What Is Jupyter?

Interactive open-source web application

Allows you to **create** and **share** documents, “notebooks,” containing:

- Live code
- Equations
- Visualizations
- Narrative text
- Interactive widgets

Things you can use Jupyter notebooks for:

- Data cleaning and data transformation
- Numerical simulation
- Statistical modeling
- Data visualization
- Machine learning
- Workflows and analytics frameworks
Why Does NERSC Care About Jupyter?

Integral part of Big (Data) Science & Superfacility:
LSST-DESC, DESI, ALS, LCLS, Materials Project, NCEM, LUX, LZ, KBase

Generational shift in data science:
UCB’s Data 8 course, entirely in Jupyter
“I’ll send you a copy of my notebook”
Training events adopting notebooks (DL)

Reproducibility and science outreach:
Open source code and open science
Jupyter notebooks alongside publications
Jupyter Usage at NERSC

For comparison, about 3000 users per month connect via ssh
NERSC and Jupyter: 7 Years Together

- JupyterHub as NERSC “science gateway” app
- Deploy hubs via Docker (Spin)
- Jupyter on Cori via JupyterHub
- JupyterLab beta becomes default
- More Cori nodes; CPU, GPU batch
- Jupyter on Perlmutter GPU
- More Cori nodes; expand batch access
- Jupyter on Perlmutter CPU
- Jupyter on Perlmutter CPU
- IPython → Jupyter
- batchspawner
- JupyterLab beta
- jupyterlab-favorites+recents
- jupyterlab-slurm
- jupyter-server-proxy
- JupyterLab 3
- Named servers UI
- NERSC hubs merged
- Cori
- Edison
- Perlmutter
- 2014
- 2015
- 2016
- 2017
- 2018
- 2019
- 2020
- 2021
- 2022
OK, How Do I Use Jupyter at NERSC?

Jupyter at NERSC is provided through a JupyterHub deployment we manage:
Redirects you to authenticate if needed
Spawns a notebook server for you somewhere at NERSC
Manages communication between you and your notebook
Keeps track of and manages your notebook process
Can provide helpful additional services

https://jupyter.nersc.gov/

Authenticate
Choose
Go!
How Do I Choose a Notebook Server to Spawn?

**Perlmutter Shared CPU:**
Notebook on Perlmutter login node
There are like 40 of those nodes!
Can see /cfs, $HOME, etc
Can see Perlmutter $SCRATCH
Same Python env as ssh login
Can submit jobs via `sbatch`

**Cori Shared CPU Node:**
Notebook on cori{13,14,16,19}
That’s right, just 4 nodes
Can see /cfs, $HOME, etc
Can see Cori $SCRATCH
Same Python env as ssh login
Can submit jobs via `sbatch`

**Cori GPU Node Options**
Enabled if you have GPU QOS
Notebook on cgpu{01-18}
Runs in a job allocation (4h for shared node)

**Perlmutter GPU Node Options**
Notebook in job allocations
CPU node or GPU node

**Other Perlmutter Options**
Notebook in job allocations
CPU node or GPU node

**NERSC resources that Jupyter depends on appear to be in maintenance or having issues. This may impact Jupyter. See the NERSC MOTD for further information. Perlmutter status: degraded**

**Use Cases**
- Visualization and analytics that are not memory intensive and can run on just a few cores.
- Visualization, analytics, and 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 changes, and more.

**Shared ⇒ Other users are on the same node as you**
JupyterLab Interface
JupyterLab Interface: NERSC Goodies

Favorites (NERSC/jupyterlab-favorites)
Bookmark your favorite places on the file system
Prepopulate with $HOME and ${C,P}SCRATCH
Add the current directory by clicking the ★ icon
Recents (NERSC/jupyterlab-recents)
Recent locations you’ve visited on the file system

Open from Path...
Jump to where you want to go on the file system
Kernels: How You Compute with Jupyter

Your Own Jupyter Kernel

Most common Jupyter question:
“How do I take a conda environment and use it from Jupyter?”

Several ways to accomplish this, here’s the easy one.

