**Docker installation guidance**

Author: Zhennong Chen, Xingyi Yao

**Preparation：**

下载Github Desktop app

在<https://github.com/zhennongchen/Standard_dockers.git>上，点击绿色的Code按钮，选择Open with Github Desktop，将它保存在你的电脑本地，推荐路径D盘，这样你就有一个D:\Standard\_dockers

**Step 1: 确认 GPU 驱动**

确认电脑里已经有 NVIDIA 驱动（最新版 Game Ready 或 Studio 都行）。

检查方法：Windows 上打开 NVIDIA 控制面板 或在命令行输入：

nvidia-smi

**Step 2: 安装 Docker Desktop**

谷歌去 Docker 官网

下载 Docker Desktop for Windows。

安装时勾选 Enable WSL2 integration 或者 安装完点击设置🡪Resources🡪WSL integration，勾选Enable integration with my default WSL distro

**Step 3: 安装 WSL2和Ubuntu 22.04**

在Windows的 PowerShell（管理员模式）里运行：

**wsl --install -d Ubuntu-22.04**

会自动安装 WSL2 + Ubuntu 22.04 子系统。

装完后重启电脑。

第一次打开 Ubuntu 会让你设置用户名/密码（务必记住）

👉 这样你就能在电脑里有一个干净的 Ubuntu 环境。

然后去Docker desktop 🡪🡪Resources🡪WSL integration, 应该可以看到多出来了一个Enable integration with additional distros: Ubuntu-22.04，勾选它。

**Step 4: 在Powershell里进入WSL Ubuntu:**

关掉powershell然后重新打开一个新的powershell

输入 wsl，应该出现以下变化（用户名后面跟着@XXXX）



最后，运行 sudo apt-get update

**Step 5: 在WSL里找到我给你的docker folder**

以你的docker folder在D:\Standard\_dockers为例

在WSL里一直cd ..，直到看到/mnt，然后再一路cd到**/mnt/d/Standard\_dockers/docker\_pytorch**

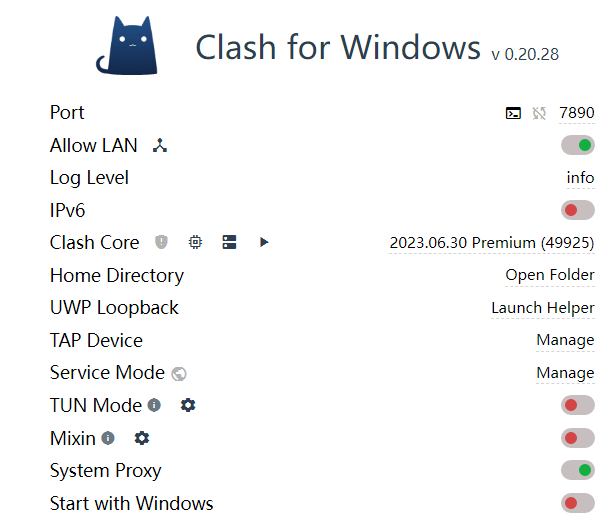
用ls应该能看到以下files

zhennongchen@5YPFKZ3:/mnt/d/Standard\_dockers/docker\_pytorch$ ls

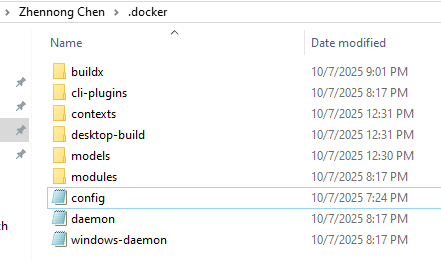
compile.sh docker\_build.sh docker\_run\_compile.sh entrypoint.sh run\_jupyter.sh

docker-compose.yml docker\_run.sh dockerfile requirements.txt

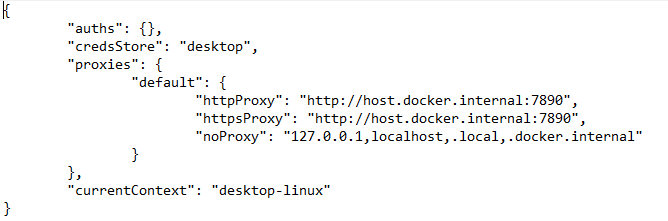
**Step 6: 连接VPN（创建docker image需要从外网下载资料包）**

