

# Using Jupyter at NERSC



New User Training  
June 16, 2020

Rollin Thomas  
Data and Analytics Services 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

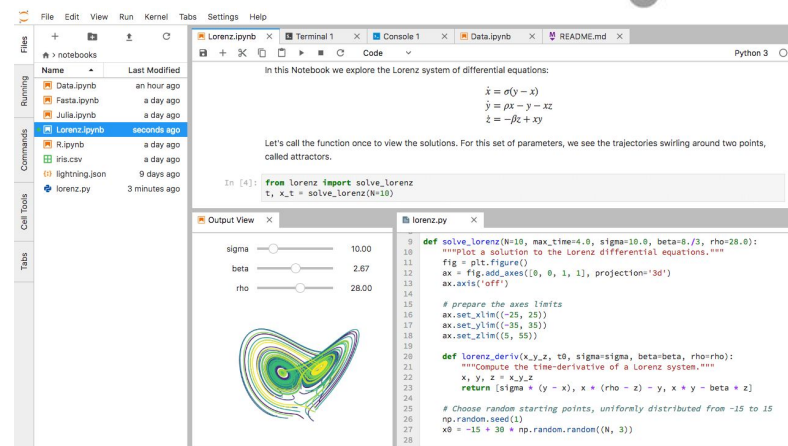
Numerical simulation

Statistical modeling

Data visualization

Machine learning

Workflows and analytics frameworks



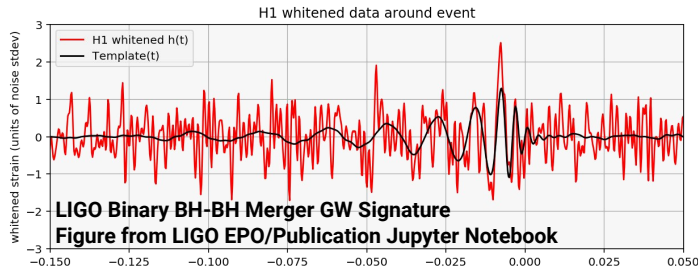
# Why Does NERSC Care About Jupyter Usage?



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

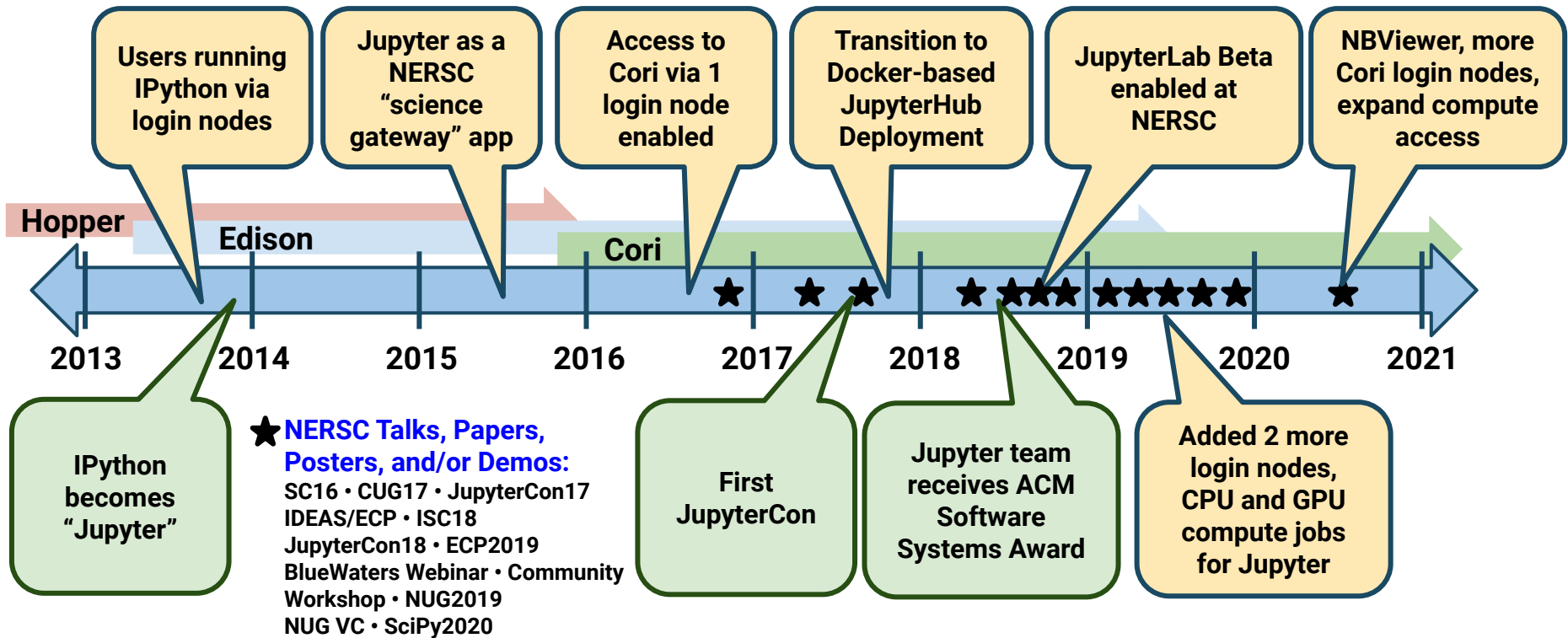
## Generational shift in data science:

