

Prepare EC2 P2 instance for tensorflow

```
1 sudo apt-get update
2 sudo apt-get upgrade
3 sudo apt-get install -y build-essential git python-pip libfreetype6-dev libxft-dev
  libncurses-dev libopenblas-dev gfortran python-matplotlib libblas-dev liblapack-dev
  libatlas-base-dev python-dev python-pydot linux-headers-generic linux-image-extra-virtual
  unzip python-numpy swig python-pandas python-sklearn unzip wget pkg-config zip g++ zlib1g-
  dev libcurl3-dev
4 sudo pip install -U pip
```

Install CUDA

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

get latest cuda

```
1 wget https://developer.nvidia.com/compute/cuda/8.0/Prod2/local_installers/cuda-repo-
  ubuntu1604-8-0-local-ga2_8.0.61-1_amd64-deb
2 sudo dpkg -i cuda-repo-ubuntu1604-8-0-local-ga2_8.0.61-1_amd64-deb
3 rm cuda-repo-ubuntu1604-8-0-local-ga2_8.0.61-1_amd64-deb
4 sudo apt-get update
5 sudo apt-get install -y cuda
```

Install CuDNN

cuDNN v5.1

You need to login to Nvidia as developer download them and upload it to the EC2 server.

```
1 sudo dpkg -i libcudnn5_5.1.5-1+cuda8.0_amd64.deb
2 sudo dpkg -i libcudnn5-dev_5.1.5-1+cuda8.0_amd64.deb

1 # add to .profile
2 export CUDA_HOME=/usr/local/cuda
3 export CUDA_ROOT=/usr/local/cuda
4 export PATH=$PATH:$CUDA_ROOT/bin
5 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CUDA_ROOT/lib64
```

Install Bazel

```
1 $ sudo apt-get install software-properties-common swig
2 $ sudo add-apt-repository ppa:webupd8team/java
3 $ sudo apt-get update
4 $ sudo apt-get install oracle-java8-installer
5 $ echo "deb http://storage.googleapis.com/bazel-apt stable jdk1.8" | sudo tee
   /etc/apt/sources.list.d/bazel.list
6 $ curl https://storage.googleapis.com/bazel-apt/doc/apt-key.pub.gpg | sudo apt-key add -
7 $ sudo apt-get update
8 $ sudo apt-get install bazel
```

Get Tensorflow

```
1 git clone --recurse-submodules https://github.com/tensorflow/tensorflow
2 git checkout -b latest-stable-version
3 ./configure
```

Do you wish to build TensorFlow with GPU support? [y/N] y
Please specify the Cuda SDK version you want to use, e.g. 7.0. [Leave empty to use system default]: 8.0

Please specify the Cudnn version you want to use. [Leave empty to use system default]: 5.1.5

Please specify the location where cuDNN 5.1.5 library is installed. Refer to README.md for more details. [Default is /usr/local/cuda]: /usr/lib/x86_64-linux-gnu

Please specify a list of comma-separated Cuda compute capabilities you want to build with.

You can find the compute capability of your device at: <https://developer.nvidia.com/cuda-gpus>.

Please note that each additional compute capability significantly increases your build time and binary size.

[Default is: "3.5,5.2"]: 3.7

```
1 bazel build -c opt --config=cuda //tensorflow/tools/pip_package:build_pip_package
2 bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
3 sudo pip install --upgrade /tmp/tensorflow_pkg/tensorflow-0.11.0rc0-py2-none-any.whl
4 # to validate
5 # checkout model repository
6 python ~/models/tutorials/image/mnist/convolutional.py
```

Reference

- <http://expressionflow.com/2016/10/09/installing-tensorflow-on-an-aws-ec2-p2-gpu-instance/>
- <http://www.nvidia.com/object/gpu-accelerated-applications-tensorflow-installation.html>