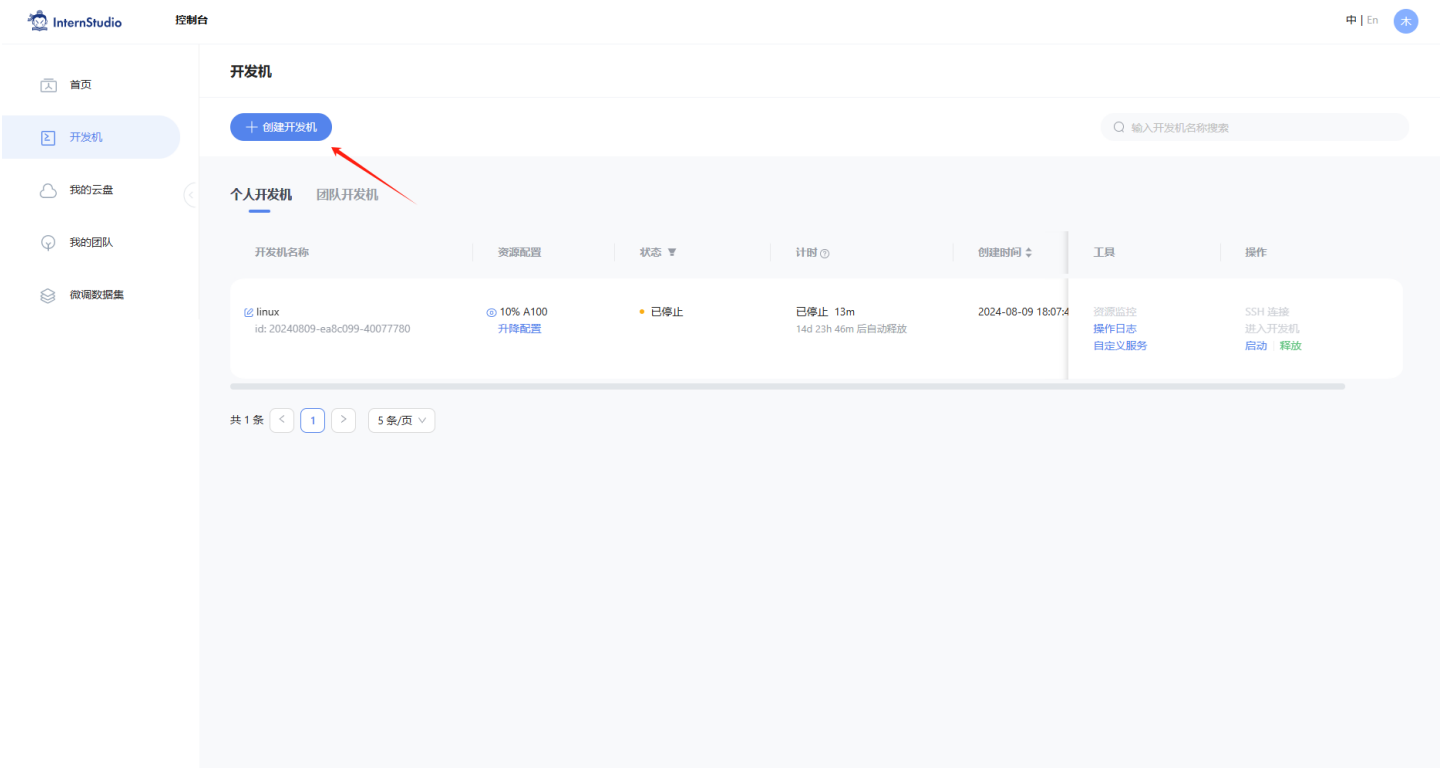
基础岛任务笔记2-8G显存玩转书生大模型Demo

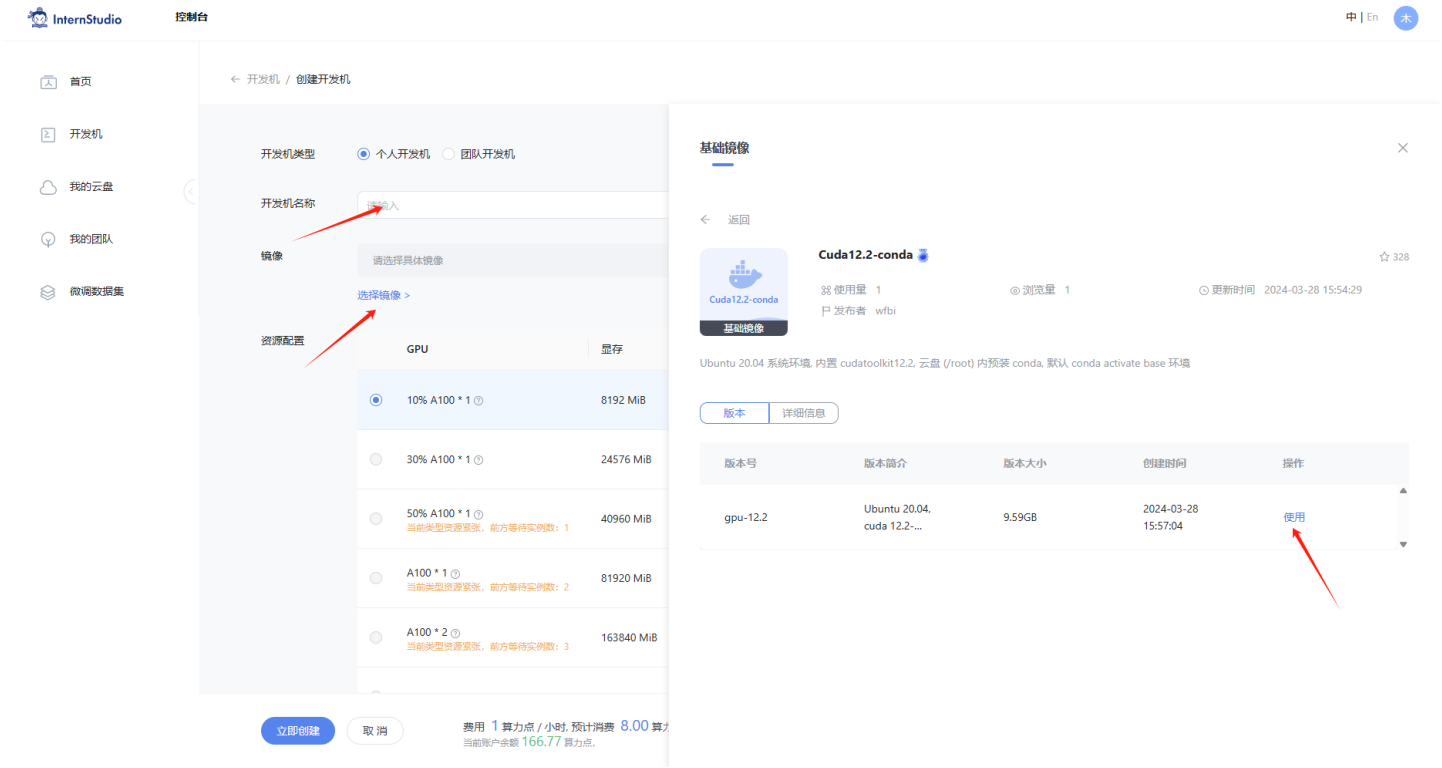
8G 显存玩转书生大模型 Demo
记录复现过程并截图
基础任务（完成此任务即完成闯关）
使用 Cli Demo 完成 InternLM2-Chat-1.8B 模型的部署，并生成 300 字小故事，记录复现过程并截图。
进阶任务（闯关不要求完成此任务）
使用 LMDeploy 完成 InternLM-XComposer2-VL-1.8B 的部署，并完成一次图文理解对话，记录复现过程并截图。
使用 LMDeploy 完成 InternVL2-2B 的部署，并完成一次图文理解对话，记录复现过程并截图。
闯关材料提交（完成任务并且提交材料视为闯关成功）
闯关作业总共分为一个任务，一个任务完成视作闯关成功。
请将作业发布到知乎、CSDN等任一社交媒体，将作业链接提交到以下问卷，助教老师批改后将获得 100 算力点奖励！！！
提交地址：https://aicarrier.feishu.cn/share/base/form/shrcnZ4bQ4YmhEtMtnKxZUcf1vd

