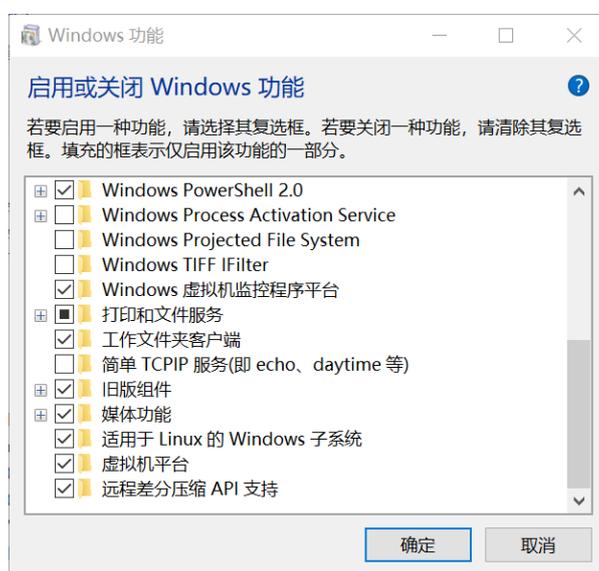


Docker 安装过程：笔者开始时使用“Docker Toolbox+Oracle VM VirtualBox”的方案进行 docker 的安装，但是出现了不兼容以及一些冲突的问题，后改用基于 WSL2 的 Docker Desktop 的方案（该方案不容易出错）。此方案要求 win10 版本为 1903 或者更高版本及 win11（通过点击“开始”按钮，打开“设置”，点击“系统”，在“关于”中可以查看）。

Windows 规格

版本	Windows 10 家庭中文版
版本号	22H2
安装日期	2022/2/15
操作系统内部版本	19045.2251
序列号	██████████
体验	██

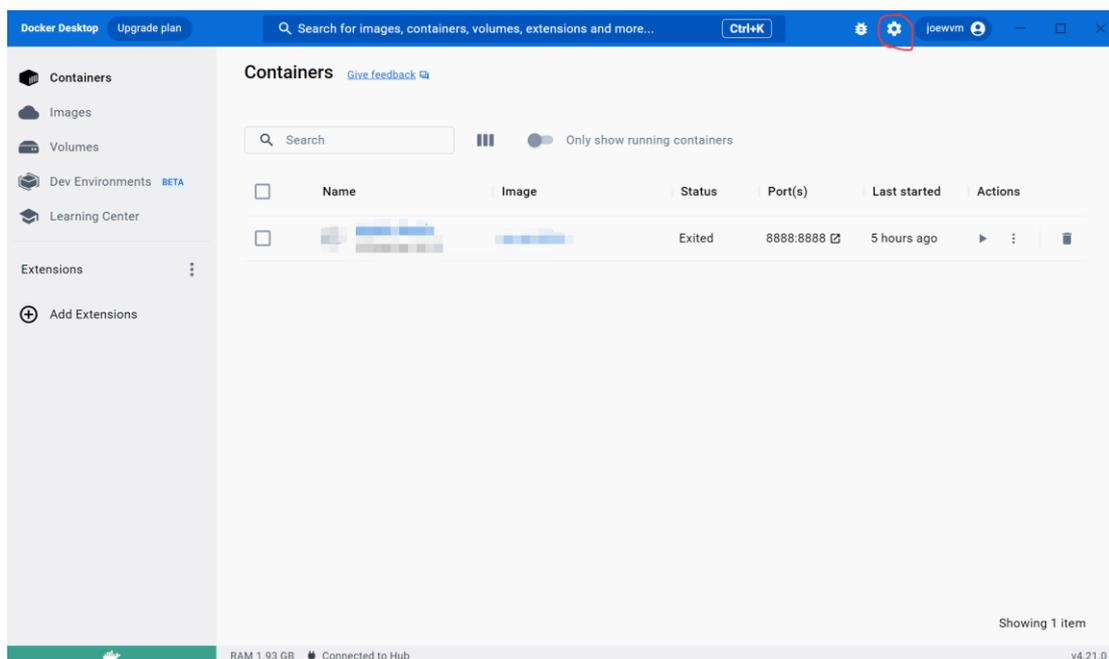
使用 WSL2 前注意在控制面板中勾选：“控制面板-程序-程序与功能-启用或关闭 Windows 功能-适用于 Linux 的 Windows 子系统与虚拟机平台”。如果是第一次勾选，勾选后要先重启电脑。



