Deepseek本地部署

本文档使用 Windows 11 + NVIDIA RTX 3080 上使用 镜像部署 DeepSeek，但需要 WSL2 + Docker + NVIDIA 容器工具 来确保 GPU 能正确被容器调用。

环境准备

检查 Windows 11 环境

启用 WSL2

在Windows中以管理员权限打开powshell

🔍 验证 WSL2 是否启用

```
PS D:\> wsl -l -v
  NAME                   STATE           VERSION
* docker-desktop         Stopped         2
  Ubuntu                 Stopped         2
  docker-desktop-data    Stopped         2
  Ubuntu-22.04           Stopped         2

# 输出中 VERSION 列应为 2
# 如果为 1，切换为 WSL2：
PS D:\> wsl --set-default-version 2
```

如果未启用WSL2，可手动强制启用WSL2

🛠️**启用所需功能**（管理员模式）：

```
PS D:\> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all
/norestart
PS D:\> dism.exe /online /enable-feature /featurename:Microsoft-Windows-
Subsystem-Linux /all /norestart
```

🔍 启用后检查

```
PS D:\> wsl --status
默认分发: docker-desktop
默认版本: 2
```

🔍 检查 Hyper-V 状态

```
Get-WindowsOptionalFeature -Online | where-Object {$_.FeatureName -like "*Hyper-V*"}

# 如果Hyper-v已启动
FeatureName : Microsoft-Hyper-V-All
State       : Enabled

FeatureName : Microsoft-Hyper-V
State       : Enabled

FeatureName : Microsoft-Hyper-V-Tools-All
State       : Enabled

FeatureName : Microsoft-Hyper-V-Management-PowerShell
State       : Enabled

FeatureName : Microsoft-Hyper-V-Hypervisor
State       : Enabled

FeatureName : Microsoft-Hyper-V-Services
State       : Enabled

FeatureName : Microsoft-Hyper-V-Management-Clients
State       : Enabled

# 说明虚拟化正常
```

补充：

启用 WSL2 会自动启用 Hyper-V（但不是完整的 Hyper-V）

当你启用 **WSL2** 时，**Windows** 会自动启用以下 **核心虚拟化组件**：

- **Hyper-V Hypervisor**（核心虚拟化功能）
- **Virtual Machine Platform**（WSL2所需）
- **Windows Hypervisor Platform**（Docker Desktop所需）

显卡驱动建议

建议安装 NVIDIA Studio Driver (SD)

虽然 **Game Ready Driver (GRD)** 也可以识别 GPU，但在 **深度学习、AI推理** 和 **Docker GPU 加速** 环境下，**Studio Driver (SD)** 提供更高的稳定性和兼容性，尤其是对 **CUDA**、**TensorRT** 和 **PyTorch** 等 AI 框架的支持更完善

更换为 Studio Driver 步骤

卸载现有驱动
打开 **设备管理器** → 显示适配器 → **NVIDIA GeForce RTX 3080** → 卸载。或使用 **DDU工具**（**推荐**，可彻底清除驱动残留）。

下载 Studio Driver

- 访问 **NVIDIA 驱动下载**
- 手动选择：
  - **产品系列**：GeForce RTX 30 Series
  - **型号**：RTX 3080
  - **操作系统**：Windows 11
  - **驱动类型**：**Studio Driver (SD)**
  - 点击 **搜索** → **下载**。

安装驱动
以 **管理员权限** 运行安装程序，选择 **自定义安装** → **执行干净安装**。

验证驱动是否生效
打开 **PowerShell**（管理员权限）：

```
nvidia-smi
```

输出中应显示：

- **Driver Version**：572.xx（Studio）或对应的最新版本。
- **CUDA Version**：大于 **12.x**（确保与 **Docker** 及 **PyTorch** 兼容）。

安装 WSL Ubuntu

默认情况下，WSL 会将 Linux 发行版（如 **Ubuntu**）安装在

```
C:\Users\<你的用户名>\AppData\Local\Packages\
```

