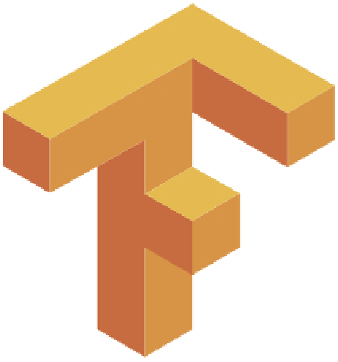
Justin's Blog

TUTORIALS

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



Date: September 8, 2016 Author: Justin 69 Comments

In this tutorial I will be going through the process of building the latest TensorFlow from sources for Ubuntu 16.04. I much prefer to install Tensorflow using Anaconda Python and you can find a tutorial for that here. 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 python-six python3-six build-essential python-pip python3-pip
python-virtualenv swig python-wheel python3-wheel libcurl3-dev libcupti-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 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
$ wget
https://developer.nvidia.com/compute/cuda/8.0/Prod2/local_installers/cuda_8.0.61_3
75.26_linux-run
$ sudo sh cuda_8.0.61_375.26_linux.run --override --silent --toolkit
```

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

To install CudNN download cudNN v6.0 for Cuda 8.0 from Nvidia website and extract into **/usr/local/cuda** via:

```
$ tar -xzvf cudnn-8.0-linux-x64-v6.0.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/CUP
TI/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
```

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://bazel.build/bazel-release.pub.gpg | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install bazel
$ sudo apt-get upgrade bazel
```

Clone Tensorflow

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

Unless you want absolute bleeding edge I highly recommend checking-out to the latest branch rather than master.

```
$ cd ~/tensorflow
$ git checkout r1.2
```

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
```

Unless you have a Radeon graphic card you can say no to OpenCL support. (has anyone tested this? ping me if so!)

Do you wish to build TensorFlow with CUDA support?

```
$ Y
```

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 --config=opt --config=cuda
//tensorflow/tools/pip_package:build_pip_package
```

For **CPU only** enter:

```
$ bazel build --config=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. Also make sure your Terminal is not in the 'tensorflow' directory.

```
$ python3 # or python for python 2
$ 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, GTX 970m & Quadro K2000M.

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

69 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:
   libsmbclient libwbclient0 samba-libs 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 istructions 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 anser, use "sudo ldconfig /usr/local/cuda/lib64" can sovle this problem, and then I can import tensorflow, but it can't open libcudnn…still finding answer 🙂

   Follow Justin's Blog 4

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 by 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/ir

   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 "libucudnn.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 ./confgure 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

    1. **Marcus says:**
       February 22, 2017 at 9:33 pm
       Will it follow the same procedure for and AMD card?

       Reply

18. **shyamkkhadka says:**
    November 23, 2016 at 3:12 am
    Does this process work for my 32 bit ubuntu machine ?

    Reply

    1. **Justin says:**
       November 25, 2016 at 8:09 pm
       Never tried but doubt it

       Reply

19. **vaibhavsahu123 says:**
    November 27, 2016 at 1:09 pm
    Can you please help me on how to install TensorFlow on another drive(on sda5 instead of sda1) on Ubuntu 16.04?

    Thanks

    Reply

20. Pingback: Installing TensorFlow on Ubuntu 16.04 | Taufiq Habib's Blog
21. Pingback: OpenAI – Universe Installation Guide Ubuntu 16.04 – Justin's Blog
22. **Gustavo Monge says:**
    December 10, 2016 at 8:10 pm
    Why is not a good idea to update the Nvidia Drivers?
    MAKE SURE YOU SAY NO TO INSTALLING NVIDIA DRIVERS!

    There is also available a run file available from Envidia to do so.
    Thanks in advance. I am totally new to Ubuntu.

Reply

1. **Justin says:**
   December 11, 2016 at 4:46 pm
   Because the cuda built in driver installation doesn't work well, best to do it yourself as per instructions.