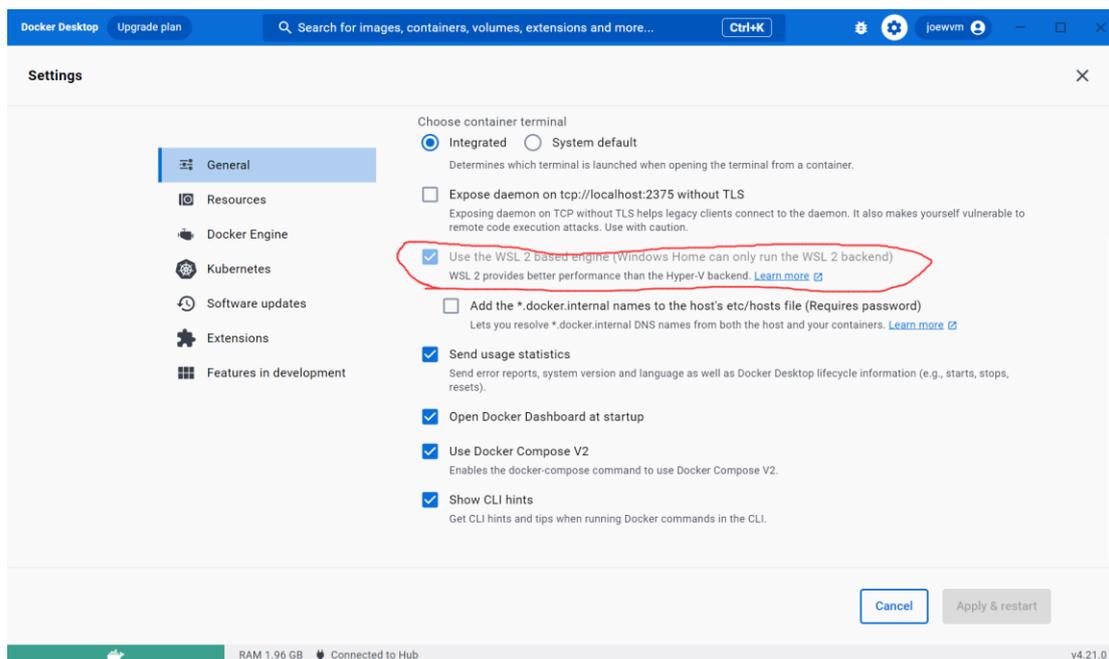
打开网址：<https://docs.microsoft.com/zh-cn/windows/wsl/install-manual#step-4---downloadthe-linux-kernel-update-package>，从步骤四开始操作。

然后就是安装 Docker Desktop，双击运行从 <https://docs.docker.com/docker->

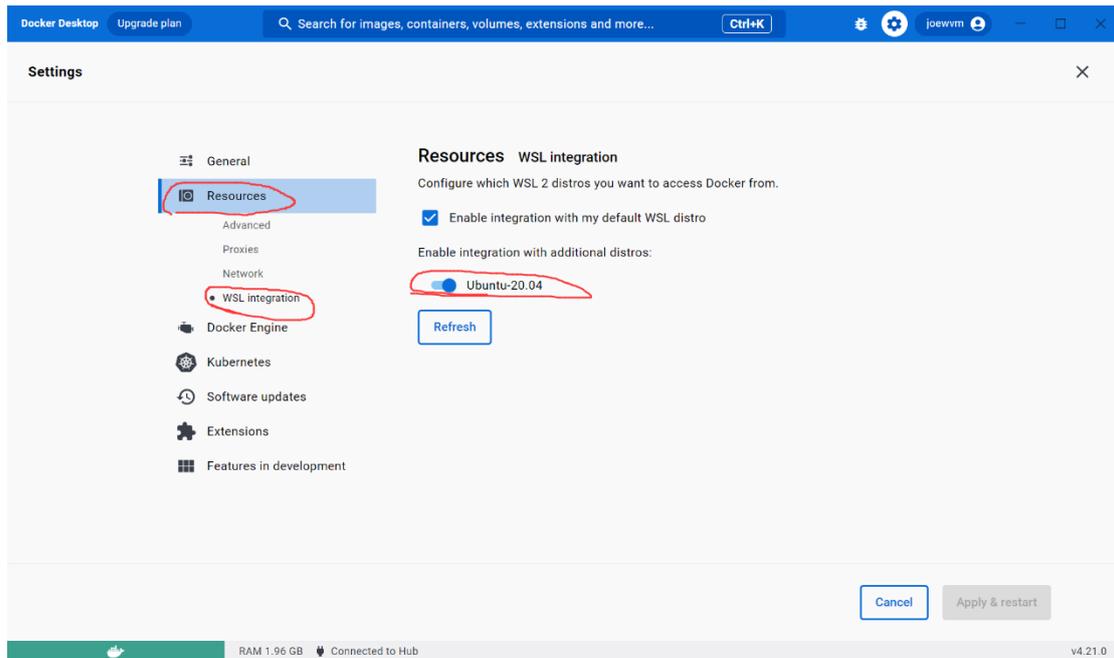
[forwindows/install/](https://forwindows.docker.com/install/)上下载下来的安装包，安装时保持默认。



