

# Jupyter For the User New to NERSC



New User Training  
September 28, 2022

Rollin Thomas  
Data Science Engagement Group



# 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

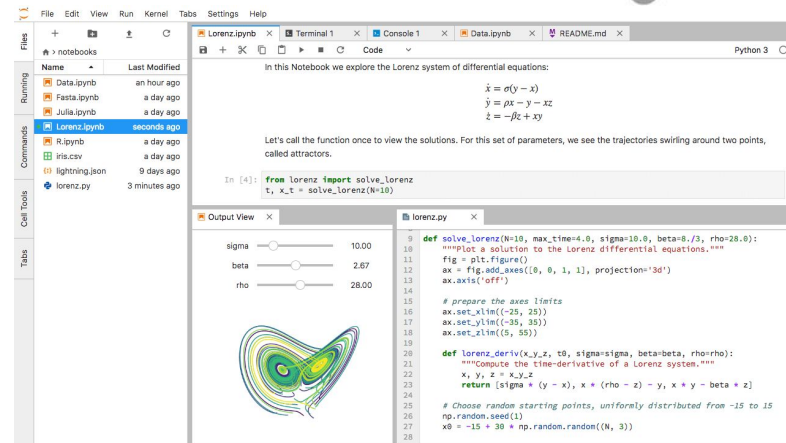
Numerical simulation

Statistical modeling

Data visualization

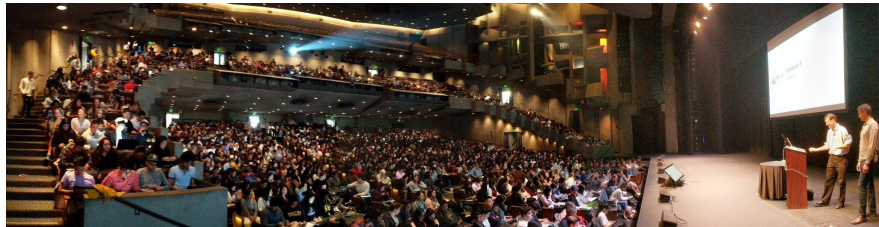
Machine learning

Workflows and analytics frameworks





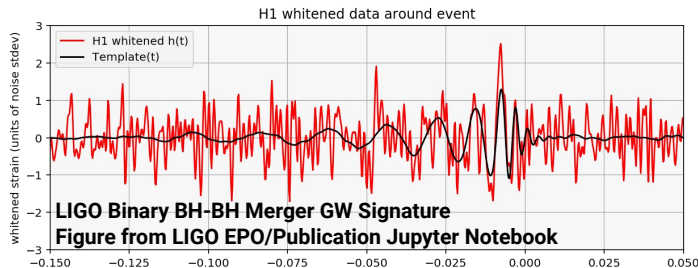
# Why Does NERSC Care About Jupyter?



Data 8: Foundations of Data Science, Fall 2018, Zellerbach Hall

## 2017 ACM Software System Award:

“... *a de facto standard for data analysis in research, education, journalism and industry*. Jupyter has broad impact across domains and use cases. Today more than **2,000,000 Jupyter notebooks are on GitHub**, each a distinct instance of a Jupyter application—covering a range of uses from technical documentation to course materials, books and academic publications.”



