# SoftGym项目安装教程

项目：[星宇-林/软健：软体操是一组用于可变形物体操作的基准环境。 (github.com)](https://github.com/Xingyu-Lin/softgym)

相关：[YunzhuLi/PyFleX：NVIDIA FleX 的定制 Python API (github.com)](https://github.com/YunzhuLi/PyFleX)

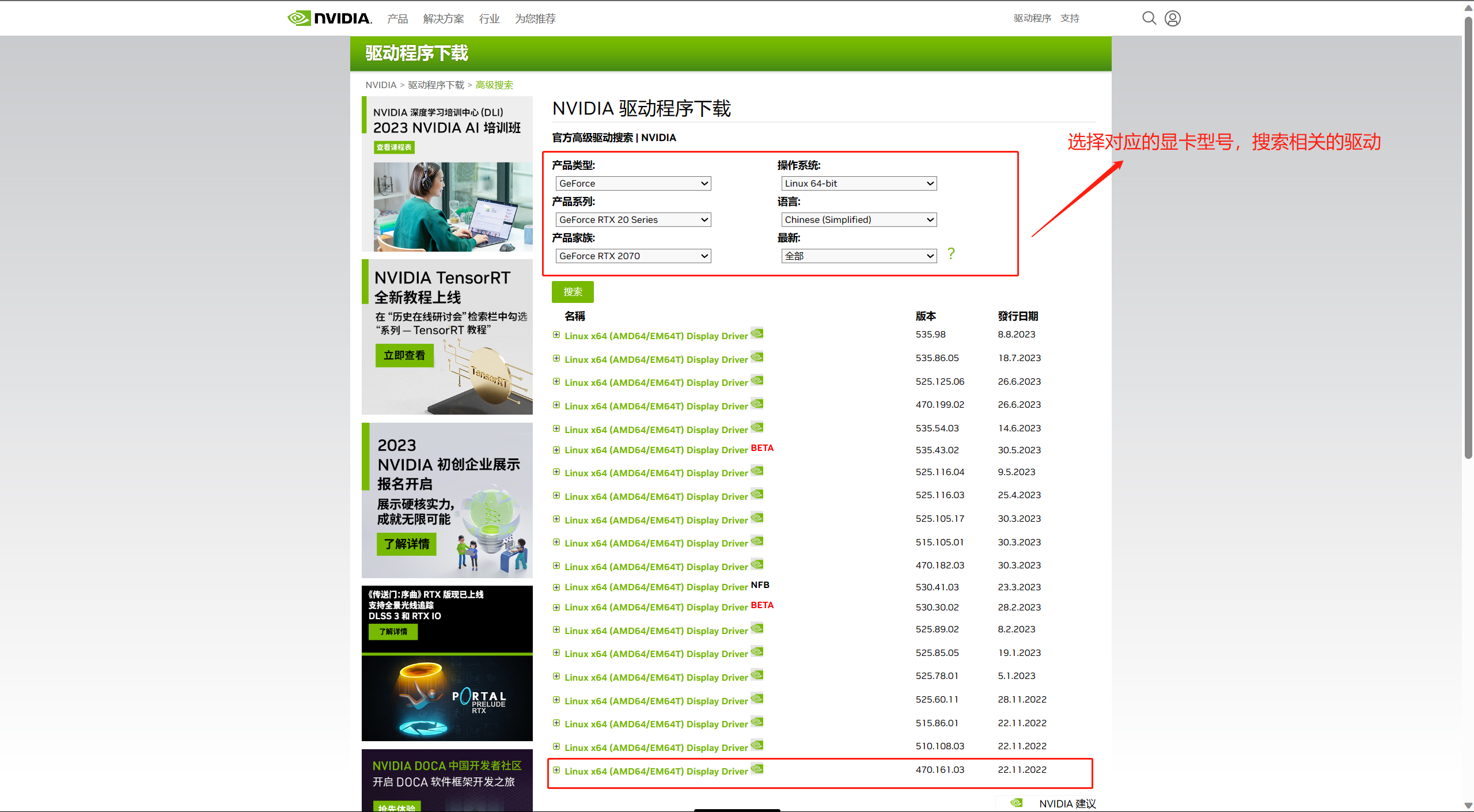
根据：[开始使用 SoftGym 进行可变形对象操作 (danieltakeshi.github.io)](https://danieltakeshi.github.io/2021/02/20/softgym/)

根据：[Install Softgym · GitBook (uc3m.es)](https://robots.uc3m.es/installation-guides/install-softgym.html" \l "install-softgym-ubuntu-1604-cuda-92-nvidia-driver-version-44064)

