Getting Started with Perlmutter

Hands-on tutorial to prepare participants for bootcamp activities

Kristy Henmueller
HPC Partnerships and Outreach Specialist
Argonne Leadership Computing Facility

5 Steps to Get Started on Perlmutter

1. Logging into Perlmutter
2. Navigating the Perlmutter File Systems
3. GitHub for Bootcamp
4. Jupyter Notebooks on Perlmutter
5. Running a Job on Perlmutter

1. Logging into Perlmutter

- **NERSC account web portal: iris.nersc.gov**
- **NERSC JupyterHub: jupyter.nersc.gov**

	Login Node	Shared GPU Node	Exclusive CPU Node	Exclusive GPU Node	Configurable Job
Perlmutter	start	start	start	start	start
Resources	Use a login node shared with other users, outside the batch queues.	Use a single GPU on a node within a job allocation using defaults.	Use your own node within a job allocation using defaults.		Use multiple compute nodes with specialized settings.
Use Cases	Visualization and analytics that are not memory intensive and can run on just a few cores.	Work that fits on a single GPU, and uses at most a quarter of a GPU node's CPU cores and host memory.	Visualization, analytics, machine learning that is compute or memory intensive but can be done on a single node.		Multi-node analytics jobs, jobs in reservations, custom project charging, and more.

Most bootcamp activities will require logging into Perlmutter via JupyterHub

2. Navigating the Perlmutter File Systems

- **Bootcamp project**: m4388

- **Storage spaces**:
  - —Home (~ 40 GB)
    - Path: `/global/homes/k/kstreu`
    - Access with: `cd $HOME`
  - —Scratch (~ 20 TB)
    - Path: `/pscratch/sd/k/kstreu`
    - Access with: `cd $PSCRATCH`
  - —Project or Community File System (CFS) (~ 20 TB)
    - Path: `/global/cfs/cdirs/m4388`
    - Access with: `cd $CFS/m4388`

- **Let's keep the project space organized!**
  - — Be careful working in the project space
  - — Do not overwrite files and folders owned by other users

- **`$PSCRATCH` for individual use**
- **`$CFS/m4388` for collaborating**

github.com/Wilber/Intro-HPC-2026

3. GitHub for Bootcamp

- **GitHub**: a website that stores code online in the form of Git repositories
- **Git**: a tool that tracks changes to files
- **Git repository (repo)**: a project folder stored online
- **Clone**: to make a copy of a repository form GitHub onto Perlmutter, or your computer, etc.)

- **A GitHub repo is like a shared Google Drive folder**
- **Cloning the repo is like downloading a working copy of the folder**

How to ‘Git’ Started

1. **Navigate to the bootcamp repository online**: https://github.com/Wilber/Intro-HPC-2026
   - Copy the repo code

2. **Open Terminal and navigate to your scratch directory (folder)**

3. **run** `git clone https://github.com/Wilber/Intro-HPC-2026.git`

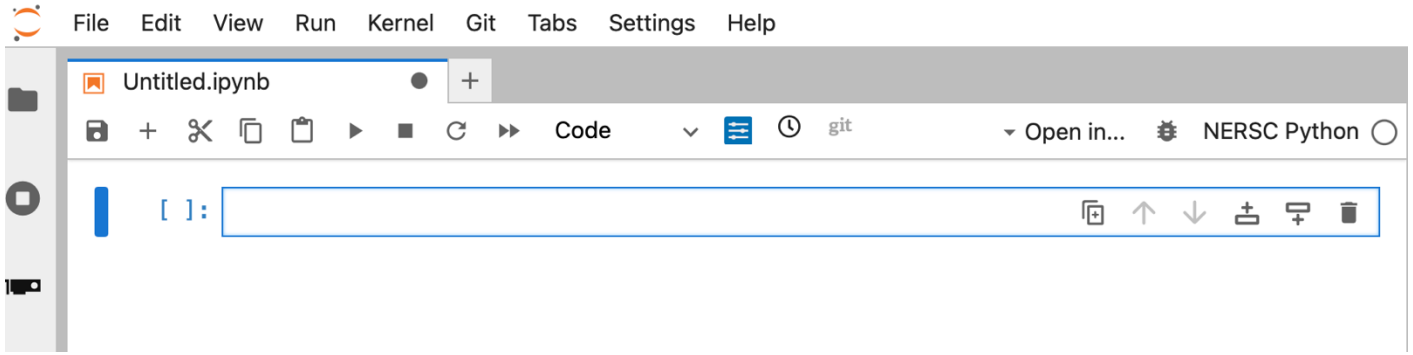
4. **Go into the cloned repo (project directory) and access files**
   - Open and edit Jupyter notebooks
   - Run and edit code

5. **Update your cloned repo as necessary with** `git pull`

- **GitHub is where the shared project lives**
- **`git clone` copies it to your computer**
- **`git pull` updated your copy if someone else makes changes**

