

Install GPU TensorFlow From Sources w/ Ubuntu 16.04 and Cuda 8.0



Date: September 8, 2016 **Author:** Justin □ 34 Comments

In this tutorial I will be going through the process of building TensorFlow 0.11rc1 from sources for Ubuntu 16.04. TensorFlow now supports using Cuda 8.0 & CuDNN 5.1 so you can use the pip's from their website for a much easier install. If you would like to install into a Anaconda environment the easiest method is to 'conda install pip' and just use the pip packages. If you prefer to build from sources using Ubuntu 14.04 please see my other tutorial.

In order to use TensorFlow with GPU support you must have a Nvidia graphic card with a minimum compute capability of 3.0.

Getting started I am going to assume you know some of the basics of using a terminal in Linux.

Install Required Packages

Open a terminal by pressing Ctrl + Alt + T

Paste each line one at a time (without the \$) using Shift + Ctrl + V

```
$ sudo apt-get install openjdk-8-jdk git python-dev python3-dev python-numpy python3-numpy build-essential python-pip python3-pip python-virtualenv swig python-wheel libcurl3-dev
```

Update & Install Nvidia Drivers

You must also have the 367 (or later) NVidia drivers installed, this can easily be done from Ubuntu's built in additional drivers after you update your driver packages.

```
$ sudo add-apt-repository ppa:graphics-drivers/ppa
$ sudo apt update
```

Once installed using additional drivers restart your computer. If you experience any troubles booting linux or logging in: try disabling fast & safe boot in your bios and modifying your grub boot options to enable nomodeset.

Install Bazel

Instructions also on Bazel website

```
$ echo "deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable jdk1.8" | sudo tee
/etc/apt/sources.list.d/bazel.list
$ curl https://storage.googleapis.com/bazel-apt/doc/apt-key.pub.gpg | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install bazel
$ sudo apt-get upgrade bazel
```

Install Nvidia Toolkit 8.0 & CudNN

Skip if not installing with GPU support

To install the Nvidia Toolkit download base installation .run file from Nvidia website. **MAKE SURE YOU SAY NO TO INSTALLING NVIDIA DRIVERS!** Also make sure you select yes to creating a symbolic link to your cuda directory.

```
$ cd ~/Downloads # or directory to where you downloaded file
$ sudo sh cuda_8.0.44_linux.run --override # hold s to skip
```

This will install cuda into: **/usr/local/cuda**

To install CudNN download cudNN v5.1 for Cuda 8.0 from Nvidia website and extract into **/usr/local/cuda** via:

```
$ sudo tar -xzf cudnn-8.0-linux-x64-v5.1.tgz
$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include
$ sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
$ sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
```

Then update your bash file:

```
$ gedit ~/.bashrc
```

This will open your bash file in a text editor which you will scroll to the bottom and add these lines:

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64"
export CUDA_HOME=/usr/local/cuda
```

Once you save and close the text file you can return to your original terminal and type this command to reload your .bashrc file:

```
$ source ~/.bashrc
```

Clone TensorFlow

```
$ cd ~
$ git clone https://github.com/tensorflow/tensorflow
```

Configure TensorFlow Installation

```
$ cd ~/tensorflow
$ ./configure
```

Use defaults by pressing enter for all except:

Please specify the location of python. [Default is /usr/bin/python]:

For Python 2 use default or If you wish to build for Python 3 enter:

```
$ /usr/bin/python3.5
```

Please input the desired Python library path to use. Default is [/usr/local/lib/python2.7/dist-packages]:

For Python 2 use default or If you wish to build for Python 3 enter:

```
$ /usr/local/lib/python3.5/dist-packages
```

Please specify the Cuda SDK version you want to use, e.g. 7.0. [Leave empty to use system default]:

```
$ 8.0
```

Please specify the Cudnn version you want to use. [Leave empty to use system default]:

```
$ 5.1.5
```

You can find the compute capability of your device at: <https://developer.nvidia.com/cuda-gpus>

If all was done correctly you should see:

INFO: All external dependencies fetched successfully.

Configuration finished

Build TensorFlow

Warning Resource Intensive I recommend having at least 8GB of computer memory.

If you want to build TensorFlow with GPU support enter:

```
$ bazel build -c opt --config=cuda //tensorflow/tools/pip_package:build_pip_package
```

For **CPU only** enter:

```
$ bazel build -c opt //tensorflow/tools/pip_package:build_pip_package
```

Build & Install Pip Package

This will build the pip package required for installing TensorFlow in your /tmp/ folder

```
$ bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
```

To Install Using Python 3 (remove sudo if using a virtualenv)

```
$ sudo pip3 install /tmp/tensorflow_pkg/tensorflow
```

```
# with no spaces after tensorflow hit tab before hitting enter to fill in blanks
```

For Python 2 (remove sudo if using a virtualenv)

```
$ sudo pip install /tmp/tensorflow_pkg/tensorflow
```

```
# with no spaces after tensorflow hit tab before hitting enter to fill in blanks
```

Test Your Installation

Close all your terminals and open a new terminal to test.

```
$ python # or python3
$ import tensorflow as tf
$ sess = tf.InteractiveSession()
$ sess.close()
```

TensorFlow also has instructions on how to do a basic test and a list of common installation problems.

There you have it, you should now have TensorFlow installed on your computer. This tutorial was tested on a fresh install of Ubuntu 16.04 with a GeForce GTX 780 and a GTX 970m.

If you want to give your GPU a workout maybe try building a massive image classifier following this tutorial.



Published by Justin

[View all posts by Justin](#)

34 thoughts on “Install GPU TensorFlow From Sources w/ Ubuntu 16.04 and Cuda 8.0”

Add Comment

1. roames says:

September 8, 2016 at 7:14 pm

Small correction – you meant to say “tested on a fresh install of Ubuntu 16.04”, correct?

Reply

1. Justin says:

September 8, 2016 at 7:38 pm

Woops yup, sorry

Reply

2. **Matthew Tsai says:**

September 9, 2016 at 12:33 am

Thank you very much!

Reply

1. **Justin says:**

September 9, 2016 at 8:09 am

You're welcome

Reply

3. **Parker says:**

September 11, 2016 at 2:46 am

(Just for those that might have encountered the same issue as me)

The final pip install did not work for me (I installed the whole thing using r0.10 branch on Ubuntu 14.04 LTS and cuda 8.0 and under an anaconda environment)... The packaged was not installed...

So I run this (as per the instructions in https://www.tensorflow.org/versions/r0.10/get_started/os_setup.html#anaconda-installation under the paragraph "Using pip" -> Finally install TensorFlow: # Python 2 (tensorflow)\$ pip install --ignore-installed --upgrade \$TF_BINARY_URL ... and changing \$TF_BINARY_URL for the path to my compiled .whl file) instead... So the command that worked is:

pip install --ignore-installed --upgrade /tmp/tensorflow_pkg/tensorflow (and hitting tab at the end to complete the name of the .whl file)... Now it works!

Thank you for the tutorial

Reply

4. **Tom Brander says:**

September 23, 2016 at 2:57 pm

Thank you so much, I'm installing on a fresh 16.04 machine but I hesitated at the upgrade bazel step as Ubuntu had just updated the drivers? is something else going on there?

bazel is already the newest version (0.3.1).

Calculating upgrade... Done

The following packages will be upgraded:

libsmclient libwbclient0 samba-ls system76-driver system76-driver-nvidia ubuntu-drivers-common

6 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

I was afraid that the install would bork system76-driver system76-driver-nvidia ubuntu-drivers-

????

I'm also trying to modify on the fly for Anaconda (already installed so any thoughts? I think it should be straightforward..

Reply

1. **Tom Brander says:**

September 24, 2016 at 8:57 am

Awesome I did go ahead and upgrade bazel and followed the rest of the instructions and got a successful install! I was able to just use pip without sudo since I had Anaconda installed with a Python 3 default at its root.

Reply

1. **Justin says:**

September 24, 2016 at 9:47 am

Great to hear!

5. **Faraz says:**

October 7, 2016 at 8:33 am

Thank you for the detailed instructions Justin, was very helpful. For those who are seeing the login-loop effect after installing the nvidia drivers, you need to make sure that the OpenGL libraries are NOT installed during the run. See this post for more detail: <https://devtalk.nvidia.com/default/topic/878117/-solved-titan-x-for-cuda-7-5-login-loop-error-ubuntu-14-04-/>

Reply

1. **Justin says:**

October 7, 2016 at 9:16 am

Thanks for sharing!

Reply

6. **derekhe1985 says:**

October 9, 2016 at 7:22 am

Thanks for this tutorial and it worked good.

There is one problem: that tensorflow says can't find library called cupti. So added CUPTI could solve this problem.

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64/"
```

