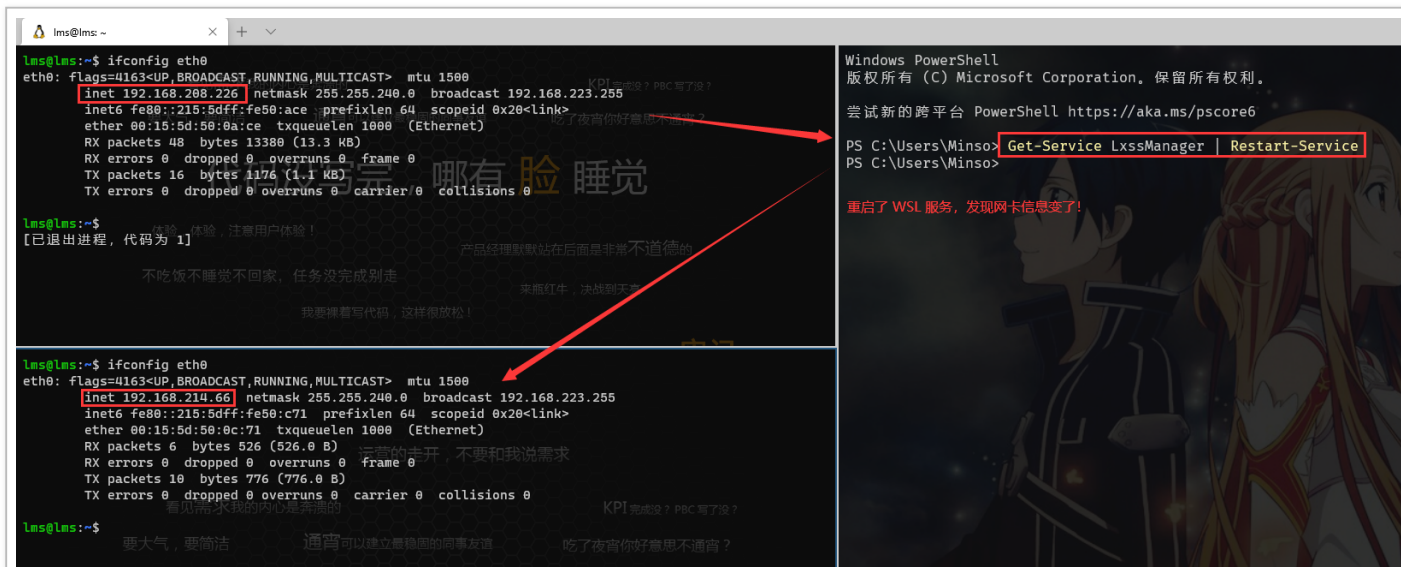


# 解决 WSL 开机网卡信息变动问题

WSL2 设置私有 IP 由于 WSL 2 新的体系结构使用虚拟化的网络组件，每次计算机重启的时候都会重置 WSL 2 的虚拟网卡，因此每次重启完计算机后 WSL2 中的 IP 地址都不一定一样。

## WSL2 设置私有 IP

由于 WSL 2 新的体系结构使用虚拟化的网络组件，每次计算机重启的时候都会重置 WSL 2 的虚拟网卡，因此每次重启完计算机后 WSL2 中的 IP 地址都不一定一样。



虽然我们可以通过 `localhost` 访问到 WSL 中的服务，但是当需要配置多个虚拟域名的时候该方法显然就不适用了，此时若要在宿主机中通过虚拟域名访问 WSL 2 中的网络服务需要每次手动对宿主机的 `hosts` 文件修改其 `IP-Domain` 映射关系。

思考后结合在网上查阅了资料，大致共两个方案：

- 方法一：通过批处理脚本，启动时自动获取 WSL 2 的网卡信息，自动修改宿主机的 `hosts` 原有的 `IP-Domain` 信息

该方法需要懂批处理脚本语言，而且实际开发情景中开发环境一般都是整个开发组统一进行维护的。包括 `hosts` 文件一般也是统一进行维护然后通过 `SwitchHosts` 从远端拉取。因此方式一比较适用个人本地项目，因为 `hosts` 一般不会变动，但并不适用协同办公的情景。

- **方法二：想办法将 WSL 的网卡信息“固定下来” - 设置私有 IP，组建 宿主机-WSL 2 局域网**

由于 WSL 2 的网卡每次都会被重置，因此设置静态 IP 的方式势必是走不通的。因此：只能在每次计算机开机或重启的时候通过脚本为 WSL 2 增置一个新的私有网卡 IP，并同时在宿主机上增设一个一样网段的以太网卡私有 IP，在宿主机与虚拟机之间组建一个小型的局域网进行服务通信。

该方法比较通用，维护成本更加小，较为推荐！

```
# `WSL 2` 新增网卡
# - 设置私有IP网段为192.168.33.10
# - 广播地址为192.168.169.15
# - 网卡名称为 eth0
# - 设置网卡标签 eth0:1
wsl -u root ip addr add 192.168.33.10 broadcast 192.168.169.15 dev eth0 label eth0:1

# 设置宿主机网卡`vEthernet (WSL)`地址为 `192.168.33.1`-需要管理员权限
netsh interface ip add address "vEthernet (WSL)" 192.168.33.1 255.255.255.240
```

方法二结合 CSDN 博主 - manbucy 的 《WSL2 固定 ip 地址》进行了修改

## 附录 - 源码

对下述两个问题的脚本进行汇总，已经做了一些优化：

- 解决 WSL 开机启动服务
- 解决 WSL 开机网卡信息变动问题

1. WSL2 开机启动 Docker 服务脚本 - 放在虚拟机中 /etc/init.wsl 位置 ``sh #!/bin/sh

```
sudo service docker start
```

```
sudo mkdir -p /run/sendsigs.omit.d/ && sudo /etc/init.d/rpcbind start
```