笔者安装的版本以及默认了使用基于 WSL2 的 Docker 引擎：



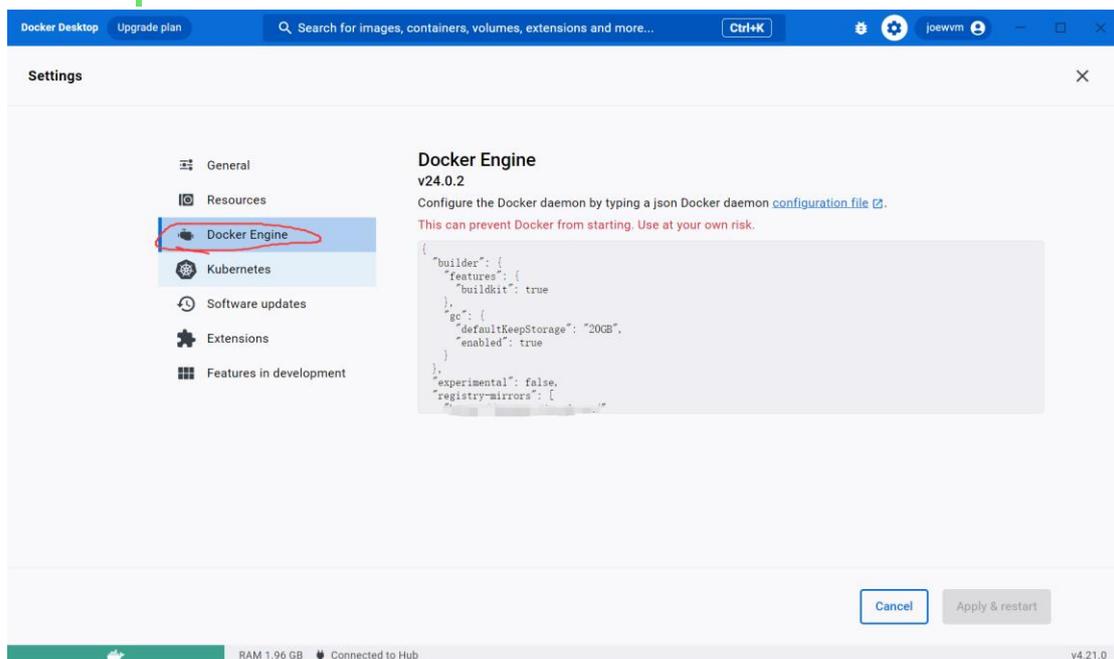
最后在 Settings-Resources-WSL Integration 中设置要从哪个 WSL2 发行版中访问 Docker：



接下来就是配置镜像加速器：

马赛克部分是镜像加速器的地址，笔者使用的是学校的镜像加速器。我们只需要在配置 Docker daemon 的 json 文件中添加如下语句（注意逗号与括号），然后 Apply&Restart 就可以了。

