

How to Install Tensorflow on Ubuntu 22.04

Learn how to install TensorFlow on Ubuntu 22.04 with our step-by-step guide. Set up your deep learning environment quickly and efficiently for your AI projects.

Contents

01	Introduction	3
02	Install TensorFlow on a GPU System	3
03	Install TensorFlow on a CPU-Only System	5
04	Test the Installation	6
05	Conclusion	8

Introduction

TensorFlow is an open-source software library that allows implementations of common algorithms used for machine learning, artificial intelligence, and deep neural networks. By using Tensorflow, you can develop and deploy deep learning models such as Stable Diffusion and Large Language Models (LLMs).

This article explains how to install Tensorflow on Ubuntu 22.04 with either a GPU or CPU-only configuration. Depending on your system capabilities, you can use a Python package or a Conda environment to install and activate Tensorflow on your server.

Prerequisites

Before you begin:

- Deploy a [fresh Ubuntu 22.04 Server on Vultr](#)
- Using [SSH](#), access the server as a non-root user with sudo privileges
- Install the latest Python3 version

```
$ sudo apt install python3
```

Install TensorFlow on a GPU System

To install TensorFlow on a Vultr Cloud GPU server, verify that the NVIDIA drivers are correctly installed, then choose your desired installation method as described below.

Install TensorFlow Using Pip (Recommended)

To install TensorFlow using Pip, verify the [required CUDA and CUDNN versions](#) to correctly activate the software package on your GPU server. The TensorFlow version `2.13` installed in this article requires the following respective packages.

- CUDA Toolkit: `11.8`
- cuDNN: `8.6.0`

Verify that you install the required CUDA toolkit and cuDNN packages natively to avoid compatibility errors as described in the following resources:

- How to install CUDA Toolkit on Ubuntu 22.04
- How to Install cuDNN on Ubuntu 22.04

1. Update the `pip` Python package manager

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

2. Using `pip`, install TensorFlow

```
$ python3 -m pip install tensorflow==2.13.*
```

The above command installs the TensorFlow version `2.13`. To install the latest version, visit the [TensorFlow releases page](#) and further verify the required dependency package versions.

Install TensorFlow Using Conda

Conda is an environment package manager that allows you to install applications together with the necessary dependencies. To install TensorFlow, you must select a source Conda channel, and install the latest `tensorflow-gpu` version as described below.

1. Activate your Conda environment if not active

```
$ conda activate env1
```

2. Install the TensorFlow GPU package

```
$ conda install tensorflow-gpu=2.12.1 -c conda-forge -y
```

The above command installs the TensorFlow version `2.12.1` from the Conda-forge channel along with CUDA Toolkit `11.8` and cuDNN `8.8`

To work with GPUs, TensorFlow requires the CUDA and cuDNN packages. The Conda installation process installs all necessary dependencies. However, they are not fully-fledged packages as compared to manually installed variants. For example, the `cuda-toolkit` package, which Conda installs as a TensorFlow dependency, does not allow you to use the `nvcc` compiler. If your use case requires these packages, explicitly install the CUDA Toolkit and cuDNN within your Conda environment.

Install TensorFlow on a CPU-Only System

If your server does not have an attached GPU interface. Install TensorFlow using the non-GPU package as described in the optional methods below.

1. Using `pip`, install the latest TensorFlow package

```
$ python3 -m pip install tensorflow==2.13.*
```

The above command installs TensorFlow version `2.13`. To install the latest version, visit the TensorFlow releases page.

2. To install TensorFlow using Conda, run the following command in your environment

```
$ conda install tensorflow=2.12.0 -y
```

The above command installs the CPU-only `tensorflow` package without any GPU dependency packages. To install the latest available Conda package, visit the [packages page](#)

Test the Installation

To verify that your TensorFlow installation is successful, perform sample computation tasks that allow you to test the application functionality using Python as described below.

TensorFlow on a GPU System

1. Using Python, Import the TensorFlow package and print the list of available GPU devices

```
$ python3 -c "import tensorflow as tf;
print(tf.config.list_physical_devices('GPU'))"
```

Output:

```
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

2. Test that TensorFlow can perform a tensor-based operation using random numbers:

```
$ python3 -c "import tensorflow as tf;
print(tf.reduce_sum(tf.random.normal([100, 100])))"
```

When successful, your output should look like the one below:

```
2023-09-06 10:18:24.938874: I tensorflow/stream_executor/platform/default/
dso_loader.cc:49] Successfully opened dynamic library libcudart.s
.
.
2023-09-06 10:18:27.066352: I tensorflow/core/common_runtime/gpu/gpu_device.cc:
1406]
Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 379
```

```
MB memory) -> physical GPU (device: 0, name: NVIDIA A16-1Q, pci bus id:
0000:06:00.0, compute capability: 8.6)

tf.Tensor(-79.17527, shape=(), dtype=float32)
```

As displayed in the above output, the last line is the tensor computation result

TensorFlow on a CPU System

1. Verify that TensorFlow can perform a basic tensor-based computation

```
$ python3 -c "import tensorflow as tf;
print(tf.reduce_sum(tf.random.normal([100, 100])))"
```

The above command executes a tensor-based operation using random numbers. When successful, your output should look like the one below:

```
.
Could not find cuda drivers on your machine, GPU will not be used.
.
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
.
tf.Tensor(-276.19254, shape=(), dtype=float32)
```

In the above output, the last line is the computation result

2. If you receive the following TensorRT prompt while using TensorFlow

```
2023-09-12 10:43:13.438987: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:
38]
TF-TRT Warning: Could not find TensorRT
```

Install the latest [TensorRT](#) to optimize your inference operations

Conclusion

In this article, you installed TensorFlow on a Ubuntu 22.04 server, you explored how to install the package on both GPU-based and CPU-only systems. For more information, please visit the following resources:

- [Official TensorFlow Linux installation guide](#)
- [Anaconda TensorFlow page](#)
- [Conda Default Channel TensorFlow page](#)
- [Conda-Forge Channel TensorFlow page](#)



VULTR

