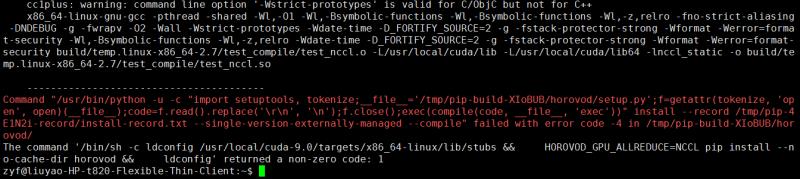
$ mkdir horovod-docker

$ wget -O horovod-docker/Dockerfile https://raw.githubusercontent.com/uber/horovod/master/Dockerfile

$ docker build -t horovod:latest horovod-docker

执行到一半是会出现如下错误：



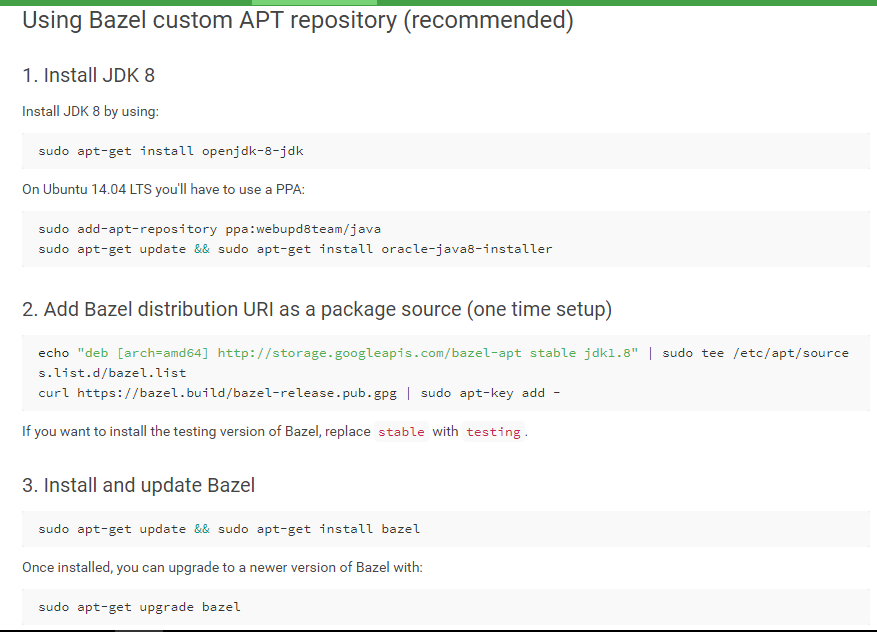
此时把错误往上翻看下是第几个step报错的，我这显示是在step 13，所以可以利用step 12建立好的镜像来创建一个容器来调试该错误，即 docker run -it step12的镜像名

进入容器后此时手动执行 ldconfig /usr/local/cuda-9.0/targets/x86\_64-linux/lib/stubs

HOROVOD\_GPU\_ALLREDUCE=NCCL pip install --no-cache-dir horovod 以及 ldconfig 发现并无报错，但是在python中import tensorflow 测试时发现之前pip install的tensorflow-gpu=1.6 与系统并不兼容(coe dump), 故pip uninstall tensorflow-gpu 准备源码安装tensorflow1.6

源码安装tensorflow过程如下：

1. 按如下所示安装Bazel



2. **pip install six numpy wheel**

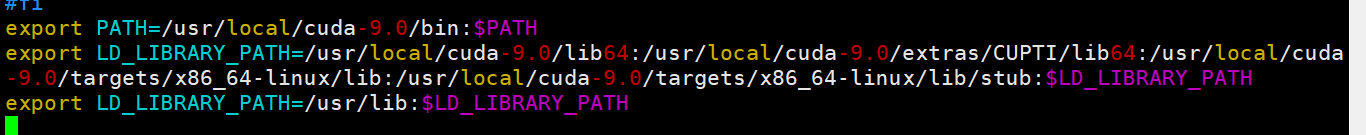
3. **git clone https://github.com/tensorflow/tensorflow**

**cd tensorflow**

**git checkout r1.6**

**apt-get install python-numpy python-dev python-pip python-wheel**

**在~/.bashrc里写入如下环境变量：**



**cd tensorflow**

**./configure**

此过程中除了自己定义cudnn的路径以及不安装OPENCL外其他都回车默认。

**bazel build --config=opt --config=cuda //tensorflow/tools/pip\_package:build\_pip\_package**

可能会遇到两个Bug:

1. 没有找到cudnn.h 。 执行find / -name “cudnn.h” 如果也没有则在/usr/include下新建一个cudnn.h文件，然后从nvivida官网下载cuda9匹配的cudnn7将其cudnn.h里的内容复制进去即可。
2. 执行bazel build时没有找到libcuda.so.1 。 虽然这个文件在/usr/local/cuda-9.0/x86\_64-linux/lib/stub下且之前也已写入到环境变量中，而且也执行过ldconfig，但既然还是报错找不到该文件，那么一个解决方案就是把该目录下的libcuda\* 两个文件复制到/usr/lib目录下再重建软链接，最后执行ldconfig

此时执行 **bazel clean 在重新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**

**pip install /tmp/tensorflow\_pkg/\*.whl**

mkdir \_python\_build

cd \_python\_build

ln -s ../bazel-bin/tensorflow/tools/pip\_package/build\_pip\_package.runfiles/org\_tensorflow/\* .

ln -s ../tensorflow/tools/pip\_package/\* .

python setup.py develop

最后执行python并import tensorflow无报错即可。

此时再手动执行一遍Dockerfile剩下的语句即可，即从ldconfig /usr/local/cuda-9.0/targets/x86\_64-linux/lib/stubs这局开始，注意把RUN去掉，

最后的 WORKDIR "/examples" 即 cd examples,测试hovorod的例子无报错即可。最后docker commit。