创建开发机

在输入开发机名称后，点击创建开发机



选择 10% 的开发机，镜像选择为 Cuda-12.2

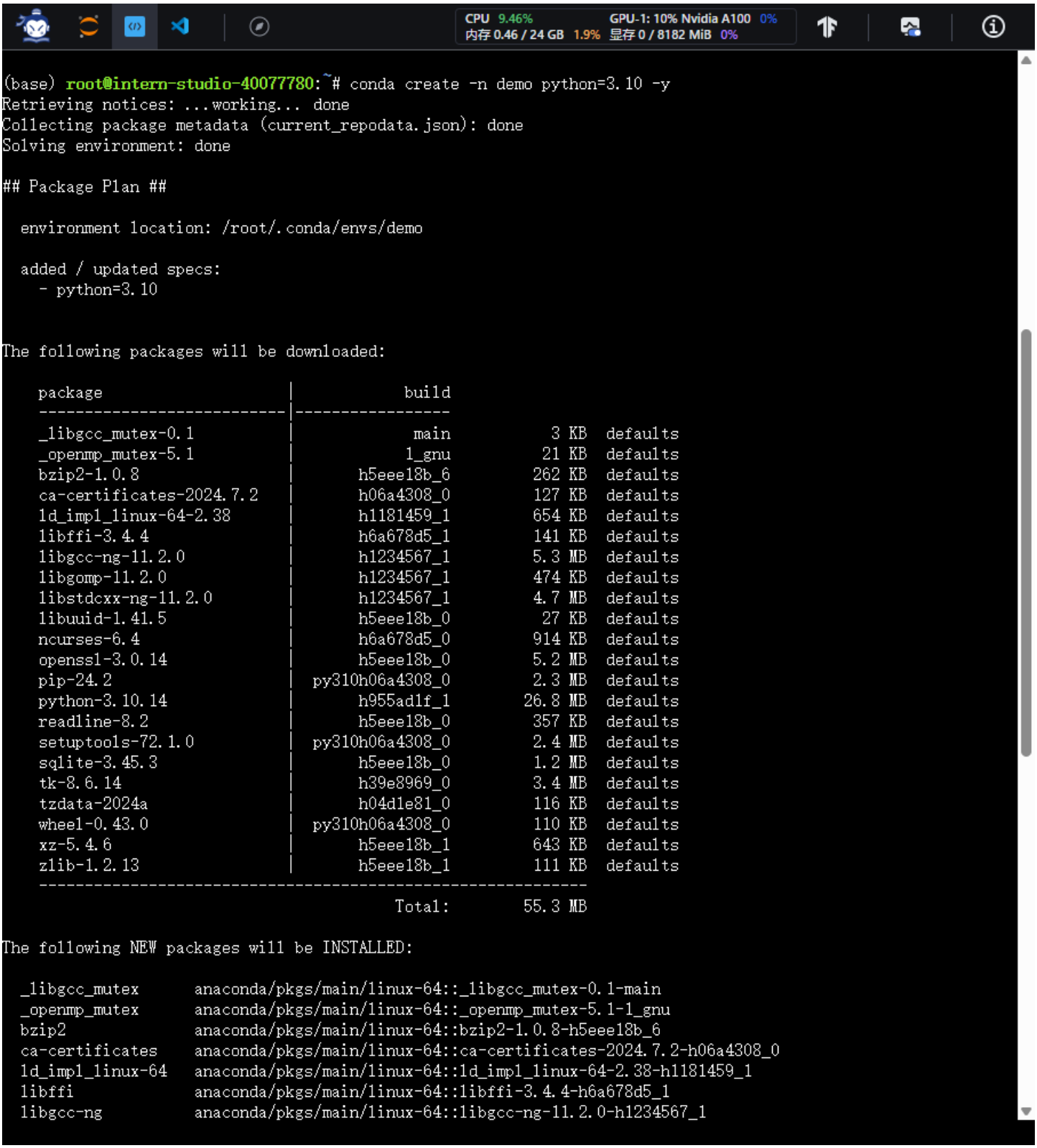


环境配置

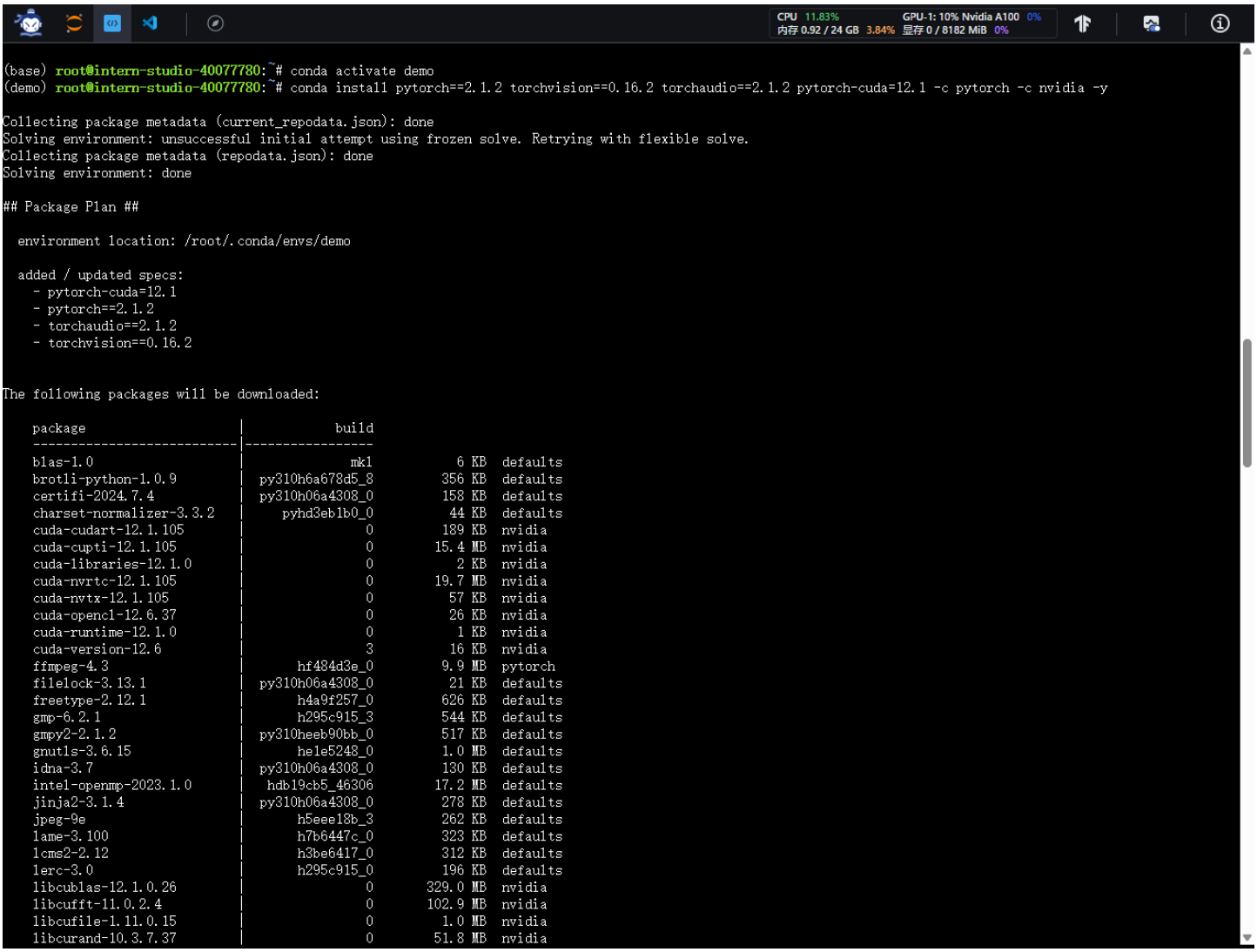
我们首先来为 Demo 创建一个可用的环境。

```
# 创建环境
conda create -n demo python=3.10 -y
# 激活环境
conda activate demo
# 安装 torch
conda install pytorch==2.1.2 torchvision==0.16.2 torchaudio==2.1.2 pytorch-cuda=12.1 -c pytorch -c nvidia -y
# 安装其他依赖
pip install transformers==4.38
pip install sentencepiece==0.1.99
pip install einops==0.8.0
pip install protobuf==5.27.2
pip install accelerate==0.33.0
pip install streamlit==1.37.0
```