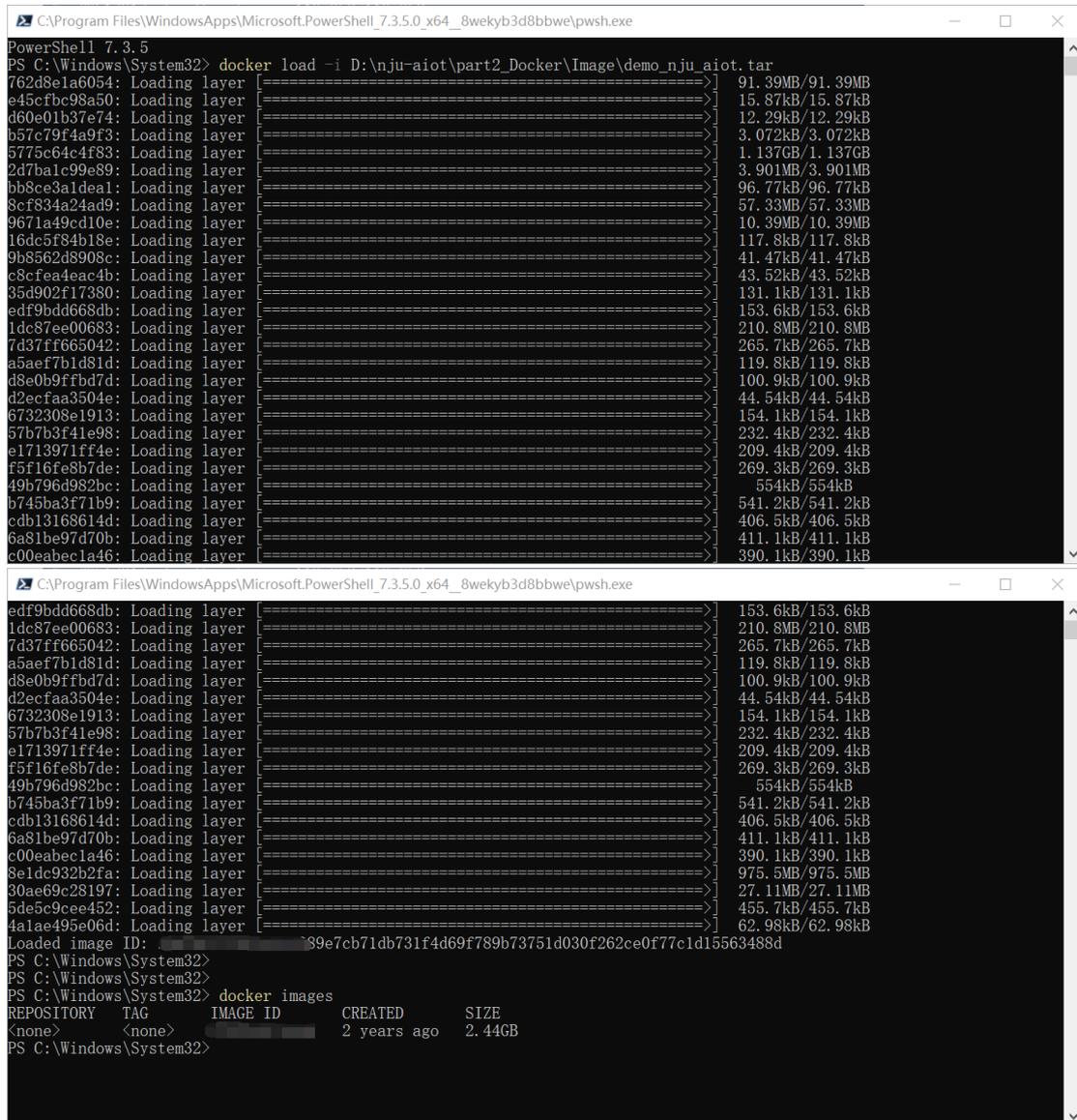
1. "registry-mirrors": [" "] 中括号中间加镜像加速器的地址。



然后我们就完成了 Docker Desktop 的安装。

最后就是镜像的加载和使用：(以下内容的指令都在 Powershell 中运行)

