**This file explains the process of checking whether your backend is being hit by the python file or not.**

**There are two servers:**

1) Build server

2) Test Server

Build server - contains the code of .cpp files.

Test server - here we write our python code to hit the backend of cpp.

**Working of build server:**

Following are steps to run full build of tops for generating x86\_64-linux-rel folder:

1. Create your workspace. (mkdir workspace\_name)
2. mention your date and create a directory name cmake\_build (You can choose any name, but generally we are preferring cmake\_build name).
3. Create today’s date folder.
4. Inside date specific folder of workspace, clone tops using the below command:

git clone "ssh://apurva@gerrit.enflame.cn:29418/tops" && (cd "tops" && curl -sL http://build.enflame.cn/.init.sh | bash)

**Note: In the above command, apurva must be renamed with the username you have logged in to your build server.**

1. Inside tops, run the following command:

* git submodule update --init –recursive

1. Inside the cmake\_build folder, run the following commands:

* cmake -S ../tops/ -G Ninja -DINCLUDE\_TOPSIDEAS=OFF 2>&1|tee ../config.log
* ninja install 2>&1|tee ../install.log
* ninja package\_all 2>&1|tee ../package\_all.log

1. REPEAT STEP 3, 4, 5, and 6 and sub once in a week.
2. Generate “.whl” file. For generation, follow the below steps:

* Go to cmake\_build folder where your x86\_64-linux-rel contains.
* Now, run command: ninja torch\_gcu3.6 2>&1 | tee ../logs/ninja.log in build folder. After running the command, your “.whl” file will be generated.

1. Now, run command: readlink -f "path-of-whl-package" and copy the path.
2. REPEAT step 8 and 9, whenever you make any changes to the .cpp file code.

/home/niraj/MS\_workspace/20\_07\_/cmake\_build/backend/torch\_dtu/lib/10/python3.6/torch\_gcu-1.10.0-2.4.1-py3.6-none-any.whl

**Working of TEST server:**

1. To go into test server, run command: ssh -p 22 [matthew.he@10.12.114.242](mailto:matthew.he@10.12.114.242)
2. Go to Aditya folder, and create your workspace, within the workspace create the today’s date folder.
3. Now, we have to copy the x86\_64-linux-rel folder path to date folder. Use the below command:

sudo scp -r [niraj@10.9.114.31:/home/niraj/MS\_workspace/1907/build/x86\_64-linux-rel ./](mailto:niraj@10.9.114.31:/home/niraj/MS_workspace/1907/build/x86_64-linux-rel%20./)

Syntax: sudo scp -r [username@ip:/path-of-x86-from-build ./](mailto:username@ip:/path-of-x86-from-build%20./)

1. Go to x86 folder.
2. install kmd. Command: ctrl + r krke kmd type krne se aa jaegi

Note: Whenever we create a new x86 folder, a new docker will be created.

1. Create a new docker, below are the steps:

* ctrl + r and type sudo docker run, and automatically enters. exit keyword is used to come out of the docker.
* sudo docker start docker\_name
* sudo docker ps -a (to check the name of your docker)
* sudo docker exec -ti docker\_name bash (to enter into your docker)

1. Go to x86 folder of test server and run the command for package install.
2. Go to one directory back and copy the whl file path from build server to this path using the scp command: scp -r niraj@10.9.114.31:/home/niraj/MS\_workspace/1407/cmake\_build/x86\_64-linux-rel .
3. Now, you need to unistall the already existing wheel package using the command:

python -m pip uninstall torch\_gcu

1. Now, run the install command within the same folder:

python -m pip install torch\_gcu-1.10.0-2.2.1-py3.6-none-any.whl

1. Run run your python test case file.