创建一个新环境，名字为demo：



激活demo环境并且安装torch及其他依赖，时间有点长


CPU 11.83%
内存 0.92 / 24 GB 3.84%
GPU-1: 10% Nvidia A100 0%
显存 0 / 8182 MiB 0%
(base) root@intern-studio-40077780:~# conda activate demo
(demo) root@intern-studio-40077780:~# conda install pytorch==2.1.2 torchvision==0.16.2 torchaudio==2.1.2 pytorch-cuda=12.1 -c pytorch -c nvidia -y
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: done
Package Plan
environment location: /root/.conda/envs/demo
added / updated specs:
- pytorch-cuda=12.1
- pytorch==2.1.2
- torchaudio==2.1.2
- torchvision==0.16.2
The following packages will be downloaded:
package | build
blas-1.0 | mk1 6 KB defaults
brotli-python-1.0.9 | py310h6a678d5_8 356 KB defaults
certifi-2024.7.4 | py310h06a4308_0 158 KB defaults
charset-normalizer-3.3.2 | pyhd3eb1b0_0 44 KB defaults
cuda-cudart-12.1.105 | 0 189 KB nvidia
cuda-cupti-12.1.105 | 0 15.4 MB nvidia
cuda-libraries-12.1.0 | 0 2 KB nvidia
cuda-nvrtc-12.1.105 | 0 19.7 MB nvidia
cuda-nvtx-12.1.105 | 0 57 KB nvidia
cuda-opencl-12.6.37 | 0 26 KB nvidia
cuda-runtime-12.1.0 | 0 1 KB nvidia
cuda-version-12.6 | 3 16 KB nvidia
ffmpeg-4.3 | hf484d3e_0 9.9 MB pytorch
filelock-3.13.1 | py310h06a4308_0 21 KB defaults
freetype-2.12.1 | h4a9f257_0 626 KB defaults
gmp-6.2.1 | h295c915_3 544 KB defaults
gmpy2-2.1.2 | py310heeb90bb_0 517 KB defaults
gnutls-3.6.15 | he1e5248_0 1.0 MB defaults
idna-3.7 | py310h06a4308_0 130 KB defaults
intel-openmp-2023.1.0 | hdb19cb5_46306 17.2 MB defaults
jinja2-3.1.4 | py310h06a4308_0 278 KB defaults
jpeg-9e | h5eee18b_3 262 KB defaults
lame-3.100 | h7b6447c_0 323 KB defaults
lcms2-2.12 | h3be6417_0 312 KB defaults
lerc-3.0 | h295c915_0 196 KB defaults
libcublas-12.1.0.26 | 0 329.0 MB nvidia
libcufft-11.0.2.4 | 0 102.9 MB nvidia
libcufile-1.11.0.15 | 0 1.0 MB nvidia
libcurand-10.3.7.37 | 0 51.8 MB nvidia