$ module load python
$ conda create -n myenv python=3.9
$ source activate myenv
(myenv) $ conda install ipykernel <other-packages>...
(myenv) $ python -m ipykernel install --user --name myenv-jupyter

Point your browser to jupyter.nersc.gov.
(You may need to restart your notebook server via control panel).
Kernel “myenv-jupyter” should be present in the kernel list.
The kernelspec File

```
(myenv) rthomas@cori01:~> cat \\
   $HOME/.local/share/jupyter/kernels/myenv-jupyter/kernel.json
{
   "argv": [
      "/global/homes/r/rthomas/.conda/envs/myenv/bin/python",
      "-m",
      "ipykernel_launcher",
      "-f",
      "{connection_file}"
   ],
   "display_name": "myenv-jupyter",
   "language": "python"
}
```
Additional Customization

```json
{
    "argv": [
        "/global/homes/r/rthomas/.conda/envs/myenv/bin/python",
        "-m",
        "ipykernel_launcher",
        "-f",
        "{connection_file}"
    ],
    "display_name": "myenv-jupyter",
    "language": "python",
    "env": {
        "PATH": ...
    },
    "LD_LIBRARY_PATH": ...
}
```
Additional Customization

```json
{
    "argv": [
        "/global/homes/r/rthomas/jupyter-helper.sh",
        "-f",
        "{connection_file}\"
    ],
    "display_name": "myenv-jupyter2",
    "language": "python",
}
```

Meanwhile, in jupyter-helper.sh:

```
#!/bin/bash
export SOMETHING=123
module load foo
exec python -m ipykernel "$@"
```

The helper script is the most flexible approach for NERSC users since it easily enables use of modules, environment variables, etc.
A Shifter Kernelspec

```json
{
    "argv": [
        "shifter",
        "--image=continuumio/anaconda3:latest",
        "/opt/conda/bin/python",
        "-m",
        "ipykernel_launcher",
        "-f",
        "{"connection_file}"  
    ],
    "display_name": "my-shifter-kernel",
    "language": "python"
}
```
Debugging Jupyter Stuff

(myenv) rthomas@cori01:~> cat ~/.jupyter-cori.log

[I 2018-03-19 16:00:08.175 SingleUserNotebookApp manager:40] [nb_conda_kernels] enabled, 5 kernels found
[I 2018-03-19 16:00:08.248 SingleUserNotebookApp extension:54] JupyterLab application directory is /global/common/cori/software/python/3.6-anaconda-4.4/share/jupyter/lab
[I 2018-03-19 16:00:09.123 SingleUserNotebookApp handlers:73] [nb_anacondacloud] enabled
[I 2018-03-19 16:00:09.129 SingleUserNotebookApp handlers:292] [nb_conda] enabled
[I 2018-03-19 16:00:09.181 SingleUserNotebookApp __init__:35] ✓ nbpresent HTML export ENABLED
[W 2018-03-19 16:00:09.181 SingleUserNotebookApp __init__:43] ✗ nbpresent PDF export DISABLED: No module named 'nbbrowserpdf'
[I 2018-03-19 16:00:09.186 SingleUserNotebookApp singleuser:365] Starting jupyterhub-singleuser server version 0.8.0.rc1
[I 2018-03-19 16:00:09.194 SingleUserNotebookApp notebookapp:1445] 0 active kernels
[I 2018-03-19 16:00:09.194 SingleUserNotebookApp notebookapp:1445] The Jupyter Notebook is running at:
[I 2018-03-19 16:00:09.194 SingleUserNotebookApp notebookapp:1445] http://0.0.0.0:56901/user/rthomas/
[I 2018-03-19 16:00:09.194 SingleUserNotebookApp notebookapp:1446] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[I 2018-03-19 16:00:09.236 SingleUserNotebookApp log:122] 302 GET /user/rthomas/ → /user/rthomas/tree/global/homes/r/rthomas? (::ffff:10.42.245.15) 0.39ms
Jupyter at NERSC

- Go to https://jupyter.nersc.gov to use Jupyter at NERSC
- Use a kernel-spec to use a conda environment in your notebook
- You can customize those kernelspec files in many ways
- We work on making Jupyter work and work better for you

Always looking for:
New ways to empower Jupyter users
Feedback, advice, and even help:
https://help.nersc.gov/
rcthomas@lbl.gov
Thank You and Welcome to NERSC!