These instructions are intended to set up a deep learning environment for GPU-powered tensorflow.  
[See here for pytorch GPU install instructions](https://github.com/williamFalcon/pytorch-gpu-install)

After following these instructions you'll have:

1. Ubuntu 16.04.
2. Cuda 9.0 drivers installed.
3. A conda environment with python 3.6.
4. The latest tensorflow version with gpu support.

**Step 0: Noveau drivers**

Before you begin, you may need to disable the opensource ubuntu NVIDIA driver called [nouveau](https://nouveau.freedesktop.org/wiki/).

**Option 1: Modify modprobe file**

1. After you boot the linux system and are sitting at a login prompt, press ctrl+alt+F1 to get to a terminal screen. Login via this terminal screen.
2. Create a file: /etc/modprobe.d/nouveau
3. Put the following in the above file...

blacklist nouveau

options nouveau modeset=0

1. reboot system

reboot

1. On reboot, verify that noveau drivers are not loaded

lsmod | grep nouveau

If nouveau driver(s) are still loaded do not proceed with the installation guide and troubleshoot why it's still loaded.

**Option 2: Modify Grub load command**  
From [this stackoverflow solution](https://askubuntu.com/questions/697389/blank-screen-ubuntu-15-04-update-with-nvidia-driver-nomodeset-does-not-work)

1. When the GRUB boot menu appears : Highlight the Ubuntu menu entry and press the E key. Add the nouveau.modeset=0 parameter to the end of the linux line ... Then press F10 to boot.
2. When login page appears press [ctrl + ALt + F1]
3. Enter username + password
4. Uninstall every NVIDIA related software:

sudo apt-get purge nvidia\*

sudo reboot

**Installation steps**

1. update apt-get

sudo apt-get update

1. Install apt-get deps

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 curl

1. install nvidia drivers

# The 16.04 installer works with 16.10.

# download drivers

curl -O http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86\_64/cuda-repo-ubuntu1604\_9.0.176-1\_amd64.deb

# download key to allow installation

sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86\_64/7fa2af80.pub

# install actual package

sudo dpkg -i ./cuda-repo-ubuntu1604\_9.0.176-1\_amd64.deb

# install cuda (but it'll prompt to install other deps, so we try to install twice with a dep update in between

sudo apt-get update

sudo apt-get install cuda-9-0

2a. reboot Ubuntu

sudo reboot

2b. check nvidia driver install

nvidia-smi

# you should see a list of gpus printed

# if not, the previous steps failed.

1. Install cudnn

wget https://s3.amazonaws.com/open-source-william-falcon/cudnn-9.0-linux-x64-v7.3.1.20.tgz

sudo tar -xzvf cudnn-9.0-linux-x64-v7.3.1.20.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\*

1. Add these lines to end of ~/.bashrc:

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64"

export CUDA\_HOME=/usr/local/cuda

export PATH="$PATH:/usr/local/cuda/bin"

4a. Reload bashrc

source ~/.bashrc

1. Install miniconda

wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86\_64.sh

bash Miniconda3-latest-Linux-x86\_64.sh

# press s to skip terms

# Do you approve the license terms? [yes|no]

# yes

# Miniconda3 will now be installed into this location:

# accept the location

# Do you wish the installer to prepend the Miniconda3 install location

# to PATH in your /home/ghost/.bashrc ? [yes|no]

# yes

5a. Reload bashrc

source ~/.bashrc

1. Create conda env to install tf

conda create -n tensorflow

# press y a few times

1. Activate env

source activate tensorflow

1. Install tensorflow with GPU support for python 3.6

pip install tf-nightly-gpu

# If the above fails, try the part below

# pip install --ignore-installed --upgrade https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\_gpu-1.2.0-cp36-cp36m-linux\_x86\_64.whl

1. Test tf install

# start python shell

python

# run test script

import tensorflow as tf

hello = tf.constant('Hello, TensorFlow!')

# when you run sess, you should see a bunch of lines with the word gpu in them (if install worked)

# otherwise, not running on gpu

sess = tf.Session()

print(sess.run(hello))