   Reply

   1. **Gustavo Monge says:**
      December 12, 2016 at 10:04 am
      Thank you.
      I am in the middle of the installation following exactly your guide:
      I am getting this:

      ```
      Please specify which C++ compiler should be used as the host C++ compiler. [Default is /usr/bin/g++]:
      Please specify which C compiler should be used as the host C compiler. [Default is /usr/bin/gcc]:
      Please specify the location where ComputeCpp for SYCL 1.2 is installed. [Default is /usr/local/computecpp]:
      Invalid SYCL 1.2 library path. /usr/local/computecpp/lib/libComputeCpp.so cannot be found
      ```

      What should I do?
      Thanks in advance

   2. **Justin says:**
      December 14, 2016 at 10:07 am
      I have never gotten those messages before when building tensorflow. Use defaults I guess. What OS are you on?

23. **Gustavo Monge says:**
    December 12, 2016 at 11:02 am
    I found this also, up in the log, even though Software and up dates / Additional drivers says:

    Using Nvidia binary driver – version 367.57 from nvidia-367(open source)

    Log:

    ```
    Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 367.48?
    (y)es/(n)o/(q)uit: n

    Install the CUDA 8.0 Toolkit?
    (y)es/(n)o/(q)uit: y

    Enter Toolkit Location
    [ default is /usr/local/cuda-8.0 ]:

    Do you want to install a symbolic link at /usr/local/cuda?
    (y)es/(n)o/(q)uit: y

    Install the CUDA 8.0 Samples?
    (y)es/(n)o/(q)uit: y

    Enter CUDA Samples Location
    [ default is /home/gustavo ]:

    Installing the CUDA Toolkit in /usr/local/cuda-8.0 …
    Missing recommended library: libGLU.so
    Missing recommended library: libXi.so
    Missing recommended library: libXmu.so

    Installing the CUDA Samples in /home/gustavo …
    Copying samples to /home/gustavo/NVIDIA_CUDA-8.0_Samples now…
    Finished copying samples.

    ===========
    = Summary =
    ===========

    Driver: Not Selected
    Toolkit: Installed in /usr/local/cuda-8.0
    Samples: Installed in /home/gustavo, but missing recommended libraries
    ```

```
Please make sure that
– PATH includes /usr/local/cuda-8.0/bin
– LD_LIBRARY_PATH includes /usr/local/cuda-8.0/lib64, or, add /usr/local/cuda-8.0/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-8.0/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-8.0/doc/pdf for detailed information on setting up CUDA.

***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of version at least 361.00 is required for CUDA 8.0 functionality to work.
To install the driver using this installer, run the following command, replacing with the name of this run file:
sudo .run -silent -driver
```

Reply

24. **Gustavo Monge says:**
December 13, 2016 at 10:02 pm

I did a new fresh Ubuntu Installation, I followed everything as per instructions and it did not work at all.

I am still getting:

```
Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 367.48?
(y)es/(n)o/(q)uit: n

Install the CUDA 8.0 Toolkit?
(y)es/(n)o/(q)uit: y

Enter Toolkit Location
[ default is /usr/local/cuda-8.0 ]:

Do you want to install a symbolic link at /usr/local/cuda?
(y)es/(n)o/(q)uit: y

Install the CUDA 8.0 Samples?
(y)es/(n)o/(q)uit: y

Enter CUDA Samples Location
[ default is /home/gustavo ]:

Installing the CUDA Toolkit in /usr/local/cuda-8.0 …
Missing recommended library: libGLU.so
Missing recommended library: libXi.so
Missing recommended library: libXmu.so

Installing the CUDA Samples in /home/gustavo …
Copying samples to /home/gustavo/NVIDIA_CUDA-8.0_Samples now…
Finished copying samples.

===========
= Summary =
===========

Driver: Not Selected
Toolkit: Installed in /usr/local/cuda-8.0
Samples: Installed in /home/gustavo, but missing recommended libraries

Please make sure that
– PATH includes /usr/local/cuda-8.0/bin
– LD_LIBRARY_PATH includes /usr/local/cuda-8.0/lib64, or, add /usr/local/cuda-8.0/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-8.0/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-8.0/doc/pdf for detailed information on setting up CUDA.
```