Reply

7. **Andrei Muntean says:**

October 9, 2016 at 8:37 pm

Works as described. Thank you!

Reply

8. **spdrnl says:**

October 11, 2016 at 12:55 am

Thanks, worked like a charm

Reply

9. **xxi says:**

October 12, 2016 at 7:57 pm

Hello, I follow this tutorial to install tensorflow with cuda 8 and when I test if install done, I got

"Error importing tensorflow.

Unless you are using bazel, you should not try to import tensorflow from its source directory; please exit the tensorflow source tree, and relaunch your python interpreter from there."

screenshot: <http://imgur.com/a/OCYu8>

can you help me, thank you very much

Reply

1. **Justin says:**

October 12, 2016 at 7:59 pm

Yeah I noticed this as well you cannot be in tensorflow directory when importing. Exit python and \$ cd ~

Reply

1. **xxi says:**

October 12, 2016 at 11:11 pm

Thanks for reply, but the same, <http://imgur.com/a/e2xE8>

and when I run "bazel build -c opt --config=cuda //tensorflow/tools/pip_package:build_pip_package"

I got some warning like this <http://imgur.com/a/GOCog>

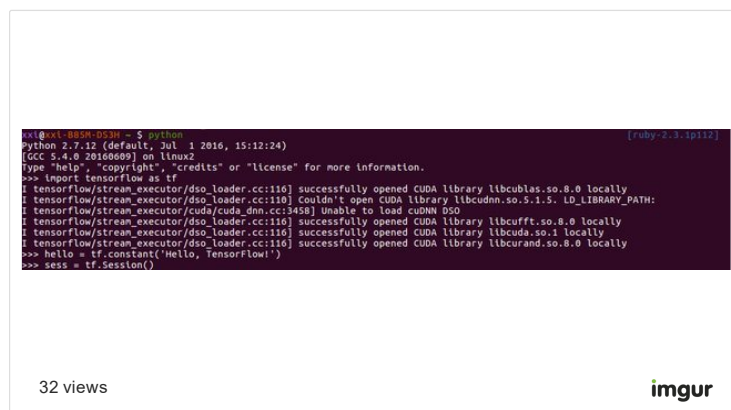
I'm not sure if it affect, thanks

2. **xxi says:**

October 13, 2016 at 12:33 am

as the screenshot said, I noticed an "importError: libcudart.so.8.0....."

I google it and found answer, use "sudo ldconfig /usr/local/cuda/lib64" can solve this problem, and then I can import tensorflow, but it can't open libcudnn...still finding answer 😊



3. **cs16 says:**

November 21, 2016 at 3:07 am

Hey, I had the same problem as this but I've fixed it now.

