Install Python on Ubuntu (Anaconda)

The way below utilizes bash scripts which is a faster way to install anaconda. This should work on Ubuntu 12.04 (precise), 14.04 (trusty), and 16.04 (xenial).

1. Open a new terminal.
2. Copy and paste the paste commands from either gist below on the terminal.

```
# Go to home directory
cd ~

# You can change what anaconda version you want at
# https://repo.continuum.io/archive/
wget https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh
bash Anaconda3-5.0.1-Linux-x86_64.sh -b -p ~/anaconda
rm Anaconda3-5.0.1-Linux-x86_64.sh
echo 'export PATH="~/anaconda/bin:$PATH"' >> ~/.bashrc

# Refresh basically
source .bashrc

conda update conda
```

3. The files are from the Anaconda installer archive.

   https://repo.continuum.io/archive/

Pip install

1. Install Python, `pip`, and `virtualenv`.

On Ubuntu, Python is automatically installed and `pip` is *usually* installed. Confirm the `python` and `pip` versions:

```
  python -V  # or: python3 -V

  pip -V     # or: pip3 -V
```

To install these packages on Ubuntu:

```
  sudo apt-get install python-pip python-dev python-virtualenv   # for Python 2.7

  sudo apt-get install python3-pip python3-dev python-virtualenv # for Python 3.n
```

We *recommend* using `pip` version 8.1 or higher. If using a release before version 8.1, upgrade `pip`:

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

If not using Ubuntu and setuptools is installed, use `easy_install` to install `pip`:

```
  easy_install -U pip
```

2. Create a directory for the virtual environment and choose a Python interpreter.

```
  mkdir ~/tensorflow  # somewhere to work out of

  cd ~/tensorflow
```

```
  # Choose one of the following Python environments for the ./venv
directory:

  virtualenv --system-site-packages venv            # Use python default
(Python 2.7)

  virtualenv --system-site-packages -p python3 venv # Use Python 3.n
```

3. Activate the Virtualenv environment.

Use one of these shell-specific commands to activate the virtual environment:

```
  source ~/tensorflow/venv/bin/activate      # bash, sh, ksh, or zsh

  source ~/tensorflow/venv/bin/activate.csh  # csh or tcsh

  . ~/tensorflow/venv/bin/activate.fish      # fish
```

When the Virtualenv is activated, the shell prompt displays as `(venv) $`.

4. Upgrade `pip` in the virtual environment.

Within the active virtual environment, upgrade `pip`:

```
(venv)$ pip install --upgrade pip
```

You can install other Python packages within the virtual environment without affecting packages outside the `virtualenv`.

5. Install TensorFlow in the virtual environment.

Choose one of the available TensorFlow packages for installation:

- `tensorflow` —Current release for CPU
- `tensorflow-gpu` —Current release with GPU support
- `tf-nightly` —Nightly build for CPU

- `tf-nightly-gpu` —Nightly build with GPU support

Within an active Virtualenv environment, use `pip` to install the package:

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

Use `pip list` to show the packages installed in the virtual environment. Validate the install and test the version:

```
(venv)$ python -c "import tensorflow as tf; print(tf.__version__)"
```