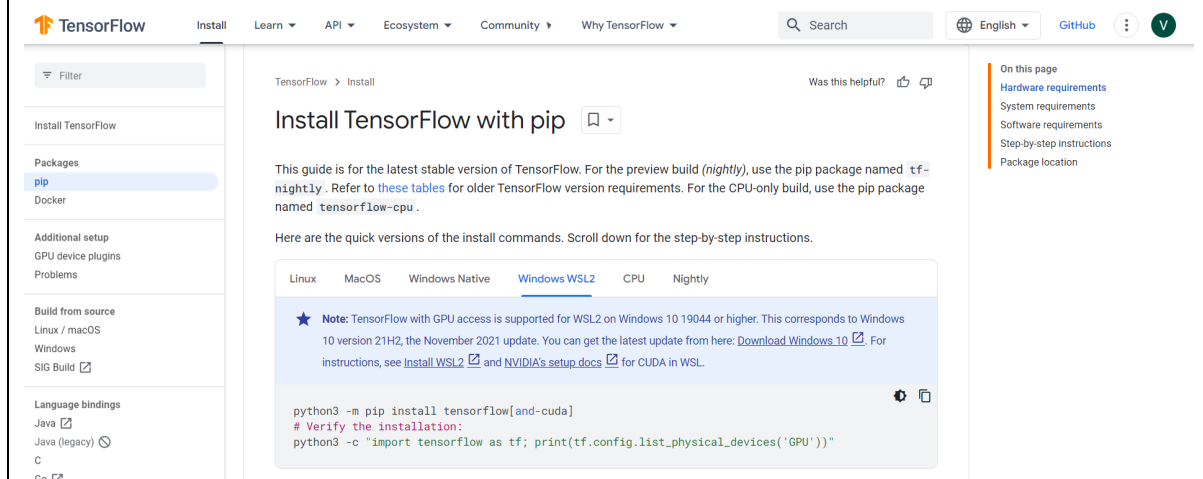


Tensorflow-GPU for Ubuntu

Version requirement:

- [WSL2 or Ubuntu 22.04 LTS](#)
- [Nvidia driver](#)
- [python 3.11.^](#)
 - This only affects the TensorFlow versions available to you. In this guide, we are downloading TensorFlow 2.14.0.
- [TensorFlow 2.14.0](#)
 - Any $\geq 2.10.1$ will work, sionna requires at least 2.10.1
- [cuda11.8](#)
- [cudnn8](#)

!! If you are following TensorFlow's guide "[Install TensorFlow with pip](#)", please disregard that guide and follow the steps in this document instead.



WSL2 / Ubuntu 22.04

**Note that ubuntu version 20.04 will also work, but remember to change to 2004 version when download Cuda and Cudnn in below sections.*

The easiest way is to download Ubuntu 22.04 LTS from Microsoft store (<https://www.microsoft.com/store/productId/9PN20MSR04DW?ocid=pdpshare>).

Alternatively, follow the guide from Microsoft to download WSL2 (<https://learn.microsoft.com/en-us/windows/wsl/install>).

Python3

Download Python version 3.11.^ (<https://www.python.org/downloads/>).

- `sudo apt install python3.11` if ubuntu 22.04/20.04
- If you want TensorFlow version other than 2.14.0, please refer to TensorFlow website for required Python version (<https://www.tensorflow.org/install/source>)

GPU

Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow-2.16.1	3.9-3.12	Clang 17.0.6	Bazel 6.5.0	8.9	12.3
tensorflow-2.15.0	3.9-3.11	Clang 16.0.0	Bazel 6.1.0	8.9	12.2
tensorflow-2.14.0	3.9-3.11	Clang 16.0.0	Bazel 6.1.0	8.7	11.8
tensorflow-2.13.0	3.8-3.11	Clang 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