注意：为保证C盘空间，建议安装到其他磁盘

在其他磁盘安装 WSL Ubuntu

在其他磁盘安装 WSL Ubuntu

首先，确保没有已安装的 **Ubuntu**。

```
wsl --list --verbose
```

如果存在（如 `Ubuntu-22.04`）：

```
wsl --unregister Ubuntu-22.04
```

创建目标目录

在你希望安装的磁盘（如 **D 盘**）下创建一个目录：

```
mkdir D:\WSL
```

下载 Ubuntu 镜像

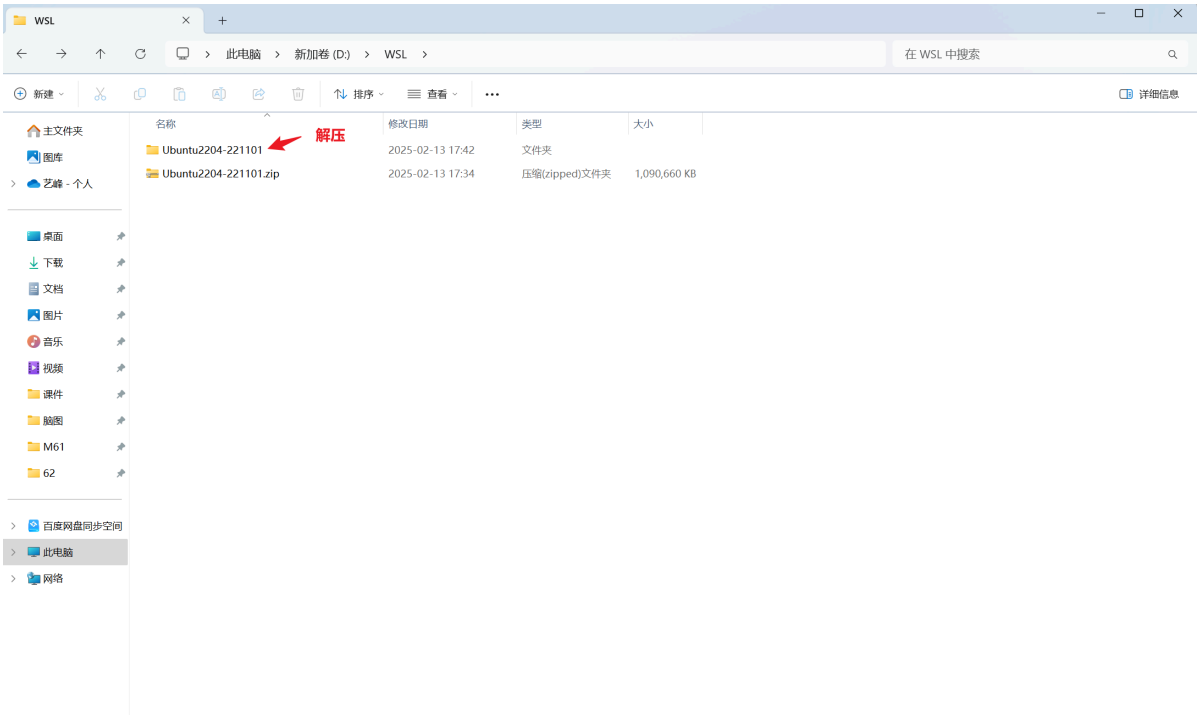
下载链接：

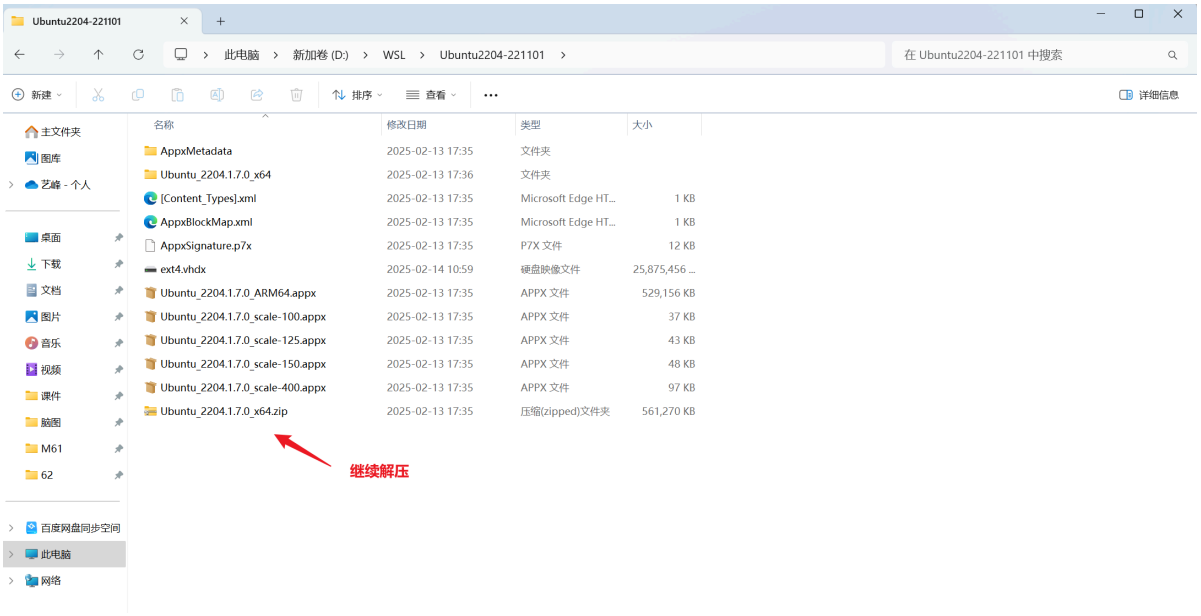
```