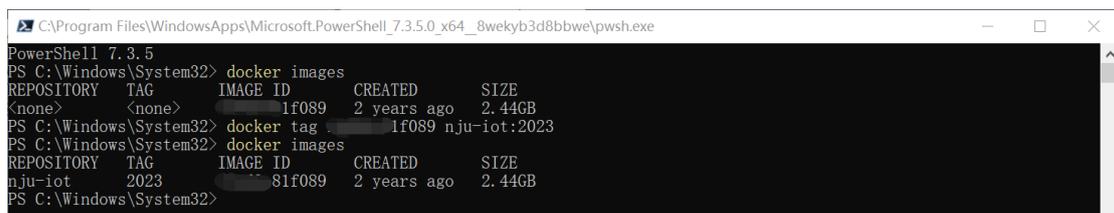
1. 首先用 `docker load -i <image/path>` 命令来实现镜像加载，`<image/path>` 是镜像文件的路径



```
PowerShell 7.3.5
PS C:\Windows\System32> docker load -i D:\nju-aiot\part2_Docker\Image\demo_nju_aiot.tar
762d8e1a6054: Loading layer [====>] 91.39MB/91.39MB
e45cfbc98a50: Loading layer [====>] 15.87kB/15.87kB
d60e01b37e74: Loading layer [====>] 12.29kB/12.29kB
b57c79f4a9f3: Loading layer [====>] 3.072kB/3.072kB
5775c64c4f83: Loading layer [====>] 1.137GB/1.137GB
2d7ba1c99e89: Loading layer [====>] 3.901MB/3.901MB
bb8ce3a1deal: Loading layer [====>] 96.77kB/96.77kB
8ef834a24ad9: Loading layer [====>] 57.33MB/57.33MB
9671a49cd10e: Loading layer [====>] 10.39MB/10.39MB
16dc5f84b18e: Loading layer [====>] 117.8kB/117.8kB
9b8562d8908c: Loading layer [====>] 41.47kB/41.47kB
c8cfea4eac4b: Loading layer [====>] 43.52kB/43.52kB
35d902f17380: Loading layer [====>] 131.1kB/131.1kB
edf9bdd668db: Loading layer [====>] 153.6kB/153.6kB
1dc87ee00683: Loading layer [====>] 210.8MB/210.8MB
7d37ff665042: Loading layer [====>] 265.7kB/265.7kB
a5aef7b1d81d: Loading layer [====>] 119.8kB/119.8kB
d8e0b9ffbd7d: Loading layer [====>] 100.9kB/100.9kB
d2ecfaa3504e: Loading layer [====>] 44.54kB/44.54kB
6732308e1913: Loading layer [====>] 154.1kB/154.1kB
57b7b3f41e98: Loading layer [====>] 232.4kB/232.4kB
e1713971ff4e: Loading layer [====>] 209.4kB/209.4kB
f5f16fe8b7de: Loading layer [====>] 269.3kB/269.3kB
49b796d982bc: Loading layer [====>] 554kB/554kB
b745ba3f71b9: Loading layer [====>] 541.2kB/541.2kB
cdb13168614d: Loading layer [====>] 406.5kB/406.5kB
6a81be97d70b: Loading layer [====>] 411.1kB/411.1kB
c00eabec1a46: Loading layer [====>] 390.1kB/390.1kB

edf9bdd668db: Loading layer [====>] 153.6kB/153.6kB
1dc87ee00683: Loading layer [====>] 210.8MB/210.8MB
7d37ff665042: Loading layer [====>] 265.7kB/265.7kB
a5aef7b1d81d: Loading layer [====>] 119.8kB/119.8kB
d8e0b9ffbd7d: Loading layer [====>] 100.9kB/100.9kB
d2ecfaa3504e: Loading layer [====>] 44.54kB/44.54kB
6732308e1913: Loading layer [====>] 154.1kB/154.1kB
57b7b3f41e98: Loading layer [====>] 232.4kB/232.4kB
e1713971ff4e: Loading layer [====>] 209.4kB/209.4kB
f5f16fe8b7de: Loading layer [====>] 269.3kB/269.3kB
49b796d982bc: Loading layer [====>] 554kB/554kB
b745ba3f71b9: Loading layer [====>] 541.2kB/541.2kB
cdb13168614d: Loading layer [====>] 406.5kB/406.5kB
6a81be97d70b: Loading layer [====>] 411.1kB/411.1kB
c00eabec1a46: Loading layer [====>] 390.1kB/390.1kB
8e1dc932b2fa: Loading layer [====>] 975.5MB/975.5MB
30ae69c28197: Loading layer [====>] 27.11MB/27.11MB
5de5c9ee452: Loading layer [====>] 455.7kB/455.7kB
4a1ae495e06d: Loading layer [====>] 62.98kB/62.98kB
Loaded image ID: 39e7cb71db731f4d69f789b73751d030f262ce0f77c1d15563488d
PS C:\Windows\System32>
PS C:\Windows\System32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
<none> <none> 1f089 2 years ago 2.44GB
PS C:\Windows\System32>
```

