CS 2770: Pitt CRC Tutorial

Prof. Nils Murrugarra- Llerena

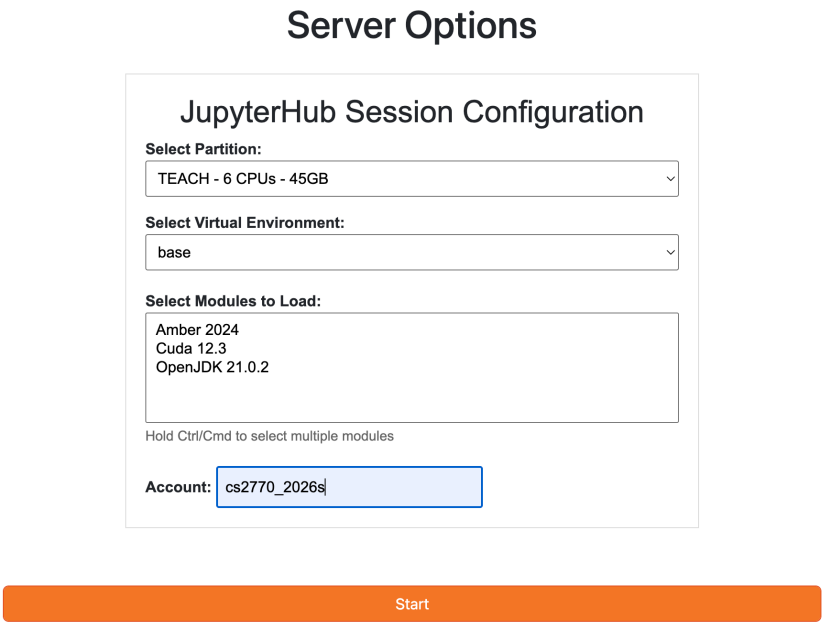
[Setup]

- Clone this Github repo: https://github.com/nineil-pitt/tutorial_pitt_CRC locally in your computer. You can use Git command line or Github Desktop.

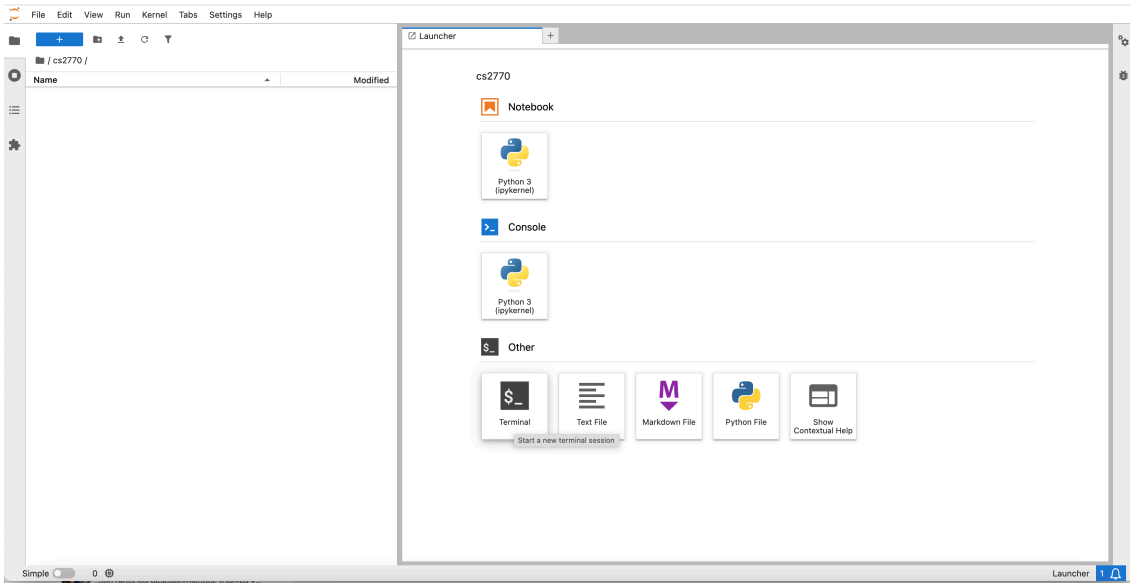
[Jupyter Notebooks]

We are following this website documentation to connect to Jupyter Notebooks on Pitt CRC.

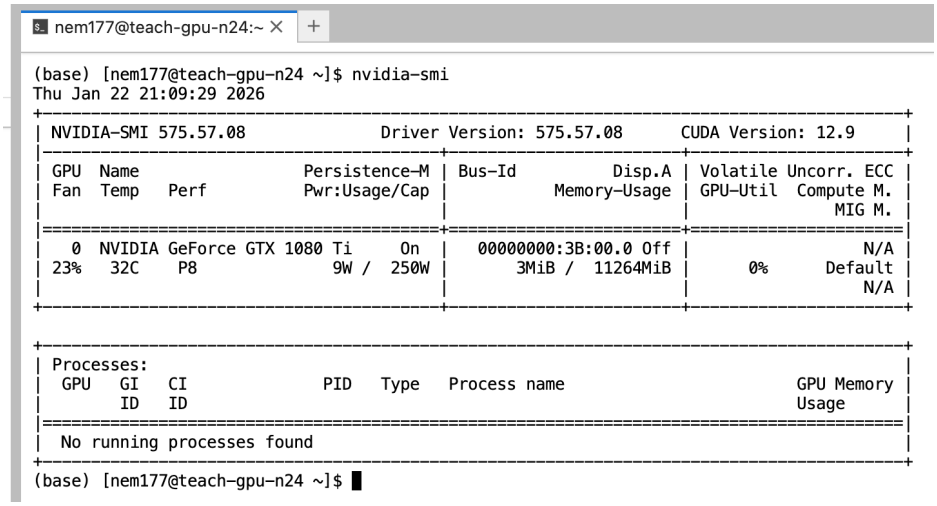
1. Login on Jupyter Hub with your Pitt Credentials
2. Select your Partition with GPU (e.g. TEACH – Nvidia GTX 1080 GPU) and account: cs2770_2026s



