

Centre de Calcul
de l'Institut National de Physique Nucléaire
et de Physique des Particules

GPUs availability via the Jupyter notebooks platform service

Bernard CHAMBON, April 30th 2021

The Jupyter notebooks platform > Reminder



- Objective of this service
 - The Jupyter notebooks platform provides notebooks with JupyterLab interface
 - It allows you to develop & run code interactively by using several programming languages; It provides a UNIX terminal
- Access to this service
 - Accessible for all CC-IN2P3 users
 - Authenticating via Keycloak SSO, by using 'computing account' credentials (same as for ssh cca.in2p3.fr)
 - Url <https://notebook.cc.in2p3.fr>
- Runtime environment
 - You will run JupyterLab with your primary and all secondary groups
 - You will access the following storage areas : /pbs/home/u/username, /pbs/throng, /pbs/software, /sps/xyz, /cvmfs/xyz
 - Available programming language : requires to setup a kernel
 - Anaconda 3 : kernel provided with Docker image, nothing to do.
 - Kernels ready to be used from /pbs/software : ROOT via Python, ROOT via C++ Cling interpreter, R, Go, Julia. See doc [Adding kernels](#)
 - Resources control
 - Memory limit set per group or per user. Default is 2GB
 - Idle notebook timeout of 72h

The Jupyter notebooks platform > Reminder

Jupyter Notebooks Platform at CCIN2P3, version 3.2.2

Launch My Notebook Server

More Information Help

CC-IN2P3 Single Sign-On

Sign in with your CC-IN2P3 account

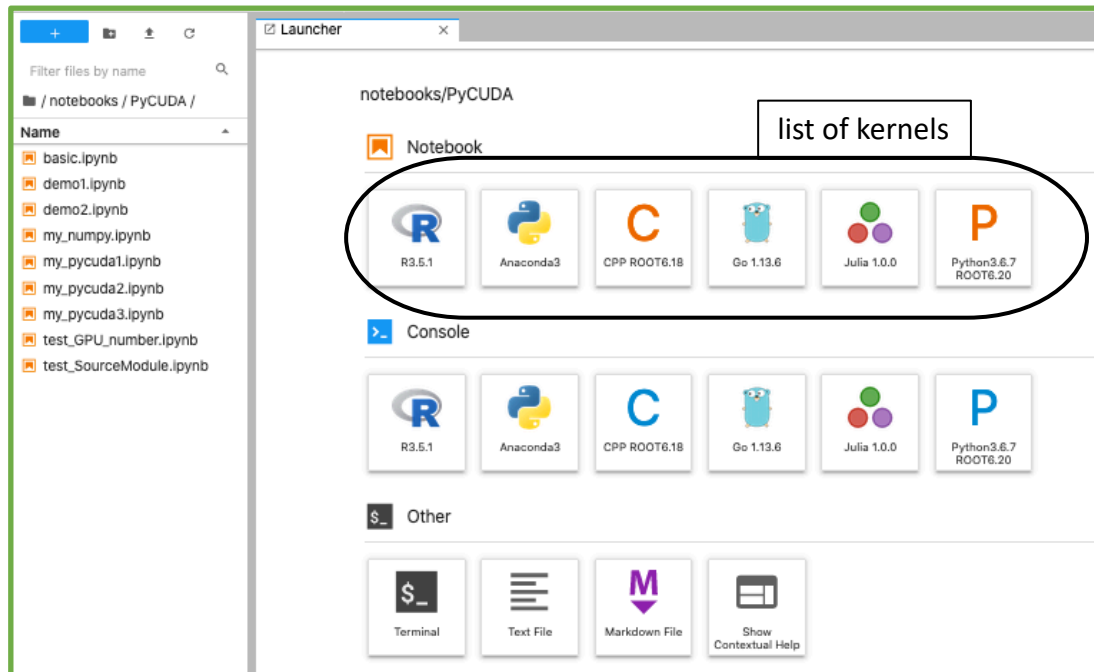
Username:

Password:

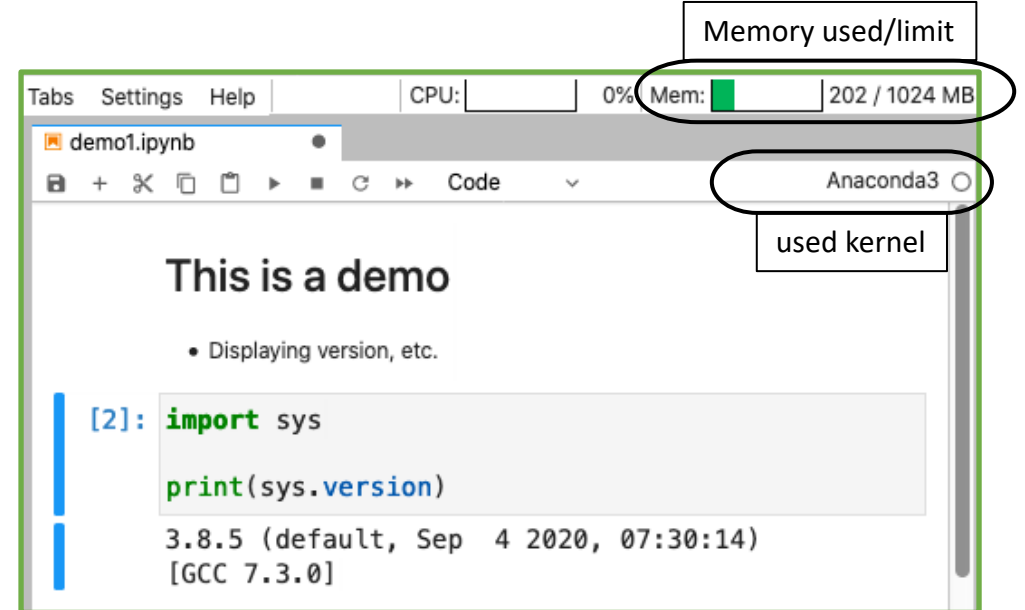
Remember me [Forgot Password?](#)

Log In

Sign in with your organization or institution account



The screenshot shows the Jupyter Notebook Launcher interface. On the left, there is a file browser showing a list of notebooks under the path "/notebooks/PyCUDA/". The main area is titled "notebooks/PyCUDA" and contains a "list of kernels" section. This section displays six kernel options: R3.5.1, Anaconda3, CPP ROOT6.18, Go 1.13.6, Julia 1.0.0, and Python3.6.7 ROOT6.20. Below the kernels, there is a "Console" section with the same six kernel options. At the bottom, there is an "Other" section with icons for Terminal, Text File, Markdown File, and Show Contextual Help.



The screenshot shows a Jupyter Notebook running a demo. The top status bar displays "CPU: 0%" and "Mem: 202 / 1024 MB". The notebook title is "demo1.ipynb" and the kernel is "Anaconda3". The main content area displays "This is a demo" and a list of bullet points: "• Displaying version, etc.". Below this, there is a code cell with the following code:

```
[2]: import sys
print(sys.version)
```

The output of the code cell is:

```
3.8.5 (default, Sep 4 2020, 07:30:14)
[GCC 7.3.0]
```

GPUs availability via the Jupyter notebooks platform service

The Jupyter notebooks platform > GPUs



- Objective of this feature
 - Provide notebook running on host with GPUs devices
 - Each user will have it's own set of GPUs (= NOT shared with other users)
- Access to this feature
 - For granted users only
 - Granted users will get an options form, allowing to select
 - The model and the numbers on GPUs, according to the instant availability
 - The memory limit for the notebooks server
- Runtime environment
 - GPU notebooks server will run on host with GPUs model K80
 - GPU software provided : CUDA, PyCUDA, cuPy, Pytorch, TensorFlow + cuDNN (details of versions from the options form)
 - Same storage areas as seen previously

The Jupyter notebooks platform > GPUs

My Notebook Server Options

Compute engines CPU Only GPU

Launch My Notebook Server

My Notebook Server Options

Compute engines CPU Only GPU

Memory (GB)

GPU model(s) K80

GPU(s) number

The GPU model K80 provides :

- Hardware
 - 4 GPUs per host and 12GiB of memory per GPU
- Software
 - CUDA 10.2 [cuda](#)
 - PyCUDA 2020.1 [pycuda](#)
 - cuPy 8.6.0 for CUDA 10.2 [cupy](#)
 - Pytorch 1.7.1 [pytorch](#)
 - TensorFlow 2.3.0 [tensorflow](#)
 - cuDNN 8.1
 - tensorflow-probability 0.11.1

Launch My Notebook Server

```
demo2.ipynb | CPU: | 0% | Mem: | 224 / 32768 MB
```

```
[1]: import pycuda
import pycuda.autotinit
import pycuda.driver as drv

[2]: print("You have selected {} GPU(s), model {}".format(drv.Device.count(), drv.Device(0).name()))
print("Using CUDA version {}".format(drv.get_version()))

You have selected 1 GPU(s), model Tesla K80
Using CUDA version (10, 2, 0)
```

Memory (32GB = selected limit)

GPUs availability via the Jupyter notebooks platform service

The Jupyter notebooks platform > GPUs



- Status (2021/04/30)
 - One bare metal host : model K80, 4 GPUs per host, RAM 135 GB (125 GiB), Ethernet 10Gbs
 - Docker image with installs for CUDA 10.2, PyCUDA 2020.1, Pytorch 1.7.1, TensorFlow 2.3 + cuDNN 8.1
 - **GPUs availability currently under test, for “beta testers”**
- Planned
 - Another K80, currently being installed
 - Add K80 hosts (from batch production farm), to provide 10 hosts by the end of the year
- How to test
 - Ask UserSupport to get a granted access
 - But for now, there is only one GPU host
 - Log in <https://notebook.cc.in2p3.fr> Log out : File > Log Out (mandatory to release GPU slot(s))
 - Documentation and Ticketing URLs are available from login page