

How to setup Cuda or Cudnn 22.04

- Một số đường link cài để kham khảo
 - [Link github refer](#)
 - [How to Install CUDA on Ubuntu 22.04 | Step-by-Step](#)
 - Version CUDA should install : [CUDA Compatibility](#)
 - Check **Capacity of GPU** each with individual GPU : [link](#)
 - Check phiên bản **Python, CUDA, cuDNN** tương thích [link](#)
 - Matrix phiên bản hỗ trợ nhau [link](#)

Cài đặt Cuda và Cudnn

- Đầu tiên phải check phiên bản **Ubuntu** tiếp theo đó là phiên bản **python** và **tensorflow** cũng như **cuda** và **cudnn** sẽ tương thích với nhau
 - Hợp lý nhất là Ubuntu phiên bản 22.04 tiếp với đó là python phiên bản 3.11 thay vì phiên bản mới nhất. Vì các lý do sau đây
 - Check phiên bản python, cuda, cudnn tương thích [link](#)

GPU

Phiên bản	Phiên bản Python	Trình biên dịch	Xây dựng công cụ	cuDNN	CUDA
tensorflow-2.19.0	3,9-3,12	Tiếng kêu 18.1.8	Bazel 6.5.0	9,3	12,5
tensorflow-2.18.0	3,9-3,12	Tiếng kêu 17.0.6	Bazel 6.5.0	9,3	12,5
tensorflow-2.17.0	3,9-3,12	Tiếng kêu 17.0.6	Bazel 6.5.0	8,9	12,3
tensorflow-2.16.1	3,9-3,12	Tiếng kêu 17.0.6	Bazel 6.5.0	8,9	12,3
tensorflow-2.15.0	3,9-3,11	Tiếng vang 16.0.0	Bazel 6.1.0	8,9	12,2
tensorflow-2.14.0	3,9-3,11	Tiếng vang 16.0.0	Bazel 6.1.0	8,7	11,8
tensorflow-2.13.0	3,8-3,11	Tiếng vang 16.0.0	Bazel 5.3.0	8,6	11,8
tensorflow-2.12.0	3,8-3,11	GCC 9.3.1	Bazel 5.3.0	8,6	11,8
tensorflow-2.11.0	3,7-3,10	GCC 9.3.1	Bazel 5.3.0	8,1	11,2
tensorflow-2.10.0	3,7-3,10	GCC 9.3.1	Bazel 5.1.1	8,1	11,2
tensorflow-2.9.0	3,7-3,10	GCC 9.3.1	Bazel 5.0.0	8,1	11,2
tensorflow-2.8.0	3,7-3,10	GCC 7.3.1	Bazel 4.2.1	8,1	11,2
tensorflow-2.7.0	3,7-3,9	GCC 7.3.1	Bazel 3.7.2	8,1	11,2
tensorflow-2.6.0	3,6-3,9	GCC 7.3.1	Bazel 3.7.2	8,1	11,2
tensorflow-2.5.0	3,6-3,9	GCC 7.3.1	Bazel 3.7.2	8,1	11,2
tensorflow-2.4.0	3,6-3,8	GCC 7.3.1	Bazel 3.1.0	8,0	11,0

- Ta sẽ ưu tiên dùng tensor 2.15 vì vậy ta sẽ dùng **Cuda 12.2** và **Cudnn 8.9**
- Trước khi muốn cài GPU

- Tắt **Secure Boot**

✓ 1. Tắt **Secure Boot** (trong BIOS/UEFI)

📌 Cách làm:

1. Khởi động lại máy
2. Khi máy đang boot, nhấn phím vào BIOS/UEFI thường là **F2**, **DEL**, **ESC**, **F12** (tùy máy)
3. Tìm đến mục: **Security** hoặc **Boot**
4. Tìm dòng **Secure Boot** → Đặt thành: **Disabled**
5. Save và khởi động lại máy

🔥 Nếu *Secure Boot* đang bật, driver *NVIDIA* sẽ không load được kernel module (*nvidia.ko*) → *nvidia-smi* báo lỗi hoặc không hiển thị GPU.

Để khởi động máy tính tự động khi có nguồn điện mà không cần nhấn nút Power (On/Off), bạn cần thay đổi thiết lập trong BIOS/UEFI của hệ thống. Cụ thể, bạn sẽ cần kích hoạt tính năng gọi là "Restore on AC Power Loss" hoặc "AC Back" (tên có thể khác tùy vào từng nhà sản xuất mainboard).

📄 Các bước thực hiện:

1. Khởi động lại máy tính và vào BIOS/UEFI bằng cách nhấn một trong các phím sau trong khi máy đang khởi động:
 - **F2**, **DEL**, **ESC**, **F12**, hoặc **F10** (tùy theo nhà sản xuất mainboard)
2. Trong BIOS/UEFI, tìm đến mục **Power Management** hoặc **Advanced** (có thể khác tùy dòng máy hoặc mainboard).
3. Tìm mục có tên là:
 - **Restore on AC Power Loss** hoặc **AC Back**
 - **Power on after power failure**
 - **After Power Loss**
4. Chọn chế độ phù hợp:
 - **Power On**: Khi có nguồn điện, máy sẽ tự động bật mà không cần nhấn nút.
 - **Last State**: Máy sẽ bật lại với trạng thái giống như khi tắt máy trước đó (nếu tắt trước đó thì nó sẽ bật, nếu đang mở thì nó vẫn mở).
 - **Off**: Máy sẽ không tự bật khi có nguồn điện (chế độ mặc định).
5. Lưu lại cấu hình và thoát khỏi BIOS: Thường thì bạn sẽ nhấn **F10** hoặc chọn **Save & Exit** để lưu thay đổi và khởi động lại.