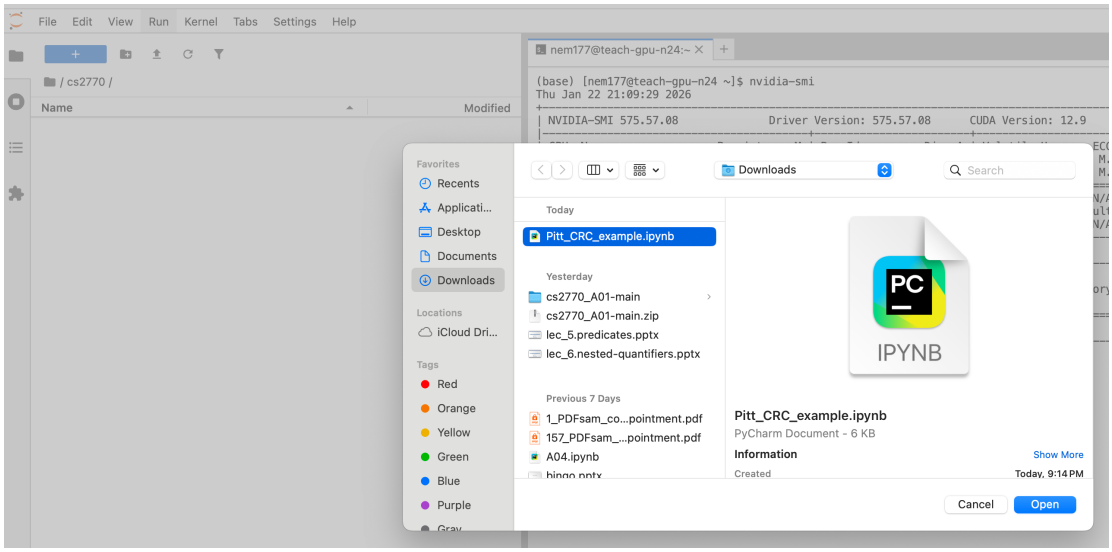
3. Once logged in, select “terminal”



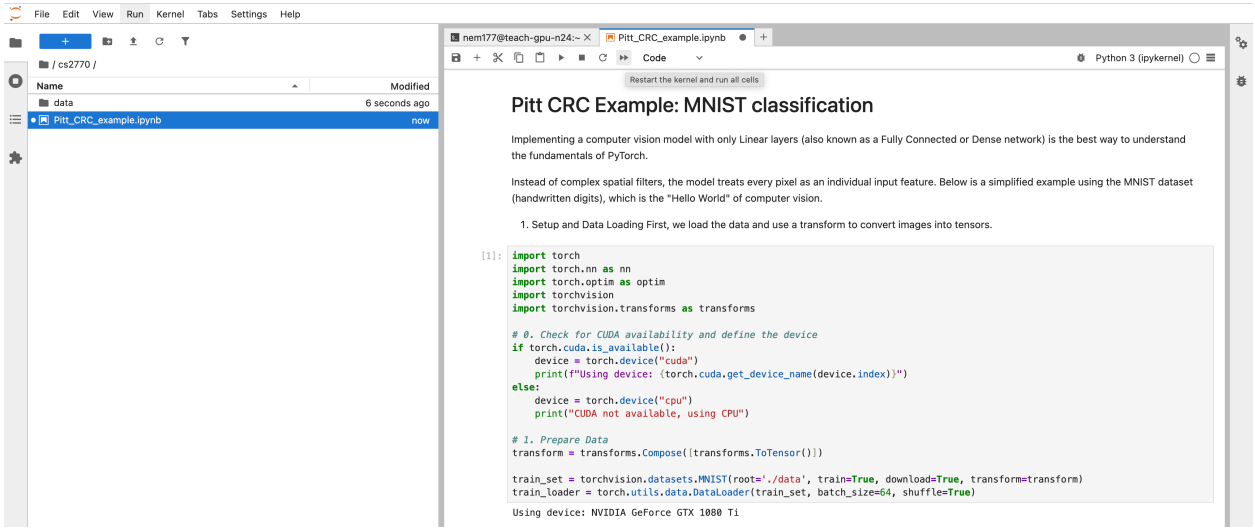
4. Verify that you have a machine with GPU using command: nvidia-smi



