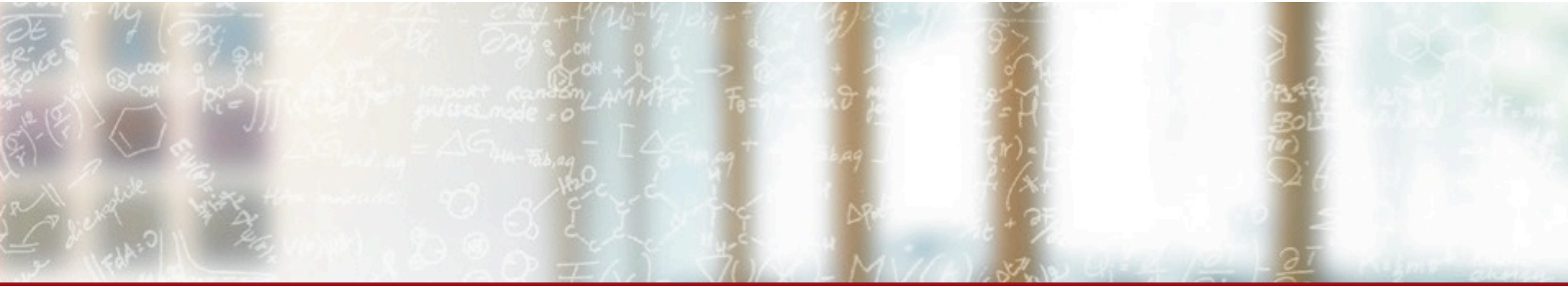


Interactive Supercomputing with Jupyter Notebooks
Tutorial on Python for HPC

CSCS-USI Summer School 2021

Tim Robinson and Rafael Sarmiento (ETH Zurich / CSCS)