2. 然后通过 `docker images` 命令来确认镜像是否加载成功。从上图可以看到笔者的镜像名字为 `none`，我们可以通过 `docker tag IMAGE ID` 命令来进行设置



```
PowerShell 7.3.5
PS C:\Windows\System32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
<none> <none> 1f089 2 years ago 2.44GB
PS C:\Windows\System32> docker tag 1f089 nju-iot:2023
PS C:\Windows\System32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nju-iot 2023 81f089 2 years ago 2.44GB
PS C:\Windows\System32>
```

遮挡部分是 IMAGE ID， 需要记录下来。

3.然后启动容器， 并执行 jupyter notebook： 使用以下命令

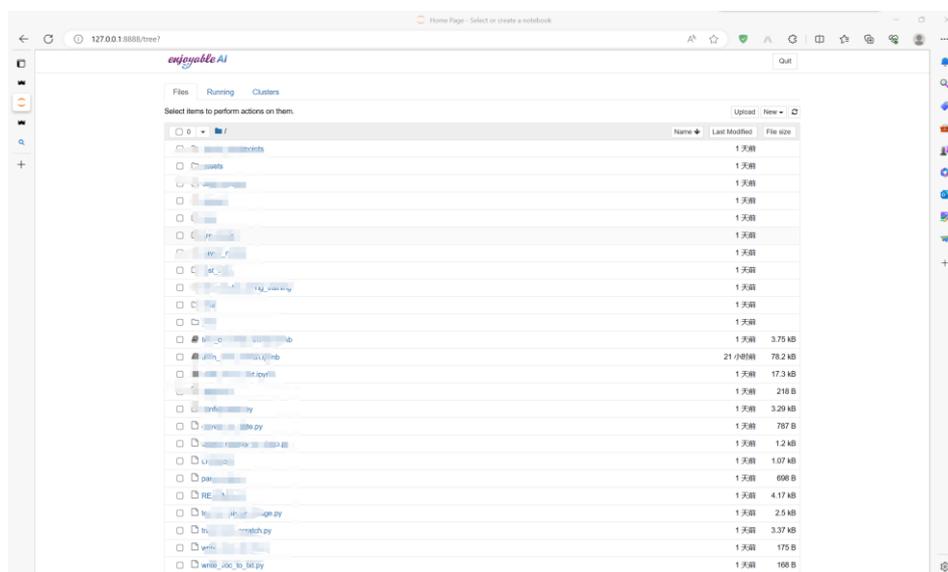
```
docker run -d -p 8888:8888 -v <program/path>:/home/assignment/ nju-iot:2023 jupyter notebook
```

注意 assignment/之后千万不要忘记加空格， 这是很容易犯的错误。

```
PS C:\Windows\System32> docker run -d -p 8888:8888 -v d:/nju-aiot/part4_model_preparation/program_aiot:/home/assignment/ nju-iot:2023 jupyter notebook
6726dd1ec12d1e539f87ae18adccc5d34d98e621c818b669d6b49aed5bd4541b
PS C:\Windows\System32>
```

4.最后在 web 端打开工程代码：

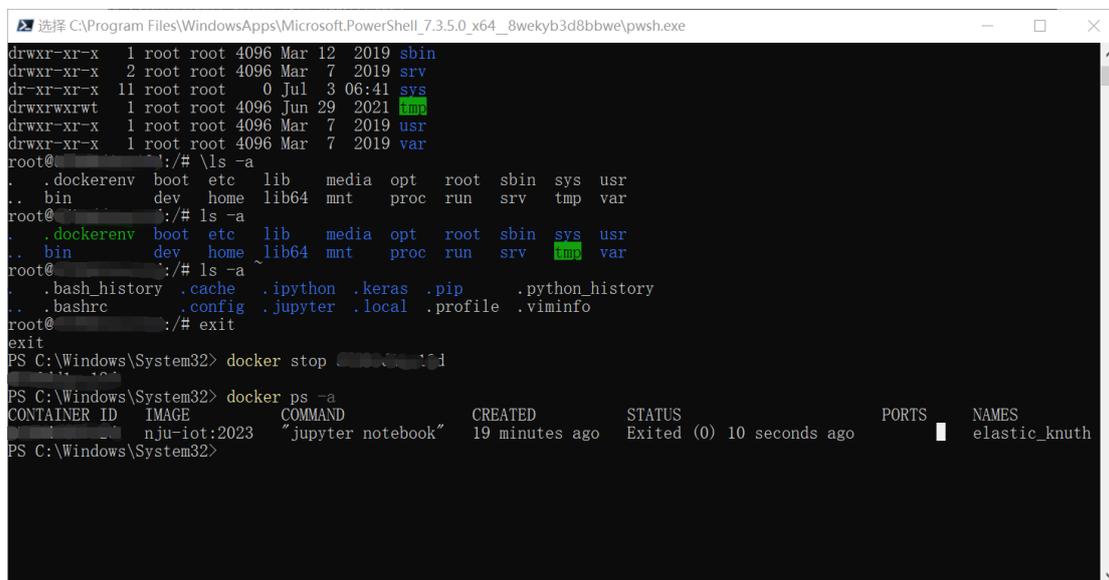
打开浏览器， 输入网址：<https://127.0.0.1:8888/>， 即可看到工程目录。