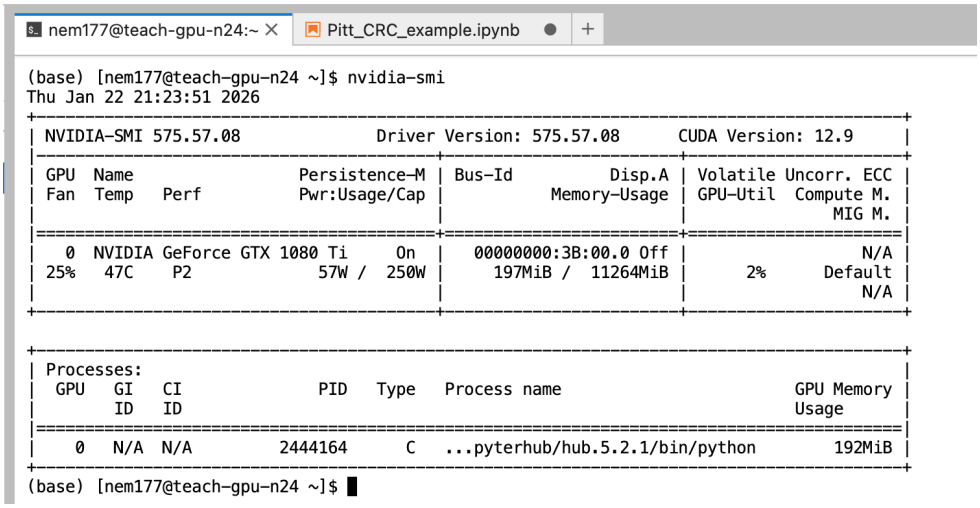
5. Upload Jupyter Notebook Pitt_CRC_example.ipynb with button



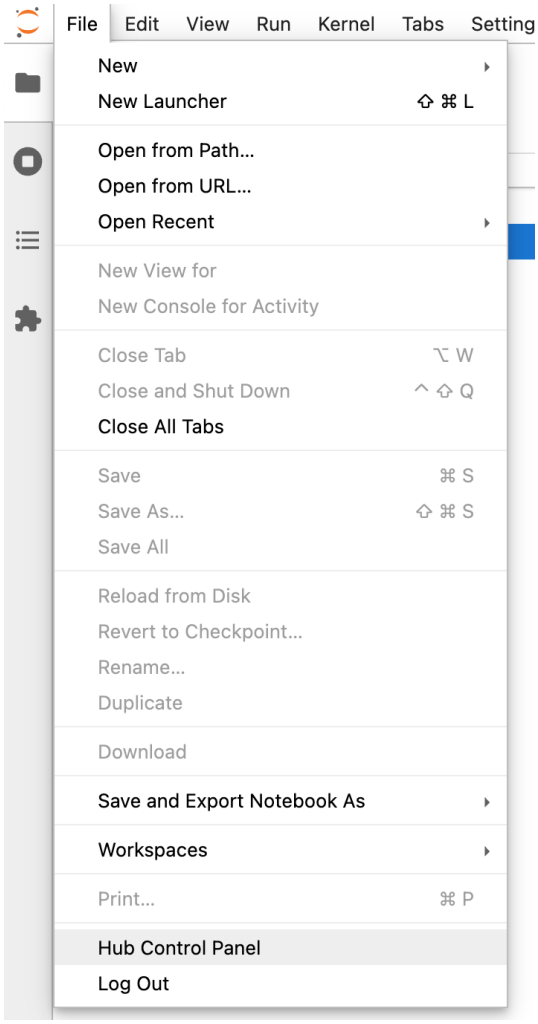
6. Open Jupyter Notebook and Execute with button ▶▶