***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of version at least 361.00 is required for CUDA 8.0 functionality to work.
To install the driver using this installer, run the following command, replacing with the name of this run file:
sudo .run -silent -driver

Logfile is /tmp/cuda_install_5406.log

Reply

1. **Justin says:**
   December 14, 2016 at 10:12 am
   That looks ok it installed the toolkit the samples are not neccessary

   Reply

25. **Gustavo Monge says:**
December 14, 2016 at 7:50 am
I found this post:
https://devtalk.nvidia.com/default/topic/827410/?comment=4513689
with this suggestion:
sudo apt-get install freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev
I used it and it solved the missing libraries.

I continued the process and used the GPU option.

This is the final log message:
At global scope:
cc1plus: warning: unrecognized command line option '-Wno-self-assign'
Target //tensorflow/tools/pip_package:build_pip_package up-to-date:
bazel-bin/tensorflow/tools/pip_package/build_pip_package
INFO: Elapsed time: 1659.310s, Critical Path: 1303.19s

I tested with the four lines of code as suggested, and I got this:
AttributeError: module 'tensorflow' has no attribute 'InteractiveSession'

Thanks in advance for any comments or help

Reply

1. **Justin says:**
   December 14, 2016 at 10:06 am
   What does it say when you import tensorflow?

   Reply

   1. **Gustavo Monge says:**
      December 14, 2016 at 11:43 am
      The last part, the compiling part, apparently worked, with a lot of "warnings" like one in the images up in the post.
      I sent you the last few lines in the previous post.

   2. **Gustavo Monge says:**
      December 14, 2016 at 11:51 am
      After adding the libraries in accordance to the link, I got this:

      ===========
      = Summary =
      ===========

      Driver: Not Selected
      Toolkit: Installed in /usr/local/cuda-8.0
      Samples: Installed in /home/gustavo

      Which I think It was right.
      I am not sure if those are the latest versions, but apparently
      worked well

      What does this means?
      AttributeError: module 'tensorflow' has no attribute 'InteractiveSession'

      Thanks for your time and comments

   3. **Gustavo Monge says:**

December 14, 2016 at 11:58 am
Excuse me.
I now understand your question.

It says nothing

```
gustavo@gus:~$ python3
Python 3.5.2 (default, Nov 17 2016, 17:05:23)
[GCC 5.4.0 20160609] on linux
Type “help”, “copyright”, “credits” or “license” for more information.
>>> import tensorflow as tf
>>> sess = tf.InteractiveSession()
Traceback (most recent call last):
File “”, line 1, in
AttributeError: module ‘tensorflow’ has no attribute ‘InteractiveSession’
>>> exit()
```

4. **Justin says:**
   December 14, 2016 at 4:06 pm
   Is your conda environment activated, that looks like your anaconda is not loading.

26. **Gustavo Monge says:**
    December 14, 2016 at 6:19 pm
    At the moment I did the installation or followed your instructions, I had not installed anaconda.

    Now, in this moment yes I have it.

    Should I do something with the anaconda?
    Do I have to create or use a special environment to run tensorFlow,
    as for the example?

    Thanks again

    Reply

    1. **Justin says:**
       December 14, 2016 at 7:31 pm
       Sorry I thought maybe you found this from the Universe guide I just made, and in that one I use anaconda.

       It looks like the pip package is not installed, did it shout back anything weird when you did the pip installation?

