CONNECT TO THE CLUSTER:

- Before starting, you need to connect to the Télécom's VPN. If you're on the Télécom's network, you can skip this part. Otherwise, you have to do it.

Setup the VPN on Windows :

https://eole.telecom-paris.fr/vos-services/services-numeriques/connexions-aux-reseaux/openvpn-avec-windows#install

Setup the VPN on Debian :

https://eole.telecom-paris.fr/vos-services/services-numeriques/connexions-aux-reseaux/openvpn-avec-debian-gnulinux

Setup the VPN on Ubuntu :

https://eole.telecom-paris.fr/vos-services/services-numeriques/connexions-aux-reseaux/openvpn-avec-ubuntu

Setup the VPN on MacOS :

https://eole.telecom-paris.fr/vos-services/services-numeriques/connexions-aux-reseaux/openvpn-avec-macos

- From the terminal, use your Télécom username (it usually is name first letter + surname) to connect to the cluster :

  ```
  ssh username@tsicluster0.enst.fr
  ```

- Use your Télécom personnal account password to connect

- Congrats ! You're connected to the cluster !

Note : There are also other machines (username@gpuX.enst.fr with X going from 1 to 6) you can connect to in order to run code. However, they aren't working the same way once your logged in and have different GPUs and CUDA versions.
You may find usefull information here : https://services.infres.enst.fr/

USE THE CLUSTER :

We will give you a survival guide. If you wanna go deeper, you may want to check this link to have more informations about how the cluster works: https://computingdoc.telecom-paristech.fr/IdsServers

Let’s suppose you’re logged in the cluster. By default, you’re on the tsicluster0 machine. This device is used as a hub, you mustn’t run code on it !

- To run some code, you have to create a script and submit it to the queue system via the following command :

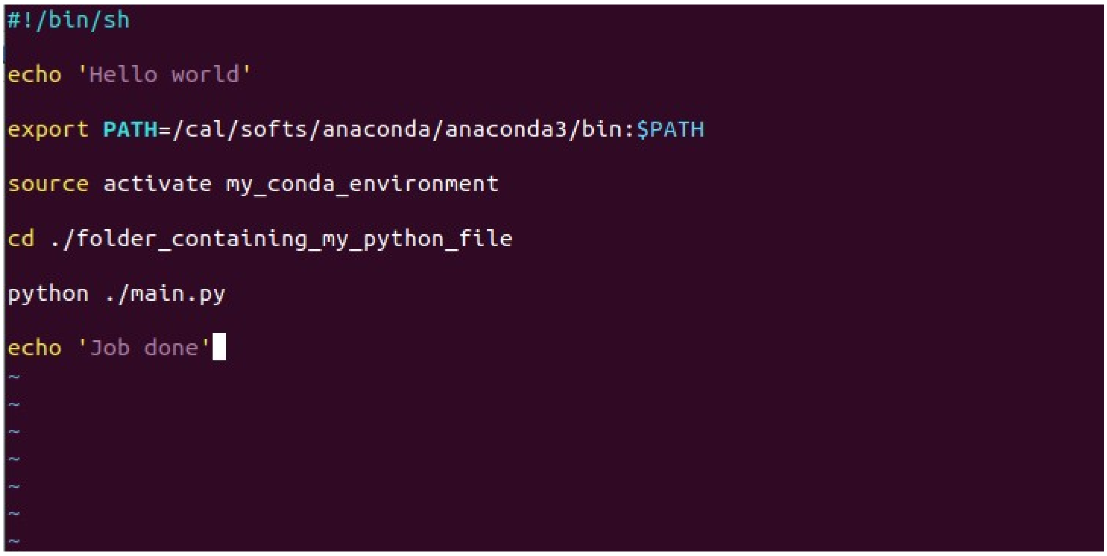
  qsub myscript

  — or —

  qsub -l gpu=X myscript (with X equals 1 or 2 if you want to enforce the number of gpu used)

WARNING : In order to allow a good sharing of ressources between all the PhD students and interns, don’t submit more than 5 jobs at a time.

Example of a basic script running python code on the cluster :



```
#!/bin/sh

echo 'Hello world'

export PATH=/cal/softs/anaconda/anaconda3/bin:$PATH

source activate my_conda_environment

cd ./folder_containing_my_python_file

python ./main.py

echo 'Job done'
~
~
~
~
~
~
~
```

The queue system will then execute the script on one of the available machines (tsiclusterXX)

- To check if your code is still running or not, you can run :

  qstat -u username

It will display all of your pending and runnning jobs with their ids and the machines on which they are assigned.

- To delete/stop a job :

  qdel -j yourjobID

- To close the ssh connection/exit the cluster :

  exit

TIPS :

- Anaconda environment

In general, you will want to install some packages. To do so, you need to create an anaconda environment.

- First, export the conda path :

  ```
  export PATH=/cal/softs/anaconda/anaconda3/bin:$PATH
  ```

- Then you can create your environment, you can find a guide here : https://towardsdatascience.com/environment-package-management-55168c56b77

- To connect to an existing environment :

  ```
  source activate yourEnvironmentName
  ```

- Debugging

In order to debug your code, the first thing you can do is to check the output file of your job.
When you submit a job, you have two files created myscript.e.jobID and myscript.o.jobID. If you open the latter, you will find all of the outputs/prints of your code written on it.

