Compiling Tensorflow from source for MacOS

1. Install Xcode and Xcode command line tools.
2. **[Optional]** Disable SIP following these instructions. Enable it back once everything is compiled and running.
3. Install homebrew.
4. Install bazel (Needs JDK).
5. Install conda.
6. Then in Terminal: `$ sudo pip install six numpy wheel`
7. `$ git clone https://github.com/tensorflow/tensorflow`
8. `$ cd tensorflow`
9. `$ git checkout r1.4` (change it to current verion)
10. `$ bazel clean`
11. `$ ./configure`
12. You'll get something like this, just install what you need (The below log is for GPU):

```
Please specify the location of python. [Default is /usr/bin/python]: /usr/bi
n/python2.7
Found possible Python library paths:
  /usr/local/lib/python2.7/dist-packages
  /usr/lib/python2.7/dist-packages
Please input the desired Python library path to use.  Default is [/usr/lib/p
ython2.7/dist-packages]

Using python library path: /usr/local/lib/python2.7/dist-packages
Do you wish to build TensorFlow with MKL support? [y/N]
No MKL support will be enabled for TensorFlow
Please specify optimization flags to use during compilation when bazel optio
n "--config=opt" is specified [Default is -march=native]:
Do you wish to use jemalloc as the malloc implementation? [Y/n]
jemalloc enabled
Do you wish to build TensorFlow with Google Cloud Platform support? [y/N]
No Google Cloud Platform support will be enabled for TensorFlow
Do you wish to build TensorFlow with Hadoop File System support? [y/N]
No Hadoop File System support will be enabled for TensorFlow
Do you wish to build TensorFlow with the XLA just-in-time compiler (experime
ntal)? [y/N]
```

```
No XLA support will be enabled for TensorFlow
Do you wish to build TensorFlow with VERBS support? [y/N]
No VERBS support will be enabled for TensorFlow
Do you wish to build TensorFlow with OpenCL support? [y/N]
No OpenCL support will be enabled for TensorFlow
Do you wish to build TensorFlow with CUDA support? [y/N] Y
CUDA support will be enabled for TensorFlow
Do you want to use clang as CUDA compiler? [y/N]
nvcc will be used as CUDA compiler
Please specify the Cuda SDK version you want to use, e.g. 7.0. [Leave empty
to default to CUDA 8.0]: 8.0
Please specify the location where CUDA 8.0 toolkit is installed. Refer to RE
ADME.md for more details. [Default is /usr/local/cuda]:
Please specify which gcc should be used by nvcc as the host compiler. [Defau
lt is /usr/bin/gcc]:
Please specify the cuDNN version you want to use. [Leave empty to default to
 cuDNN 6.0]: 6
Please specify the location where cuDNN 6 library is installed. Refer to REA
DME.md for more details. [Default is /usr/local/cuda]:
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.nvi
dia.com/cuda-gpus.
Please note that each additional compute capability significantly increases
your build time and binary size.

Do you wish to build TensorFlow with MPI support? [y/N]
MPI support will not be enabled for TensorFlow
Configuration finished
```

1. Leave this answer in its default (Just press enter when this is asked) : `Please specify optimization flags to use during compilation when bazel option "--config=opt" is specified [Default is -march=native]`

2. Then :

   ```
   $ bazel build -c opt --copt=-mavx --copt=-mavx2 --copt=-mfma --copt=-mss
   e4.1 --copt=-msse4.2 --config=opt -k //tensorflow/tools/pip_package:buil
   d_pip_package
   ```

   It is recommended to use `--config=opt` as it would build specifically for the host cpu. So the build command would be:

   ```
   $ bazel build --config=opt //tensorflow/tools/pip_package:build_pip_pack
   age
   ```

If this fails due to bazel issues, append `--incompatible_load_argument_is_label=false` to the build command. It would become:

```
$ bazel build --config=opt --incompatible_load_argument_is_label=false /
/tensorflow/tools/pip_package:build_pip_package
```

3. Build the wheel file:

```
$ bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorfl
ow_pkg
```

4. Finally (change the .whl file as generated by your cpu, its found in tmp) :

```
$ pip install --upgrade --ignore-installed /tmp/tensorflow_pkg/tensorflo
w-1.4.1-cp36-cp36m-macosx_10_7_x86_64.whl
```

For GPU installation:

Follow this guide.

Debug Stuff:

Compiled Wheel File is found in:

> Macintosh HD->tmp->tensorflow_pkg
> `/private/tmp/tensorflow_pkg`

Some permission problems can be solved using:

```
$ sudo chown -R USERNAME /Users/USERNAME/FOLDER
```

References:

Official Guide
Stackoverflow