4. What is a Jupyter Notebook?

- **A Jupyter notebook is an interactive document that combines:**
  - —**Text**: explanations, notes, equations
  - —**Code**: typically Python
  - —**Output**: numbers, tables, plots, images
  - —**Exploration**: change something → run it → see what happens



- **Google Colab is a Jupyter-style notebook and the basic ideas are the same**
  - —Check it out if you're curious: colab.research.google.com/notebooks/intro.ipynb

Anatomy of a Jupyter Notebook

Markdown cell

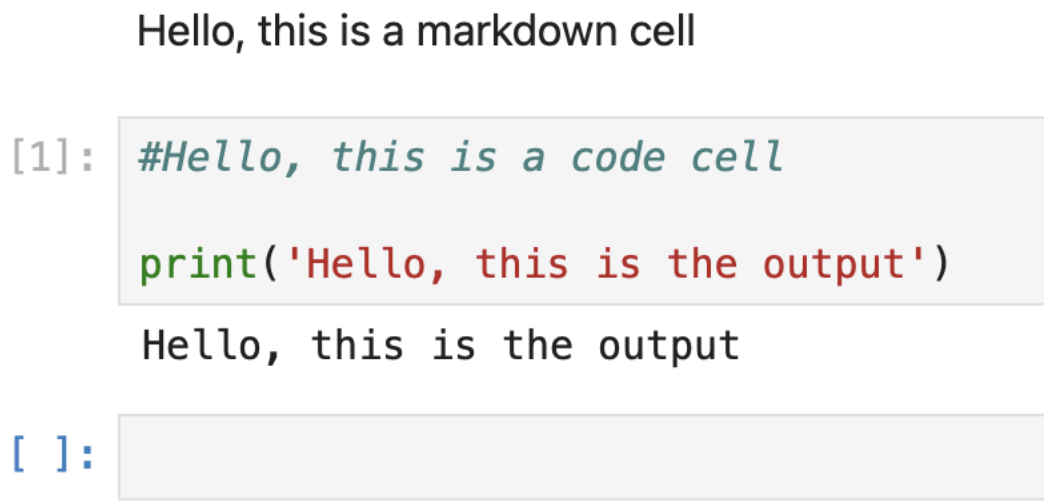
- Explains what we're doing

Code cell

- Instructions for Python to execute

Output

- The result produced by a code cell



Cells run independently, but variables persist between cells

Basic Workflow of a Jupyter Notebook

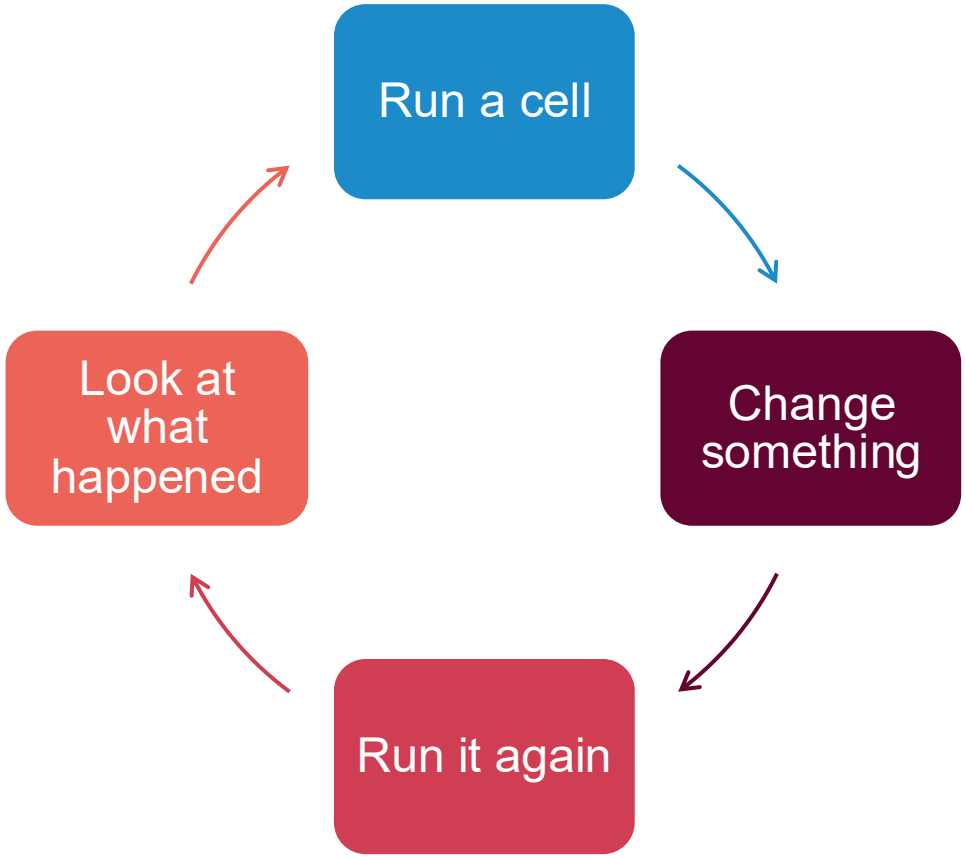
Don’t just read a Jupyter notebook.
Change things and see what happens.

