

WSL2 安装 Docker

原创 老李家的打字机 Leaders工作室 1周前

收录于话题

#Docker

1个

WSL2 相较于 WSL 除了 I/O 性能有了巨大的改进，最主要的：WSL2 的底层是一个跑在 Hyper-V 上的完整的 Linux 系统，而不是像 WSL 一样是穿着 Linux 的外衣和 Windows 打交道的系统。

因此，在 WSL2 没有出现之前如果要在 Windows 上玩 Docker 很多人是不建议的。

- 因为即使是使用 Docker 官方的提供的 Docker for Windows 也是经常出些奇奇怪怪的错误并迟迟得不到解决。
- 通过虚拟机安装 boot2docker.iso，在虚拟出来的 Linux 系统下玩，但启动虚拟机是真慢。而且随着 WSL2 的推出 boot2docker 也不再维护。

因此 WSL2，对于经常需要使用一些办公程序软件、又想玩 Docker、还没钱买 Mac 的用户来说，确实是一个福音！

安装 WSL2

查看系统版本

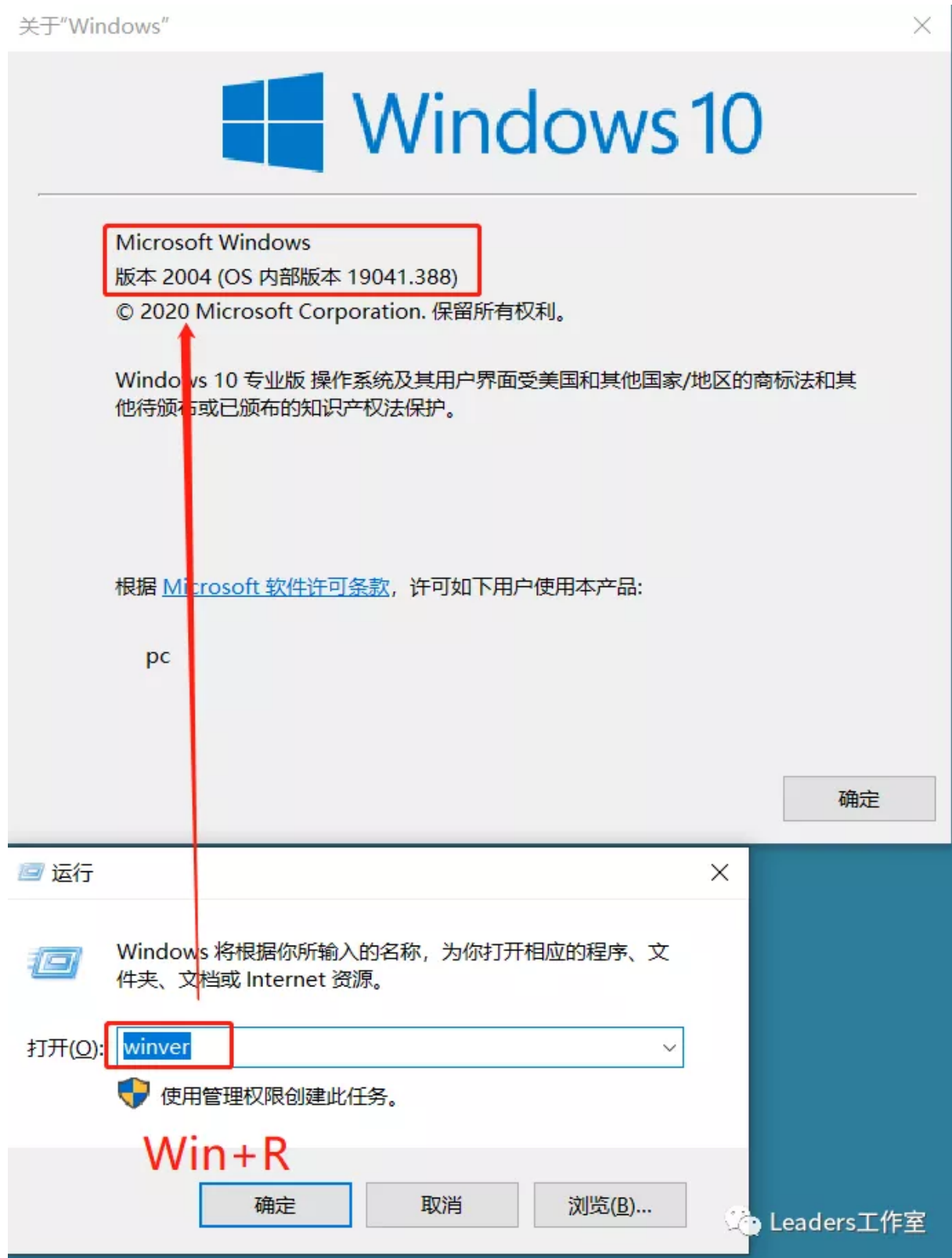
Requirements

- For x64 systems: Version 1903 or higher, with Build 18362 or higher.
- For ARM64 systems: Version 2004 or higher, with Build 19041 or higher.
- Builds lower than 18362 do not support WSL 2. Use the Windows Update Assistant to update your version of Windows.

Update to WSL2 Requirements

即：最新的 WSL2 特性要求在 Windows 10 x64 Version 1903 及以上版本的系统或 ARM 64 systems: Version 2004 及以上版本的系统。

Win+R 输入 winver，即可查看到当前自己 Windows 版本信息：



winver

- 更新 [Windows](https://www.microsoft.com/zh-hk/software-download/windows10) 工具: <https://www.microsoft.com/zh-hk/software-download/windows10>
- [Windows 1909](https://www.win10gw.com/win10wenzhang/6507.html) 升级失败问题:
<https://www.win10gw.com/win10wenzhang/6507.html>

安装 WSL 并升级到 WSL2

一定要确保系统已经达到标准，不然通过 WSL 使用 Docker 会遇到各种不知道为啥的坑，而且启动 Docker 也非常慢。

WSL2 更新特性: <https://docs.microsoft.com/en-us/windows/wsl/compare-versions#whats-new-in-wsl-2>

下列命令都需要使用管理员身份运行 PowerShell，运行下列脚本！！！！

1. 安装 WSL：开启 Windows-Subsystem-Linux 特性

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

2. 升级 WSL2 -开启Windows的虚拟平台特性

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

升级完毕之后，需要重启计算机！！！！

3. 下载安装 Linux 内核更新包

查看系统类型，在 PowerShell 中输入 systeminfo 即可查看，如下图是 x64 的版本：

```
PS C:\Users\Administrator> systeminfo

主机名:          DESKTOP-MMFTD6M
OS 名称:         Microsoft Windows 10 专业版
OS 版本:         10.0.19041 暂缺 Build 19041
OS 制造商:       Microsoft Corporation
OS 配置:         独立工作站
OS 构建类型:     Multiprocessor Free
注册的所有人:   pc
注册的组织:
产品 ID:         00331-10000-00001-AA426
初始安装日期:   2020/6/18, 1:11:29
系统启动时间:   2020/11/2, 20:15:53
系统制造商:     LENOVO
系统型号:       81HX
系统类型:       x64-based PC
```

Leaders工作室

systeminfo

- WSL2 Linux kernel update package for x64 machines
- WSL2 Linux kernel update package for ARM64 machines