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

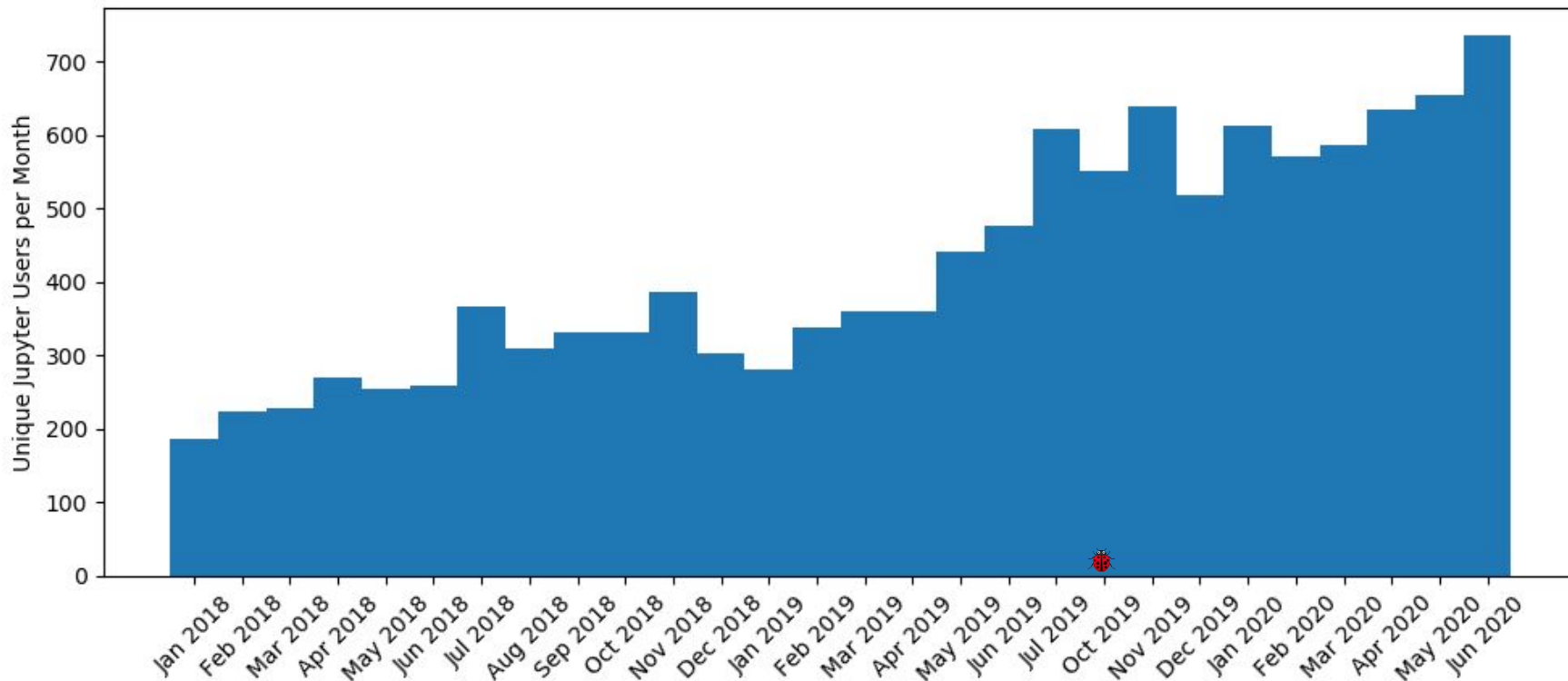
## Reproducibility and science outreach:

Open source code and open science  
Jupyter notebooks alongside publications

# Jupyter at NERSC Timeline



# Number of Jupyter Users per Month



🐛: Bug in monitoring, data missing Aug, Sep 2019.

# OK, How Do I Use Jupyter at NERSC?

- Jupyter at NERSC is provided through a JupyterHub deployment we manage:
- Authenticates you (username, password, and OTP)
- 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>

Sign in

Username:  
juptest

Password:  
.....

OTP:  
123456

Sign In

**Authenticate**

	Shared CPU Node	Shared GPU Node
Cori	<input type="button" value="start"/>	<input type="button" value="start"/>
Spin	<input type="button" value="start"/>	
Resources	Use a node shared with other users' notebooks but outside the batch queues.	
Use Cases	Visualization and analytics that are not memory intensive and can run on just a few cores.	

**Choose**

```
[1]: import os
     | workdir = os.path.join(os.environ["SCRATCH"], "dask")
     | mkdir -p $workdir
     | cd $workdir
     | rm -rf *
     | /global/u1/r/rthomas
     | /global/gscratch1/sd/rthomas/dask
[2]: %load_ext slurm_magic
[3]: %sbatch
     | #SBATCH --constraint=haswell
     | #SBATCH --nodes=16
     | #SBATCH --qos=regular
     | #SBATCH --sdn
     | #SBATCH --time=30
     | echo $SLURM_IP_ADDR
     | export PATH=/global/common/gerty/software/python/3.6--anaconda-
     | python -u $(which dask=scheduler) --scheduler=file_scheduler.i
```

**Go!**

# How Do I Choose a Notebook Server to Spawn?



## Cori Shared CPU Node:

- Notebook on cori{13,14,19}
- Can see /cfs, \$HOME, etc
- Can see Cori \$SCRATCH
- Same Python env as ssh login
- Can submit jobs via `%batch`

## Spin Shared CPU Node:

- External to Cori, in Spin
- Can't see \$SCRATCH
- Can't run jobs
- But *can* see /cfs, \$HOME

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

## Cori Shared GPU Node:

- Notebook on cgpu{01-18}
- Like Cori Shared CPU
- Runs in a 4h job
- Enabled *if* you have GPU QOS

	Shared CPU Node	Shared GPU Node
Cori	<input type="button" value="start"/>	<input type="button" value="start"/>
Spin	<input type="button" value="start"/>	
Resources	Use a node shared with other users' notebooks but outside the batch queues.	
Use Cases	Visualization and analytics that are not memory intensive and can run on just a few cores.	

# Hub Services: Announcement & NBViewer

## NERSC uses JupyterHub's Services feature

A process that interacts with the Hub's REST API

May perform a specific action or task:

- Shutting down idle notebook servers (16 hours)

- Posting announcements on the hub

- Rendering or sharing notebooks

## Announcement

- Notices about upcoming maintenances

- Communication about known issues

(Not a replacement for NERSC MOTD)

