

Install Docker With WSL2

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。

方式一： Docker Desktop + WSL2 运行 Docker

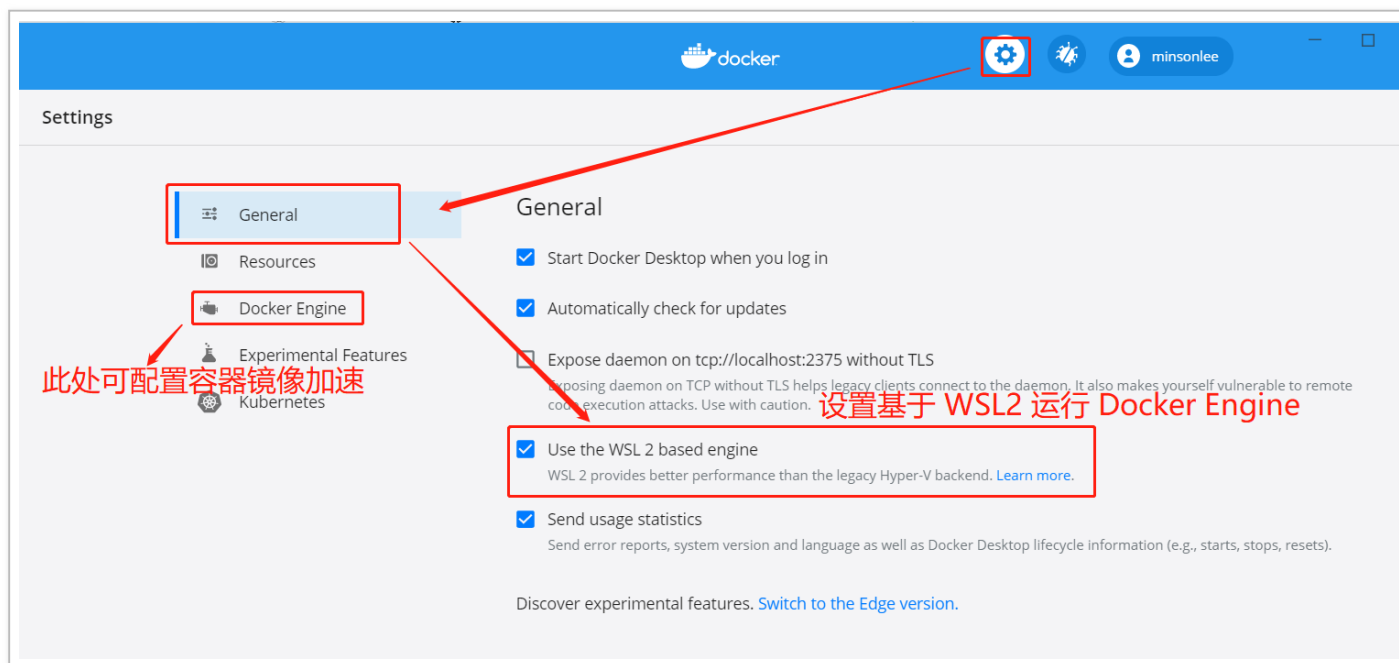
Docker Desktop 将 Docker CE 、 Docker Compose 、 Kubernetes 等软件整合在了一起进行安装，省去了一一安装的烦恼。

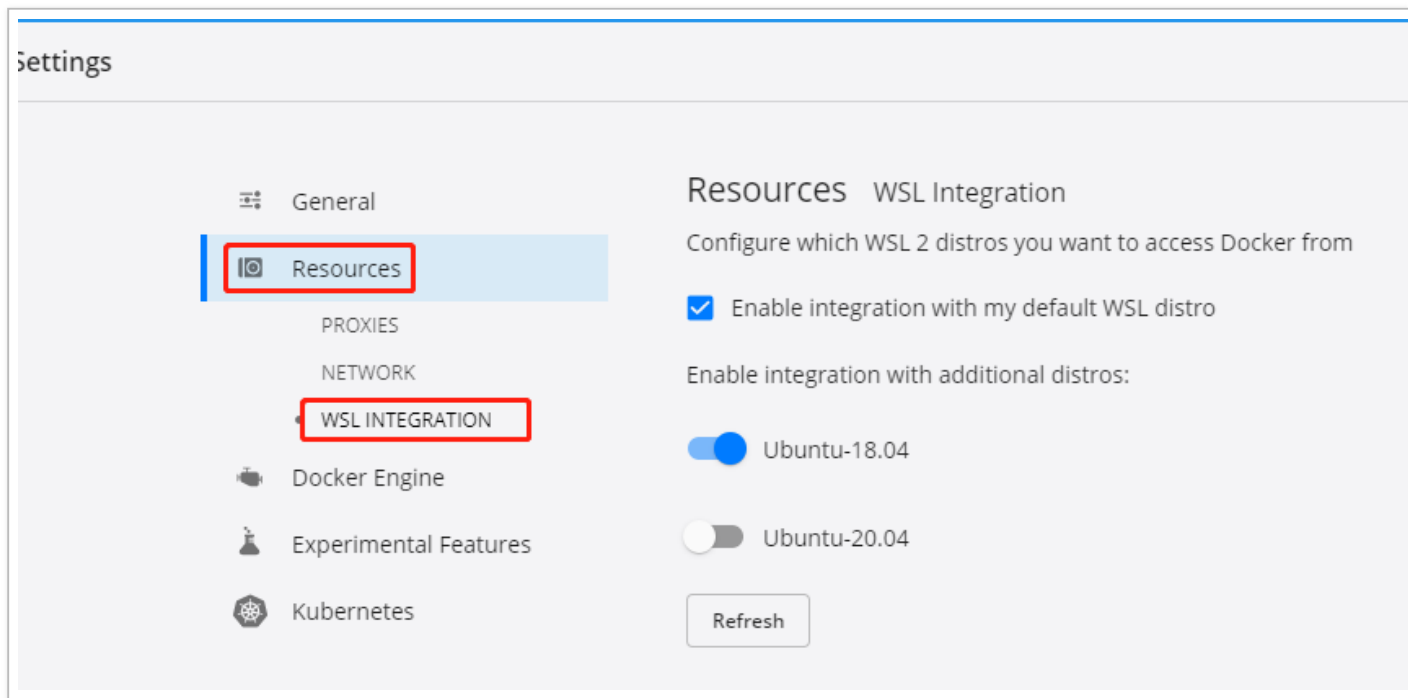
下载安装 Docker Desktop 运行 Docker，可以让你在 Windows 中方便的管理配置 Docker