打开你的VPN，勾选**system proxy（连接VPN）以及Allow LAN**，并且记下你的port number（下面例子里port = 7890）

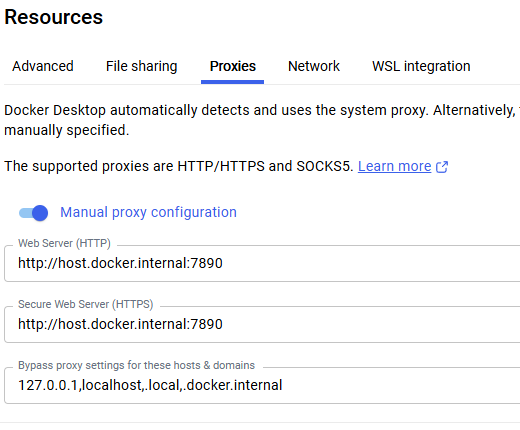
然后在windows的File Explorer里，进入这个路径：C:\Users\Zhennong.Chen\.docker （改成你自己的user name），找到**config file，然后将它用我给你的docker folder里的config file替换**。



**替换后的config应该内容如下**（注意port number要根据你VPN实际的port number来填写，我的例子里port number是7890）



最后到docker desktop的设置🡪Resources🡪Proxies里，勾选**manual proxy configuration，**然后填写以下内容（port number要注意）



**Step 7: 构建docker image**

在文件夹中打开D:/Standard\_dockers/docker\_pytorch/dockerfile，找到第一行有一个FROM

比如FROM pytorch/pytorch:2.0.1-cuda11.7-cudnn8-runtime

然后去wsl里，**输入sudo docker pull pytorch/pytorch:2.0.1-cuda11.7-cudnn8-runtime**

**即sudo docker pull <FROM之后的内容>**

等成功pull之后，再输入**docker build -t pytorch:3.0 .(注意3.0之后是 一个空格+一个点)**

其中**pytorch:3.0就是你给你的image取的名字，可以改成其他的**

等运行结束，在terminal输入sudo docker image ls,看到有一个image的名字是pytorch:3.0即表示docker image安装成功