下载对应补丁包安装补丁！

4. 设置默认通过 WSL2 方式安装 Linux 系统

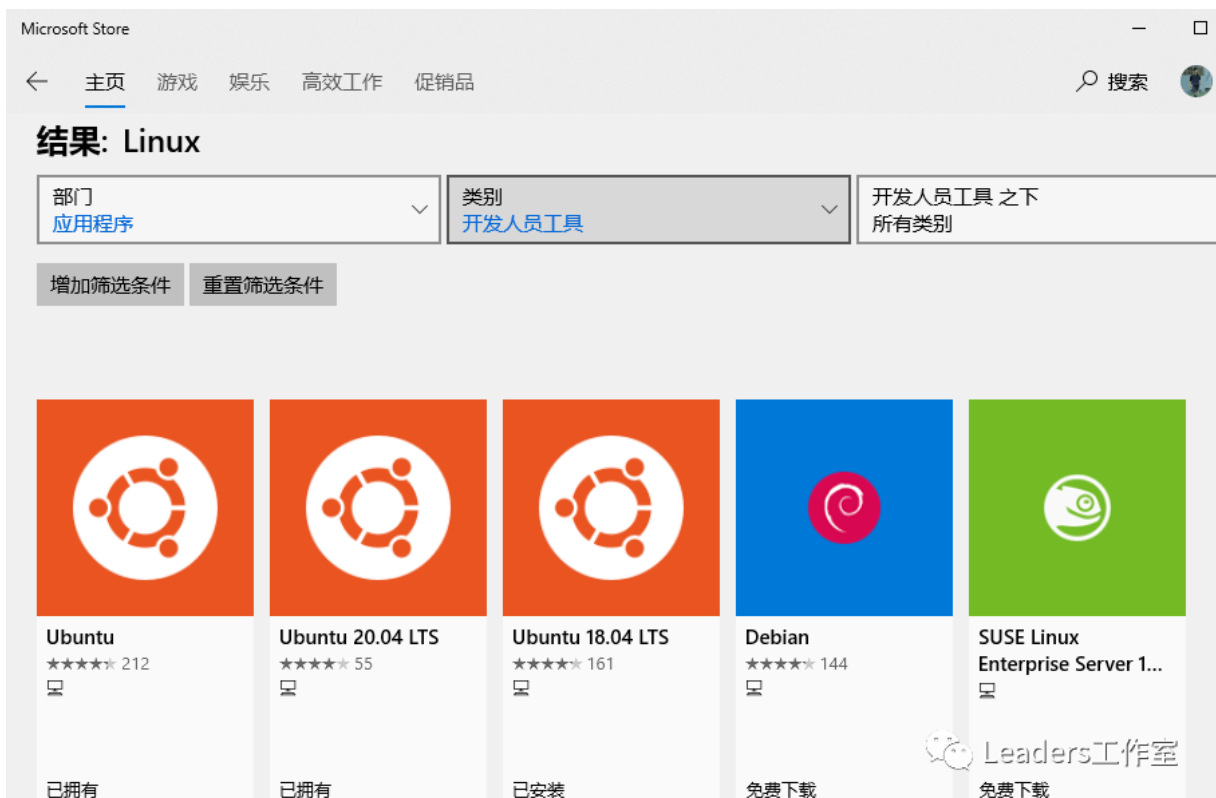
要使用管理员身份运行 PowerShell，运行下列脚本：

```
wsl --set-default-version 2
```

如果你之前已经通过 WSL1 安装了 Linux，现在想转为 WSL2 运行，可参考：

<https://docs.microsoft.com/en-us/windows/wsl/install-win10#set-your-distribution-version-to-wsl-1-or-wsl-2>

5. 安装 Linux 子系统 在 Microsoft Store 上搜索一个你自己喜欢的子系统进行安装



Microsoft Store Result for search Linux

打开 PowerShell，输入 `wsl -l -v` 可以看到你的子系统运行版本

```
PS C:\Users\Administrator> wsl -l -v
```

NAME	STATE	VERSION
* Ubuntu-18.04	Running	2
docker-desktop-data	Running	

Leaders工作室

wsl -l -v

如果你想重启 WSL 系统，使用管理员身份打开 [PowerShell](#) 执行下列命令

```
Get-Service LxssManager | Restart-Service
```

安装Windows Terminal

[Windows](#) 的命令行向来是以丑著称，[Windows Terminal](#) 总算是挽救了一点它的颜值。

Windows Terminal

- 安装: <https://github.com/microsoft/terminal>
- 使用指南: <https://docs.microsoft.com/zh-cn/windows/terminal/customize-settings/global-settings>
- 如何给 Windows Terminal 增加一个新的终端

下述是我将 [Git-Bash](#) 添加到 [Windows Terminal](#) 的配置

```
{
  // guid 可以通过网上查询 guid 生成器生成: http://tool.pfan.cn/guidgen
  "guid": "{ce7a80b8-da75-4628-a2e1-663af0f3ce7c}", // 终端唯一标识号, 用于设置 defaultProc
  "name": "Git Bash", // 终端名称
  "icon": "C:\\Self\\code\\CVS\\Git\\git-for-windows.ico", // 终端ICON图标路径
  "commandline": "C:\\Self\\code\\CVS\\Git\\bin\\bash.exe --login -i", // 终端所在绝对路径
  "startingDirectory": "D:\\htdocs", // 打开终端默认打开路径, 如果为null则默认上一次退出时路径
  "hidden": false, // 是否隐藏不展示
  "useAcrylic" : true, // 是否使用磨砂效果
  "acrylicOpacity" : 0.8, // 磨砂效果
  "colorScheme" : "Campbell", // 配色方案
  "cursorColor" : "#FFFFFFD", // 光标颜色
  "fontFace" : "Fira Code", // 终端字体
  "backgroundImage": "C:\\Users\\Minso\\Pictures\\Camera Roll\\view.jpg", // 终端背景图
  "backgroundImageOpacity": 0.75, // 背景图透明度
}
```