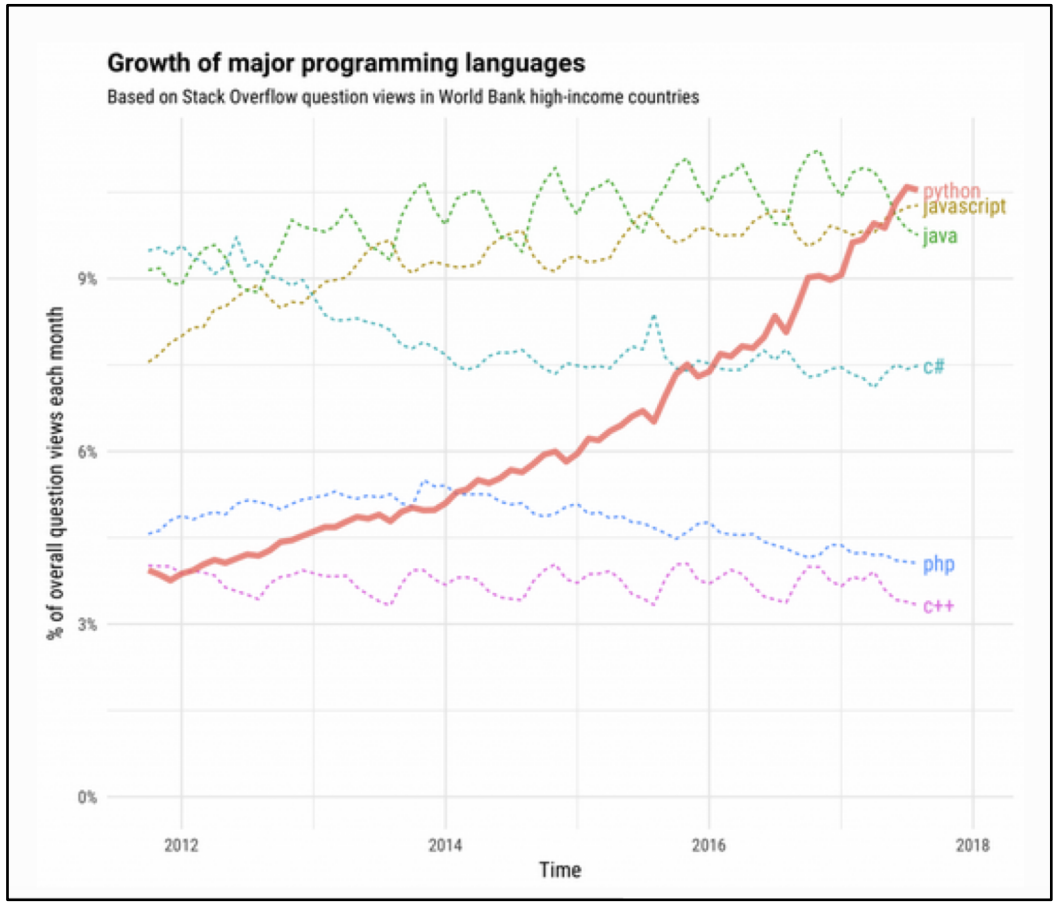
July 26, 2021

Schedule for today

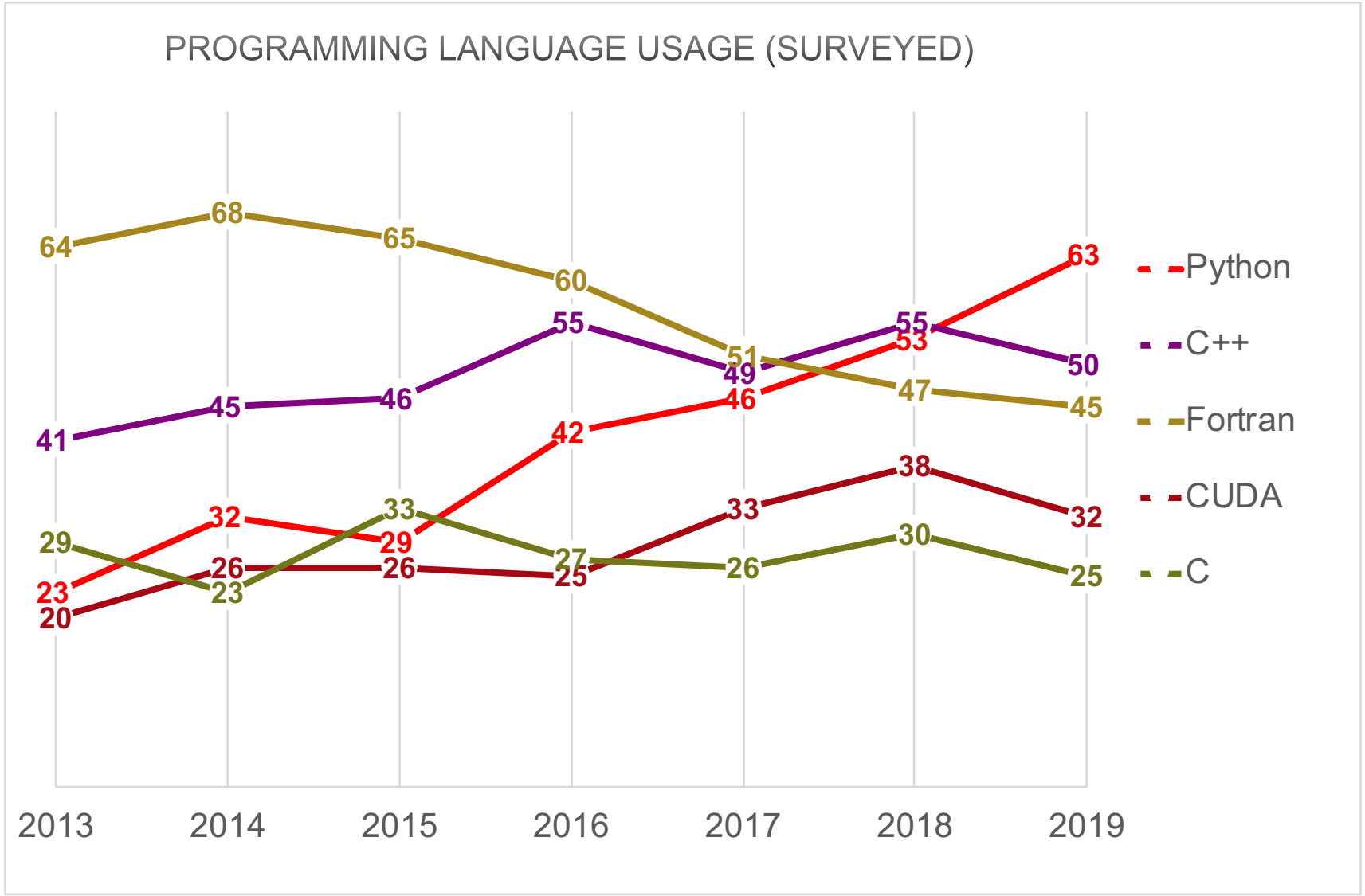
- 09:00 – 10:30 Interactive Supercomputing with JupyterLab
- 10:30 – 11:00 Break
- 11:00 – 12:00 Python for HPC
- 12:00 – 13:00 Lunch
- 13:00 – 13:30 Networking on Spatial Chat (?)
- 13:30 – 15:30 Python for HPC (cont.)
- 15:30 – 16:00 Break
- 16:00 – 17:00 Q&A (Zoom)

The rise of Python

- Python has grown to become a dominant language both in data analytics and general programming
- Rise fueled by computational libraries like Numpy, Pandas, and Scikit-Learn and libraries for visualization, interactive notebooks, collaboration, etc.
- Python long used as glue, for pre- and/or post-processing.. but increasingly used for simulation as well