https://wslstorestorage.blob.core.windows.net/wslblob/Ubuntu2204-221101.AppxBundle
```

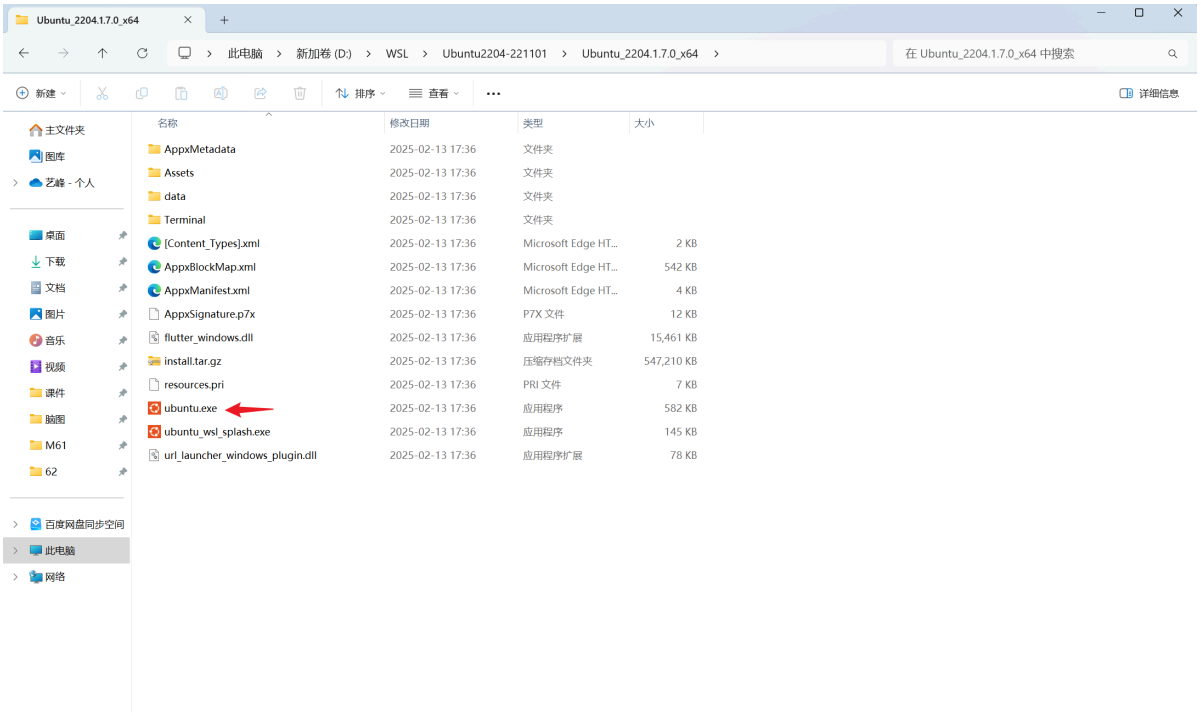
手动安装

将下载的文件重命名为 `.zip` 文件并解压到 `D:\WSL\Ubuntu-22.04`

进入解压目录，找到 `ubuntu.exe` 文件







指定 WSL 安装目录（提前注册）

如果你想要提前在 `D:\WSL\Ubuntu2204-221101` 注册 **Ubuntu**，可以通过以下命令完成

```
wsl --import Ubuntu-22.04 D:\WSL\Ubuntu2204-221101 D:\WSL\Ubuntu2204-221101\Ubuntu_2204.1.7.0_x64\install.tar.gz
```