根据：[我的研究工作流程：Conda、深度学习、CUDA、存储和 SSH (danieltakeshi.github.io)](https://danieltakeshi.github.io/2021/09/05/workflow/)

# 安装Ubuntu18.04.6

## 安装nvidia显卡驱动：[官方高级驱动搜索 | NVIDIA](https://www.nvidia.cn/Download/Find.aspx?lang=cn)

* 1. lspci | grep -i nvidia 或 lspci -vnn | grep VGA （#查看显卡型号）
  2. nvidia-smi 或 lshw -c video （#查看有显卡驱动，有则显示，无则不显示）
  3. 注意：默认的Ubuntu18.04.6是开源的nvidia显卡驱动，我们这里必须要去nvidia官网下载显卡驱动安装，特别注意：安装时留意是否有提示安装OpenGL（一定要安装这个，因为这个项目有用到OpenGL，如果你没有，就算部署成功就会报：OpenGL初始化失败，段错误等问题）
  4. 
     1. 扩展：安装NVIDIA显卡驱动有3个方法分别：软件更新，PAA源安装，runfile安装（这里用runfile安装）
  5. 下载安装
     1. wget <https://www.nvidia.cn/content/DriverDownloads/confirmation.php?url=/XFree86/Linux-x86_64/470.161.03/NVIDIA-Linux-x86_64-470.161.03.run&lang=cn&type=TITAN>
     2. chmod a+x NVIDIA-Linux-x86\_64-470.161.03.run
     3. 打开禁用列表，禁用nouveau，前面说了默认自带的是开源的nvidia显卡驱动（如果不关闭则会于nouweau冲突，安装不成功，安装失败等问题）
     4. vim /etc/modprobe.d/blacklist.conf
     5. 在最后一行添加：（添加完成后保存退出）
        1. blacklist nouveau
        2. options nouveau modeset=0
     6. 更新重启
        1. sudo update-initramfs -u
        2. reboot
     7. 再次查看是否禁用nouveau（没有任何输出则表示禁用成功）
        1. lsmod | grep nouveau
     8. init 2
     9. ./NVIDIA-Linux-x86\_64-470.161.03.run
     10. 输入完成后根据提示安装步骤，一步一步走下去
     11. 安装完成后，检查是否安装显卡驱动成功
         1. nvidia-smi

## 安装CUDA

* 1. 下载连接[CUDA Toolkit 10.0 Download | NVIDIA Developer --- CUDA 工具包 10.0 下载 |英伟达开发者](https://developer.nvidia.com/cuda-10.0-download-archive)
     1. 下载命令
        1. wget <https://developer.nvidia.com/compute/cuda/10.0/Prod/local_installers/cuda_10.0.130_410.48_linux>
        2. sudo sh cuda\_10.0.130\_410.48\_linux.run
           1. 可以设置环境变量，用sudo find / -name nvcc 走到后，添加到.bashrc 里面export PATH=$PATH:/usr/local/cuda-xx.x/bin/nvcc

cuda-xx.x 表示你的cuda版本，例如cuda10.0

## 下载conda并安装

* 1. 下载链接：<https://repo.anaconda.com/miniconda/>
  2. 我这里下载的是Miniconda3-latest-Linux-x86\_64.sh
     1. 下载好了后给执行权限，然后安装
        1. chmod a+x Miniconda3-latest-Linux-x86\_64.sh
        2. ./ Miniconda3-latest-Linux-x86\_64.sh
           1. 设置国内源（下载速度快）

在家目录执行一下内容

conda config --remove-key channels

vim ~/.condarc

看这里面[Linux安装Miniconda和换源 - Oraer - 博客园 (cnblogs.com)](https://www.cnblogs.com/Oraer/p/17431614.html)

## 安装docker-ce（下面的docker-ce官方提供的一键脚本安装）

* 1. 参考docker文档链接：[Install Docker Engine on Ubuntu | Docker Documentation --- 在 Ubuntu 上安装 Docker 引擎 |码头工人文档](https://docs.docker.com/engine/install/ubuntu/)
  2. curl https://get.docker.com | sh **\**

&& sudo systemctl --now enable docker

* 1. sudo systemctl start docker
  2. 测试是否安装docker成功
  3. sudo docker run hello-world

## 安装nvidia-container-toolkit（前身nvida-docker）

* 1. 参考nvidia文档链接：[Installation Guide — container-toolkit 1.13.5 documentation --- 安装指南 — 容器工具包 1.13.5 文档 (nvidia.com)](https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html)
  2. 设置软件包存储库和 GPG 密钥：
     1. distribution=$(. /etc/os-release;echo $ID$VERSION\_ID) \

&& curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \

&& curl -s -L https://nvidia.github.io/libnvidia-container/experimental/$distribution/libnvidia-container.list | \

sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | \

sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list

* 1. 更新包列表
     1. sudo apt-get update
  2. 安装nvidia-container-toolkit包（依赖项）
     1. sudo apt-get install -y nvidia-container-toolkit
  3. 配置 Docker 守护程序以识别 NVIDIA 容器运行时：
     1. sudo nvidia-ctk runtime configure --runtime=docker
  4. 在设置默认运行时后重新启动 Docker 守护程序以完成安装：
     1. sudo systemctl restart docker
  5. 可以通过运行基本 CUDA 容器来测试工作设置
  6. sudo docker run --rm --runtime=nvidia --gpus all nvidia/cuda:11.6.2-base-ubuntu20.04 nvidia-smi
  7. 安装nvidia-docker2
     1. sudo apt install nvidia-docker2
     2. （会提示是否覆盖，输入Y即可）

## 克隆softgym项目到本地以及安装依赖

* 1. sudo apt-get install build-essential libgl1-mesa-dev freeglut3-dev libglfw3 libgles2-mesa-dev
  2. git clone <https://github.com/Xingyu-Lin/softgym.git>

## 创建conda以及通过哦nvidia-docker2编译安装softgym

* 1. cd softgym
  2. conda env create -f environment.yml
  3. conda activate softgym
  4. conda install pybind11
  5. docker pull xingyu/softgym
  6. docker images
  7. 启动容器
     1. nvidia-docker run \