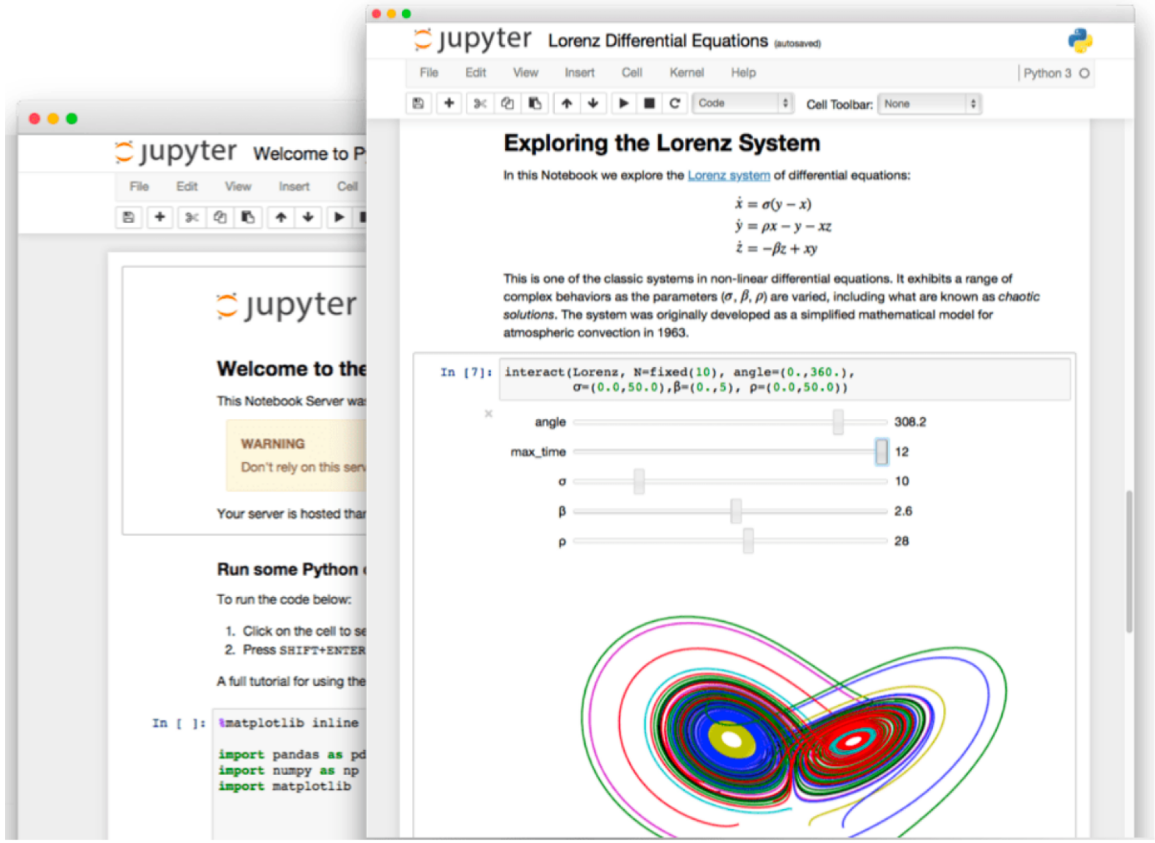
Python usage at CSCS



Jupyter Notebook

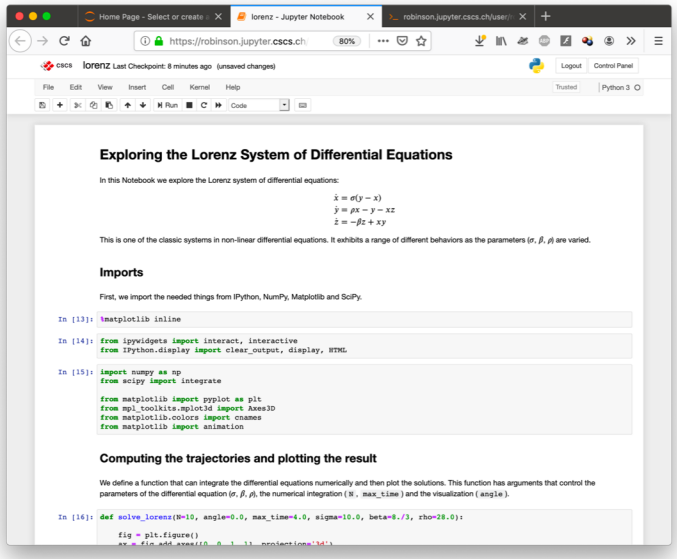
IPython

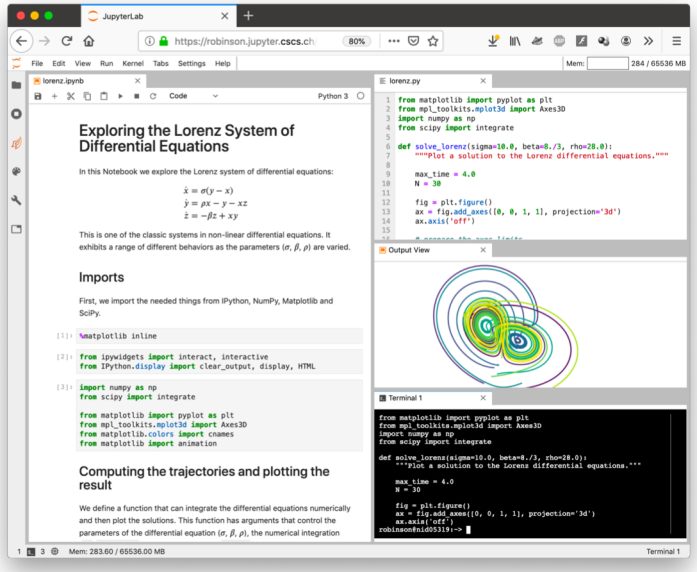
- Jupyter Notebook is an open-source web app for creating reproducible computational narratives
- Documents can contain live code, equations, narrative text, visualizations, and rich media
- Documents can be shared or exported to PDF, HTML, LaTeX, etc.
- Working environment includes
  - File browser
  - Terminal
  - Support for many languages: Python, R, Julia, C++, …
  - Many server/client plugins
- Notebook is attached to a “kernel”, which does the actual computation



JupyterLab

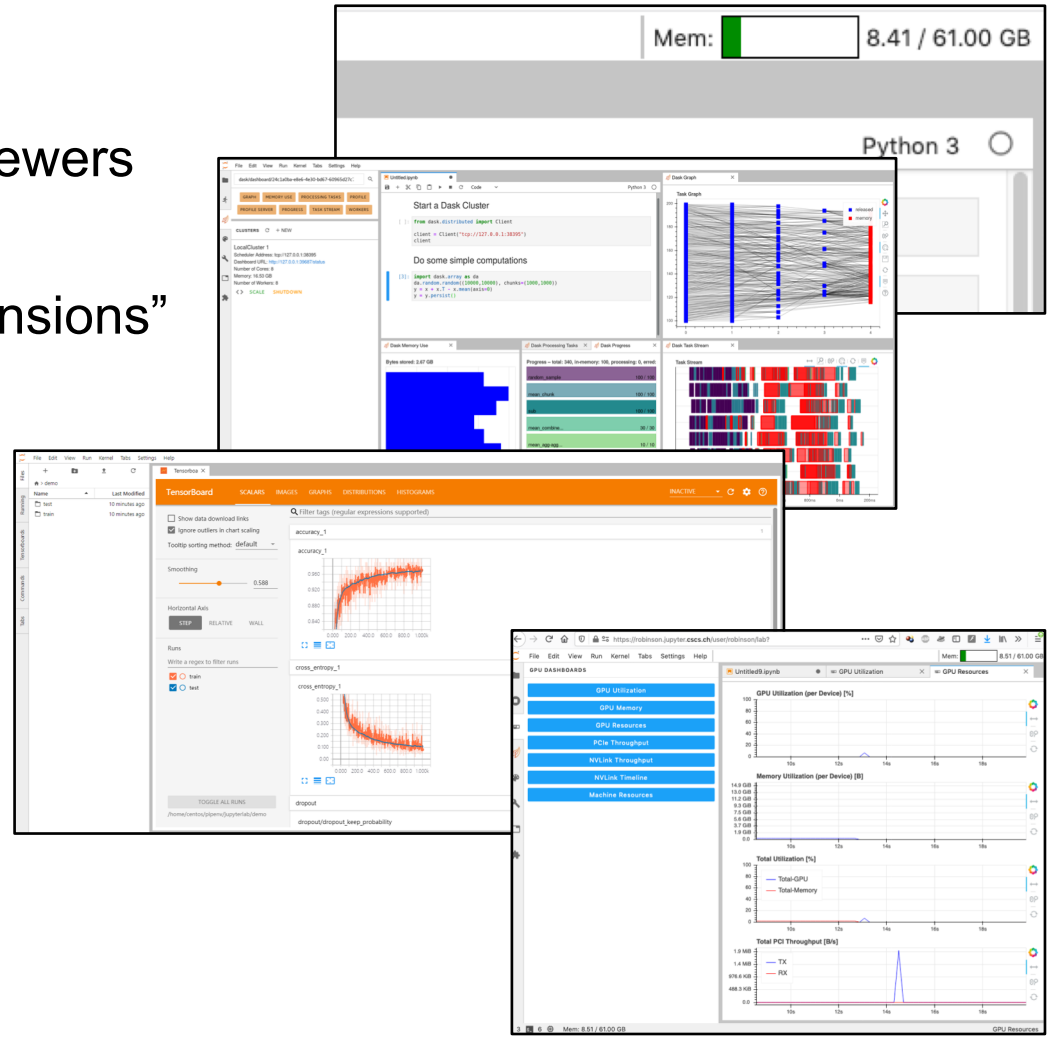
- JupyterLab is the next-generation user interface for Jupyter
- Higher degree of interaction between notebooks, documents, text editors and other activities (arrange with tabs/splitters)
- More advanced interactive development environment
- Uses the same notebook document format
- JupyterLab is the default interface at CSCS
- Classic notebook interface is still accessible by modifying the URL from /lab to /tree
  - Or: Help -> Launch Classic Notebook