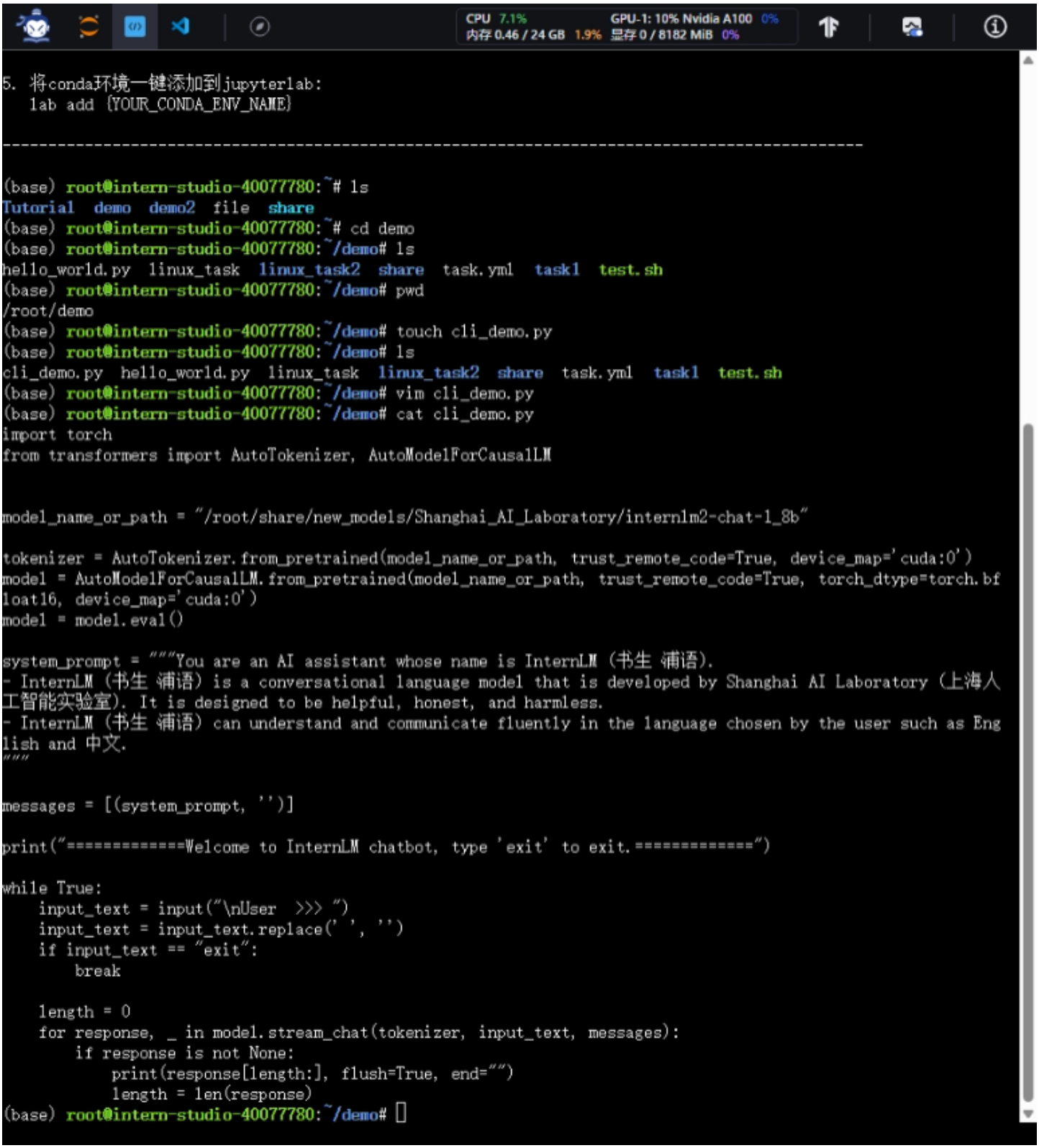
Cli Demo 部署 InternLM2-Chat-1.8B 模型

第一步：在之前已经创建的demo文件夹中创建一个 `cli_demo.py` 。

```
mkdir -p /root/demo
touch /root/demo/cli_demo.py
```