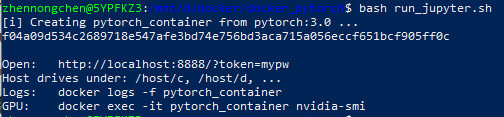


**Step 8：创建docker container**

在wsl里运行：

bash run\_jupyter.sh

出现以下结果则表示安装成功



可以用docker container ls来确认你已经创建好了docker container：

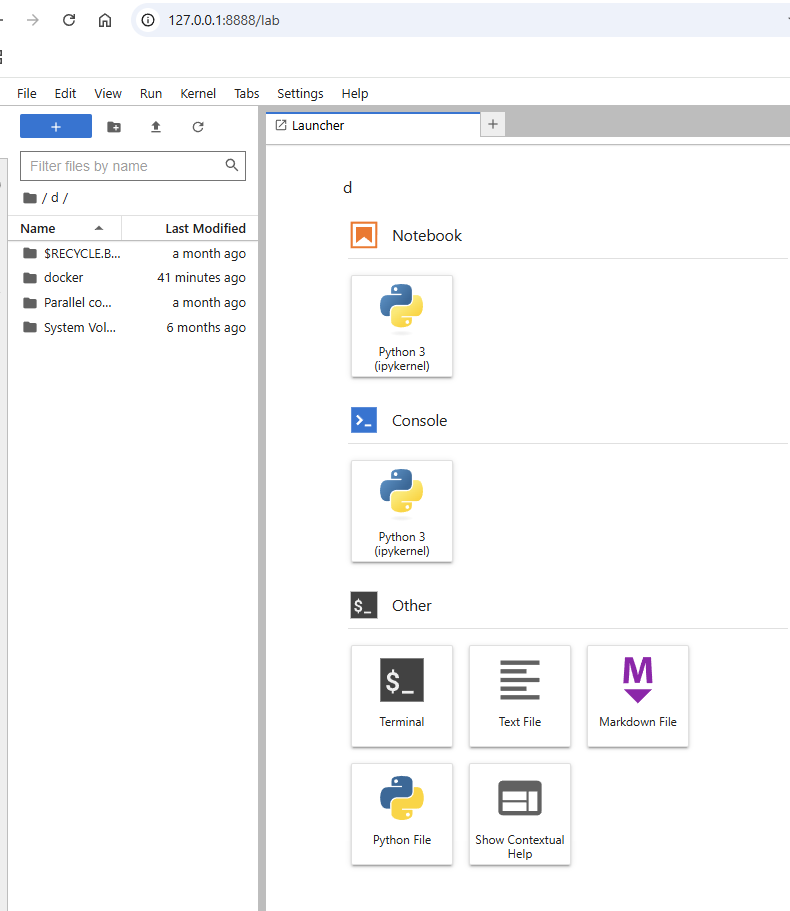


**在上面这个例子里，这个container的ID是f04a09d534c2,名字是pytorch\_container，**名字可以在D:/Standard\_dockers/docker\_pytorch/run\_jupyter.sh里修改

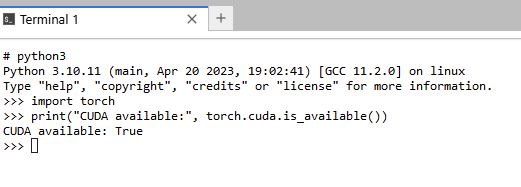
**这个container是可以access to整个电脑本地的数据的，地址是/host/c（C盘）和/host/d（D盘），相当于它是你整台电脑的镜像**

**Step 9：在浏览器里打开这个container**

打开浏览器，**地址栏输入**[**http://127.0.0.1:8888/lab**](http://127.0.0.1:8888/lab)**，密码是mypw**，出现以下界面



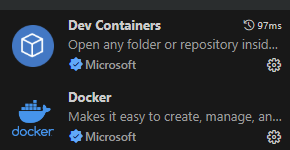
点击terminal，然后在里面尝试以下代码，如果没有报错以及CUDA = True说明一切成功



**Step 10：在VSCode里打开container**

下载VScode桌面版，**这是我们写python代码的地方**。

在左边栏找到Extensions，下载Dev Containers和Docker



下载完之后左边栏会出现一个新图标，点击它就能看到此时我们电脑里存在的docker image以及docker container



在VSCode界面点击Ctrl+Tab+P，然后在出现的下拉菜单里找到**Dev Container: Attach to Running Container**，点击，选择你想要的container后进入

此时会新跳出来一个vscode的窗口，**它的左下角应该是ContainerXXX，表示已进入成功**

在左边栏Open file里打开/host，就会看到我们的C盘和D盘所有文件，说明此时在vscode里我们已经能够同步access和编辑本地的文件

Q&A

**1. 什么时候使用VScode里的container，什么时候使用网页里的container**

VScode是用来写代码的，当你进入vscode container后，你可以直接在左边的文件栏里access/create/delete/edit文件，这就是你写代码的过程。

它的另一个很有用的地方在于可以运行jupyter notebook，对于一些需要可视化结果的代码，可以直接在vscode container里创建.ipynb文件，然后写代码。

网页里的container是用来运行文件的，当你在VScode里写好你的文件后（除了jupyter notebook直接在VScode里运行），你的电脑本地+Vscode container+网页container都已经real-time地更新了这个文件，然后你就可以转去网页里的terminal运行这个文件。

2. **怎么查看，删除docker image**

查看：sudo docker image ls

删除：sudo docker rmi image\_ID

**3. 怎么查看，删除docker container**

查看： sudo docker container ls

删除：先sudo docker container stop container\_ID, 然后再sudo docker container rm container\_ID

**4. 怎么在container里安装additional的python package**

先sudo docker container ls获得当前container的ID

然后在wsl terminal里sudo docker exec -it -u 0 ID bash

然后就可以进入container的root，用pip install安装

安装好后ctrl+A+D退出root