JupyterLab Extensions

- JupyterLab is designed as a customizable, extensible environment
- Extensions can provide new themes, file viewers and editors, and renderers for rich output
- >100 GitHub repos tagged “jupyterlab-extensions”
- Examples include:
  - Memory and CPU usage, simple profiling
  - Dask dashboard
  - Tensorboard
  - GPU dashboards



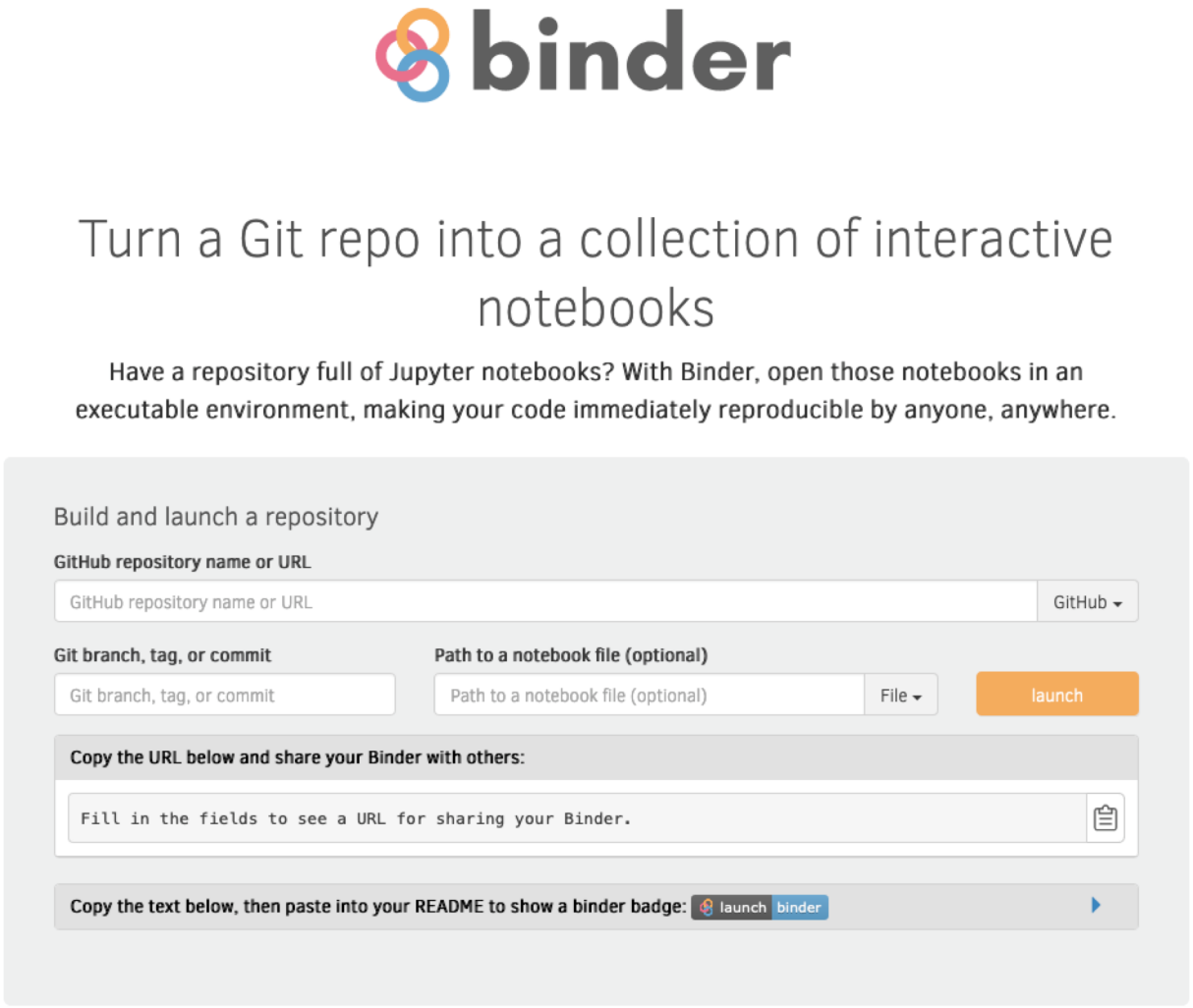
Using notebooks in the cloud…

Sharing static notebooks

https://nbviewer.jupyter.org/

Sharing live notebooks

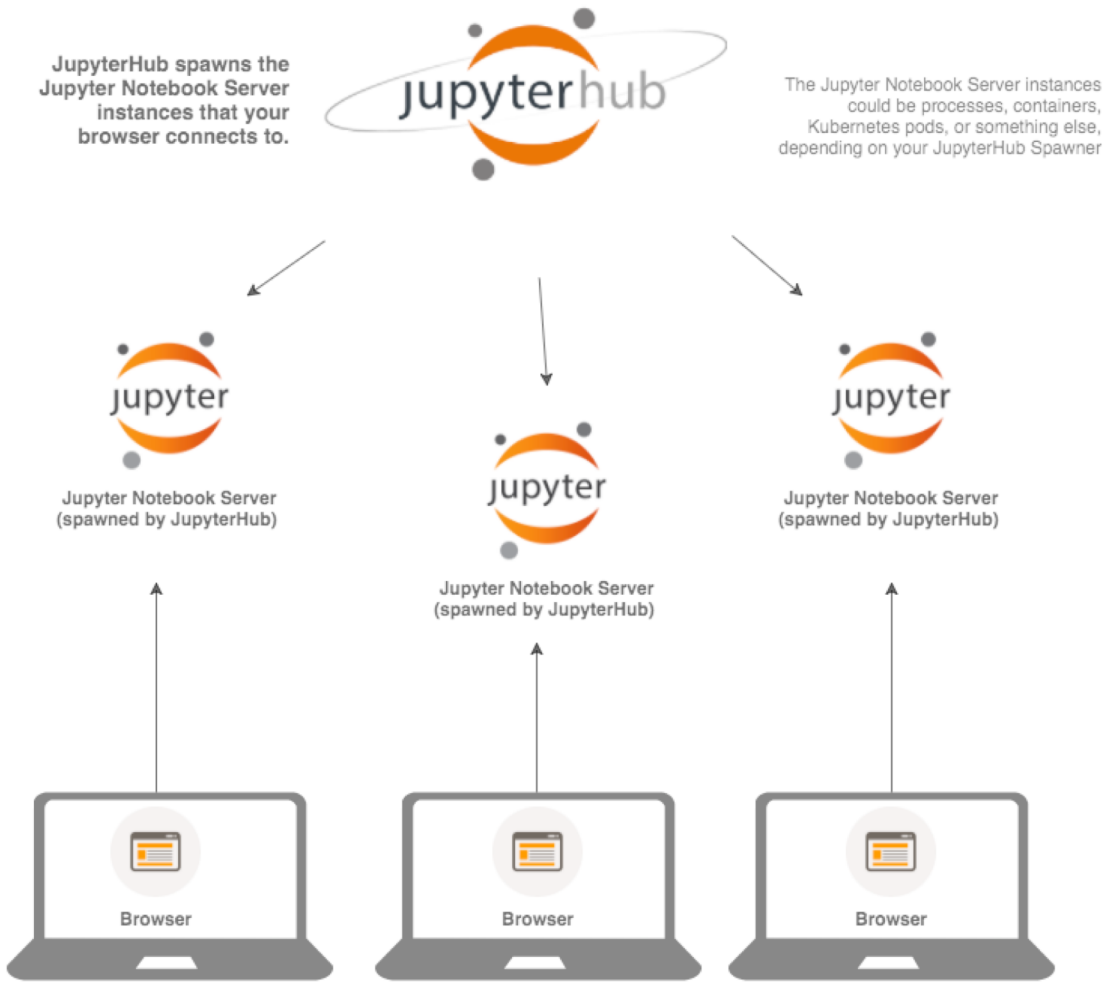
https://mybinder.org/



Using notebooks on Piz Daint: JupyterHub

