

Install TensorFlow with pip

TensorFlow 2 packages are available

- `tensorflow` —Latest stable release with CPU and GPU support (*Ubuntu and Windows*)
- `tf-nightly` —Preview build (*unstable*). Ubuntu and Windows include GPU support.

Older versions of TensorFlow

For TensorFlow 1.x, CPU and GPU packages are separate:

- `tensorflow==1.15` —Release for CPU-only
- `tensorflow-gpu==1.15` —Release with GPU support (*Ubuntu and Windows*)

System requirements

- Python 3.5–3.8
 - Python 3.8 support requires TensorFlow 2.2 or later.
- pip 19.0 or later (requires `manylinux2010` support)
- Ubuntu 16.04 or later (64-bit)
- macOS 10.12.6 (Sierra) or later (64-bit) (*no GPU support*)
- Windows 7 or later (64-bit)
 - Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019
- Raspbian 9.0 or later
- GPU support requires a CUDA®-enabled card (*Ubuntu and Windows*)

Note: Installing TensorFlow 2 requires a newer version of [pip](#).

Hardware requirements

- Starting with TensorFlow 1.6, binaries use [AVX instructions](#) which may not run on older CPUs.
- Read the [GPU support guide](#) to set up a CUDA®-enabled GPU card on Ubuntu or Windows.

1. Install the Python development environment on your system

Check if your Python environment is already configured:

Requires Python 3.5–3.8, pip and venv >= 19.0

```
python3 --version  
pip3 --version
```

If these packages are already installed, skip to the next step. Otherwise, install [Python](#), the [pip package manager](#), and [venv](#):

[Ubuntu](#)[macOS](#)[Windows](#)[Raspberry Pi](#)[Other](#)

```
sudo apt update  
sudo apt install python3-dev python3-pip python3-venv
```

Caution: Upgrading the system `pip` can cause [problems](#).

If *not* in a virtual environment, use `python3 -m pip` for the commands below. This ensures that you upgrade and use the *Python pip* instead of the *system pip*.

2. Create a virtual environment (recommended)

Python virtual environments are used to isolate package installation from the system.

[Ubuntu](#) / [macOS](#) [Windows](#) [Conda](#)

Create a new virtual environment by choosing a Python interpreter and making a `./venv` directory to hold it:

```
python3 -m venv --system-site-packages ./venv
```

Activate the virtual environment using a shell-specific command:

```
source ./venv /bin/activate # sh, bash, or zsh  
./venv /bin/activate.fish # fish  
source ./venv /bin/activate.csh # csh or tcsh
```

When the virtual environment is active, your shell prompt is prefixed with `(venv)`.

Install packages within a virtual environment without affecting the host system setup. Start by upgrading pip:

```
pip install --upgrade pip
```

```
pip list # show packages installed within the virtual envir
```

And to exit the virtual environment later:

```
deactivate # don't exit until you're done using TensorFlow
```

3. Install the TensorFlow pip package

Choose one of the following TensorFlow packages to install [from PyPI](#):

- `tensorflow` —Latest stable release with CPU and GPU support (*Ubuntu and Windows*).
- `tf-nightly` —Preview build (*unstable*). Ubuntu and Windows include GPU support.
- `tensorflow==1.15` —The final version of TensorFlow 1.x.

Package dependencies are automatically installed. These are listed in the [setup.py](#) file under **REQUIRED_PACKAGES**.

[Virtual environment install](#)[System install](#)

```
pip install --upgrade tensorflow
```

Verify the install:

```
python -c "import tensorflow as tf;print(tf.reduce_sum(tf.rand
```

Success: TensorFlow is now installed. Read the [tutorials](#) to get started.

Package location

A few installation mechanisms require the URL of the TensorFlow Python package. The value you specify depends on your Python version.

Version	URL
Linux	
Python 3.5 GPU support	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-2.3.0-cp35-cp35m-manylinux2010_x86_64.whl
Python 3.5 CPU-only	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow_cpu-2.3.0-cp35-cp35m-manylinux2010_x86_64.whl

Python 3.6	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_2.3.0-cp36-cp36m-manylinux2010_x86_64.whl
GPU support	
Python 3.6	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow_2.3.0-cp36-cp36m-manylinux2010_x86_64.whl
CPU-only	
Python 3.7	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_2.3.0-cp37-cp37m-manylinux2010_x86_64.whl
GPU support	
Python 3.7	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow_2.3.0-cp37-cp37m-manylinux2010_x86_64.whl
CPU-only	
Python 3.8	https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_2.3.0-cp38-cp38-manylinux2010_x86_64.whl
GPU support	
Python 3.8	https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow_2.3.0-cp38-cp38-manylinux2010_x86_64.whl
CPU-only	
macOS (CPU-only)	
Python 3.5	https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-2.3.0-cp35-cp35m-macosx_10_6_intel.whl
Python 3.6	https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-2.3.0-cp36-cp36m-macosx_10_9_x86_64.whl
Python 3.7	https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-2.3.0-cp37-cp37m-macosx_10_9_x86_64.whl
Python 3.8	https://storage.googleapis.com/tensorflow/mac/cpu/tensorflow-2.3.0-cp38-cp38-macosx_10_14_x86_64.whl
Windows	
Python 3.5	https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_2.3.0-cp35-cp35m-win_amd64.whl
GPU support	
Python 3.5	https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow_2.3.0-cp35-cp35m-win_amd64.whl
CPU-only	

Python 3.6	https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow-2.3.0-cp36-cp36m-win_amd64.whl
Python 3.6	https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-2.3.0-cp36-cp36m-win_amd64.whl
Python 3.7	https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow-2.3.0-cp37-cp37m-win_amd64.whl
Python 3.7	https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-2.3.0-cp37-cp37m-win_amd64.whl
Python 3.8	https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow-2.3.0-cp38-cp38-win_amd64.whl
Python 3.8	https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-2.3.0-cp38-cp38-win_amd64.whl

Raspberry PI (CPU-only)

Python 3, Pi0 or Pi1	https://storage.googleapis.com/tensorflow/raspberrypi/tensorflow-cp35-none-linux_armv6l.whl
Python 3, Pi2 or Pi3	https://storage.googleapis.com/tensorflow/raspberrypi/tensorflow-cp35-none-linux_armv7l.whl

