1. 更新最新的Nvidia驱动

```
# 检查机器驱动建议
ubuntu-drivers devices

# 装12.0驱动
sudo apt install nvidia-driver-525

# 重启
sudo reboot
```

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            On   | 00000000:00:08.0 Off |                    0 |
| N/A   38C    P8     9W /  70W |      2MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

2. Docker的安装官方版本

https://docs.docker.com/engine/install/ubuntu/

Set Up

```
# 删掉之前的docker
sudo apt-get remove docker docker-engine docker.io containerd runc

# Update the apt package index and install packages to allow apt to use a
repository over HTTPS:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg

# Add Docker's official GPG key:
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

```
# Use the following command to set up the repository:
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Install Docker Engine

```
# Update the apt package index:
sudo apt-get update

# To install the latest version, run:
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

# Verify that the Docker Engine installation is successful by running the hello-
world image.
sudo docker run hello-world
```

安装nvidia cuda tookit

```
# 安装nvidia tookit
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/$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

sudo apt-get update
sudo apt-get install -y nvidia-container-toolkit

# 安装完toolkit要重启
sudo systemctl restart docker
```

4. 把docker加入User Group(选做)

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

加入了之后重启了才能使用

```
sudo reboot
```

5. 查看docker 挂载路径的目录

```
docker info | grep 'Docker Root Dir'
```

`Docker Root Dir: /data/docker`
我的docker数据的挂载就是在/data/docker下面的

6. 更改挂载的目录(选做)

```
# 查看路径
docker info | grep 'Docker Root Dir'
# 更改文件
sudo mkdir /data/docker
sudo vim /etc/docker/daemon.json


# 添加
{
    "data-root": "/data/docker"
}

# 重启docker
sudo systemctl restart docker
```

7. 拉取Pytorch训练的镜像

```
docker run --gpus all -it --name env_pyt_1.12 -v $(pwd):/app
nvcr.io/nvidia/pytorch:22.03-py3
```

8. 拉取TensorRT的镜像

```
docker run --gpus all -it --name env_trt -v $(pwd):/app
nvcr.io/nvidia/tensorrt:22.08-py3
```

9. 拉取DeepStream的镜像

```
docker run --gpus all -v `pwd`:/app -p 8556:8554  --name deepstream_env -it
nvcr.io/nvidia/deepstream:6.1.1-devel bash
```

10. 打包镜像上传到DockerHub

登陆自己的账号

```
docker login --username easonbob
```

标记自己的镜像

```
docker tag nvcr.io/nvidia/pytorch:22.03-py3 easonbob/my_torch1-pytorch:22.03-py3

docker tag nvcr.io/nvidia/tensorrt:22.08-py3 easonbob/my_trt-tensorrt:22.08-py3
```

推送自己的镜像

```
docker push easonbob/my_torch1-pytorch:22.03-py3
docker push easonbob/my_trt-tensorrt:22.08-py3
```

12. 在当前目录下拉取自己打包好的镜像

13. 更改自己的镜像

1. 提交容器更改：首先，您需要使用 docker commit 命令将更改后的容器提交为新的镜像。将 container-id 替换为您要提交的容器 ID，将 new-image-name 替换为新的镜像名称。

```
docker commit container-id new-image-name
```

2. 标记新的镜像：使用 docker tag 命令为新的镜像添加标签，以便将其推送到 Docker Hub。将 new-image-name 替换为您在上一步中使用的名称，将 your-username 和 your-repo-name 替换为您的 Docker Hub 用户名和仓库名称。

```
docker tag new-image-name your-username/your-repo-name:new-image-tag
```

3. 推送新的镜像：使用 docker push 命令将新的镜像推送到 Docker Hub。这将覆盖之前的镜像。

```
docker push your-username/your-repo-name:new-image-tag
```

当您将新的镜像推送到 Docker Hub 时，它将覆盖具有相同标签的现有镜像。请注意，如果您希望保留旧版本的镜像，可以为新的镜像使用不同的标签。