此外用如下命令可以了解镜像文件的目录结构与文件。

```
PS C:\Windows\System32> docker exec -it [redacted] /bin/bash
root@6726dd1ec12d:/# ls -l
total 68
drwxr-xr-x  2 root root 4096 Mar  7  2019 bin
drwxr-xr-x  2 root root 4096 Apr 24  2018 boot
drwxr-xr-x  5 root root  340 Jul  3  06:41 dev
drwxr-xr-x  1 root root 4096 Jul  3  06:41 etc
drwxr-xr-x  1 root root 4096 Jun 29  2021 home
drwxr-xr-x  1 root root 4096 Apr 22  2019 lib
drwxr-xr-x  2 root root 4096 Mar  7  2019 lib64
drwxr-xr-x  2 root root 4096 Mar  7  2019 media
drwxr-xr-x  2 root root 4096 Mar  7  2019 mnt
drwxr-xr-x  2 root root 4096 Mar  7  2019 opt
dr-xr-xr-x 298 root root  0 Jul  3  06:41 proc
drwx----- 1 root root 4096 Jun 29  2021 root
drwxr-xr-x  1 root root 4096 Mar 12  2019 run
drwxr-xr-x  1 root root 4096 Mar 12  2019 sbin
drwxr-xr-x  2 root root 4096 Mar  7  2019 srv
dr-xr-xr-x 11 root root  0 Jul  3  06:41 sys
drwxrwxrwt  1 root root 4096 Jun 29  2021 tmp
drwxr-xr-x  1 root root 4096 Mar  7  2019 usr
drwxr-xr-x  1 root root 4096 Mar  7  2019 var
root@6726dd1ec12d:/# \
```

docker container ls 可以显示容器的目录，我们可以看到容器的 ID； docker exec 命令表示在运行的容器中执行命令。 docker stop <Container ID> 命令杀死正在运行的容器。



```
选择 C:\Program Files\WindowsApps\Microsoft.PowerShell_7.3.5.0_x64_8wekyb3d8bbwe\pwsh.exe
drwxr-xr-x 1 root root 4096 Mar 12 2019 sbin
drwxr-xr-x 2 root root 4096 Mar 7 2019 srv
dr-xr-xr-x 11 root root 0 Jul 3 06:41 sys
drwxrwxrwt 1 root root 4096 Jun 29 2021 tmp
drwxr-xr-x 1 root root 4096 Mar 7 2019 usr
drwxr-xr-x 1 root root 4096 Mar 7 2019 var
root@elastic_knuth:/# \ls -a
. .dockerenv boot etc lib media opt root sbin sys usr
.. bin dev home lib64 mnt proc run srv tmp var
root@elastic_knuth:/# ls -a
. .dockerenv boot etc lib media opt root sbin sys usr
.. bin dev home lib64 mnt proc run srv tmp var
root@elastic_knuth:/# ls -a
. .bash_history .cache .ipython .keras .pip .python_history
.. .bashrc .config .jupyter .local .profile .viminfo
root@elastic_knuth:/# exit
exit
PS C:\Windows\System32> docker stop elastic_knuth
PS C:\Windows\System32> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS          NAMES
elastic_knuth  nju-iot:2023  "jupyter notebook"     19 minutes ago Exited (0) 10 seconds ago          elastic_knuth
PS C:\Windows\System32>
```

至此，镜像的加载与使用结束。欢迎留言讨论！