方式一: Docker Desktop+WSL2 运行 Docker

Docker Desktop 将 Docker CE、Docker Compose、Kubernetes 等软件整合在了一起进行安装, 省去了一一安装的烦恼。

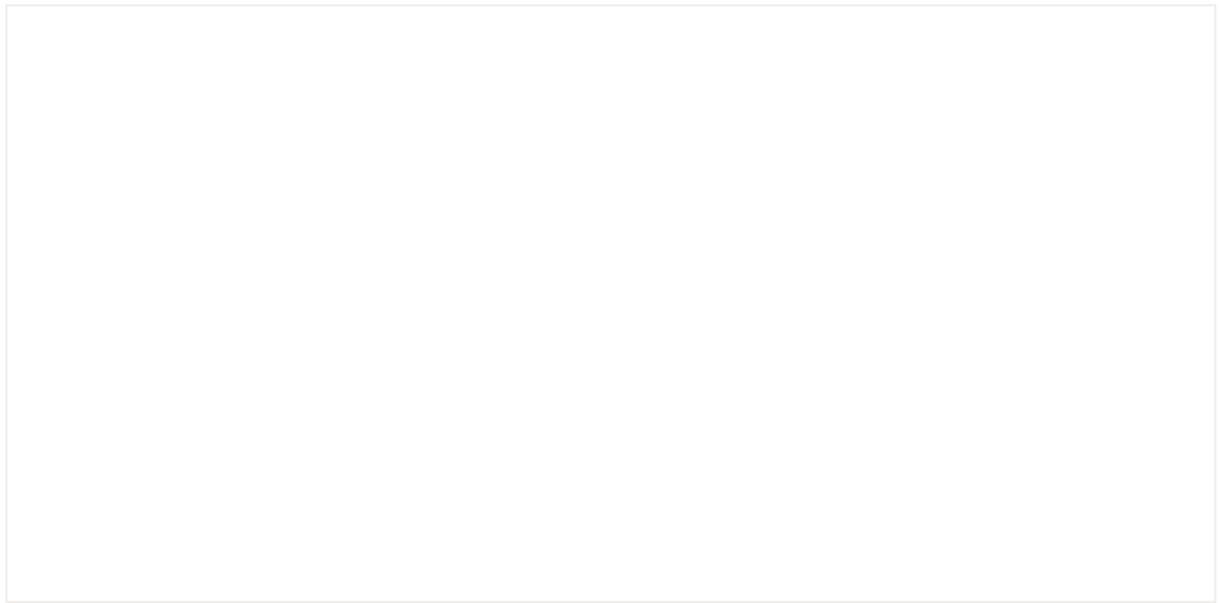
Docker Daemon 由于是安装在宿主机上的, 因此可以直接使用宿主机的网卡信息对容器进行访问。