- Docker for Windows: [https://desktop.docker.com/win/stable/Docker Desktop Installer.exe](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





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

安装最新 Docker CE

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

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
```

```
lms@lms:~$ docker info
Client:
 Debug Mode: false
 Plugins:
  app: Docker Application (Docker Inc., v0.8.0)
  buildx: Build with BuildKit (Docker Inc., v0.4.2-tp-docker)

Server:
 Containers: 3
  Running: 0
  Paused: 0
  Stopped: 3
 Images: 3
 Server Version: 19.03.13
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
```

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:
errors pretty printing info
```

```
lms@LMS:/tmp$ docker info
Client:
 Debug Mode: false

Server:
ERROR: Got permission denied while trying to connect to the Docker dae
mon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fd
ocker.sock/v1.40/info: dial unix /var/run/docker.sock: connect: permis
sion denied
errors pretty printing info
```

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

```
lms@LMS:/tmp$ ll /var/run/docker.sock
srw-rw---- 1 root docker 0 Nov  4 10:46 /var/run/docker.sock=
```

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。

```
lms@LMS:/tmp$ id
uid=1000(lms) gid=1000(lms) groups=1000(lms),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugin),117(netdev)
```

1. 创建 `docker` 组

1. 将当前用户加入到 `docker` 组

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

1. 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：

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>

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

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

exec $SHELL -l

docker-compose --version
```

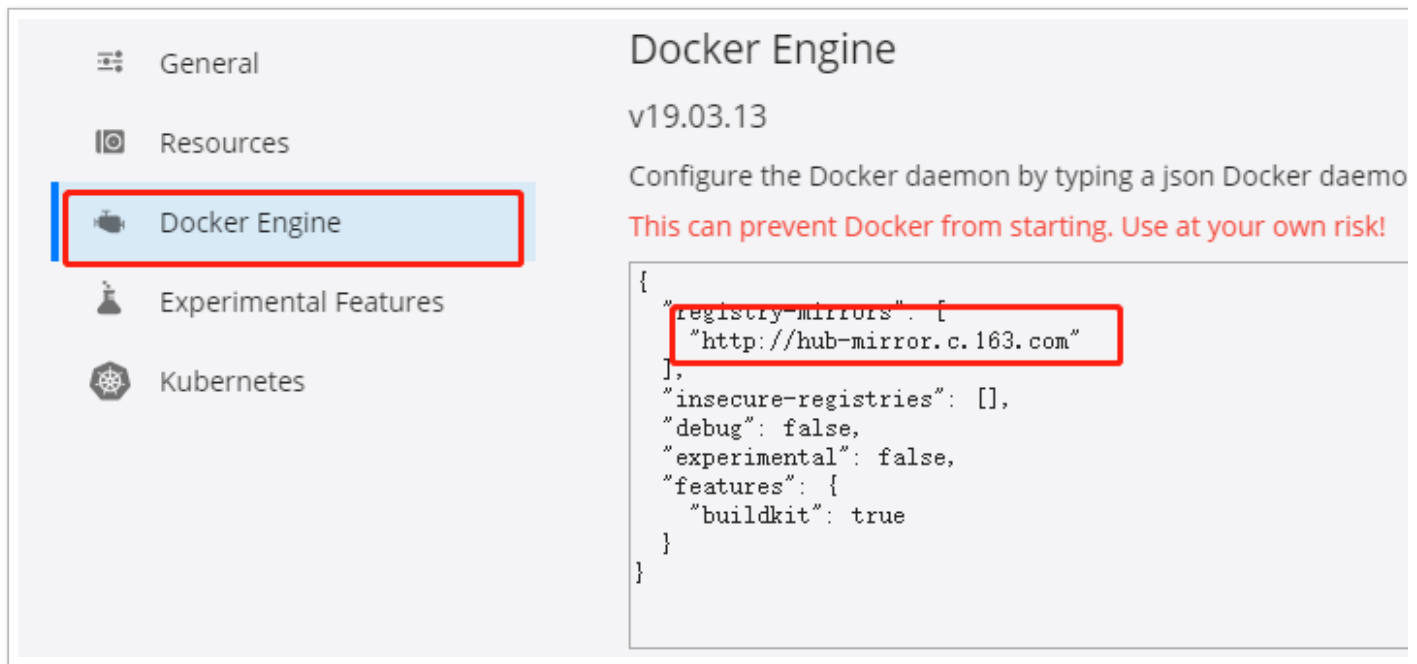
设置 Docker Server 镜像源加速

- 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！



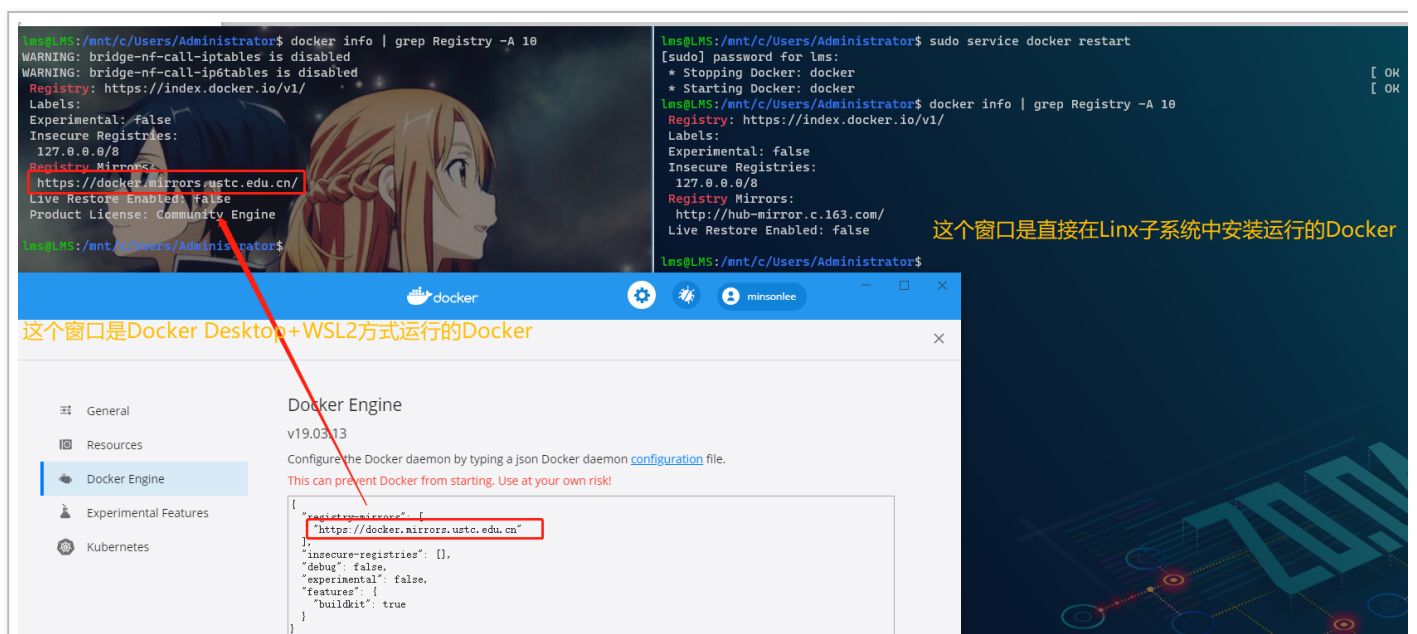
配置 WSL2 中的 Docker 镜像加速地址

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



遗留问题

在安装之后，留下了两个问题：

1. 设置开机自启服务异常

WSL2 中设置开机 `docker` 服务自启，我尝试了使用注册列表、通过 `VBScript` 启动服务、创建文件夹，但是失败了！

问题已解决，参考文章：[解决 WSL 开机启动服务](#)

1. 访问容器服务不方便

由于每次重启 `WSL` 的网卡都会被重置，在公司的网络环境中 `Docker` 的 IP 网段也会不断的变化。因此，通过直接在 `WSL2` 中直接安装 `docker` 的方式，若在网络复杂的环境中在宿主机访问 `WSL` 内部容器服务需要自己手动变更 `IP-Domain` 绑定信息。如果通过 `Docker Desktop` + `WSL2` 的方式则不会有这样的问题，因为该方式 `WSL2` 内部访问的 `daemom` 服务仍然是 `Windows` 中，但该方式安装的 `docker` 文件挂载异常，偶尔会出现挂载目录为空的情况。

问题已解决，参考文章：[解决 WSL 开机网卡信息变动问题](#)

全文完

本文由 简悦 SimpRead 转码，用以提升阅读体验，原文地址