       Reply

       1. **Gustavo Monge says:**
          December 14, 2016 at 8:25 pm
          I don think so. Nothing else but warnings.
          Here is the beginning and the end of the pip:

          Start:

          ```
          gustavo@gus:~/tensorflow$ bazel build -c opt –config=cuda //tensorflow/tools/pip_package:build_pip_package
          INFO: Found 1 target…
          INFO: From Compiling external/protobuf/python/google/protobuf/pyext/message_factory.cc:
          external/protobuf/python/google/protobuf/pyext/message_factory.cc: In function ‘PyObject* google::protobuf::python::message_factory::New(PyTypeObject*, PyObject*, PyObject*)’:
          external/protobuf/python/google/protobuf/pyext/message_factory.cc:78:37: warning: ISO C++ forbids converting a string constant to ‘char*’ [-Wwrite-strings]
          static char* kwlist[] = {“pool”, 0};
          ^
          ```

          End:

          ```
          INFO: From Compiling tensorflow/python/pywrap_tensorflow.cc:
          bazel-out/local_linux-py3-opt/bin/tensorflow/python/pywrap_tensorflow.cc: In function ‘PyObject* _wrap_PyRecordReader_New(PyObject*, PyObject*)’:
          bazel-out/local_linux-py3-opt/bin/tensorflow/python/pywrap_tensorflow.cc:4713:138: warning: ‘arg2’ may be used uninitialized in this function [-Wmaybe-uninitialized]
          result = (tensorflow::io::PyRecordReader *)tensorflow::io::PyRecordReader::New((string const &)*arg1,arg2, (string const &)*arg3,arg4);
          ^
          At global scope:
          cc1plus: warning: unrecognized command line option ‘-Wno-self-assign’
          ```

```
Target //tensorflow/tools/pip_package:build_pip_package up-to-date:
bazel-bin/tensorflow/tools/pip_package/build_pip_package
INFO: Elapsed time: 1659.310s, Critical Path: 1303.19s
```

27. **Max says:**
    December 19, 2016 at 6:25 am
    I done all above, and thanks for instructions but I always get problem with bazel…

    ```
    [Default is: "3.5,5.2"]:
    INFO: Options provided by the client:
    Inherited 'common' options: –isatty=1 –terminal_columns=103
    INFO: Reading options for 'clean' from /home/victorem/tensorflow/tools/bazel.rc:
    Inherited 'build' options: –force_python=py3 –host_force_python=py3 –python3_path=/usr/bin/python3.5 –define=use_fast_cpp_protos=true –define=allow_oversize_protos=true –define PYTHON_BIN_PATH=/usr/bin/python3.5 –spawn_strategy=standalone –genrule_strategy=standalone
    INFO: Reading options for 'clean' from /etc/bazel.bazelrc:
    Inherited 'build' options: –action_env=PATH –action_env=LD_LIBRARY_PATH –action_env=TMPDIR –test_env=PATH –test_env=LD_LIBRARY_PATH
    Unrecognized option: –action_env=PATH
    ```

    Thanks for any help!

    Reply

    1. **Justin says:**
       December 19, 2016 at 12:04 pm
       Sorry never seen that one before, try stackoverflow or just use the pip installations.

       Reply

28. **Andrew Selle says:**
    January 9, 2017 at 10:22 am
    Max, Just ran into this myself. The cause of that error for me was that there were two versions of bazel installed on my machine (one in my home directory that was older) and one installed with apt-get. I'd check to see if you have two bazel binaries installed. Good luck.

    Reply

29. **clark says:**
    February 7, 2017 at 2:42 am
    I managed to get it installed and can run tensorflow but it seems like the CUDA isnt working….this is the log/error, i ran the script for image recognition, it did give me results correctly. But it seems like its not using the CUDA/GPU?