I did "sudo ldconfig /usr/local/cuda/lib64" and got the same libcudnn error but then I edited my bashrc file again and changed the line

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64"
```

to

```
export
```

```
LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64:/usr/local/cuda/include"
```

and now everything works.

10. **Robert says:**

October 14, 2016 at 11:05 am

I have exactly the same problem. Any ideas?

Reply

1. **Richard Butt says:**

October 16, 2016 at 9:15 pm

firstly, try:

pip install /tmp/tensorflow_pkg/tensorflow (note that it is not sudo)

just in case there's an issue with permission/access/location of the files, then try again.

IE:

"sudo which python" points to /usr/bin/python

"which python" in my virtual env named tfgpu points to /home/dickbutt/tfgpu/bin/python

If that fails, you can try rebuilding. I had a different issue with bazel the last time that I configured and built tensorflow. I recall that I had to delete the cached files created by bazel to make changes.

Reply

2. **xxi says:**

October 16, 2016 at 9:16 pm

The libcudnn problem?

When I run "sudo ldconfig /usr/local/cuda/lib64", I got a message "libcudnn.so.5 is not a symbolic link", I think this might be the reason but I don't know how to fixed it

If you haven't installed, you can try <http://askubuntu.com/questions/767269/how-can-i-install-cudnn-on-ubuntu-16-04> use "cp -p"

so anyone knows how to fixed it after installed?

Reply

11. **Richard Butt says:**

October 16, 2016 at 9:06 pm

Awesome writeup. Thanks a lot for putting together the requirements and JDK install. I rolled it successfully on:

Ubuntu 16.04 (Desktop version)

Nvidia driver 367.44

CUDA 8.0

CUDNNv5.1

Python 2.7.12

inside of a virtualenv (sudo pip install virtualenv)

Tensorflow on master (10/16/16)

The step I replaced was due to install path issue (might be virtualenv issue that was specific to me):

sudo pip install /tmp/tensorflow_pkg/tensorflow(tab-complete)

Reply

1. **Justin says:**

October 16, 2016 at 10:35 pm

I primarily use Ubuntu 14.04 but I learned today that with Ubuntu 16.04 you have to update nvidia drivers to a version newer than the additional drivers tab provides.

Unless you created your virtualenv as SU no need to use sudo with a virtualenv, I think it might actually defeat the whole purpose of a virtualenv.

Reply

12. **Chad Taljaardt says:**

October 20, 2016 at 4:39 pm

Hello,

Im having a issue when using tensorflow, :

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcublas.so.8.0 locally

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcudnn.so.5.1.5 locally

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcufft.so.8.0 locally

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcuda.so.1 locally

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcurand.so.8.0 locally

I tensorflow/core/common_runtime/gpu/gpu_device.cc:953] Found device 0 with properties:

name: GeForce GTX 1070
major: 6 minor: 1 memoryClockRate (GHz) 1.683
pciBusID 0000:01:00.0
Total memory: 7.92GiB
Free memory: 7.59GiB

I tensorflow/core/common_runtime/gpu/gpu_device.cc:974] DMA: 0

I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] 0: Y

I tensorflow/core/common_runtime/gpu/gpu_device.cc:1043] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1070, pci bus id: 0000:01:00.0)

E tensorflow/stream_executor/cuda/cuda_dnn.cc:385] could not create cudnn handle: CUDNN_STATUS_INTERNAL_ERROR

E tensorflow/stream_executor/cuda/cuda_dnn.cc:352] could not destroy cudnn handle: CUDNN_STATUS_BAD_PARAM

F tensorflow/core/kernels/conv_ops.cc:532] Check failed: stream->parent()->GetConvolveAlgorithms(&algorithms) Aborted (core dumped)

Do you know what the issue is?

Reply

13. **Shahbaz Chaudhary says:**

October 23, 2016 at 6:11 pm

I'm attempting to follow these instructions for Ubuntu 16.10. Here are a couple of notes you might want to add to this guide...and an error at the bottom

1. 16.10 installs gcc 6 by default, which will throw error because CUDA doesn't support anything higher than 5 (the error message says as much). In order to resolve this, I installed gcc-5 and used update-alternatives to change my system gcc to gcc-5 (this way, I can switch back to 6 if I want). Here is a good reference: <https://gist.github.com/beci/2a2091f282042ed20cda>. This is an alternative way of getting cuda to work with gcc 6, but it didn't seem as clean to me: https://www.reddit.com/r/archlinux/comments/4iczja/a_work_around_if_you_are_using_cuda_and_have/

2. The ./configure script also asks if google cloud, hadoop hdfs and GPU support should be built. I picked yes for all of them.

Error:

After everything, when I try to run a test python (3.5) script, I get this error:

```
>>> import tensorflow as tf
```

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcublas.so.8.0 locally

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcudnn.so.5.1.5 locally

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcufft.so.8.0 locally

I tensorflow/stream_executor/dso_loader.cc:120] Couldn't open CUDA library libcuda.so.1. LD_LIBRARY_PATH: /usr/local/cuda/lib64

I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:162] hostname: shahbaz-XPS-15-9550

I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:186] libcuda reported version is: Not found: was unable to find libcuda.so DSO loaded into this program

I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:190] kernel reported version is: Permission denied: could not open driver version path for reading: /proc/driver/nvidia/version

I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:1091] LD_LIBRARY_PATH: /usr/local/cuda/lib64

I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:1092] failed to find libcuda.so on this system: Failed precondition: could not dlopen DSO: libcuda.so.1; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory

I tensorflow/stream_executor/dso_loader.cc:126] successfully opened CUDA library libcurand.so.8.0 locally

So it can't find libcuda.so.1 ...what? Is it a permission error or is the LD_LIBRARY_PATH string messed up (due to that colon)?

Reply

1. **Justin says:**

October 23, 2016 at 6:46 pm

The tutorial is for 16.04

Reply

2. **Aaron says:**

November 17, 2016 at 8:06 am

Have you found a solution for the problem?

Reply

14. **Giridhur says:**

October 25, 2016 at 5:57 am

After generating python wheel file "Requirement already satisfied (use --upgrade to upgrade): setuptools in /usr/lib/python2.7/dist-packages (from protobuf==3.1.0->tensorflow==0.11.0rc1)

Installing collected packages: funcsigs, pbr, mock, protobuf, tensorflow

Successfully installed funcsigs-1.0.2 mock-2.0.0 pbr-1.10.0 protobuf-3.1.0 tensorflow-0.11.0rc1

..

I try "import tensorflow" in my python, and It says " no module named tensorflow. "

Reply

1. **Justin says:**

October 25, 2016 at 7:14 am

Try rebooting?

Reply

1. **Giridhur Sriraman says:**

October 25, 2016 at 8:54 am

Removed anaconda entirely, seems to work now. I think 'pip' pointed to the system python's and not anaconda's .

15. **JeeSok Lee says:**

November 20, 2016 at 5:43 pm

Hi Justin. Thanks to you, I was able to install tensorflow using CUDA 8.0. I was hoping to translate this post into Korean in my blog, if that's okay with you (I do not yet have a blog, and this will be my first post). If so, your site will be linked on the top of the post as a reference.

Reply

1. **Justin says:**

November 20, 2016 at 5:53 pm

Sounds great there is this in mandarin as well.

<https://www.oreilly.com.cn/ideas/?p=690>

Reply

16. **E says:**

November 20, 2016 at 7:03 pm

Hey, Just wanted to say thanks! I have been stuck for a week dealing with getting this installed and finding your website has really helped.

Reply

17. **Justin says:**

November 20, 2016 at 8:54 pm

Glad to hear I saved you some time and frustration. Also good news the latest TensorFlow build now seems to support OpenCL & AMD Cards.

Reply