Nvidia driver

Pre-Installation

```
echo blacklist nouveau | sudo tee -a /etc/modprobe.d/nouveau-kms.conf
echo options nouveau modeset=0 | sudo tee -a /etc/modprobe.d/nouveau-kms.conf
sudo update-initramfs -u

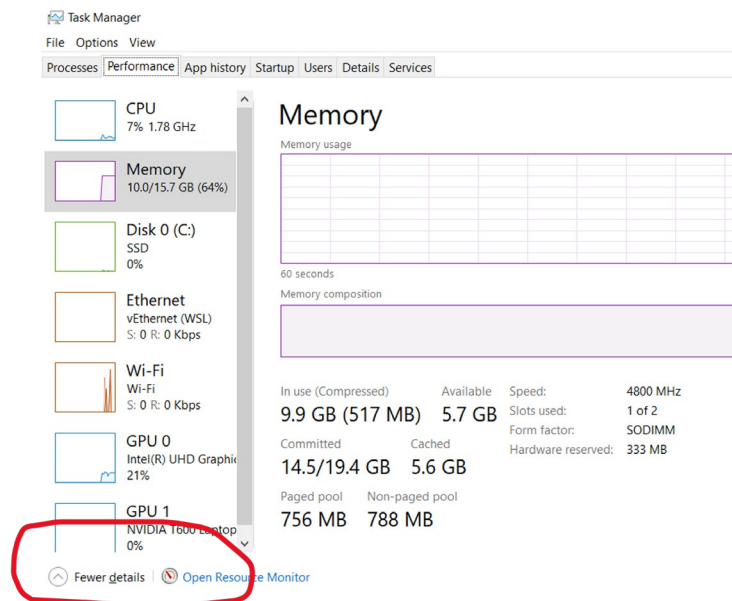
sudo reboot

# to register driver in case of kernel change, we use dkms
sudo apt-get install dkms
```

- (Updated 1/2025) This is tested on Ubuntu 22.04/20.04, if you have the same spec, this is the fastest way to download Nvidia driver (**choose compatible version with your kernel!!**)

```
wget https://us.download.nvidia.com/tesla/520.61.05/NVIDIA-Linux-x86_64-520.61.05.run
sudo sh NVIDIA-Linux-x86_64-520.61.05.run --dkms
```

- Download driver from Nvidia (<https://www.nvidia.com/download/index.aspx>)
 - This depends on your system specification, quickest way to check your graphics card is from task manager - performance



- Test Nvidia driver is successfully installed. Note that `nvidia-smi` is showing the highest CUDA version that the driver can support, it is not the CUDA version that you installed.

```
nvidia-smi
dkms status
```

```
vivian@LAPTOP-91QIKE7S:/mnt/c/Users/Vivian Ha$ nvidia-smi
Fri Mar 1 15:03:35 2024
```

NVIDIA-SMI 535.133				Driver Version: 537.79		CUDA Version: 12.2		
GPU	Name	Perf	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC
Fan	Temp		Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute	M. MIG M.
0	NVIDIA T600 Laptop GPU	P0	7W / 30W	00000000:01:00.0	off	0%	Default	N/A
N/A	0C			0MiB / 4096MiB				N/A

Processes:							GPU Memory Usage
GPU	GI ID	CI ID	PID	Type	Process name		
No running processes found							

Cuda11.8

- (Updated 1/2025) Current fastest way is to run the CUDA toolkit file provided below. Deselect the Nvidia 520.61.05 driver option in the menu.

```
wget
https://developer.download.nvidia.com/compute/cuda/11.8.0/local_installers/cuda_11.8.0_520.61.05_linux.run
sudo sh cuda_11.8.0_520.61.05_linux.run
```

- Install Cuda11.8. If the commands below do not work, refer to alternative in Appendix A.

```
# if you have different ubuntu version like 20.04, just change 2204
to 2004
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/11.8.0/local_inst
allers/cuda-repo-ubuntu2204-11-8-local_11.8.0-520.61.05-1_amd64.deb

sudo dpkg -i cuda-repo-ubuntu2204-11-8-local_11.8.0-520.61.05-
1_amd64.deb
sudo cp /var/cuda-repo-ubuntu2204-11-8-local/cuda-*-keyring.gpg
/usr/share/keyrings/

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

***Post-Installation**

```
# setup your paths, these two needed to be added to ~/.bashrc
export PATH=/usr/local/cuda-11.8/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-11.8/lib64:$LD_LIBRARY_PATH
# check path
echo $PATH
sudo ldconfig
```

Cudnn8

- (Updated 1/2025) cudnn8.7.0 archived make file