    ```
    I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcublas.so locally
    I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcudnn.so locally
    I tensorflow/stream_executor/dso_loader.cc:128] successfully opened CUDA library libcufft.so locally
    I tensorflow/stream_executor/dso_loader.cc:119] Couldn't open CUDA library libcuda.so.1. LD_LIBRARY_PATH: :/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64
    I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:165] hostname: clark-VirtualBox
    I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:189] 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:193] 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:1092] LD_LIBRARY_PATH: :/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64
    I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:1093] 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:128] successfully opened CUDA library libcurand.so locally
    E tensorflow/stream_executor/cuda/cuda_driver.cc:509] failed call to cuInit: CUDA_ERROR_NO_DEVICE
    I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:145] kernel driver does not appear to be running on this host (clark-VirtualBox): /proc/driver/nvidia/version does not exist
    ```

    Reply

30. **Shirsh Zibbu says:**
    February 20, 2017 at 7:06 pm
    when using in python , i get the following warnings:

    ```
    W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.
    ```

W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.

W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.

W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up CPU computations.

W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.

how can i (and should i?) build tensorflow with these instructions?

Reply

31. **neama rakha says:**
February 26, 2017 at 6:28 am
please specify c++ compiler will be used as the host c++ compiler

Reply

32. **Chris Pernoud says:**
March 2, 2017 at 3:12 pm
Thanks for this writeup. Have you looked at a way to use the Nvidia GPU for compute only and not display? i.e. integrated graphics for graphical monitor display, probably nouveau driver, and nvidia driver loaded on the GPU for dedicated Tensorflow? That way there is no contention for GPU resources.

I've seen a few articles under compute-only, but no happy endings. In my effort so far, I've installed 16.04 desktop, but have had a heck of a time trying to get nouveau to drive the integrated card without interfering with the GPU and nvidia drivers.

Remote development still seems hard enough that you have to have a local display for eclipse and matplotlib output, so I can't run it headless. I'm about to try installing the server software and trying to build up. However, at some point X will have to be installed and nouveau will probably try to take control again.

Any ideas are appreciated.

Reply

33. **Henry says:**
March 9, 2017 at 9:35 pm
I build package, but it does not install. Pls help.

sudo -H pip3 install /tmp/tensorflow_pkg/tensorflow-1.0.1-cp36-cp36m-linux_x86_64.whl

Error:
tensorflow-1.0.1-cp36-cp36m-linux_x86_64.whl is not a supported wheel on this platform.

Reply

34. **Kiran Prakash says:**
April 2, 2017 at 7:32 am
Works perfect! Geforce 820M Ubuntu16.04. The only place I had an issue while installing the

***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of version at least 361.00 is required for CUDA 8.0 functionality to work.
To install the driver using this installer, run the following command, replacing with the name of this run file:
sudo .run -silent -driver ***"
exactly similar to what Gustavo Monge had discussed. But I think I could ignore it for now and get my hands on programming. I nearly spent a 18 hours figuring out how to install and tried several others. They all are same but too much of explanation for each step for a beginner in the field of Linux and AI like me. You just made my day with this awesome blog, Justin!

Reply

35. **PRAVEEN SINGH says:**
May 14, 2017 at 9:18 pm
Thanks for posting this tutorial. I am following your instructions to install TensorFlow on Ubuntu 17.04. However when I try to configure using "./configure" I get an error –
ERROR: It appears that the development version of libcurl is not available. Please install the libcurl3-dev package.
I purged libcurl4 and installed libcurl3 and tried again, getting the same error.
Can you kindly let me know how to proceed?

Reply

1. **Justin says:**
   May 15, 2017 at 5:45 am
   If using 17.04 best use pip packages building from sources I havnt attempted nor see the need for 17.04.

   Reply

36. **ryeingoddard says:**
    August 31, 2017 at 1:38 pm
    Good stuff…I used this with anaconda to avoid conflicts and improve upon their implementation which doesn't use any optimizations.

