

# Installing Tensorflow GPU on Ubuntu 18.04 LTS

I wanted to detail here what I did to get tensorflow-gpu working with my fresh Ubuntu 18.04 LTS install. NVIDIA doesn't have any official downloads for Ubuntu 18.04 yet, but you can get things to work with the available files for Ubuntu 17.04.

## Check your NVIDIA driver version

The first thing you should check is that you have an Nvidia driver installed for your graphics card. Your graphics card must support at least Nvidia compute 3.0 to install tensorflow-gpu.

You can check what graphics driver you have installed with the `nvidia-smi` command. You should see some output like the following:

```
→ nvidia-smi
Fri Apr 27 15:43:16 2018
+-----+
| NVIDIA-SMI 390.48                  Driver Version: 390.48          |
+-----+-----+-----+-----+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+=====+=====+=====+=====+
|    0  GeForce GTX 1050      Off   | 00000000:01:00.0 Off  |          N/A         |
| N/A   49C    P0      N/A /  N/A |    538MiB /  4042MiB |         1%      Default |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:                         GPU Memory |
|   GPU       PID    Type    Process name      Usage   |
|=====+=====+=====+=====+=====+=====+
|    0       1569     G   /usr/lib/xorg/Xorg        28MiB   |
+-----+-----+-----+-----+-----+-----+

```

The driver version you have installed is near the top left next to "NVIDIA-SMI". I've got nvidia-390 installed.

If you don't have a proper driver installed, go do that now.

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

```
$ sudo apt install nvidia-390
```

```
$ reboot
```

## Install CUDA Toolkit 9.0

- gcc/g++ --version 확인 후 6 이하로 설정할 것

Head over to <https://developer.nvidia.com/cuda-toolkit> and grab the the runfile download for Ubuntu 17.04. While this is for a different version of Ubuntu, you can get it to install what you need. You'll have to go to the legacy downloads archive page to find version 9.0.

[Home](#) > [ComputeWorks](#) > [CUDA Toolkit](#) > [CUDA Toolkit Archive](#) > [CUDA Toolkit 9.0 Downloads](#)

## CUDA Toolkit 9.0 Downloads

### Select Target Platform ?

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX			
Architecture <span>?</span>	x86_64	ppc64le				
Distribution	Fedora	OpenSUSE	RHEL	CentOS	SLES	Ubuntu
Version	17.04	16.04				
Installer Type <span>?</span>	runfile (local)	deb (local)	deb (network)			

### Download Installers for Linux Ubuntu 17.04 x86\_64

The base installer is available for download below.  
There are 2 patches available. These patches require the base installer to be installed first.

> Base Installer

Download (1.6 GB) 

Installation Instructions:

1. Run `sudo sh cuda_9.0.176_384.81_linux.run`
2. Follow the command-line prompts

Runfile download page. Grab the file from the Base Installer link.

Once you've got that file, navigate to where the file was downloaded in your terminal and do

```
sudo chmod +x cuda_9.0.176_384.81_linux.run
./cuda_9.0.176_384.81_linux.run --override
```

Accept the terms and conditions, say  to installing with an unsupported configuration, and  to "Install NVIDIA Accelerated Graphics Driver for Linux-x86\_64 384.81?". **Make sure you don't agree to install the new driver.** In my experience, doing so often leads to system instability issues. Follow the prompts to install the toolkit using the default install locations.

## Install CUDNN 7.0

Next, head to <https://developer.nvidia.com/cudnn> to get CUDNN 7.0. Go to the downloads archive page again and find version 7.0 for CUDA 9.0 that you just installed. Download the link that says "[cuDNN v7.0.5 Library for Linux](#)". This will download an archive that you can unpack and move the contents the correct locations.

[Download cuDNN v7.1.2 \[Mar 21, 2018\], for CUDA 9.0](#)

[Download cuDNN v7.1.2 \[Mar 21, 2018\], for CUDA 8.0](#)

[Download cuDNN v7.1.1 \[Feb 28, 2018\], for CUDA 9.1](#)

[Download cuDNN v7.1.1 \[Feb 28, 2018\], for CUDA 9.0](#)

[Download cuDNN v7.1.1 \[Feb 28, 2018\], for CUDA 8.0](#)

[Download cuDNN v7.0.5 \[Dec 11, 2017\], for CUDA 9.1](#)

[Download cuDNN v7.0.5 \[Dec 5, 2017\], for CUDA 9.0](#)

[cuDNN Developer Guide](#)

[cuDNN Install Guide](#)

[cuDNN Release Notes](#)

[cuDNN v7.0.5 Library for Linux](#)

[cuDNN v7.0.5 Library for Linux \(Power8\)](#)

[cuDNN v7.0.5 Library for Windows 7](#)

[cuDNN v7.0.5 Library for Windows 10](#)

[cuDNN v7.0.5 Runtime Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN v7.0.5 Developer Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN v7.0.5 Code Samples and User Guide for Ubuntu16.04 \(Deb\)](#)

[cuDNN v7.0.5 Runtime Library for Ubuntu14.04 \(Deb\)](#)

[cuDNN v7.0.5 Developer Library for Ubuntu14.04 \(Deb\)](#)

There are lots of options on the archive downloads page for CUDNN. Get the Library for Linux file for CUDA 9.0.

Once downloaded, unpack the archive and move it the contents into the directory where you install CUDA 9.0:

```
# Unpack the archive
tar -zxvf cudnn-9.0-linux-x64-v7.tgz
```

```
# Move the unpacked contents to your CUDA directory
sudo cp -P cuda/lib64/libcudnn* /usr/local/cuda-9.0/lib64/
sudo cp  cuda/include/cudnn.h /usr/local/cuda-9.0/include/
```

```
# Give read access to all users
sudo chmod a+r /usr/local/cuda-9.0/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
```

## Install libcupti

This one is easy.

```
sudo apt-get install libcupti-dev
```

## Do the CUDA post-install actions

So Tensorflow can find your CUDA installation and use it properly, you need to add these lines to the end of you `~/.bashrc` or `~/.zshrc`.

```
export PATH=/usr/local/cuda-9.0/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

Restart your terminal before proceeding to the next step.

```
$ nvcc --version
```

## Install Tensorflow GPU

Finally, to install tensorflow-gpu run

```
pip install --upgrade tensorflow-gpu

sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
```

I recommend installing tensorflow in a virtualenv to prevent having to muck around with your system Python packages. The [official Tensorflow install instructions](#) give various options, so you can choose what works best for you. If you choose the virtualenv route, I highly recommend using [virtualenvwrapper](#), which makes using virtualenv far easier.

<https://medium.com/@taylordenouden/installing-tensorflow-gpu-on-ubuntu-18-04-89a142325138>