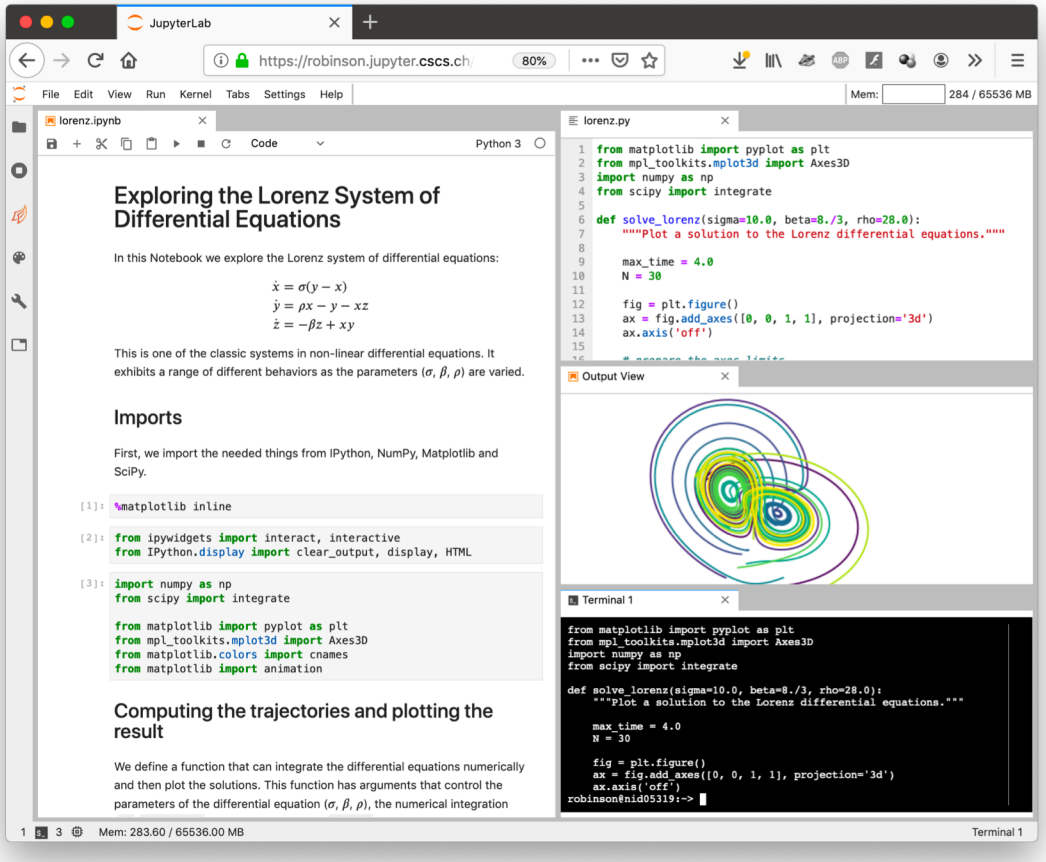
JupyterHub

- Multi-user server for Jupyter Notebooks (designed for classrooms, research labs, Universities...)
- Spawns, manages and proxies multiple instances of the single-user Jupyter Notebook server
- Main components:
  - **Hub** (tornado process)
  - **configurable http proxy** (node-http-proxy) that receives the requests from the client's browser
  - multiple **single-user Jupyter notebook servers** (Python/IPython/tornado) that are launched and monitored by **Spawners**
  - an **authentication class** that manages how users can access the system