What can you do in a Jupyter Notebook?

1. Markdown cell
   —Not code. Allows us to use notebooks to explain our thinking as well as run code.
2. Import a library
   —Libraries give us tools that Python doesn't provide automatically.
3. Create some data to explore
   ```
   df
   ```
4. Explore the data
   ```
   df.head()
   df.describe()
   ```
5. Change something
   ```
   head() to tail()
   head() to head(3)
   ```
6. Calculate something
   ```
   mean()
   ```
7. Create a Visualization
   ```
   plt.scatter()
   ```

Jupyter Notebook Survival Guide

- **Run Cells**
  - —Shift + Enter
    - Runs the current cell and moves to the next one
  - —Ctrl/Cmd + Enter
    - Runs the current cell and stays there
- **Move quickly**
  - —Esc → A
    - Add a cell above
  - —Esc → B
    - Add a cell below
  - —Esc → M
    - Change cell to Markdown
  - —Esc → Y
    - Change cell to Code
- **Explore**
  - —Place cursor after a function and press Shift + Tab
    - See information about the function



Have fun using Jupyter to experiment!

5. Running a Job on Perlmutter

- **Command for running a job in terminal**:

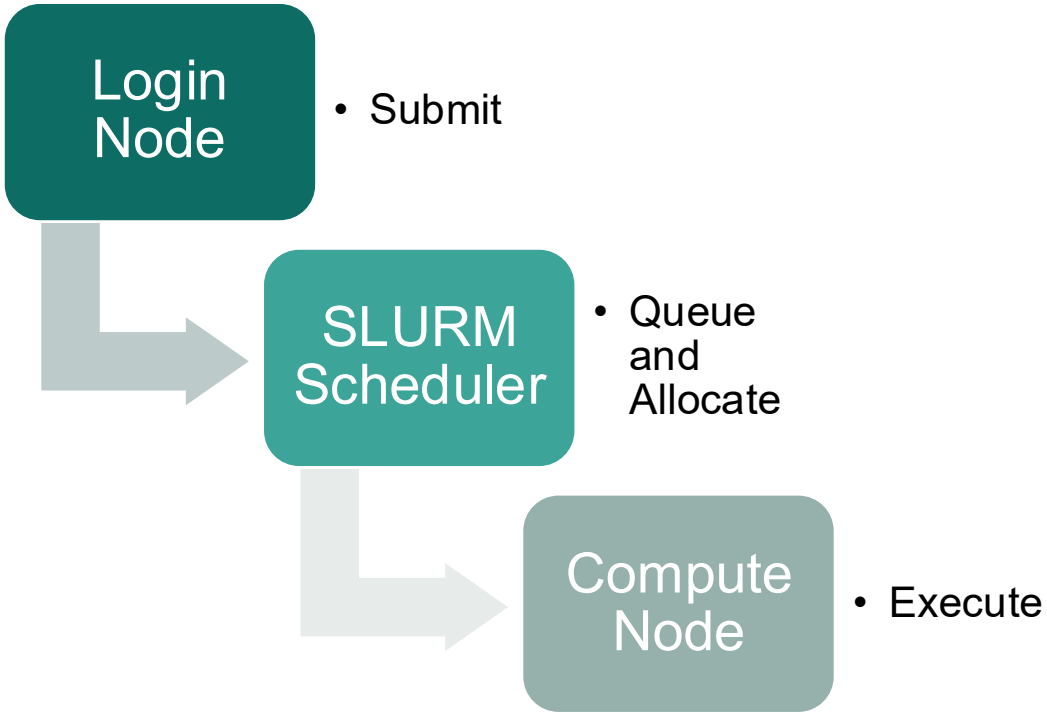
  ```
  sbatch filename.sh
  ```

- **Let's try it with `01_hello_res.sh`**

  —You can also use `01_hello.sh` when there is no reservation available.

What is going on when you run a job on Perlmutter?

1. **Write and submit**
   - Package your code and inputs into a **job script** and submit it to the scheduler.
2. **Queue**
   - The scheduler (Slurm) waits until the requested resources are available.
3. **Allocate**
   - Perlmutter assigns you **compute nodes / GPUs / CPUs / memory / time.**
4. **Run**
   - Your program executes in parallel across the allocated hardware.
5. **Communicate**
   - Processes exchange data over the supercomputer's **high-speed network** and access shared storage.
6. **Finish and store**
   - Results, logs, and output files are written back to **parallel storage**; resources are released.



Resources

- NERSC/Perlmutter
  - —docs.nersc.gov/beginner-guide
  - —docs.nersc.gov/getting-started
- Jupyter
  - —docs.nersc.gov/services/jupyter
  - —docs.jupyter.org/en/latest/
- GitHub
  - —docs.github.com/en/get-started/using-github/hello-world