## 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)

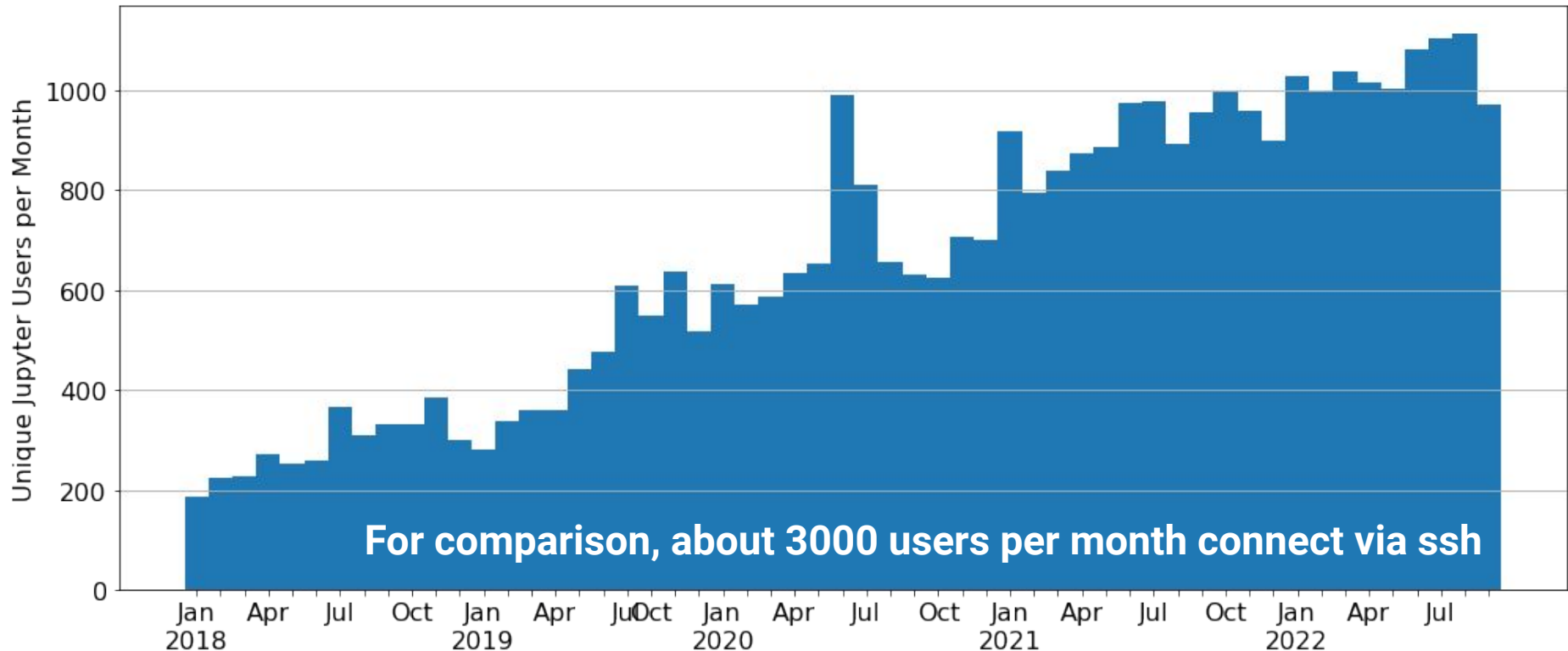
## Reproducibility and science outreach:

Open source code and open science

Jupyter notebooks alongside publications

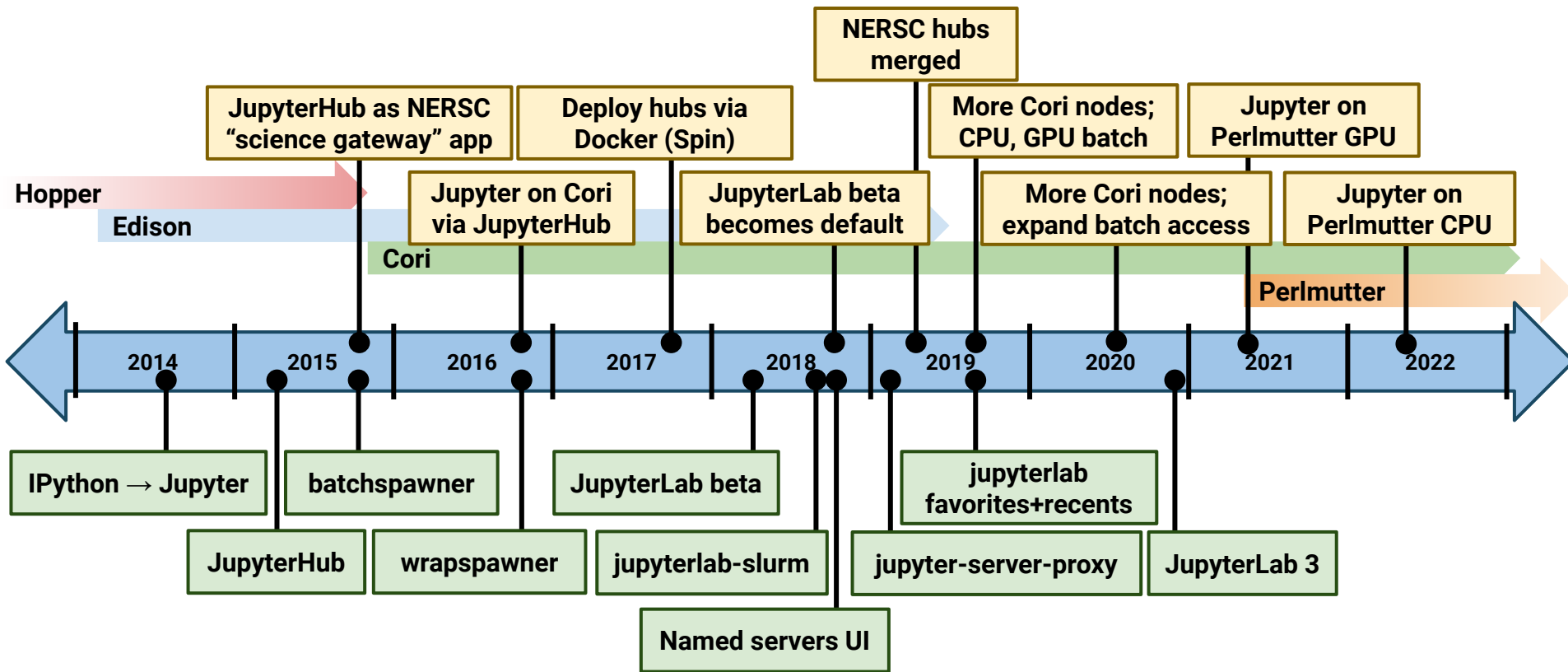


# Jupyter Usage at NERSC





# NERSC and Jupyter: 7 Years Together





# 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**

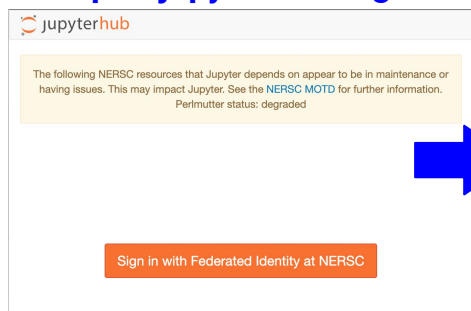
**Spawns a notebook server for you somewhere at NERSC**

**Manages communication between you and your notebook**

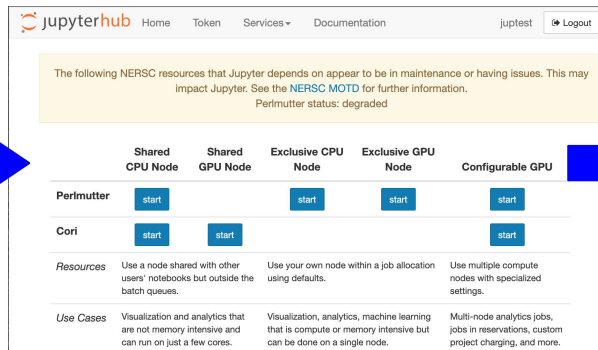
**Keeps track of and manages your notebook process**

**Can provide helpful additional services**

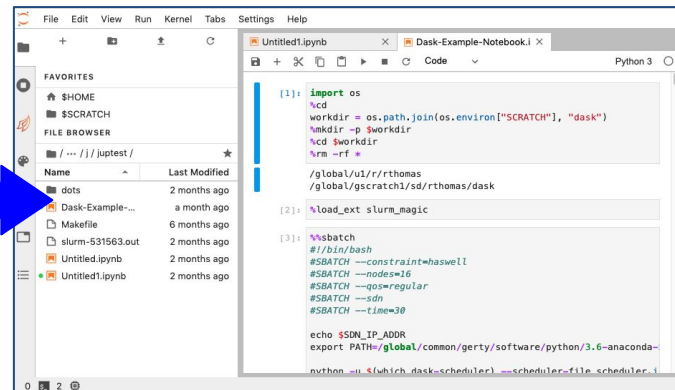
<https://jupyter.nersc.gov/>



**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 **lsbatch**

## Cori Shared CPU Node:

Notebook on cori{13,14,16,19}

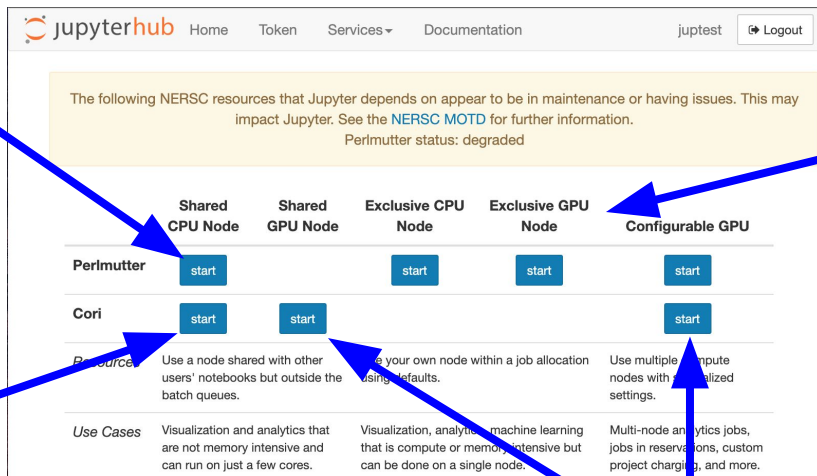
That's right, just **4** nodes

Can see /cfs, \$HOME, etc

Can see Cori \$SCRATCH

Same Python env as ssh login

Can submit jobs via **lsbatch**



	Shared CPU Node	Shared GPU Node	Exclusive CPU Node	Exclusive GPU Node	Configurable GPU
Perlmutter	<a href="#">start</a>		<a href="#">start</a>	<a href="#">start</a>	<a href="#">start</a>
Cori	<a href="#">start</a>	<a href="#">start</a>			<a href="#">start</a>