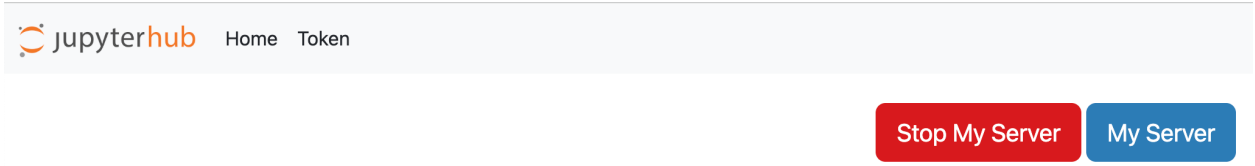


7. Open a new terminal window, and monitor the use of GPU



8. Logout using File, Hub Control Panel, Stop My Server

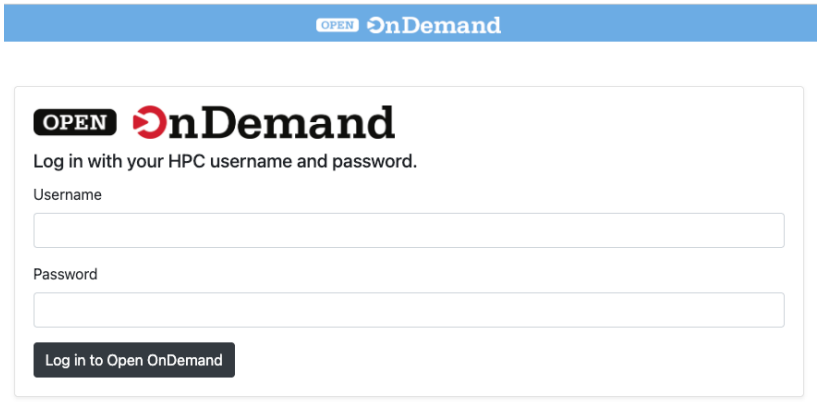




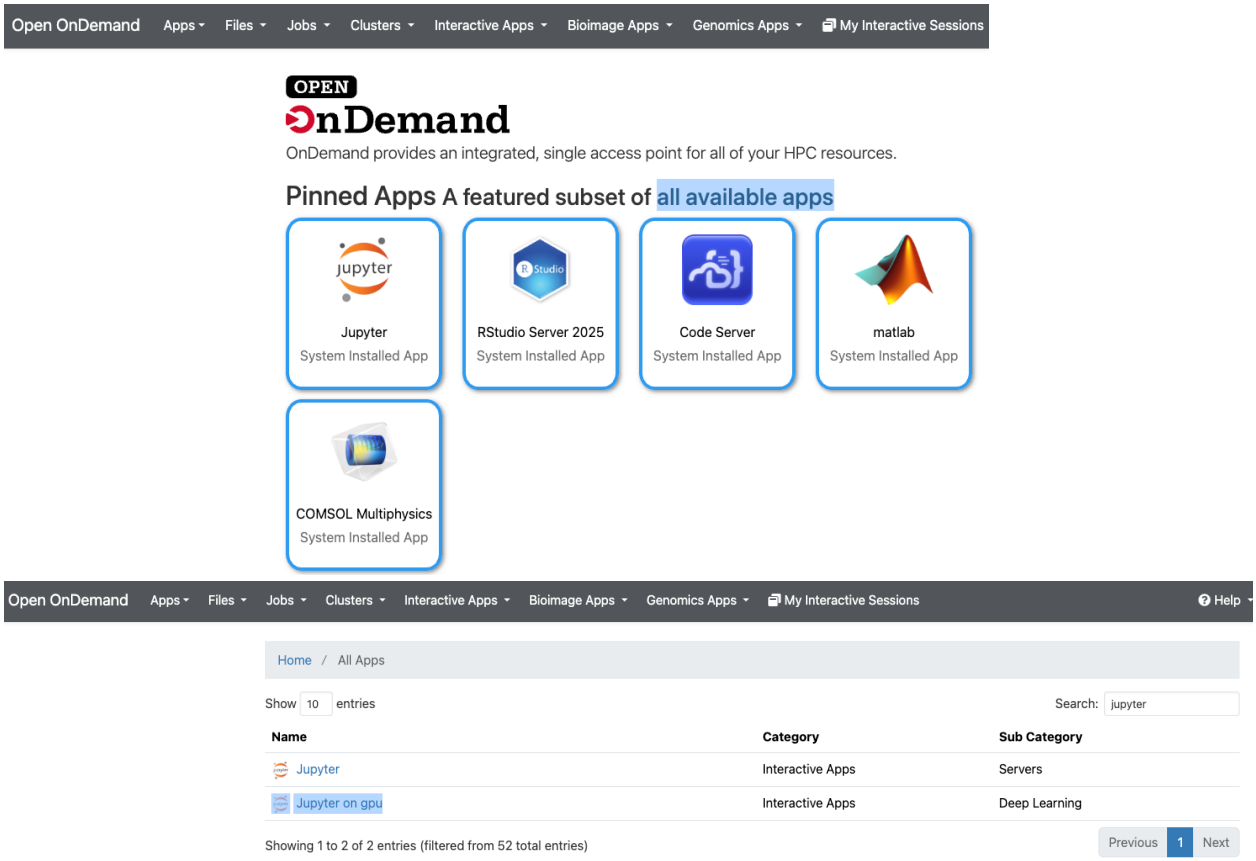
[Jupyter Ondemand]

We are following this website documentation to connect to Jupyter OnDemand on Pitt CRC.

1. Login to Open OnDemand with your Pitt Credentials



2. Select on “all available apps” and search for “Jupyter on GPU”



3. Select your configuration (e.g. # of hours, # of cores, GPU type, # of GPUs, etc) and add account to “cs2770_2026s” and click on launch.

Bioimage Apps

GUIs

- CellProfiler
- Fiji
- Napari
- QuPath
- ilastik

fMRI

- 3D-Slicer
- AFNI
- FSL
- Freeview
- ITK-SNAP
- MRIcron
- MRtrix3
- dsi_studio

Jupyter on gpu

This app will launch a Jupyter server using Python on the [gpu cluster].

☑ Use JupyterLab instead of Jupyter Notebook?

JupyterLab is the next generation of Jupyter with an IDE-like experience, and is completely compatible with existing Jupyter Notebooks.

Python version

pytorch 2.5.1 python 3.11 cuda 12.4

This defines the version of python you want to load.

