

# CUDA as the Prerequisite for Tensor Flow

<https://www.tensorflow.org/tutorials/>

1. Find the version of your ubuntu

`uname -m && cat /etc/*release`

or

`lsb_release -a`

```
ubuntu@ubuntu-ThinkPad-Yoga-14: ~  
ubuntu@ubuntu-ThinkPad-Yoga-14:~$ lsb_release -a  
No LSB modules are available.  
Distributor ID: Ubuntu  
Description:    Ubuntu 14.04.5 LTS  
Release:        14.04  
Codename:       trusty  
ubuntu@ubuntu-ThinkPad-Yoga-14:~$
```

2. find the graphics card your machine support

`sudo lshw -C video | grep product:`

```
ubuntu@ubuntu-ThinkPad-Yoga-14: ~  
ubuntu@ubuntu-ThinkPad-Yoga-14:~$ sudo lshw -C video | grep product:  
product: Sky Lake Integrated Graphics  
product: GM108M [GeForce 940M]  
ubuntu@ubuntu-ThinkPad-Yoga-14:~$
```

Since I have GPU (GeForce 940M), to install CUDA tool kit 8.0, will have to check if the GPU is CUDA capable.

<http://docs.nvidia.com/cuda/cuda-installation-guide-linux/#axzz4VZnqTJ2A>

Run lshw to find out your CPU feature

Ubuntu 14.04 support CUDA tool kit 8.0 :

X86\_64 Ubuntu 14.04 kernel: 3.13; gcc: 4.8.2;  
glibc: 2.19

For ARMv8 CPU (aarch64) is the same version requirements.

GPU GeForce 940M is CUDA capable (compute capability 5.0, see NVDA website

<https://developer.nvidia.com/cuda-gpus>

If needs GPU driver download:

<http://www.nvidia.com/Download/index.aspx?lang=en-us>

# Cuda Compute Capability Explained (1)

<https://stackoverflow.com/questions/10961476/what-are-the-differences-between-cuda-compute-capabilities>

Feature Support	Compute Capability						
(Unlisted features are supported for all compute capabilities)	1.0	1.1	1.2	1.3	2.x	3.0	3.5, 5.0
Atomic functions operating on 32-bit integer values in global memory ( <b>Atomic Functions</b> )	No	Yes					
atomicExch() operating on 32-bit floating point values in global memory ( <b>atomicExch()</b> )							
Atomic functions operating on 32-bit integer values in shared memory ( <b>Atomic Functions</b> )	No	Yes					
atomicExch() operating on 32-bit floating point values in shared memory ( <b>atomicExch()</b> )							
Atomic functions operating on 64-bit integer values in global memory ( <b>Atomic Functions</b> )							
Warp vote functions ( <b>Warp Vote Functions</b> )							
Double-precision floating-point numbers	No			Yes			

Note: my laptop GeForce 940M, supports 5.0

# Cuda Compute Capability Explained (2)

<https://stackoverflow.com/questions/10961476/what-are-the-differences-between-cuda-compute-capabilities>

Feature Support	Compute Capability						
(Unlisted features are supported for all compute capabilities)	1.0	1.1	1.2	1.3	2.x	3.0	3.5, 5.0
Atomic functions operating on 64-bit integer values in shared memory ( <b>Atomic Functions</b> )	No				Yes		
Atomic addition operating on 32-bit floating point values in global and shared memory ( <b>atomicAdd()</b> )							
<b>__ballot()</b> ( <b>Warp Vote Functions</b> )							
<b>__threadfence_system()</b> ( <b>Memory Fence Functions</b> )							
<b>__syncthreads_count()</b> , <b>__syncthreads_and()</b> , <b>__syncthreads_or()</b> ( <b>Synchronization Functions</b> )							
Surface functions ( <b>Surface Functions</b> )							
3D grid of thread blocks							
Unified Memory Programming	No					Yes	
Funnel shift (see reference manual)	No						Yes
Dynamic Parallelism							

# CUDA Installation Guide

<http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#axzz4oHtT15GP>

Find ubuntu gcc compiler version:

```
gcc -v
```

Note: the guideline for installation

Verify the system has a CUDA-capable GPU.

Verify the system is running a supported version of Linux.

~~Verify the system has gcc installed.~~

Verify the system has the correct kernel headers and development packages installed

Download the NVIDIA CUDA Toolkit.

Handle conflicting installation methods.