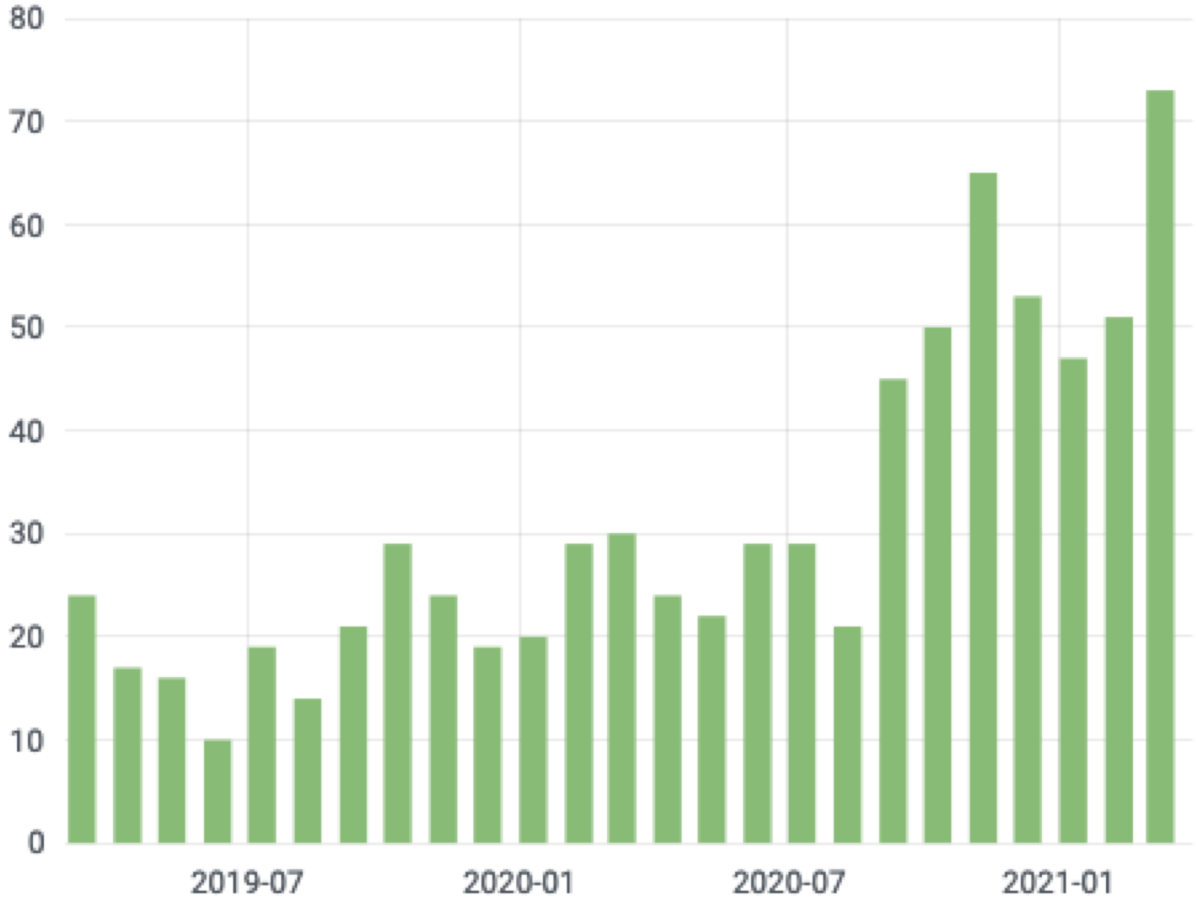


JupyterHub for Piz Daint

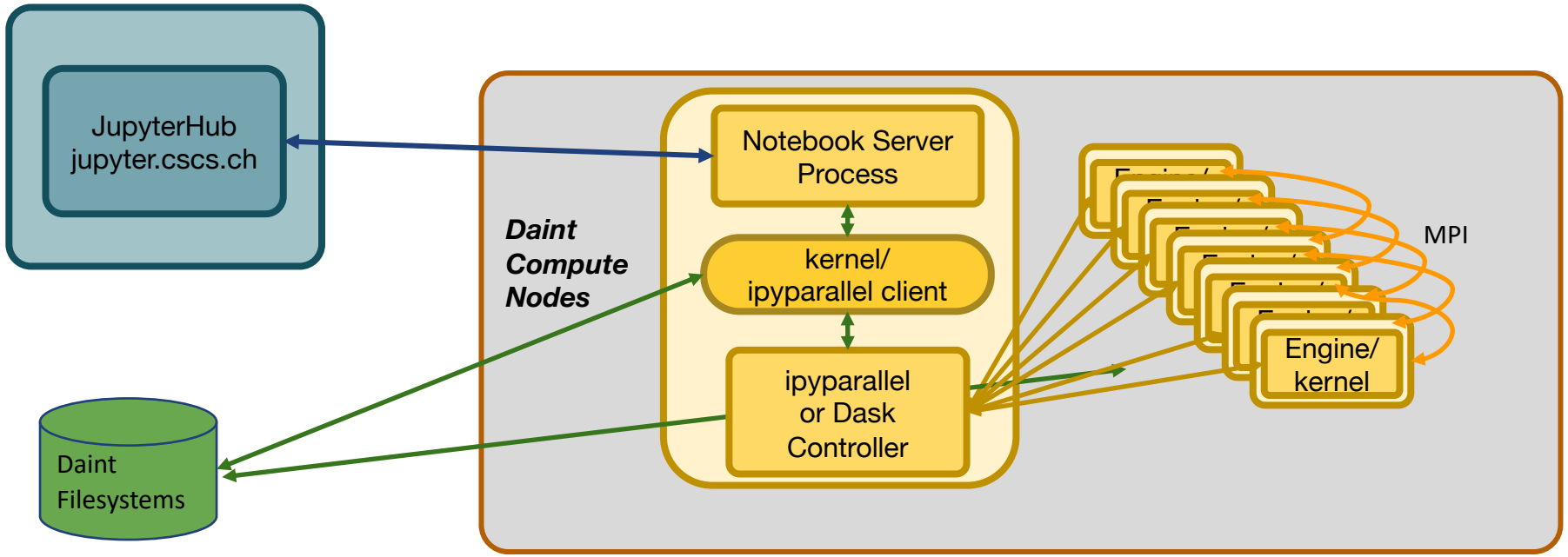
- JupyterHub for Piz Daint is accessible at:
  https://jupyter.cscs.ch
- Available to anyone with a compute allocation on Piz Daint
- Servers are launched on the compute nodes (GPU or multicore)
- You have dedicated access to the full compute capability of the node(s)
- Special Slurm reservations for interactive computing grow and shrink automatically according to demand
- In most cases, users will wait less than five minutes for a notebook server