    Thanks

    Reply

37. **MIftah says:**
    September 13, 2017 at 6:42 pm
    Thanks for sharing .. I followed the steps here all the way to successfully configuring tensorflow , but the steps for building with bazel failed to build and i got the following error message :

```
/home/miftah/tensorflow/tensorflow/contrib/gdr/BUILD:52:1: C++ compilation of rule
'//tensorflow/contrib/gdr:gdr_memory_manager' failed (Exit 1)
tensorflow/contrib/gdr/gdr_memory_manager.cc:28:27: fatal error: rdma/rdma_cma.h: No such file or directory
compilation terminated.
Target //tensorflow/tools/pip_package:build_pip_package failed to build
Use –verbose_failures to see the command lines of failed build steps.
INFO: Elapsed time: 7.314s, Critical Path: 3.32s
FAILED: Build did NOT complete successfully
```

and I appended verbose rebuild and i got the following ;

```
ERROR: /home/miftah/tensorflow/tensorflow/contrib/verbs/BUILD:118:1: C++ compilation of rule
'//tensorflow/contrib/verbs:rdma_mgr' failed (Exit 1): crosstool_wrapper_driver_is_not_gcc failed: error executing
command
(cd /home/miftah/.cache/bazel/_bazel_miftah/a00fd82407b7696f248094caeae3a544/execroot/org_tensorflow && \
exec env – \
CUDA_TOOLKIT_PATH=/usr/local/cuda \
CUDNN_INSTALL_PATH=/usr/local/cuda-8.0 \
GCC_HOST_COMPILER_PATH=/usr/bin/gcc \
PWD=/proc/self/cwd \
PYTHON_BIN_PATH=/usr/bin/python \
PYTHON_LIB_PATH=/usr/local/lib/python2.7/dist-packages \
TF_CUDA_CLANG=0 \
TF_CUDA_COMPUTE_CAPABILITIES=3.5,5.2 \
TF_CUDA_VERSION=8.0 \
TF_CUDNN_VERSION=6 \
TF_NEED_CUDA=1 \
TF_NEED_OPENCL=0 \
external/local_config_cuda/crosstool/clang/bin/crosstool_wrapper_driver_is_not_gcc -U_FORTIFY_SOURCE '-
D_FORTIFY_SOURCE=1' -fstack-protector -fPIE -Wall -Wunused-but-set-parameter -Wno-free-nonheap-object -fno-
omit-frame-pointer -g0 -O2 -DNDEBUG -ffunction-sections -fdata-sections '-march=native' '-std=c++11' '-march=native'
-MD -MF bazel-out/local_linux-opt/bin/tensorflow/contrib/verbs/_objs/rdma_mgr/tensorflow/contrib/verbs/rdma_mgr.pic.d
'-frandom-seed=bazel-out/local_linux-
opt/bin/tensorflow/contrib/verbs/_objs/rdma_mgr/tensorflow/contrib/verbs/rdma_mgr.pic.o' -fPIC -
DEIGEN_MPL2_ONLY -DTENSORFLOW_USE_JEMALLOC -DSNAPPY -DTENSORFLOW_USE_VERBS -
DTENSORFLOW_USE_GDR -iquote . -iquote bazel-out/local_linux-opt/genfiles -iquote external/protobuf_archive -
iquote bazel-out/local_linux-opt/genfiles/external/protobuf_archive -iquote external/bazel_tools -iquote bazel-
out/local_linux-opt/genfiles/external/bazel_tools -iquote external/grpc -iquote bazel-out/local_linux-
opt/genfiles/external/grpc -iquote external/zlib_archive -iquote bazel-out/local_linux-opt/genfiles/external/zlib_archive -
iquote external/boringssl -iquote bazel-out/local_linux-opt/genfiles/external/boringssl -iquote external/nsync -iquote bazel-
out/local_linux-opt/genfiles/external/nsync -iquote external/jemalloc -iquote bazel-out/local_linux-
opt/genfiles/external/jemalloc -iquote external/eigen_archive -iquote bazel-out/local_linux-
opt/genfiles/external/eigen_archive -iquote external/local_config_sycl -iquote bazel-out/local_linux-
opt/genfiles/external/local_config_sycl -iquote external/gif_archive -iquote bazel-out/local_linux-
opt/genfiles/external/gif_archive -iquote external/jpeg -iquote bazel-out/local_linux-opt/genfiles/external/jpeg -iquote