```
CUDNN_TAR_FILE="cudnn-linux-x86_64-8.7.0.84_cuda11-archive.tar.xz"
sudo wget
https://developer.download.nvidia.com/compute/redist/cudnn/v8.7.0/1
ocal_installers/11.8/cudnn-linux-x86_64-8.7.0.84_cuda11-
archive.tar.xz
sudo tar -xvf ${CUDNN_TAR_FILE}
sudo mv cudnn-linux-x86_64-8.7.0.84_cuda11-archive cuda
```

- Install Cudnn8. If the commands do not work, refer to alternative in Appendix A.

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

***Post-Installation**

```
sudo cp -P cuda/include/cudnn.h /usr/local/cuda-11.8/include
sudo cp -P cuda/lib/libcudnn* /usr/local/cuda-11.8/lib64/
sudo chmod a+r /usr/local/cuda-11.8/lib64/libcudnn*
```

TensorFlow

```
# any version >=2.10.1 is compatible with sionna, but check your cuda
version and compatible TensorFlow version
pip3 install tensorflow==2.14.0

# test if GPU is detected
python -c "import tensorflow as tf;
print(tf.config.list_physical_devices('GPU'))"
```

Appendix

A. Alternative way to download CUDA and CUDNN

```
# add Nvidia CUDA repository to Ubuntu22.04
# if you have different ubuntu version like 20.04, just change 2204 to
2004
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_6
4/cuda-ubuntu2204.pin
sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-pin-
600
sudo apt-key adv --fetch-keys
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_6
4/3bf863cc.pub
sudo add-apt-repository "deb
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_6
4/ /"

# use following command to download specific version
sudo apt-get install libcudnn8=${cudnn_version}-1+${cuda_version}
sudo apt-get install libcudnn8-dev=${cudnn_version}-1+${cuda_version}

# example: for cuda11.8 and cudnn8.9.7
sudo apt-get install libcudnn8=8.9.7.29-1+cuda11.8
sudo apt-get install libcudnn8-dev=8.9.7.29-1+cuda11.8
```

- 3bf863cc.pub is the known public key, this is the fastest method
 - If does not work, flow Nvidia guide here to add the key:
<https://forums.developer.nvidia.com/t/notice-cuda-linux-repository-key-rotation/212772>

B. TensorFlow official guide issues

If you are following TensorFlow's guide "[Install TensorFlow with pip](#)", the software requirements listed are the same:

The following NVIDIA® software are only required for GPU support.

- [NVIDIA® GPU drivers](#) [🔗](#) version 450.80.02 or higher.
- [CUDA® Toolkit 11.8](#) [🔗](#).
- [cuDNN SDK 8.6.0](#) [🔗](#).
- (Optional) [TensorRT](#) [🔗](#) to improve latency and throughput for inference.

!! Note:

- `cuDNN SDK 8.6.0` link will direct to the latest version of cuDNN on Nvidia website. For consistency, please follow the [Cudnn](#) section to download `cuda11`.
- `python3 -m pip install tensorflow[and-cuda]` will download the latest `cuda12`. For consistency, please follow the [Cuda](#) section to download `cuda11`.

Install TensorFlow with pip

This guide is for the latest stable version of TensorFlow. For the preview build (*nightly*), use the pip package named `tf-nightly`. Refer to [these tables](#) for older TensorFlow version requirements. For the CPU-only build, use the pip package named `tensorflow-cpu`.

Here are the quick versions of the install commands. Scroll down for the step-by-step instructions.

Linux macOS Windows Native Windows WSL2 CPU Nightly

★ **Note:** TensorFlow with GPU access is supported for WSL2 on Windows 10 19044 or higher. This corresponds to Windows 10 version 21H2, the November 2021 update. You can get the latest update from here: [Download Windows 10](#). For instructions, see [Install WSL2](#) and [NVIDIA's setup docs](#) for CUDA in WSL.

```
python3 -m pip install tensorflow[and-cuda]
# Verify the installation:
python3 -c "import tensorflow as tf; print(tf.config.list_physical_devices('GPU'))"
```