下载安装Docker Desktop运行 Docker, 可以让你在Windows中方便的管理配置Docker

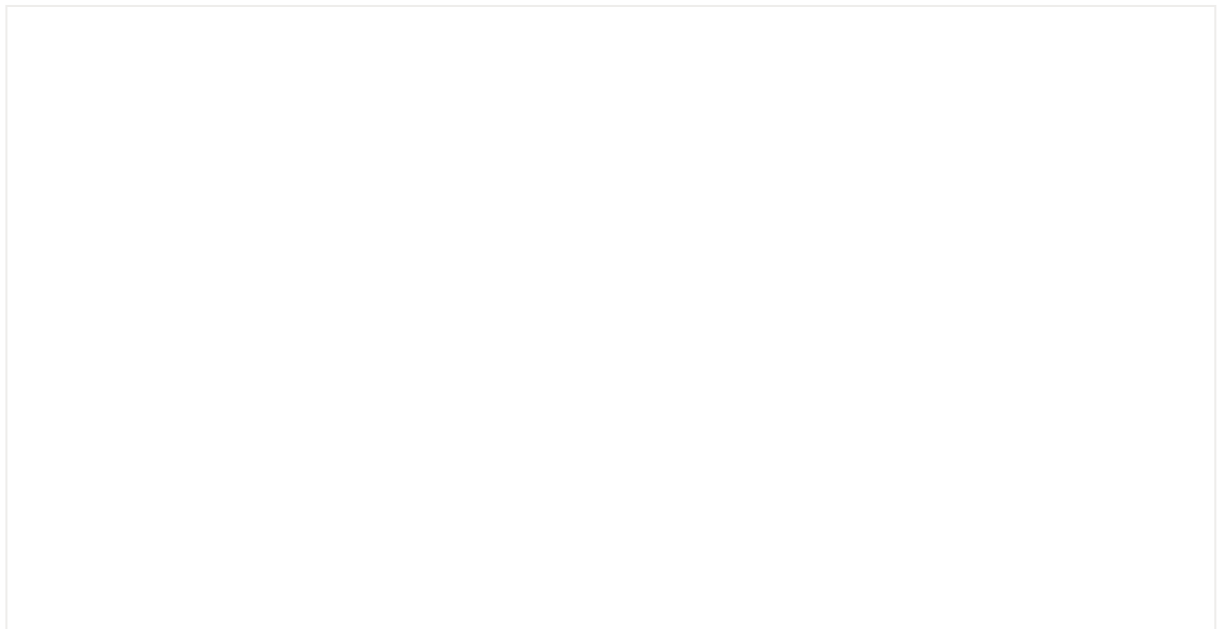
- Docker for Windows:
<https://desktop.docker.com/win/stable/Docker%20Desktop%20Installer.exe>
- Docker for Mac: <https://desktop.docker.com/mac/stable/Docker.dmg>

傻瓜式安装即可!

配置Docker Desktop使用WSL2运行Docker engine



Use the WSL 2 based engine



Configure which WSL 2 distros you want to access Docker from

现在你就可以在对应的子系统中玩 **Docker** 了，打开安装好的子系统，输入 **docker info** 就可以查看到对应的系统信息了。

方式二：WSL2子系统中安装**Docker**

该方式是利用 **Windows Hyper-v** 将 **Docker** 装在子系统虚拟机中的，但该方式装的虚拟机启动、读写性能方面要比使用 **Virtual Box** 、 **VM** 的方式更好。

安装最新Docker CE



```
curl -fsSL get.docker.com -o /tmp/get-docker.sh && sudo /bin/sh /tmp/get-docker.sh --mirror
```

启动并检验安装是否成功

```
# 启动 docker 服务
```

```
sudo service docker start # 注意: 有的Linux系统时通过`systemctl start docker`启动
```

```
# 打印docker系统详细信息
```

```
docker info
```

```
docker info
```

看到以上信息证明已经安装完成并成功启动了服务！

踩坑预警

使用 `docker info` 打印信息时可能会遇到下述信息，提示 `/var/run/docker.sock` 权限不足

```
Client:
```



```
Debug Mode: false
```

```
Server:
```

```
ERROR: Got permission denied while trying to connect to the Docker daemon socket at unix:/var/run/docker.sock: Get http://localhost/v1.24/info: net/http: request canceled (Client.Timeout exceeded while awaiting headers)  
errors pretty printing info
```

```
/var/run/docker.sock connect permission denied
```

通过 `ls -al /var/run/docker.sock` 查看该文件权限信息，发现文件是 `root` 用户或 `docker` 组用户才能进行读写

```
ls -al show files info
```

可以通过 `id` 或 `groups ${USER}` 查看当前用户所在的组信息，发现当前用户的附加组信息里并没有 `docker`，那么问题就好解决了：只需将当前用户加入到 `docker` 组即可。

```
id show ${USER} groups
```

1. 创建 `docker` 组

```
sudo groupadd docker
```

2. 将当前用户加入到 `docker` 组

```
sudo usermod -aG docker ${USER}
```

3. 刷新用户的用户组信息

你需要先登出再重新登录一次，这样当前系统会重新刷新你的身份信息。或者，使用下列方式不用退出即可刷新当前进程的用户身份信息：

```
exec su -l ${USER}
```

此时再运行 `docker info` 应该就正常打印信息了！

安装最新Docker Compose

可参考：<https://docs.docker.com/compose/install/#install-compose-on-linux-systems>

```
# 下载当前最新版本是：1.27.4
```

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$"
```

```
# 赋予 docker-compose 可执行权限
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
# 刷新当前进程信息
```