external/com_googlesource_code_re2 -iquote bazel-out/local_linux-opt/genfiles/external/com_googlesource_code_re2 -
iquote external/farmhash_archive -iquote bazel-out/local_linux-opt/genfiles/external/farmhash_archive -iquote
external/fft2d -iquote bazel-out/local_linux-opt/genfiles/external/fft2d -iquote external/highwayhash -iquote bazel-
out/local_linux-opt/genfiles/external/highwayhash -iquote external/png_archive -iquote bazel-out/local_linux-
opt/genfiles/external/png_archive -iquote external/snappy -iquote bazel-out/local_linux-opt/genfiles/external/snappy -
iquote external/local_config_cuda -iquote bazel-out/local_linux-opt/genfiles/external/local_config_cuda -iquote
external/curl -iquote bazel-out/local_linux-opt/genfiles/external/curl -iquote external/jsoncpp_git -iquote bazel-
out/local_linux-opt/genfiles/external/jsoncpp_git -isystem external/protobuf_archive/src -isystem bazel-out/local_linux-
opt/genfiles/external/protobuf_archive/src -isystem external/bazel_tools/tools/cpp/gcc3 -isystem external/grpc/include -
isystem bazel-out/local_linux-opt/genfiles/external/grpc/include -isystem external/zlib_archive -isystem bazel-
out/local_linux-opt/genfiles/external/zlib_archive -isystem external/boringssl/src/include -isystem bazel-out/local_linux-
opt/genfiles/external/boringssl/src/include -isystem external/nsync/public -isystem bazel-out/local_linux-
opt/genfiles/external/nsync/public -isystem external/jemalloc/include -isystem bazel-out/local_linux-
opt/genfiles/external/jemalloc/include -isystem external/eigen_archive -isystem bazel-out/local_linux-
opt/genfiles/external/eigen_archive -isystem external/gif_archive/lib -isystem bazel-out/local_linux-
opt/genfiles/external/gif_archive/lib -isystem external/farmhash_archive/src -isystem bazel-out/local_linux-
opt/genfiles/external/farmhash_archive/src -isystem external/png_archive -isystem bazel-out/local_linux-
opt/genfiles/external/png_archive -isystem external/local_config_cuda/cuda -isystem bazel-out/local_linux-
opt/genfiles/external/local_config_cuda/cuda -isystem external/local_config_cuda/cuda/cuda/include -isystem bazel-
out/local_linux-opt/genfiles/external/local_config_cuda/cuda/cuda/include -isystem external/curl/include -isystem bazel-
out/local_linux-opt/genfiles/external/curl/include -isystem external/jsoncpp_git/include -isystem bazel-out/local_linux-
opt/genfiles/external/jsoncpp_git/include -no-canonical-prefixes -Wno-builtin-macro-redefined ‘-D__DATE__=”redacted”‘
‘-D__TIMESTAMP__=”redacted”‘ ‘-D__TIME__=”redacted”‘ -fno-canonical-system-headers -c
tensorflow/contrib/verbs/rdma_mgr.cc -o bazel-out/local_linux-
opt/bin/tensorflow/contrib/verbs/_objs/rdma_mgr/tensorflow/contrib/verbs/rdma_mgr.pic.o)
In file included from ./tensorflow/contrib/verbs/rdma_mgr.h:24:0,
from tensorflow/contrib/verbs/rdma_mgr.cc:18:
./tensorflow/contrib/verbs/rdma.h:21:30: fatal error: infiniband/verbs.h: No such file or directory
compilation terminated.
Target //tensorflow/tools/pip_package:build_pip_package failed to build
INFO: Elapsed time: 142.589s, Critical Path: 49.90s
FAILED: Build did NOT complete successfully
```

I took 2 days to figure out what went wrong , i came here if folks can help , I thank you in advance …

Reply