解释：

- **Ubuntu-22.04** → 在 **WSL** 中的发行版名称。
- **D:\WSL\Ubuntu2204-221101** → 安装目录。
- **D:\WSL\Ubuntu2204-221101\install.tar.gz** → **Ubuntu 根文件系统。**

启动 Ubuntu

安装完成后，启动 **Ubuntu**

```
wsl -d Ubuntu-22.04
```

启用 WSL 的 systemd

WSL 默认不支持 `systemd`，而是使用 `sysvinit` 或 `upstart`。

为方便后续服务管理，可以启用WSL的systemd

编辑 WSL 配置文件

```
vim /etc/wsl.conf
[boot]
systemd=true
```

重启WSL

```
wsl --shutdown
```

重新进入 Ubuntu

```
wsl -d Ubuntu-22.04
```

配置 WSL 代理

如果你的 Windows 代理无法关闭，或者你需要在 WSL 里使用代理，请手动添加代理设置

进入WSL

```
wsl -d Ubuntu-22.04
```

在 WSL 内部配置代理

```
echo 'export http_proxy="http://127.0.0.1:8080"' >> ~/.bashrc
echo 'export https_proxy="http://127.0.0.1:8080"' >> ~/.bashrc
echo 'export no_proxy="127.0.0.1,localhost"' >> ~/.bashrc
source ~/.bashrc
```

重启WSL

```
wsl --shutdown
wsl -d Ubuntu-22.04
```

配置并运行DeepSeek

安装 NVIDIA WSL2 工具

在 **WSL2 的 Ubuntu** 终端中运行：

```
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-keyring_1.0-1_all.deb
sudo dpkg -i cuda-keyring_1.0-1_all.deb
sudo apt update
sudo apt install -y cuda-toolkit-12-2
```

然后 **重启 WSL2**

```
wsl --shutdown
wsl -d Ubuntu-22.04
```

安装 Docker & NVIDIA 容器工具

在 **WSL2 Ubuntu** 中

```
# 安装 Docker
sudo apt update && sudo apt install -y docker.io
sudo usermod -aG docker $USER
newgrp docker

# 安装 NVIDIA Container Toolkit
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
   && curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add
- \
   && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-
docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list
```

```
sudo apt update
sudo apt install -y nvidia-container-toolkit
sudo systemctl restart docker

# 配置Docker代理
```

拉取DeepSeek镜像

```
#docker run -d -p 3030:8080 --gpus all --add-host=host.docker.internal:host-gateway -v open-webui:/app/backend/data  -e OLLAMA_API_BASE_URL=http://host.docker.internal:11434  --name open-webui-test --restart always open-webui-custom:cuda

docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:main

docker run -d --gpus all -p 11434:11434 -v /data/ollama-data/:/root/.ollama --name ollama ollama/ollama
```

注意:

```
ghcr.io/open-webui/open-webui:main这个镜像内的文件建议修改，否则启动很慢
```

傻瓜式部署

1 下载 WSL 整个实例

```
通过网盘分享的文件：ubuntu_backup.tar.gz
链接: https://pan.baidu.com/s/1HpmobQaxhdEyIL1XXxlStQ?pwd=4tcm 提取码: 4tcm
--来自百度网盘超级会员v7的分享

# 下载后解压，后续导入WSL
```

2️⃣ 在目标主机导入 `WSL`

```
# 创建目标目录
mkdir D:\WSL_Migration

# 导入 WSL 实例
wsl --import Ubuntu-22.04 D:\WSL_Migration D:\WSL_Backup\ubuntu_backup.tar

# 参数说明
Ubuntu-22.04：导入的实例名称（可自定义）。
D:\WSL_Migration：指定 WSL 实例的根目录。
D:\WSL_Backup\ubuntu_backup.tar：前面导出的备份文件。

# 启动导入的 Ubuntu
wsl -d Ubuntu-22.04
```