Cuda version

cuda 12.9.0

This defines the version of cuda you want to load. **You must choose None if you are using PyTorch and the associated PyTorch-CUDA packages.**

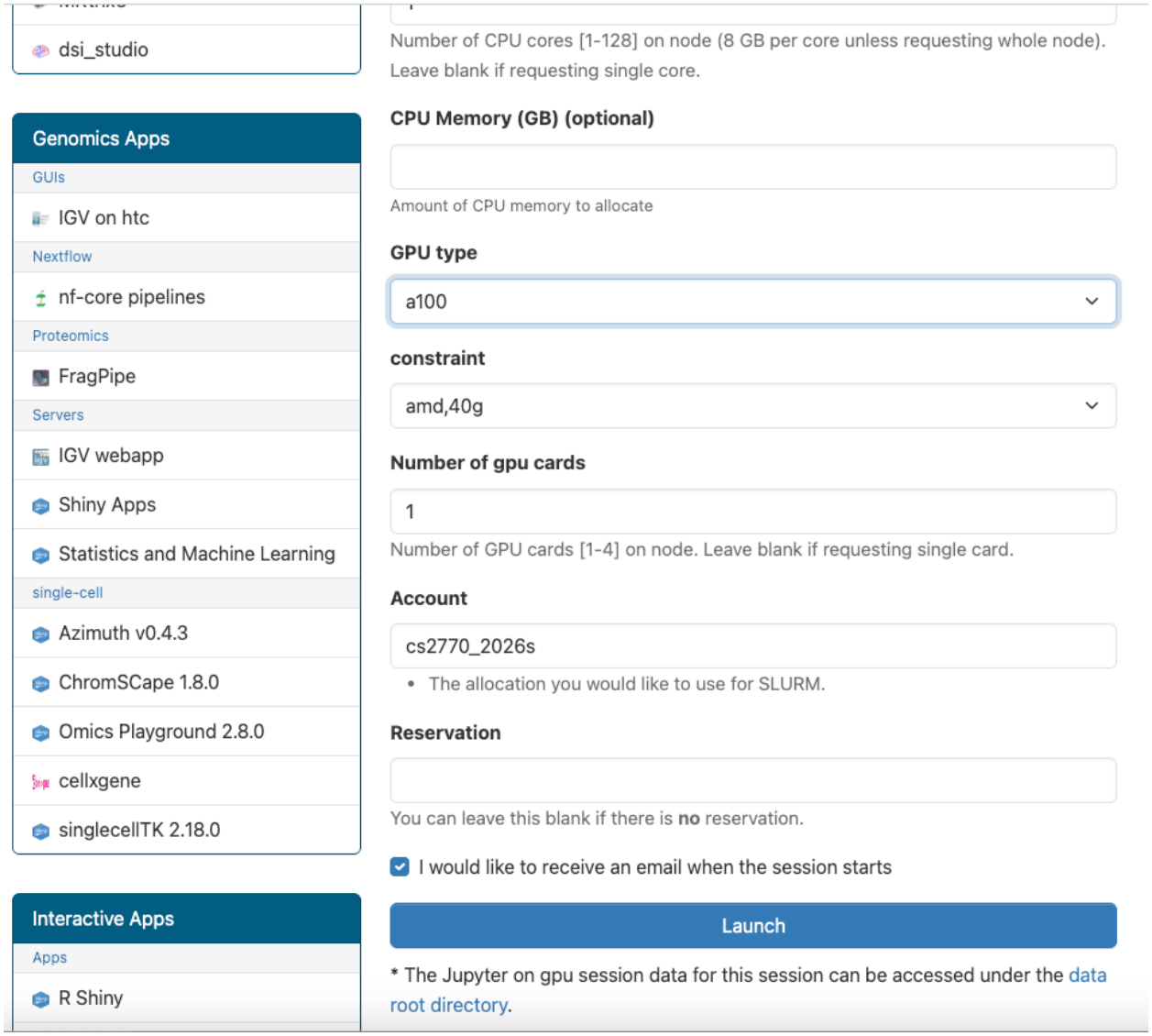
Number of hours

1

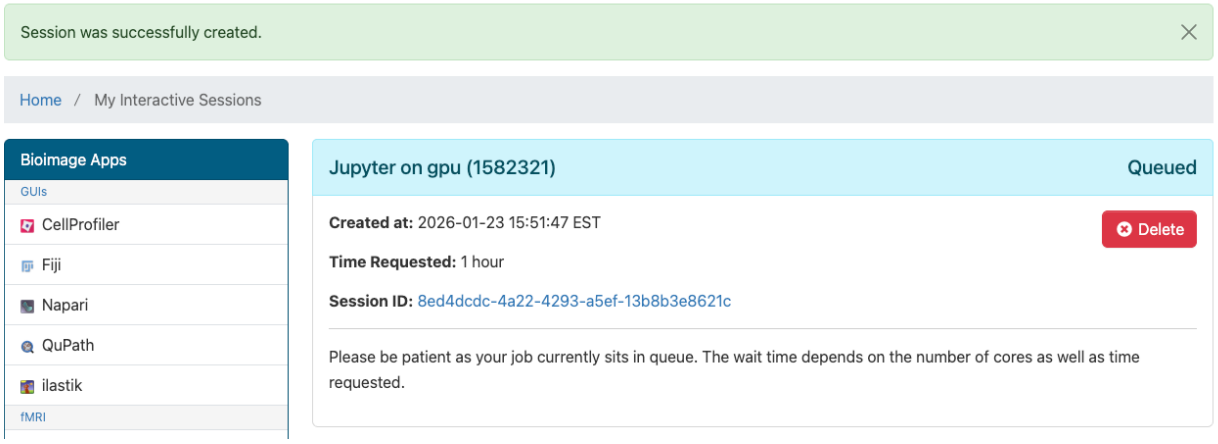
Number of cores

1

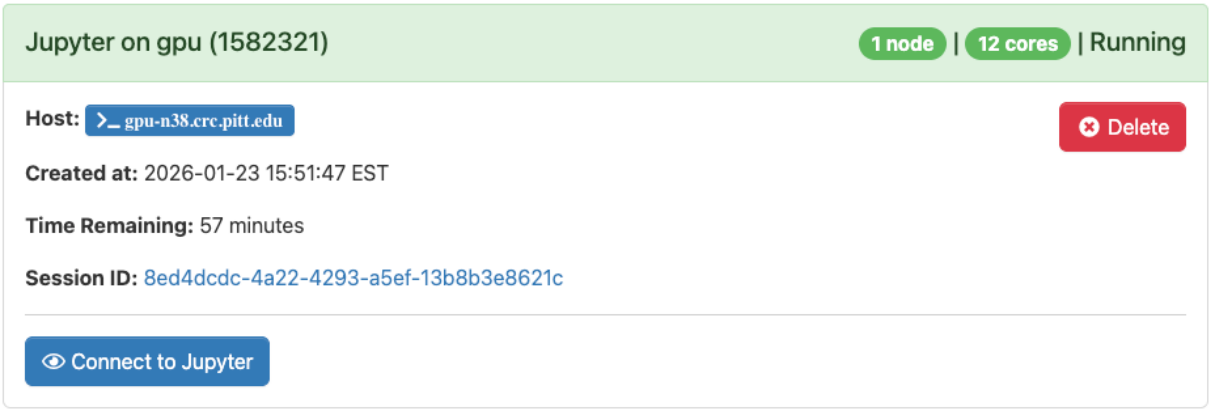
Number of CPU cores [1-128] on node (8 GB per core unless requesting whole node). Leave blank if requesting single core.



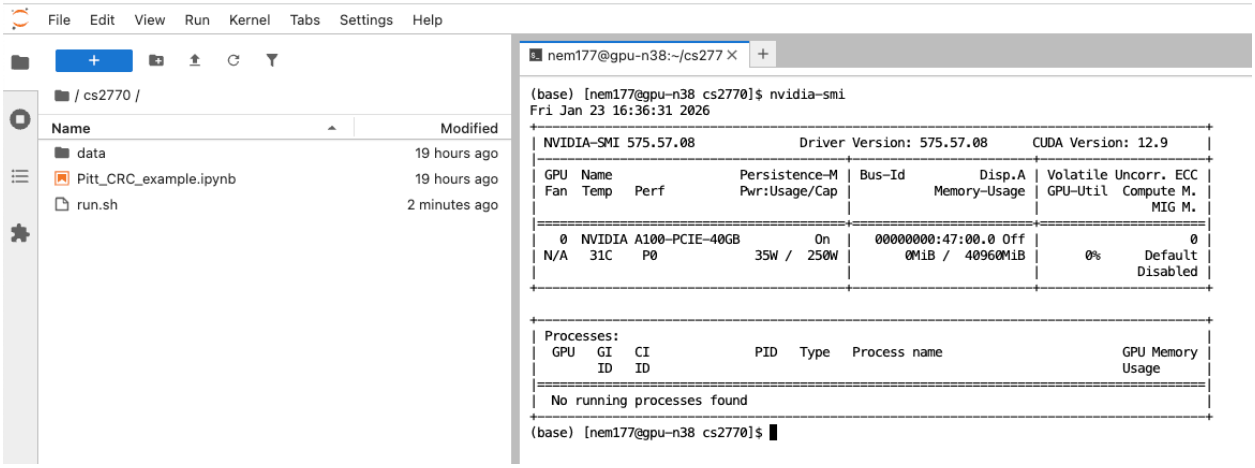
4. Wait while the system assigns you resources, you will receive an email.



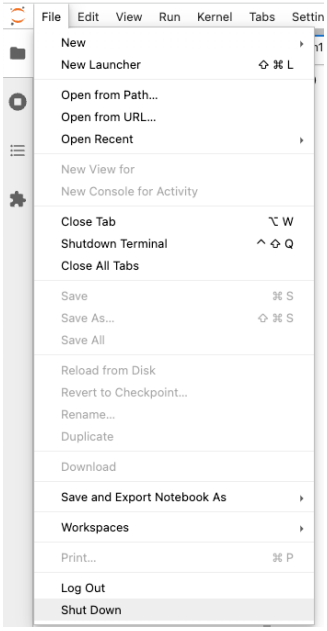
5. Once you received the email, you can connect using Open OnDemand interface. Click on “Connect to Jupyter”



6. Then, you can do same steps as our previous section (working with Jupyter Notebooks)



7. Once you finish, you can shut down this machine.



[Command Line: Interactive Job]

We are following this website documentation from Pitt CRC.