Read more at: <http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#ixzz4oCNBURHX>

Follow us: @GPUComputing on Twitter | NVIDIA on Facebook

# Verify Correct Kernel Header CUDA

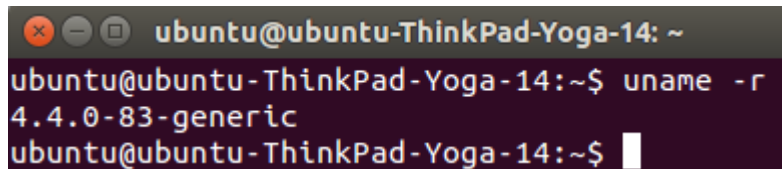
<http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#axzz4oHtT15GP>

The kernel headers and development packages has to be installed.

1. the Runfile is not good, Runfile installation performs no package validation; 2. the RPM and Deb installations of the driver will make an attempt to install the kernel header;

## 2. Find the version of your linux kernel

`uname -r`



```
ubuntu@ubuntu-ThinkPad-Yoga-14: ~  
ubuntu@ubuntu-ThinkPad-Yoga-14:~$ uname -r  
4.4.0-83-generic  
ubuntu@ubuntu-ThinkPad-Yoga-14:~$
```

This is the version of the kernel headers and development packages that must be installed prior to installing the CUDA Drivers.

To install the header and the package:

```
sudo apt-get install linux-headers-$(uname -r)
```

The NVIDIA CUDA Toolkit is available at .

Installation Instructions:

```
`sudo dpkg -i cuda-repo-ubuntu1404-8-0-local-  
ga2_8.0.61-1_amd64.deb`
```

```
`sudo apt-get update`
```

```
`sudo apt-get install cuda`
```

Make sure do download verification:  
? ? ?

add CUDA to the PATH

```
export PATH=/usr/local/cuda-8.0/bin$  
{PATH:+:${PATH}}
```

# Export PATH and Profiler Installation

PATH is a global OS variable contains names of files to be executed without specifying the whole path. For example You can just write startx to start graphic environment instead of /bin/some other folders/startx

Run `. ~/.profile` for changes to take immediate effect

To add a directory to your \$PATH, follow any one of the below. (example, to add '/usr/hitech/picc/9.82/bin/picc')

simply edit ~/.profile

```
gedit ~/.profile
find the following line:
PATH="$HOME/bin:$PATH"
and change it to:
PATH="$HOME/bin:
$PATH:/usr/hitech/picc/9.82/bin"
```

Or run the below command in terminal  
`export PATH=$PATH:/usr/hitech/picc/9.82/bin`

```
export PATH=/usr/local/cuda-
8.0/bin${PATH:+:${PATH}}
```

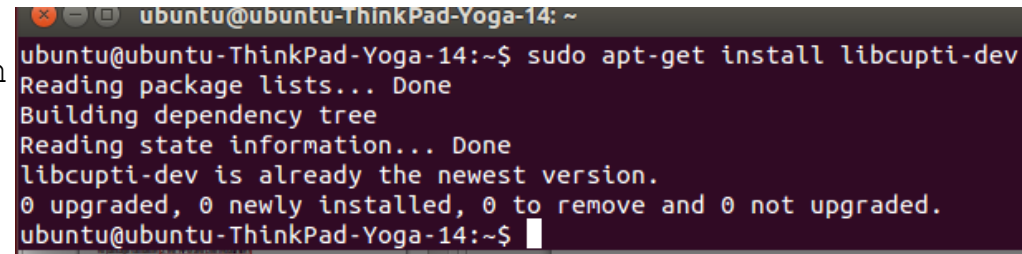
The libcupti-dev library, which is the NVIDIA CUDA Profile Tools Interface. This library provides advanced profiling support. To install this library, issue the following command

[https://www.tensorflow.org/install/install\\_linux](https://www.tensorflow.org/install/install_linux)

Note: when I did the above for libcupti-dev, I have got a some kind symbolic link error message, after searching google, I did:

```
sudo apt-get update
sudo apt-get upgrade
```

Then retry install libcupti-dev, the error is gone.



```
ubuntu@ubuntu-ThinkPad-Yoga-14: ~
ubuntu@ubuntu-ThinkPad-Yoga-14:~$ sudo apt-get install libcupti-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libcupti-dev is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntu-ThinkPad-Yoga-14:~$
```

# cuDNN v5.1 Installation

Deep Neural Network library (cuDNN)

cuDNN accelerates widely used deep learning frameworks, including , , , , and . See for more details.