```
exec $SHELL -l
```

```
# 检查docker-compose是否安装成功
```

```
docker-compose --version
```

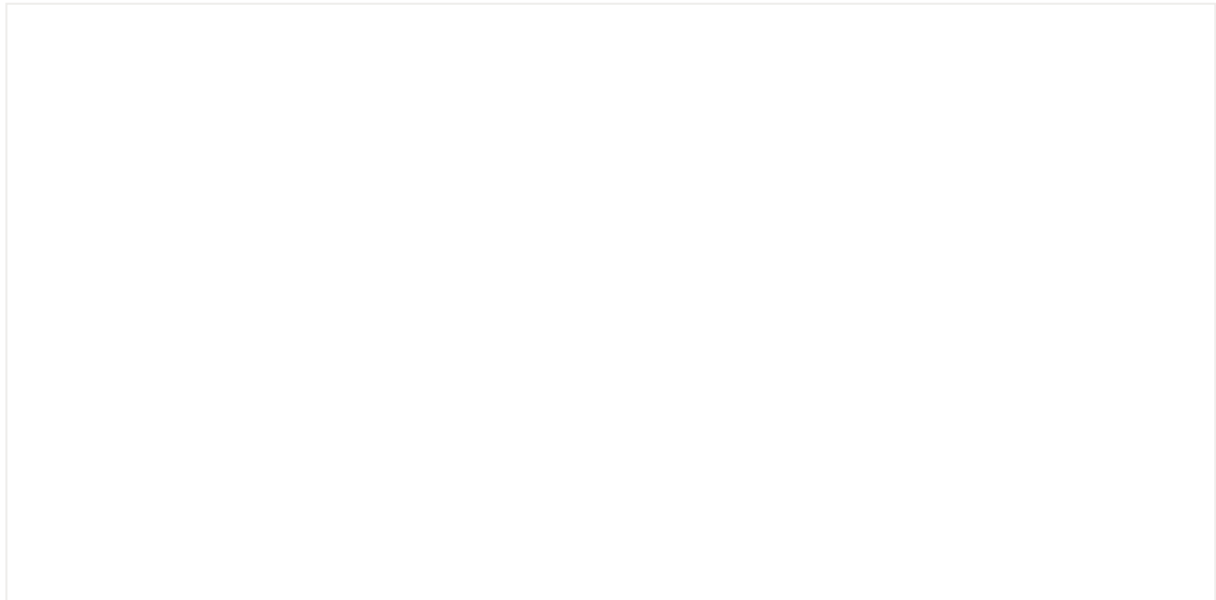
设置Docker Server镜像源加速

- Docker 官方中国区镜像地址：<https://registry.docker-cn.com>
- 网易镜像地址：<http://hub-mirror.c.163.com>
- ustc镜像地址：<https://docker.mirrors.ustc.edu.cn>
- 阿里云镜像地址设置参考文章底部：<https://cr.console.aliyun.com/cn-hangzhou/instances/mirrors>

如果想使用阿里云镜像地址需要有阿里云账号，但我在广州试用了上述的镜像地址，确实设置了阿里云的镜像加速是效果最好的

通过Docker Desktop配置registry-mirrors

如果是使用 [Docker Desktop](#) + [WSL](#) 方式安装 [Docker](#) 直接使用面板即可修改！



set registry-mirrors for Docker Desktop

配置 WSL2 中的 Docker 镜像加速地址

当你的docker版本较新（**Docker Version** \geq 1.10）时，建议直接通过 [/etc/docker/daemon.json](#) 进行配置（若没有该文件则直接新建，写入下述配置即可）

```
{
  "registry-mirrors": ["http://hub-mirror.c.163.com"]
}
```

若 [Docker](#) 版本比较旧，则需根据系统的不同而修改不同位置的配置文件，详细参考[此处](#)

修改完成后，执行 `sudo service docker restart` 重启 [docker](#)

update setting registry-mirrors

解决 WSL2 安装 Docker 的一些问题

1. 开机启动Docker服务

WSL 的是一个基于 Windows 系统的 Hyper V 服务运行的 Linux 系统，但没有对应的开机自检程序，因此在 WSL 中设置服务开机启动是没有用的。

因此，通过实现 Windows 的开机启动项执行一段脚本调用 WSL 内的服务，从而达到 WSL 开机启动服务的目的。

查阅并试验了网上的一些文章，大致分为两类：

- 通过将 VBScript 脚本的快捷方式放到 `C:\Users\{USER}\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup` 下，从而实现开机自启【试验结果：Windows 10 似乎由于权限问题，不能将 *.vbs 格式脚本放置到该目录下】
- 通过修改注册列表的实现开机自动执行一段命令【试验结果：可行】

1. 脚本如下

shell-script-init.wsl

2. 设置 `sudo` 执行时不需要输入密码

配置 `/etc/sudoers` 文件，将下述信息加入到文件末端