1. Connect via ssh with command: "ssh <yourpittID>@h2p.crc.pitt.edu" in your terminal for linux or Mac. If you are using windows, please try MobaXterm or Putty.

```
(base) nem177@Nilss-MacBook-Pro ~ % ssh nem177@h2p.crc.pitt.edu
nem177@h2p.crc.pitt.edu's password:
################################################################################

                        Welcome to h2p.crc.pitt.edu!

      Documentation can be found at https://crc-pages.pitt.edu/user-manual/

--------------------------------------------------------------------------------

                              IMPORTANT REMINDERS

 Don't run jobs on login nodes! Use interactive jobs: `crc-interactive --help`

 Slurm is separated into 'clusters', e.g. if `scancel <jobnum>` doesn't work
      try `crc-scancel <jobnum>`. Try `crc-sinfo` to see all clusters.

--------------------------------------------------------------------------------

################################################################################
Last login: Fri Jan 23 16:01:11 2026 from 10.25.197.87
(base) [nem177@login0 ~]$
```

2. Clone this Github repo: https://github.com/nineil-pitt/tutorial_pitt_CRC on Pitt CRC with command "git clone https://github.com/nineil-pitt/tutorial_pitt_CRC.git" and move to tutorial_pitt_CRC/ with command "cd tutorial_pitt_CRC"
3. Observe state of GPU resources with command: "sinfo --clusters=gpu". Partition l40s seems available.

```
(base) [nem177@login1 ~]$ sinfo --clusters=gpu
CLUSTER: gpu
PARTITION          AVAIL  TIMELIMIT  NODES  STATE NODELIST
a100*                 up   infinite     12    mix gpu-n[33-44]
a100_multi            up   infinite      1  down* gpu-n53
a100_multi            up   infinite      9  alloc gpu-n[45-52,54]
preempt               up   infinite      1  down* gpu-n53
preempt               up   infinite     16    mix gpu-n[58-59,61-73],mems-n0
preempt               up   infinite     13  alloc gpu-n[45-52,54-57,60]
a100_nvlink           up   infinite      4    mix gpu-n[29-32]
a100_nvlink           up   infinite      1  alloc gpu-n28
a100_nvlink_multi     up   infinite      4    mix gpu-n[29-32]
a100_nvlink_multi     up   infinite      1  alloc gpu-n28
l40s                  up   infinite     15    mix gpu-n[58-59,61-73]
l40s                  up   infinite      4  alloc gpu-n[55-57,60]
```

4. Request interactive node: "srun --clusters=gpu --gres=gpu:1 --partition=l40s --nodes=1 --time=01:00:00 --pty bash"

```
(base) [nem177@login0 cs2770]$ srun --clusters=gpu --gres=gpu:1 --partition=l40s --nodes=1 --time=01:00:00 --pty bash
srun: job 1582339 queued and waiting for resources
srun: job 1582339 has been allocated resources
(base) [nem177@gpu-n55 cs2770]$ nvidia-smi
Fri Jan 23 17:40:00 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 575.57.08              Driver Version: 575.57.08      CUDA Version: 12.9     |
|-----------------------------------------+------------------------+----------------------+
| 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 L40S                    On  |   00000000:E1:00.0 Off |                    0 |
| N/A   22C    P8             32W /  350W |       0MiB /  46068MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

5. Execute your code
    a. Jupyter Notebook Option
        i. Execute Jupyter Notebook with command: “papermill Pitt_CRC_example.ipynb out.ipynb”

```
(base) [nem177@gpu-n55 cs2770]$ papermill Pitt_CRC_example.ipynb out.ipynb
Input Notebook:  Pitt_CRC_example.ipynb
Output Notebook: out.ipynb
Executing:   0%|
Executing notebook with kernel: python3
Executing: 100%|
```

        ii. The generated output is stored in out.ipynb notebook
    b. Python Code Option
        i. Execute Python code with command: “python Pitt_CRC_example.py”

```
(base) [nem177@gpu-n55 cs2770]$ python Pitt_CRC_example.py
Using device: NVIDIA L40S
Epoch 0
Batch 0: Loss = 2.2995
Batch 500: Loss = 1.0016
Epoch 1
Batch 0: Loss = 0.5489
Batch 500: Loss = 0.5350
Epoch 2
Batch 0: Loss = 0.3408
Batch 500: Loss = 0.3494
Epoch 3
Batch 0: Loss = 0.3059
Batch 500: Loss = 0.2231
Epoch 4
Batch 0: Loss = 0.4119
Batch 500: Loss = 0.2242
Epoch 5
Batch 0: Loss = 0.2699
Batch 500: Loss = 0.4217
Epoch 6
Batch 0: Loss = 0.4221
Batch 500: Loss = 0.3276
Epoch 7
Batch 0: Loss = 0.3909
Batch 500: Loss = 0.3017
Epoch 8
Batch 0: Loss = 0.3090
Batch 500: Loss = 0.2182
Epoch 9
Batch 0: Loss = 0.2445
Batch 500: Loss = 0.3315
```

[Command Line: Batch Job]

We are following this website documentation from Pitt CRC.

1. Connect via ssh with command: “ssh <pittID>@h2p.crc.pitt.edu” in your terminal for linux or Mac. If you are using windows, please try MobaXterm or Putty.

```
(base) nem177@Nilss-MacBook-Pro ~ % ssh nem177@h2p.crc.pitt.edu
nem177@h2p.crc.pitt.edu's password:
################################################################################

                        Welcome to h2p.crc.pitt.edu!

      Documentation can be found at https://crc-pages.pitt.edu/user-manual/

--------------------------------------------------------------------------------

                              IMPORTANT REMINDERS

 Don't run jobs on login nodes! Use interactive jobs: `crc-interactive --help`

 Slurm is separated into 'clusters', e.g. if `scancel <jobnum>` doesn't work
      try `crc-scancel <jobnum>`. Try `crc-sinfo` to see all clusters.