2. `Windows` 宿主机开机需自动执行的 `init-wsl.bat` 脚本

```
``bat
@echo off
:: run task as administrator
%1 %2
ver|find "5.">nul&&goto :Admin
mshta vbscript:createobject("shell.application").shellexecute("%~s0","goto :Admin","","runas",1)(window
:Admin
```

```

:: set WSL distribution
set distribution=Ubuntu-20.04

:: 设置WSL初始化脚本位置
set initWSL=/etc/init.wsl

:: 配置宿主机私有网络IP、WSL2私有网络IP、广播地址、子网掩码
set hostOSIP=192.168.33.1
set WSL2IP=192.168.33.10
set WSL2Broadcast=192.168.33.15
set subnetMask=255.255.255.240

:: init WSL2 services
wsl -d %distribution% -u root -e %initWSL%
if %ERRORLEVEL% NEQ 0 (
    echo init WSL services error OR Already init WSL!
    pause
) else (
    echo init WSL success!
)

:: set wsl2 ip
wsl -d %distribution% -u root ip addr | findstr "%WSL2IP%" > nul
if !errorlevel! equ 0 (
    echo wsl ip has set!
) else (
    wsl -d %distribution% -u root ip addr add %WSL2IP% broadcast %WSL2Broadcast% dev eth0 label eth0:1
    echo set wsl ip success: %WSL2IP%
)

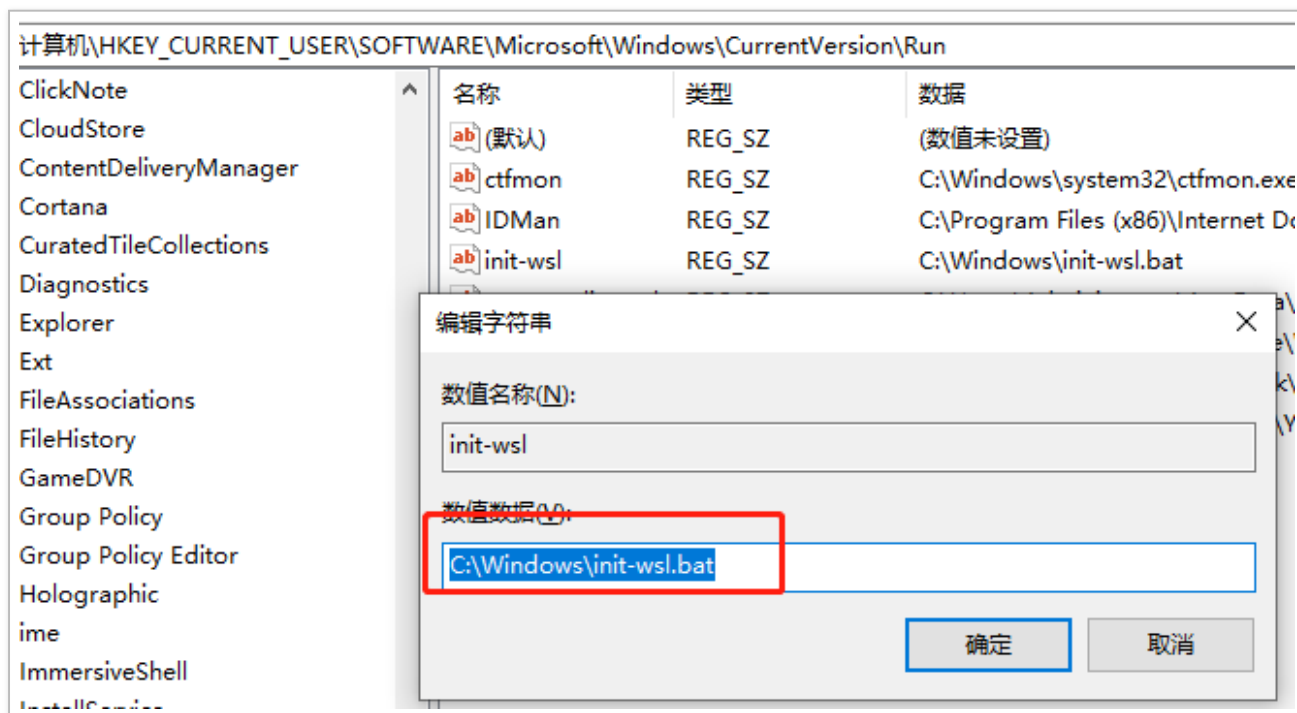
:: set windows ip
ipconfig | findstr %hostOSIP% > nul
if !errorlevel! equ 0 (
    echo windows ip has set!
) else (
    netsh interface ip add address "vEthernet (WSL)" %hostOSIP% %subnetMask%
    echo set windows ip success: %hostOSIP%
)

```

### 1. 添加 Windows 开机自动执行 init-wsl.bat 脚本

Windows 键 + R , 输入 regedit 打开注册列表, 在地址栏输入 计算机

\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run 定位到对应注册表位置, 右键新建一个字符串值, 键入 init-wsl.bat 脚本绝对路径即可, 如下图!



重启计算机即可发现，WSL 中的 Docker 服务已经被启动，通过 192.168.33.10 可以直接访问到 WSL 内部的网络！

全文完

本文由 简悦 SimpRead 转码，用以提升阅读体验，原文地址