**Resource**

Shared CPU Node	Use a node shared with other users' notebooks but outside the batch queues.	Shared GPU Node	Use your own node within a job allocation using defaults.	Exclusive CPU Node	Use multiple compute nodes with specialized settings.
-----------------	---	-----------------	---	--------------------	---

**Use Cases**

Shared CPU Node	Visualization and analytics that are not memory intensive and can run on just a few cores.	Shared GPU Node	Visualization, analytics, machine learning that is compute or memory intensive but can be done on a single node.	Exclusive GPU Node	Multi-node analytics jobs, jobs in reservations, custom project charging, and more.
-----------------	--	-----------------	--	--------------------	---

## Other Perlmutter Options

Notebook in job allocations  
CPU node or GPU node

**Shared ⇒ Other users  
are on the same node  
as you**

## Cori GPU Node Options

Enabled if you have GPU QOS

Notebook on cgpu{01-18}

Runs in a job allocation

(4h for shared node)



# JupyterLab Interface

The screenshot displays the JupyterLab interface with a dark theme. On the left is a sidebar with a 'FAVORITES' section containing links to '\$SPSCRATCH', '\$CSCRATCH', and '\$HOME'. Below this is a 'FILE BROWSER' showing a directory tree for '/ / r / rthomas /'. The main area is divided into two panes. The top pane is a notebook titled 'analyze-jupyter-users.ipynb' with a 'No Kernel' status. It contains a title 'Analyze Jupyter Users', a paragraph of text, and a code cell with the following Python code:

```
[ ]: import os

import dask
from dask.distributed import Client

scheduler_file = os.path.join(os.environ["SCRATCH"], "scheduler.json")
dask.config.config["distributed"]["dashboard"]["link"] = (
    "{JUPYTERHUB_SERVICE_PREFIX}proxy/{host}:{port}/status"
)

[ ]: client = Client(scheduler_file=scheduler_file)
client

[ ]: client.restart()

[ ]: import glob

tasks = client.map(
    lambda host: glob.glob(f"{os.environ['SCRATCH']}/{host}/*.top"),
    ["cori13", "cori14", "cori16", "cori19"]
)

[ ]: from dask.distributed import as_completed

paths = list()
for future, result in as_completed(tasks, with_results=True):
    paths += result
paths.sort()
len(paths)
```

The bottom pane is a terminal window titled 'Terminal 1' showing the command prompt 'rthomas@login34:/global/ui/r/rthomas>'.

At the bottom of the interface, a status bar shows 'Simple', 'python-volta | Disconnected', 'Mem: 245.59 MB', 'Mode: Command', and 'Ln 1, Col 1 analyze-jupyter-users.ipynb'.



# JupyterLab Interface: NERSC Goodies

The screenshot displays the JupyterLab interface. On the left, the 'FAVORITES' sidebar is highlighted with a red box and a red arrow pointing to the 'FILE BROWSER' section. The 'FILE BROWSER' shows a list of files and directories, including '2022', 'add\_pm', 'dask-worker-space', 'has-toml', 'julia-1.4.0', 'jupytercon-jupyter...', 'mapper', 'meltdown-spectre', 'mingyuli', 'mmedford', 'sw', 'tmp', 'tmp1', 'tmp2', 'tmp3', 'tmp4', 'toast', 'vm-config', 'work', 'xalt', '10GB.dat', and '2021-09-03-dask...'. The main area shows a Jupyter notebook titled 'Analyze Jupyter Users' with Python code for interacting with Dask. The terminal at the bottom shows the command prompt 'rthomas@login34:/global/ui/r/rthomas>'.

