEthernet (WS
)" -Action Allow
```

```
PS C:\Users\75987> New-NetFirewallRule -DisplayName "WSL" -Direction Inbound -InterfaceAlias "vEthernet (WSL)" -Action Allow

Name                : {78642c3a-c0ca-4bcf-8ef9-d1ce39650b21}
DisplayName          : WSL
Description          :
DisplayGroup        :
Group                :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Inbound
Action               : Allow
EdgeTraversalPolicy  : Block
LooseSourceMapping   : False
LocalOnlyMapping     : False
Owner                :
PrimaryStatus        : OK
Status               : 已从存储区成功分析规则。 (65536)
EnforcementStatus    : NotApplicable
PolicyStoreSource    : PersistentStore
PolicyStoreSourceType : Local
RemoteDynamicKeywordAddresses : {}
```

主机或局域网中访问wsl中的服务

主机中访问wsl的服务有两种方式，一种直接使用localhost访问，不过这种方式缺点就是有些调试工具不支持，而且不支持。第二种方式就是使用端口转发，这种比较麻烦的部分是wsl每次启动IP都会变而用localhost经常会连接不上。我们可以获取wsl2的IP并写入到hosts中，感谢shayne大佬提供了[工具](#)载后使用管理员启动powershell执行如下命令

```
.\wsl2host.exe install
```

输入计算机用户名，密码，重启wsl后就会自动讲wsl的ip写入到hosts，ubuntu.wsl,然后设置端口转发，在powershell中执行

```
netsh interface protproxy add v4tov4 listenport=监听的端口 connectaddress=ubuntu.wsl connectport=监听的端口 listenaddress=* protocol=tcp
```

wsl中开机自启

我们经常会在wsl中安装一些服务，并希望这些服务开机自启，我们可以在wsl中的/etc/init.wsl写入机启动的服务比如redis

```
#!/bin/bash
service redis-server start
```

在windows下创建启动的vbs脚本 linux-start.vbs

```
Set ws = WScript.CreateObject("WScript.Shell")
ws.run "wsl -d ubuntu -u root /etc/init.wsl"
```

按win+r键输入： shell:startup

将linux-start.vbs拖入到这个目录就可以了，这样就实现了开机启动