Step -1 :- Enter in CCF environment (open the terminal)

```
$ ssh username   $
$ password       $
```

This will provide by your guide only.
Now you will be in CCF virtual environment

Step-2 :- Set proxy setup there

```
export HTTP_PROXY=http://pro2017001:2G@ur@v8@172.31.2.3:8080/
export HTTPS_PROXY=http://pro2017001:2G@ur@v8@172.31.2.3:8080/
```

Step-3:- Create a virtual environment there for your requirements

```
$ conda create -n envname python=3.6
```

Step-4:- Activate your virtual environment

```
$ conda activate file_name
```

Step-5:- Install all dependencies required for your code

Step-6:-  Create a folder there

```
$ mkdir foldername
```

Step-7:- import youfile from local directory to CCF folder

```
$ rsync -avz pathoffolderinlocal directory usernameofCCF destination_path
```

Like for me
If you don't know how to know destination_path simply type ther $ pwd

```
$ rsync -avz /home/gaurav/Desktop/ResearachProgress/July-August/papers/ohmp/code/hmp gcnandi@172.20.70.12:/home/gcnandi/Gaurav/
```

Step-8:- for running your code

```
$ qsub submit.sh
```

Make a bash file there and write down script of programm which you want to run

Step-9 for checking your programe is running or not

```
$ qstat or qstat -n
```

Step 10:- Export your code from ccf to local machine directory
Open new terminal

```
$ rsync -avz  gcnandi@172.20.70.12:/home/gcnandi/Gaurav /home/gaurav/Desktop/ResearachProgress/July-August/papers/ohmp/code/hmp
$ rsync -avz gcnandi@172.20.70.12:/home/gcnandi/Gaurav/hmp/samples.h5 /home/gaurav/Desktop/ResearachProgress/July-August/papers/ohmp/code/results_of_paper/100000/our_results/epochs=0.6/
```

Step -11:-
To kill the job:
Syntax: `qdel -Wforce <jobid>` (Example: `qdel -Wforce 22276.agni` )

Step-12
To check allotment of GPU

```
ssh nodeNO
nvidia-smi
```

For running code in ccf you have need to add some line in your code
If your code in Tensorflow then write down below lines

```
------------------------------------------------------------------------------------------------------------------------
session_conf = tf.ConfigProto(intra_op_parallelism_threads=1, inter_op_parallelism_threads=1)
tf.set_random_seed(1234)
tf.Session(graph=tf.get_default_graph(), config=session_conf)
------------------------------------------------------------------------------------------------------------------------
```

If your code in Keras

```
------------------------------------------------------------------------------------------------------------------------
import tensorflow as tf
from keras import backend as K
session_conf = tf.ConfigProto(intra_op_parallelism_threads=1, inter_op_parallelism_threads=1)
tf.set_random_seed(1234)
sess = tf.Session(graph=tf.get_default_graph(), config=session_conf)
K.set_session(sess)
------------------------------------------------------------------------------------------------------------------------
```

If your code in pytorch

```
----------------------------------------------------------------------------------------------------------------------
device = torch.device(“cuda” if torch.cuda.is_available() else “cpu”)
----------------------------------------------------------------------------------------------------------------------
```