JupyterHub monthly users on Piz Daint



JupyterHub on Piz Daint

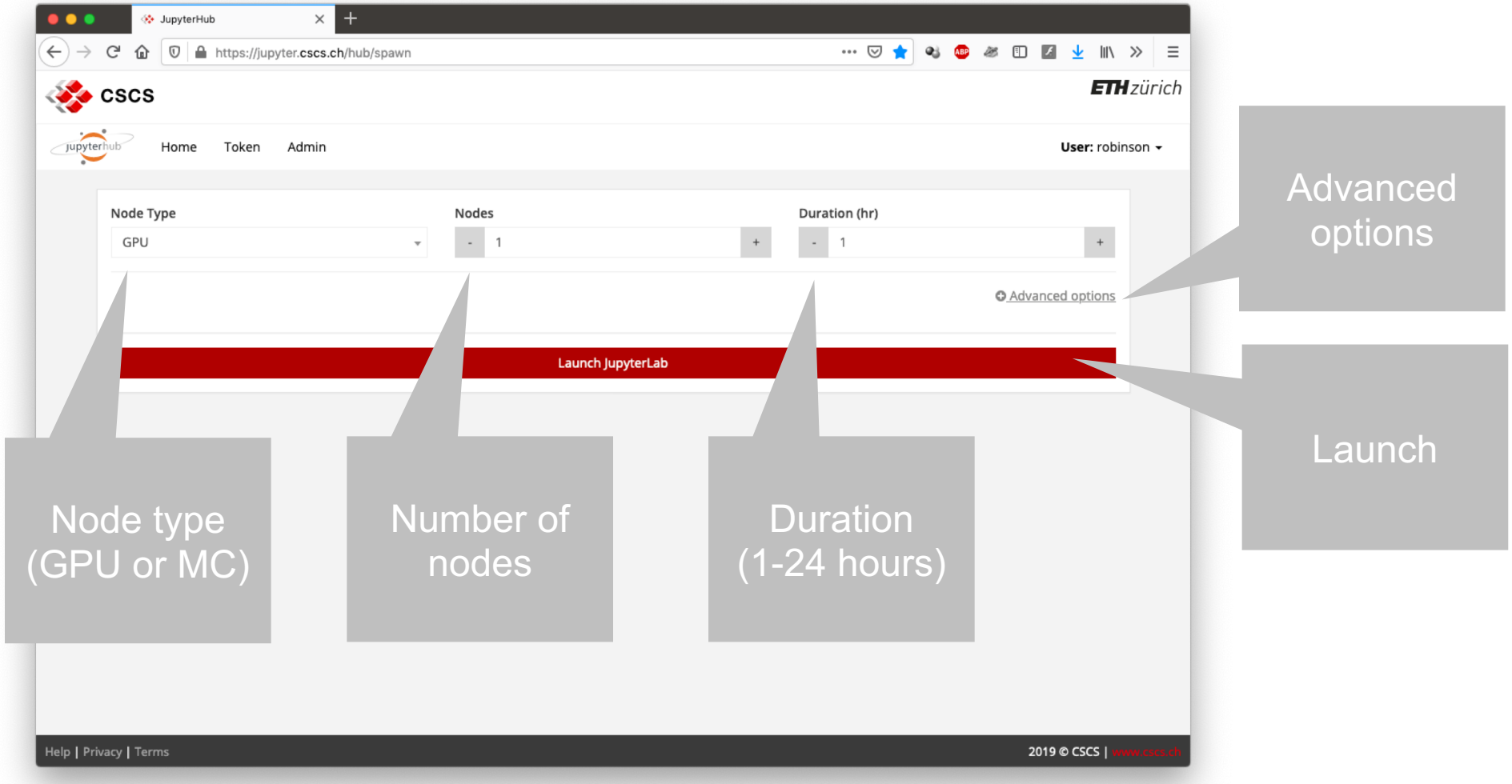


Challenges with interactive computing on HPC

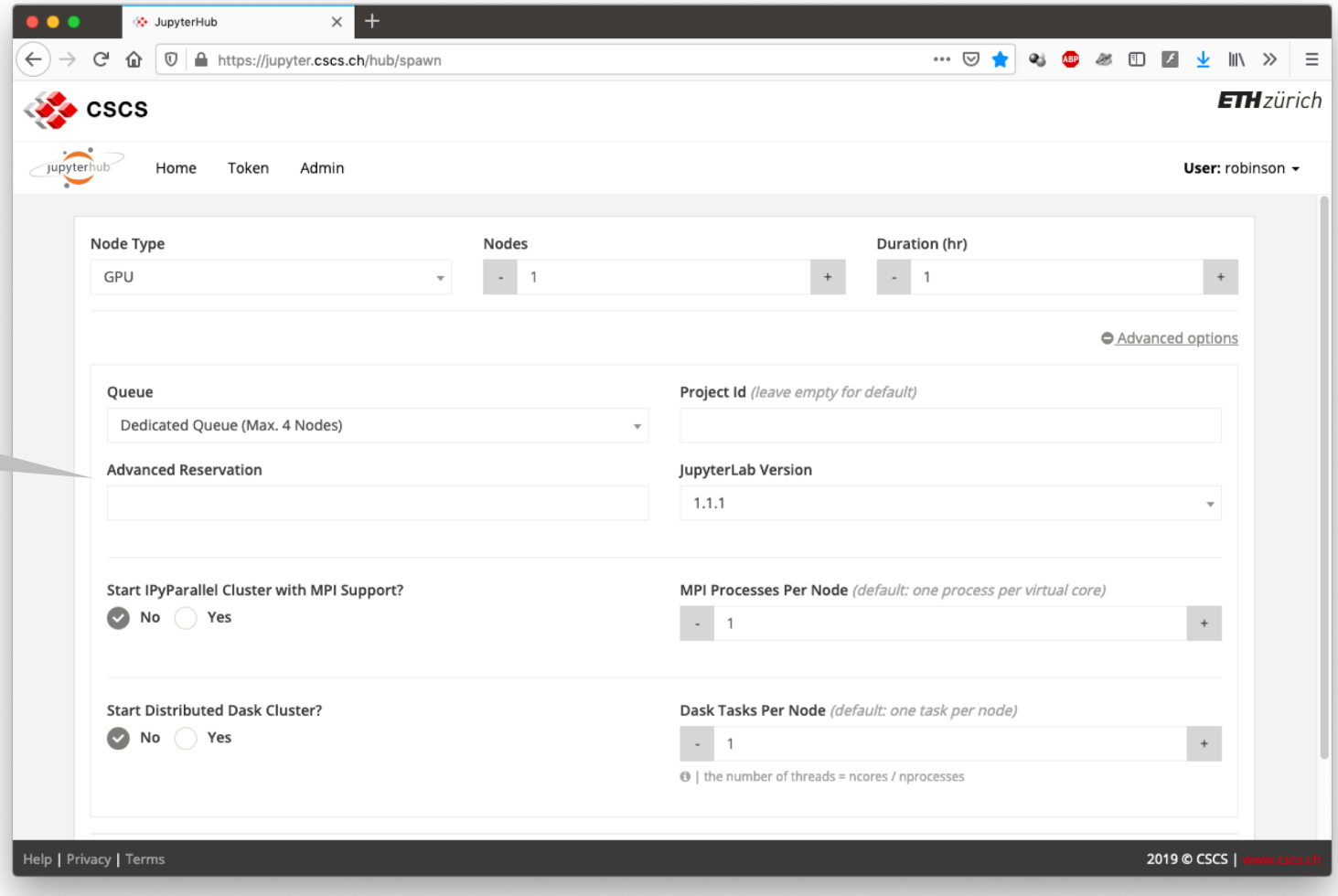
- Shared parallel filesystems
- Network connectivity
- Slurm
- Warm swapping
- All kinds of other possible issues - usually these are temporal!