NB : this file contains the prints of your python code. However, it is not dynamic : If you just call print('…') in your python file, the prints will stack and will only be written on your output file at the end of the execution.
If you want this file to be written dynamically, you have to import sys and call sys.stdout.flush() every time after a print.

WARNING :
The following command must be ONLY used for debugging ! (to know why, go to the end of this section)

Right, it may tell you if the code has finished correctly or not but you won't know what's the issue. For that, you can connect dynamically to a machine via :

```
qlogin
```

— or —

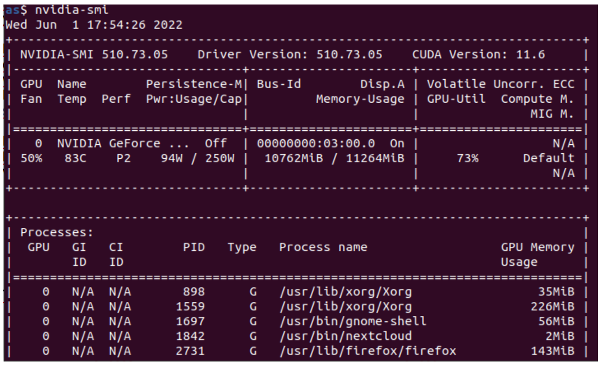
`qlogin -l gpu=X` (with X equals 1 or 2)

You will be connected to a machine tsiclusterXX. Then, you have to check if the cluster's GPU is used or not with :

```
nvidia-smi
```

Example :

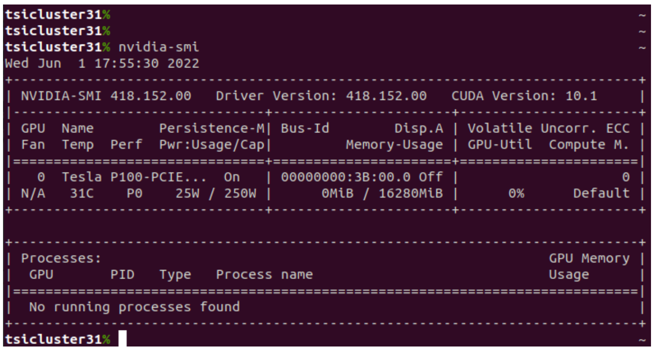
Cluster GPU is used



```
as$ nvidia-smi
Wed Jun  1 17:54:26 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 510.73.05    Driver Version: 510.73.05    CUDA Version: 11.6     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA GeForce ...  Off  | 00000000:03:00.0  On |                  N/A |
| 50%   83C    P2    94W / 250W |  10762MiB / 11264MiB |     73%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A       898      G   /usr/lib/xorg/Xorg                 35MiB |
|    0   N/A  N/A      1559      G   /usr/lib/xorg/Xorg                226MiB |
|    0   N/A  N/A      1697      G   /usr/bin/gnome-shell               56MiB |
|    0   N/A  N/A      1842      G   /usr/bin/nextcloud                  2MiB |
|    0   N/A  N/A      2731      G   /usr/lib/firefox/firefox          143MiB |
```

Cluster GPU is free



```
tsicluster31%                                                                 ~
tsicluster31%                                                                 ~
tsicluster31% nvidia-smi                                                      ~
Wed Jun  1 17:55:30 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.152.00   Driver Version: 418.152.00   CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  On   | 00000000:3B:00.0 Off |                    0 |
| N/A   31C    P0    25W / 250W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
tsicluster31%                                                                 ~
```

- If the cluster is used, disconnect to the cluster using :

  exit

  And try to connect again till obtaining a free cluster.

Once you’re on a free cluster, you can export the conda path, activate your conda environment and run some python code :

```
export PATH=/cal/softs/anaconda/anaconda3/bin:$PATH

source activate yourEnvironmentName

python myfile.py
```

When you finally have debugged your code, you can exit the cluster and run the whole thing with the qsub command.

Why can’t I use qlogin to run my code ?

qlogin is a command that doesn’t take into account the queue system of the cluster. Hence, when you connect to a cluster using it and run some code which requires GPU ressources, if someone is already using the GPU, this person will be kicked out of the GPU.

- Files and Folders

- On tsicluster0 your default directory will be */tsi/clusterhome/username*. Here you can copy your files (code, data, etc.)

- To copy files/folder and their content from your machine to tsicluster0 use the scp command from your local machine:

```
scp file_path username@tsicluster0.enst.fr:~/file_path

scp -r folder_path username@tsicluster0.enst.fr:~/folder_path
```

- Inversely, if you want to copy data from tsicluster0 to you local machine, run the following command from your local machine :

```
scp username@tsicluster0.enst.fr:~/file_path file_path

scp -r username@tsicluster0.enst.fr:~/folder_path folder_path
```

- Interpreter/IDE

- To configure remote interpreter (I didn’t try it) :

https://www.jetbrains.com/help/pycharm/configuring-remote-interpreters-via-ssh.html#ssh