--------------------------------------------------------------------------------

################################################################################
Last login: Fri Jan 23 16:01:11 2026 from 10.25.197.87
(base) [nem177@login0 ~]$
```

2. In Pitt CRC, execute your code
    a. Jupyter Notebook Code
        i. Type “sbatch run.sh” command.

```
(base) [nem177@gpu-n55 cs2770]$ sbatch run.sh
Submitted batch job 1582341 on cluster gpu
```

        ii. While executing, run a command to the state of your job using “squeue --clusters=gpu --partition=l40s --user=<yourPittID>”

```
(base) [nem177@login1 cs2770]$ squeue --clusters=gpu --partition=l40s --user=nem177
CLUSTER: gpu
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1582339      l40s     bash   nem177  R      37:27      1 gpu-n55
           1582341      l40s test_277   nem177  R       0:09      1 gpu-n61
```

    b. Python Code
        i. Type “sbatch run_python.sh” command.

```
(base) [nem177@login0 cs2770]$ sbatch run_python.sh
Submitted batch job 1582364 on cluster gpu
```

        ii. After the command finishes, you can inspect files: code_gpu-1582364.err and code_gpu-1582364.out with cat command

```
(base) [nem177@login0 cs2770]$ cat code_gpu-1582364.err
Lmod Warning:
--------------------------------------------------------------------------------
The following dependent module(s) are not currently loaded: libxml2/2.10.3
(required by: cuda/12.1.1)
--------------------------------------------------------------------------------


The following have been reloaded with a version change:
  1) libiconv/1.17-b47av4d => libiconv/1.17-oygrtg5
  2) libxml2/2.10.3 => libxml2/2.13.5-ftxdjfd
  3) xz/5.4.1 => xz/5.4.6-anmq7m4
  4) zlib-ng/2.1.5-g64y7ia => zlib-ng/2.2.3-sp4tqsg


[notice] A new release of pip is available: 24.0 -> 25.3
[notice] To update, run: pip3 install --upgrade pip
```

```
(base) [nem177@login0 cs2770]$ cat code_gpu-1582364.out
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: Pillow in /ihome/nllerena/nem177/.local/lib/python3.11/site-packages (12.1.0)
Executing in node: gpu-n61.crc.pitt.edu
Date: Fri Jan 23 06:51:59 PM EST 2026
Working Directory: /ihome/nllerena/nem177/cs2770
GPU information:
Fri Jan 23 18:51:59 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 575.57.08              Driver Version: 575.57.08      CUDA Version: 12.9     |
|-----------------------------------------+------------------------+----------------------+
| 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 L40S                    On  |   00000000:4A:00.0 Off |                    0 |
| N/A   25C    P8             33W /  350W |       0MiB /  46068MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
Initializing code...
Input: /ihome/nllerena/nem177/cs2770//Pitt_CRC_example.py
Using device: NVIDIA L40S
Epoch 0
Batch 0: Loss = 2.3222
Batch 500: Loss = 0.9446
Epoch 1
Batch 0: Loss = 0.5379
Batch 500: Loss = 0.4536
Epoch 2
Batch 0: Loss = 0.5789
Batch 500: Loss = 0.3803
Epoch 3
Batch 0: Loss = 0.2842
Batch 500: Loss = 0.2730
Epoch 4
Batch 0: Loss = 0.2942
Batch 500: Loss = 0.3258
Epoch 5
Batch 0: Loss = 0.4418
Batch 500: Loss = 0.1103
Epoch 6
Batch 0: Loss = 0.3798
Batch 500: Loss = 0.2339
Epoch 7
Batch 0: Loss = 0.5312
Batch 500: Loss = 0.3298
Epoch 8
Batch 0: Loss = 0.3859
Batch 500: Loss = 0.3477
Epoch 9
Batch 0: Loss = 0.2134
Batch 500: Loss = 0.3049
Code executed correctly!
Job finished on: Fri Jan 23 06:52:27 PM EST 2026
```

3. Finally, generate errors for our examples and run again
    a. For Jupyter Notebook: run.sh script, comment the line “# unset LD_LIBRARY_PATH” and submit the job.
    b. For Python code: run_python.sh script, update line “python3 "$INPUT_FILE"” to “python "$INPUT_FILE"”
    c. What are the responses from Slurm?
        i. Do you receive an email?
        ii. Can you inspect error and output files?
        iii. What about output Jupyter notebook?

[FAQ]

Q: How to transfer files to Pitt CRC?

A: To transfer files to Pitt CRC, you can follow this documentation.

Q: How to send a program in background in an interactive session?

A: You can use screen Linux command.

Q: Where can I find more documentation of slurm?

A: You can review slurm documentation and Pitt CRC website.

Q: How can I cancel a job on Slurm?

A: Use command “scancel <jobID>”. See documentation.

Q: How can I know the state of a job on Slurm?

A: Use command “scontrol --clusters=gpu show job <jobID>”. See documentation.

Q: How to create a conda virtual environment?

A: Follow the guide from Pitt CRC.

Q: What modules/packages/programs are available to load?

A: Please, review them from this website.

Q: Where can I find more help from Pitt CRC?

A: Please, review their website https://crc.pitt.edu/