Một số lưu ý:

- Tính năng này chỉ hoạt động khi máy tính của bạn có nguồn điện và nguồn điện cung cấp là ổn định. Nếu điện bị ngắt quãng hoặc bị tắt trong thời gian dài, bạn cần bật lại nguồn bằng tay (hoặc dùng thiết bị UPS để ổn định nguồn).
- Một số máy tính (đặc biệt là những máy không phải server hoặc workstation) có thể không hỗ trợ tính năng này hoặc không có tùy chọn trong BIOS.

Kiểm tra lại sau khi khởi động:

```
bash
```

[Sao chép](#)[Chỉnh sửa](#)

```
mokutil --sb-state
```

- Nếu trả về: `SecureBoot disabled` → bạn đã thành công

• Tắt Nouveau

```
sudo nano /etc/modprobe.d/blacklist-nouveau.conf
```

– Rồi thêm lệnh sau

```
blacklist nouveau
```

```
options nouveau modeset=0
```

– Chạy lệnh

```
sudo update-initramfs -u # lệnh này để xác nhận đã có sự thay đổi  
sudo reboot
```

– Kiểm tra xem đã vô hiệu hóa được chưa dùng lệnh

```
lsmod | grep nouveau
```

• Các lệnh sẽ là

- Cập nhật

```
sudo apt update && sudo apt upgrade
```

- Remove previous NVIDIA, cuda, cudnn installation

```
sudo apt autoremove nvidia* --purge
```

```
sudo apt purge -y 'nvidia-*'
```

```
sudo apt autoremove -y
```

```
sudo apt-get --purge remove '*cuda*'
sudo apt-get --purge remove '*cudnn*'
```

- List the recommended NVIDIA drivers

```
sudo apt install ubuntu-drivers-common
sudo ubuntu-drivers devices
```

- Xem nó recommended thẳng nào

```
sudo apt install nvidia-driver-535
```

- Reboot lại hệ thống tiếp đến
- Để check driver đã được cài

```
nvidia-smi
```

- Update tiếp (lặp lại đoạn trên cùng)

```
sudo apt update && sudo apt upgrade
```

- Install GCC

```
sudo apt install gcc
gcc -v
```

- Tải Cuda từ trang chủ và mình sẽ tải version 22.04 [Link](#)

```
wget
```

```
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-ubuntu2204.pin
```

```
sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-pin-600
```

```
wget
```

```
https://developer.download.nvidia.com/compute/cuda/12.2.2/local_installers/cuda-repo-ubuntu2204-12-2-local_12.2.2-535.104.05-1_amd64.deb
```

```
sudo dpkg -i cuda-repo-ubuntu2204-12-2-local_12.2.2-535.104.05-1_amd64.deb
```

```
sudo cp /var/cuda-repo-ubuntu2204-12-2-local/cuda-*-keyring.gpg
/usr/share/keyrings/
```

```
sudo apt-get update
```

```
sudo apt-get -y install cuda
```

- Tải phiên bản Cudnn 8.9 về rồi nhập lệnh

```
sudo apt install ./<filename.deb>
sudo cp /var/cudnn-<something>.gpg /usr/share/keyrings/
```

- Reboot lại
- Environment setup với

```
nano ~/.bashrc
```

– Và nhập

```
export PATH=/usr/local/cuda/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-12.2/lib64\
    ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}} ## không
nên sai thẳng này mà nên sai thẳng bên dưới và đúngn 100% rồi
export LD_LIBRARY_PATH=/usr/local/cuda-
12/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}# nhớ cho nó thẳng hàng
```

– After:

```
. ~/.bashrc
```

- Test Cuda toolkit `nvcc -V` or `nvcc --version`
- My cuDNN version is 8, adapt the following to your version

```
sudo apt update
sudo apt install libcudnn8
sudo apt install libcudnn8-dev
sudo apt install libcudnn8-samples
```

- Một số mã cần nhớ
 - 1
 - 123456789
 - viethung

Để gỡ hoàn toàn CUDA và cuDNN trên Ubuntu (hoặc các hệ Debian-based)

- Gỡ Cuda
 - Xác định các gói Cuda đã cài đặt

```
dpkg -l | grep cuda
```

- Gỡ toàn bộ Cuda

```
sudo apt-get --purge remove '*cuda*'
```

- Xác định các gói Cudnn đã cài đặt

```
dpkg -l | grep cudnn
```

- Gỡ toàn bộ Cudnn

```
sudo apt-get --purge remove '*cudnn*'
```

• XÓA CÁC THU' MỤC LIÊN QUAN

- Dọn sạch sẽ

```
sudo rm -rf /usr/local/cuda*  
sudo rm -rf /usr/include/cudnn*  
sudo rm -rf /usr/lib/x86_64-linux-gnu/libcudnn*
```

- Kiểm tra thêm

```
ls /usr/local/ | grep cuda
```

- Nếu còn bất kì thu mục cuda nào xóa nốt

```
sudo rm -rf /usr/local/cuda
```

• LÀM SẠCH PATH & LD_LIBRARY_PATH (nếu bạn đã từng thêm vào)

- Mở file .bashrc hoặc .zshrc

```
nano ~/.bashrc
```

- Tìm và xóa những dòng có chứa

```
export PATH=/usr/local/cuda-.../bin:$PATH  
export LD_LIBRARY_PATH=/usr/local/cuda-.../lib64:$LD_LIBRARY_PATH
```

- Rồi run lệnh sau để xác nhận

```
source ~/.bashrc
```