```
# 设置sudo脚本-无需输入密码验证
%sudo ALL=NOPASSWD: /etc/init.wsl
```

3. PowerShell 如下

```
# -u root 通过root用户启动wsl
# -e /etc/init.wsl 启动执行 /etc/init.wsl脚本

wsl -u root -e /etc/init.wsl
```

powershell 调用 wsl内部脚本

2. 解决 WSL2 内网IP变动问题

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。

Get-Servcie-LxssManager-Restarat-Service.png

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。

于是，想了两个方案：

- 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` 从远端获取。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。

```
# `WSL 2` 新增网卡
# - 设置私有IP网段为192.168.33.10-192.168.33.28
# - 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 192.168.33.10/28 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
```

3. 修改注册列表实现宿主机开机自动执行初始化脚本

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

```
#!/bin/sh

# 启动docker服务
sudo service docker start

# 开启 rpcbind 服务
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

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

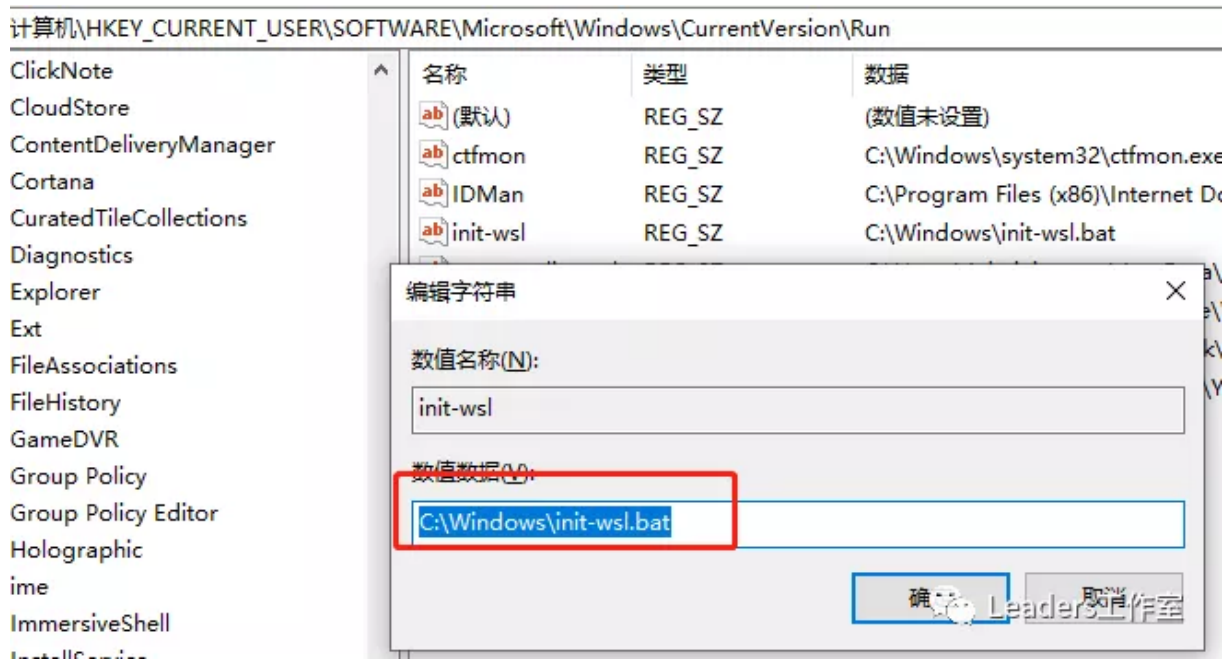
:: init WSL2 services
wsl -d %distribution% -u root -e %initWSL%
if %ERRORLEVEL% NEQ 0 (
    echo init WSL services error OR Already init WSL!
    pause
) else (
    echo init WSL success!
)

:: 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
    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%
)
```

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，键入对应命令即可，如下图！



set register service

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！

[阅读原文](#)