theano

<https://developer.nvidia.com/rdp/form/cudnn-download-survey>



# Tensor Flow Installation on Ubuntu 14.04

<https://www.tensorflow.org/tutorials/>

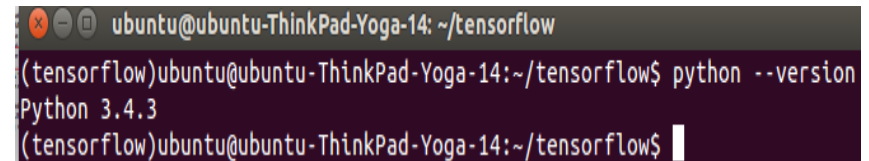
After all the prerequisite, now follow the tensor flow installation recommendation, go with virtualenv installation for isolated python environment. 5 Steps from the tensor flow tutorial:  
Step 1: `sudo apt-get install python3-pip python3-dev python-virtualenv`

Step 2: `virtualenv --system-site-packages -p python3 ~/tensorflow`

Step 3: `$source ~/tensorflow/bin/activate` # bash, sh, ksh, or zsh (note, you can use the following for your choice:  
`$ source ~/tensorflow/bin/activate.csh` # csh or tcsh) if it is successful, then The preceding source command should change your prompt to the following:  
(tensorflow)\$

Note: check your python version

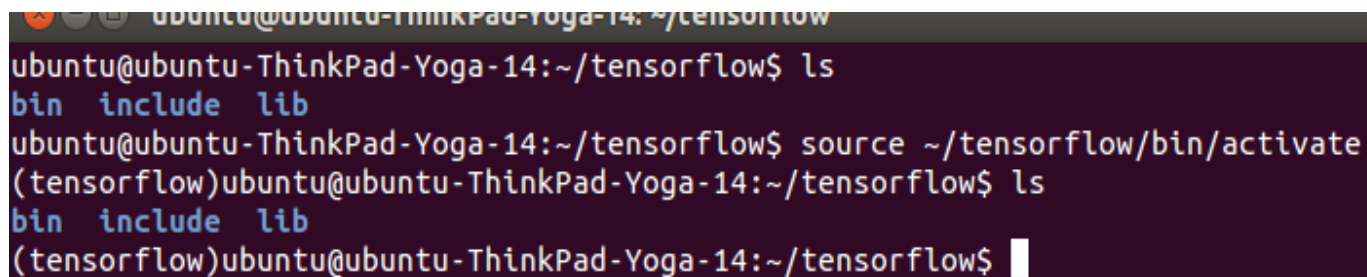
`$python --version`



```
ubuntu@ubuntu-ThinkPad-Yoga-14: ~/tensorflow
(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/tensorflow$ python --version
Python 3.4.3
(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/tensorflow$
```

Step 4: `pip3 install --upgrade tensorflow-gpu`  
But had error message of no download was found, so upgrade pip and tensorflow as follows, then install again, the error was gone.

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



```
ubuntu@ubuntu-ThinkPad-Yoga-14: ~/tensorflow
ubuntu@ubuntu-ThinkPad-Yoga-14:~/tensorflow$ ls
bin  include  lib
ubuntu@ubuntu-ThinkPad-Yoga-14:~/tensorflow$ source ~/tensorflow/bin/activate
(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/tensorflow$ ls
bin  include  lib
(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/tensorflow$
```



# Activate and Deactivate Tensor Flow

To activate:

```
$ source ~/tensorflow/bin/activate      # bash, sh, ksh, or zsh  
$ source ~/tensorflow/bin/activate.csh # csh or tcsh
```

To deactivate:

```
$deactivate
```

Your prompt will become the following to indicate that your tensorflow environment is active:

```
(tensorflow) $
```

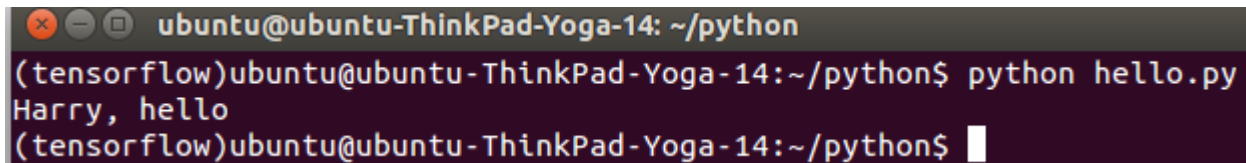
# Your First Python Program

The program first.py

```
print("Harry, hello")
```

To run it:

\$python first.py

A terminal window with a dark background and light-colored text. The window title is 'ubuntu@ubuntu-ThinkPad-Yoga-14: ~/python'. The prompt is '(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/python\$'. The command 'python hello.py' has been entered and executed, resulting in the output 'Harry, hello'. The prompt is now '(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/python\$' with a cursor at the end.

```
(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/python$ python hello.py
Harry, hello
(tensorflow)ubuntu@ubuntu-ThinkPad-Yoga-14:~/python$
```