**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



# JupyterLab Interface: NERSC Goodies

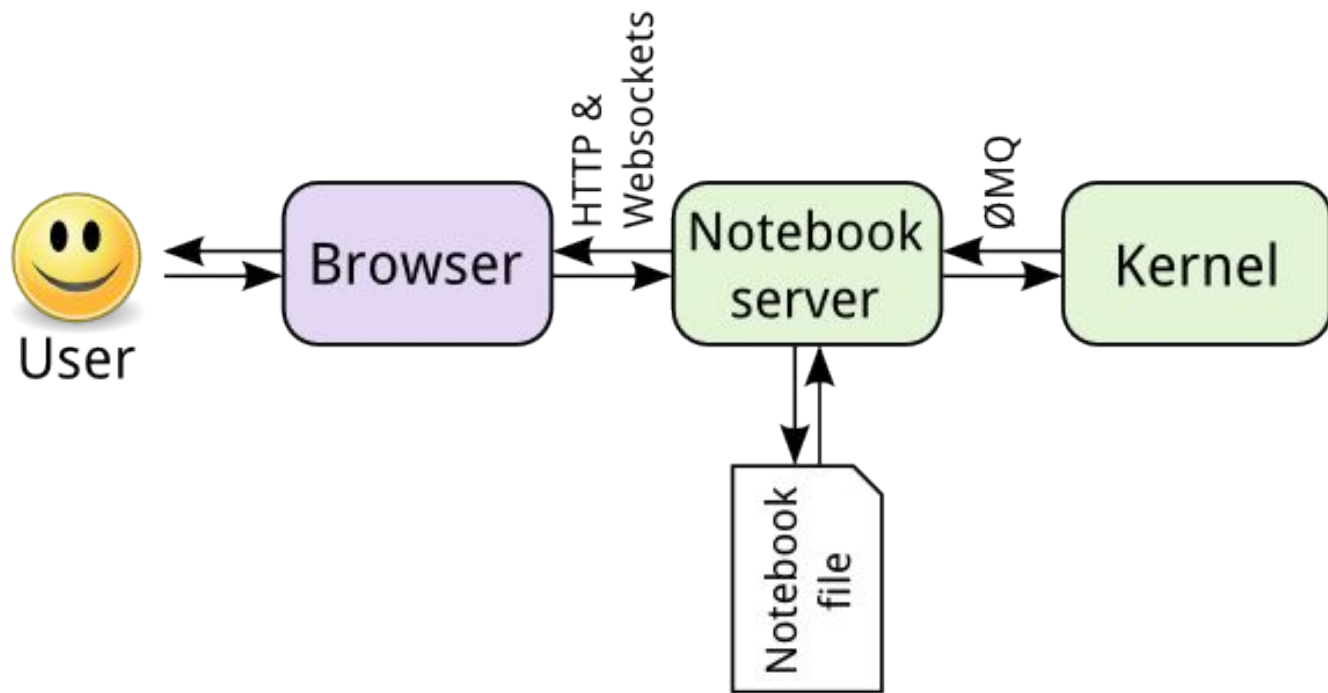
The screenshot shows the JupyterLab interface with a notebook titled "Analyze Jupyter Users". The left sidebar contains a "FILE BROWSER" and a "RECENTS" list. The "FILE BROWSER" shows a tree view of the file system. The "RECENTS" list shows a list of recent files and folders. The notebook editor displays a Python script. The terminal at the bottom shows the command prompt.

**Open from Path...**  
Jump to where you want to go on the file system

**Recents (NERSC/jupyterlab-recents)**  
Recent locations you've visited on the file system



# Kernels: How You Compute with Jupyter



[https://jupyter.readthedocs.io/en/latest/architecture/how\\_jupyter\\_ipython\\_work.html](https://jupyter.readthedocs.io/en/latest/architecture/how_jupyter_ipython_work.html)



# 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](http://jupyter.nersc.gov).

(You may need to restart your notebook server via control panel).

Kernel “myenv-jupyter” should be present in the kernel list.

This creates a  
“kernelpec” file.



# 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

```
{  
  "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

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

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



**Meanwhile, in jupyter-helper.sh:**

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



# A Shifter Kernelspec

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

Image name

Path to Python in the image



SHIFTER



# Debugging Jupyter Stuff

➡ **YOUR FRIEND!!!**

```
(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:53] JupyterLab beta preview extension loaded from
/usr/common/software/python/3.6-anaconda-4.4/lib/python3.6/site-packages/jupyterlab
[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 16:00:09.190 SingleUserNotebookApp log:122] 302 GET /user/rthomas/ →
/user/rthomas/tree/global/homes/r/rthomas? (@128.55.206.24) 0.62ms
[I 2018-03-19 16:00:09.194 SingleUserNotebookApp notebookapp:1445] Serving notebooks from local directory: /
[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](https://help.nersc.gov/rcthomas@lbl.gov)



Thank You and  
Welcome to  
NERSC!