- Debugging problems:
  - Check for any maintenance announcements on the spawner page
  - Check your own internet connectivity
  - Check for old JupyterHub sessions in browser tabs (especially if kernels are not starting or are slow to connect or restart)
  - Inspect the Jupyter log file in your $SCRATCH directory

Basic options



Advanced options

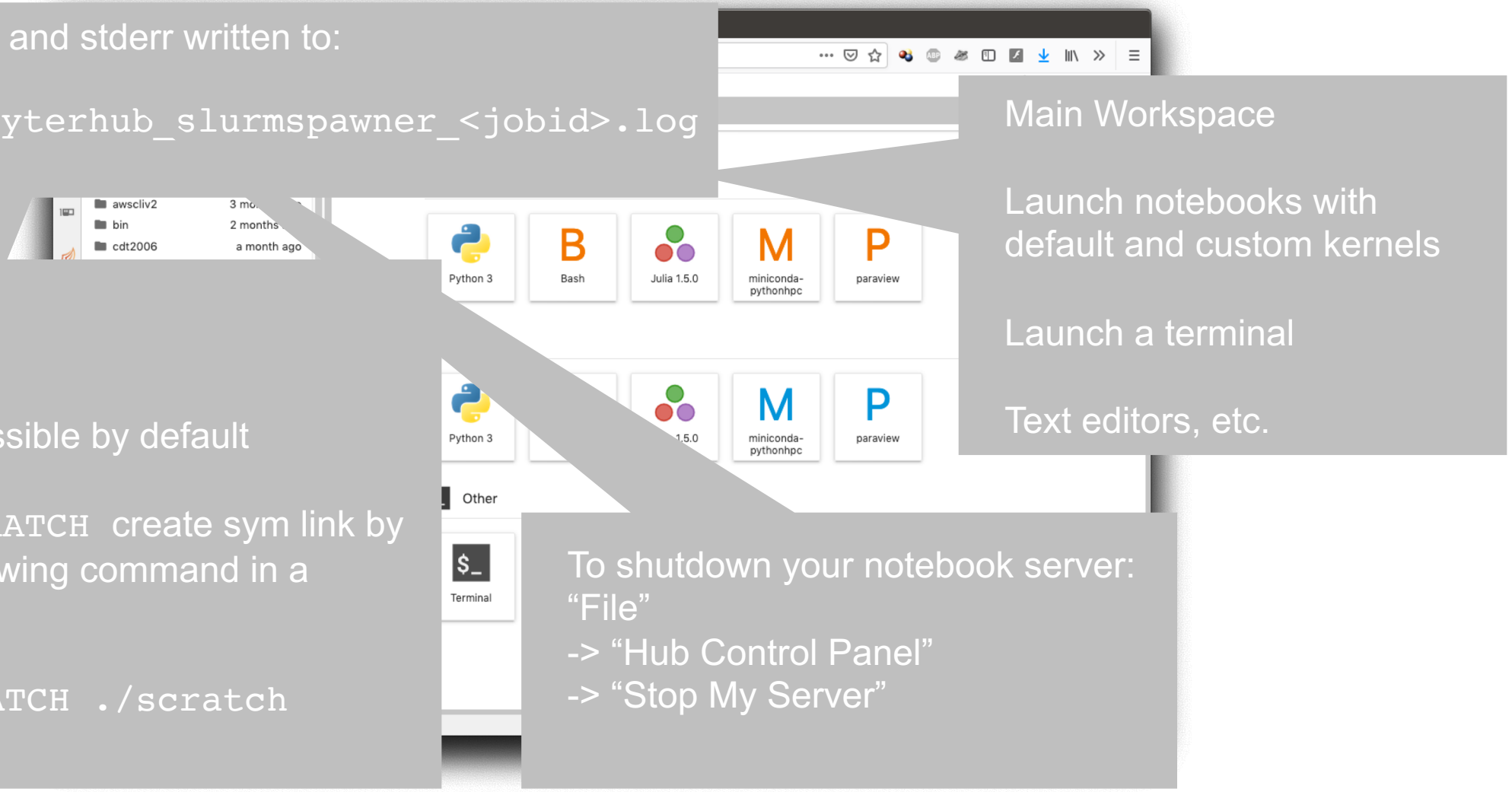


Reservation

JupyterLab interface

Standard output and stderr written to:

`$SCRATCH/jupyterhub_slurmspawner_<jobid>.log`



File Browser

`$HOME` is accessible by default

To access `$SCRATCH` create sym link by issuing the following command in a notebook cell:

```
!ln -s $SCRATCH ./scratch
```

Main Workspace

Launch notebooks with default and custom kernels

Launch a terminal

Text editors, etc.

To shutdown your notebook server:
“File”
-> “Hub Control Panel”
-> “Stop My Server”

Demo of JupyterLab interface

Let's get going…

Please open a browser (**Firefox** or **Chrome**, no guarantees for the others…) and visit…

https://jupyter.cscs.ch