-v /home/seita/softgym:/workspace/softgym \

-v /home/seita/miniconda3:/home/seita/miniconda3 \

-v /tmp/.X11-unix:/tmp/.X11-unix \

--gpus all \

-e DISPLAY=$DISPLAY \

-e QT\_X11\_NO\_MITSHM=1 \

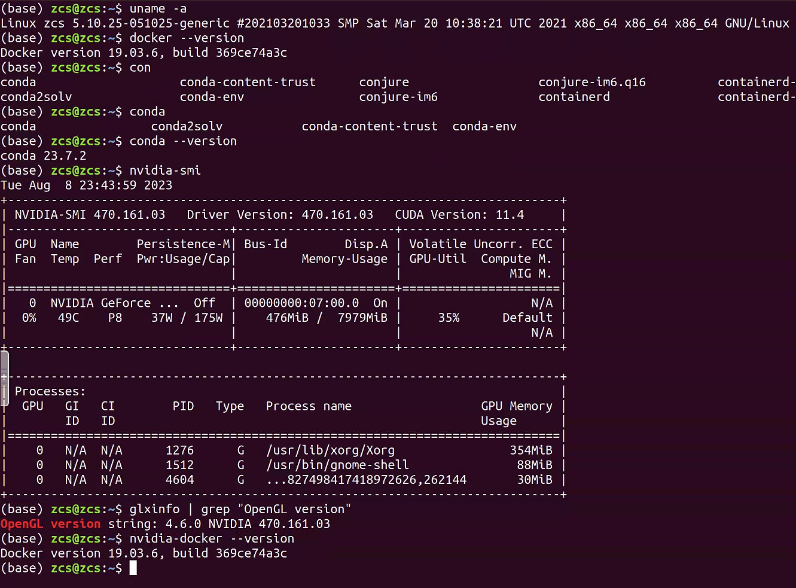
-it xingyu/softgym:latest bash

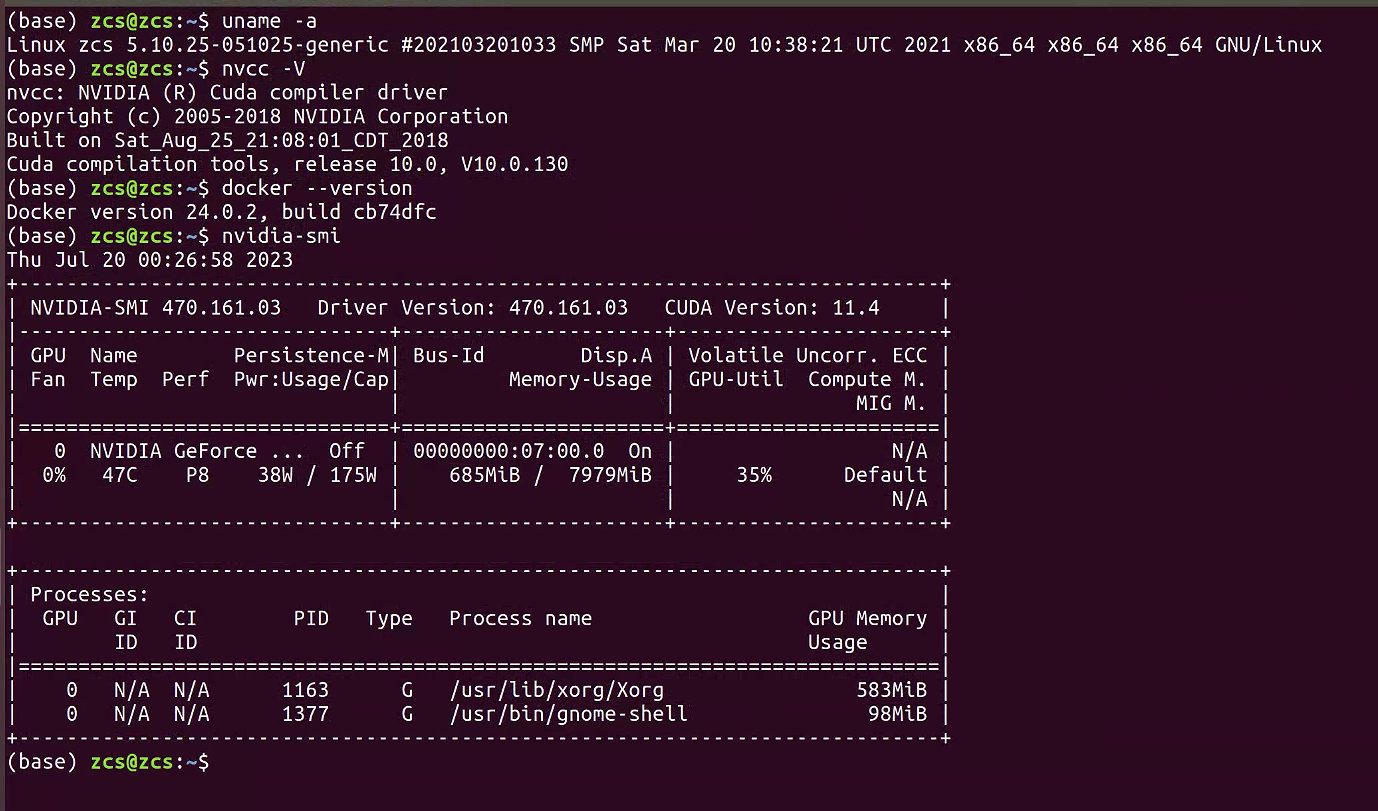
* + 1. 注意这里是seita得是原机器的目录路径
    2. cd softgym/
    3. export PATH="/home/seita/miniconda3/bin:$PATH"
    4. . ./prepare\_1.0.sh
    5. . ./compile\_1.0.sh
    6. 等待编译安装好

## 测试softgym

* 1. conda activate softgym
  2. export PYFLEXROOT=${PWD}/PyFlex
  3. export PYTHONPATH=${PYFLEXROOT}/bindings/build:$PYTHONPATH
  4. export LD\_LIBRARY\_PATH=${PYFLEXROOT}/external/SDL2-2.0.4/lib/x64:$LD\_LIBRARY\_PATH
  5. python examples/random\_env.py --env\_name PassWater

## 系统信息相关





## 扩展

1. nvidia-container-toolkit是一个软件包，用于在Docker容器内设置和配置GPU环境。它提供了Docker运行时所需的组件，以便在容器内部访问主机上的GPU设备。使用nvidia-container-toolkit，你可以确保容器中的应用程序可以利用GPU进行计算加速。

nvidia-docker是nvidia-container-toolkit的一个旧版本，它提供了一组命令行工具，用于管理和运行支持GPU的Docker容器。它通过在Docker容器中挂载NVIDIA驱动和相关的库文件，使得容器内的应用程序能够和主机上的GPU进行通信。

这两个工具的功能相似，但nvidia-container-toolkit是nvidia-docker的继任者，并添加了一些新功能和改进。因此，建议在使用时优先选择nvidia-container-toolkit。

但是因为，你要部署的项目是使用nvidia-docker，所以我才安装了这个包

1. NVIDIA显卡驱动里面都会自带安装OpenGL（一定确保OpenGL安装成功）在安装显卡驱动的时候会一同安装
2. 重点就是
   1. nvidia驱动（核心）
   2. opengl（核心，运行softgym项目的时候需要的）
   3. pybind11（编译安装softgym的时候需要的）
   4. nvidia-docekr（编译安装的时候需要）
   5. conda（整个softgym项目需要的虚拟环境）
   6. cuda（核心）