

## NVIDIA Docker で TensorFlow

Nvidia の Docker イメージ `nvidia/cuda:9.1-cudnn7-devel-ubuntu16.04` で TensorFlow をビルドしようとしたら若干手間どってしまったのでメモ

(0) 準備 (ここは問題ない)

```
apt update && apt upgrade -y
apt install git python python-dev python3 python3-dev python3-pip -y
apt install curl openjdk-8-jdk sudo vim -y
pip3 install --upgrade pip
pip3 install tensorflow-gpu
pip3 remove tensorflow-gpu
```

で必要なものをいれて、

<https://docs.bazel.build/versions/master/install-ubuntu.html>

に従って Bazel をインストール。

で、[https://www.tensorflow.org/versions/master/install/install\\_sources](https://www.tensorflow.org/versions/master/install/install_sources) に従ってビルド ... すると、`libcuda.so.1` がない、とかいわれた。

みると、`/usr/local/cuda/lib64/stubs` 以下に

いくつかのライブラリファイルが格納されているので、`/etc/ld.so.conf/nvidia.conf` に

```
/usr/local/cuda/lib64
/usr/local/cuda/lib64/stubs
```

を追加して、`ldconfig` を実行。再び、

```
./configure
bazel build --config=opt --config=cuda //tensorflow/tools/pip_package:build_pip_package
bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
```

とすると、とりあえずビルドできたので、よかった。

## RISC-V のベクの命令

[RISC V Vector Extension Proposal](#) を視聴してのメモ。

- 32bit\*8 なベクトルレジスタを 32 本もつ (... の?)
- VL で、演算対象のレジスタを選択できる
- SIMD じゃないので、利用可能な演算器を再利用しながらぐるぐるまわす
  - 全部を 1 クロックでできる = SIMD (08:20 あたり)
- mask で if もベクトル演算で効率良く
  - ベクトルレジスタの対応するスロットの LSB でマスク、など (12:10 あたり)
- ストア / ロードで工夫
  - ex. `vlds` で、メモリからストライド指定してベクトルレジスタにデータをストアできる (14:20 あたり)