第二步：利用vim编辑 `cli_demo.py` 文件,也可将下面的代码直接复制到 `cli_demo.py` 中。

```
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM


model_name_or_path = "/root/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-1_8b"

tokenizer = AutoTokenizer.from_pretrained(model_name_or_path, trust_remote_code=True, device_map='cuda:0')
model = AutoModelForCausalLM.from_pretrained(model_name_or_path, trust_remote_code=True, torch_dtype=torch.bfloat16, device_map='cuda:0')
model = model.eval()

system_prompt = """You are an AI assistant whose name is InternLM (书生·浦语).
- InternLM (书生·浦语) is a conversational language model that is developed by Shanghai AI Laboratory (上海人工智能实验室). It is designed to be helpful, honest, and harmless.
- InternLM (书生·浦语) can understand and communicate fluently in the language chosen by the user such as English and 中文.
"""

messages = [(system_prompt, '')]

print("=============Welcome to InternLM chatbot, type 'exit' to exit.=============")

while True:
    input_text = input("\nUser  >>> ")
    input_text = input_text.replace(' ', '')
    if input_text == "exit":
        break

    length = 0
    for response, _ in model.stream_chat(tokenizer, input_text, messages):
        if response is not None:
            print(response[length:], flush=True, end="")
            length = len(response)
```



最后一步：通过 `python /root/demo/cli_demo.py` 启动 Demo（注意环境用“demo”虚拟环境）。问了俩问题，小模型有点不理解问题很正常。

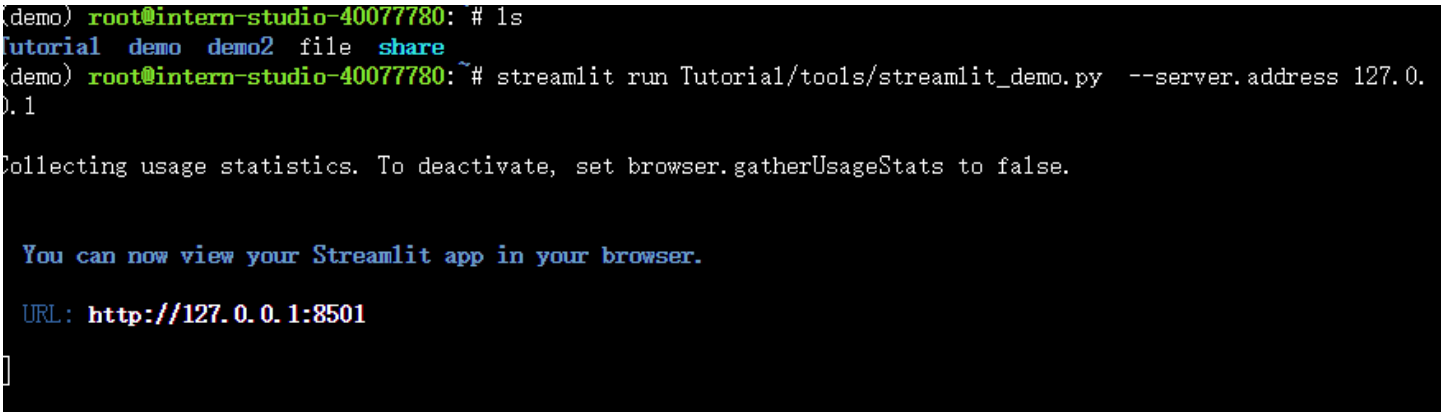
Streamlit Web Demo 部署 InternLM2-Chat-1.8B 模型

首先把教程仓库 clone 到本地（以执行后续的代码）

```
cd /root/demo
git clone https://github.com/InternLM/Tutorial.git
```

然后，启动一个 Streamlit 服务。

```
cd /root/demo
streamlit run /root/demo/Tutorial/tools/streamlit_demo.py --server.address 127.0.0.1 --server.port 6006
```



接下来，在**本地**的 PowerShell 中输入以下命令，将端口映射到本地。

`ssh -CNg -L 6006:127.0.0.1:6006 root@ssh.intern-ai.org.cn -p 你的 ssh 端口号`

然后将 SSH 密码复制并粘贴到 PowerShell 中，回车，即可完成端口映射。

在完成端口映射后，通过浏览器访问 `http://localhost:6006` 来启动 Demo。