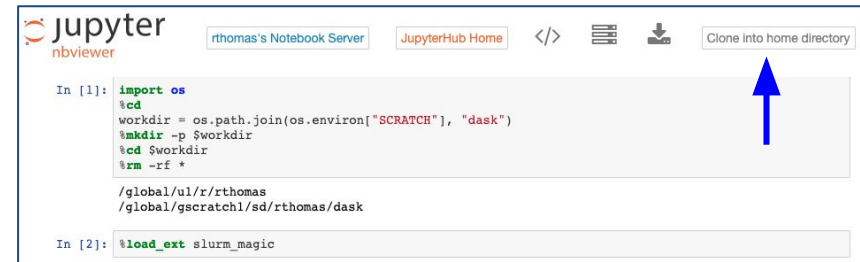
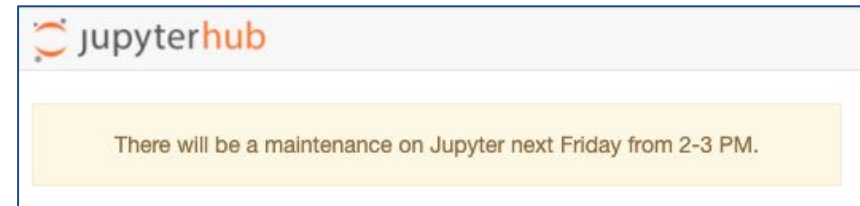
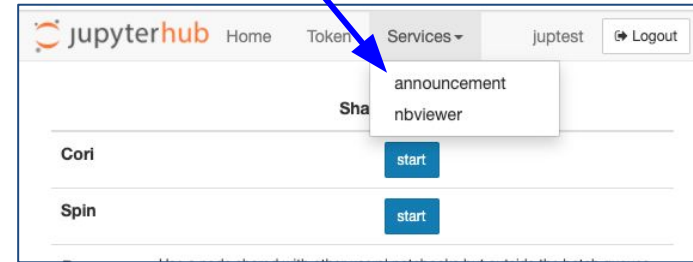
## NBViewer (Coming Soon)

- Render a notebook as static HTML

- Copy a notebook to your server and start it up

- Can copy the kernel used with the notebook

(Service Links, New Feature Coming Soon)





# JupyterLab Interface

The screenshot displays the JupyterLab interface. On the left, there is a sidebar with 'FAVORITES' (HOME, SCRATCH), a 'FILE BROWSER' showing a directory structure, and a list of files. The main area is a code editor for 'Untitled1.ipynb' in Python 3. The code includes shell commands for directory creation and removal, followed by a Slurm batch script for running a Python program. The output shows the job submission details and a table of job information.

```
[1]: import os
      %cd
      workdir = os.path.join(os.environ["SCRATCH"], "dask")
      %mkdir -p $workdir
      %cd $workdir
      %rm -rf *

/global/u1/r/rthomas
/global/gscratch1/sd/rthomas/dask

[2]: %load_ext slurm_magic

[3]: %%sbatch
      #!/bin/bash
      #SBATCH --constraint=haswell
      #SBATCH --nodes=16
      #SBATCH --qos=regular
      #SBATCH --sdn
      #SBATCH --time=30

      echo $SDN_IP_ADDR
      export PATH=/global/common/gerty/software/python/3.6-anaconda-5.2/bin:$PATH

      python -u $(which dask-scheduler) --scheduler-file scheduler.json --local-directory local-scheduler-dir &
      sleep 5

      srun -u -n 512 python -u $(which dask-worker) --scheduler-file scheduler.json --local-directory local-worker-dir --nthreads 1 --reconnect

[3]: 'Submitted batch job 1001599\n'

[4]: job_id = _.split()[-1]
      job_id

[4]: '1001599'

[5]: %squeue -u rthomas

[5]:
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
0	1001599	regular sbatch	rthomas	R	0:04	16	nid00[384-391,440-447]

```
[6]: from dask.distributed import Client
```

# JupyterLab Interface: NERSC Goodies

The screenshot shows the JupyterLab interface with a sidebar on the left and a code editor on the right. The sidebar has a 'FAVORITES' section at the top, which is highlighted with a red box. Below it are sections for '\$HOME', '\$SCRATCH', and 'FILE BROWSER'. The 'FILE BROWSER' section shows a list of files and directories, with a star icon next to the current directory. A blue arrow points from the 'FAVORITES' section to a text box on the right. The text box contains the following text:

**Favorites (NERSC/jupyterlab-favorites)**  
**Bookmark your favorite places on the file system**  
**Prepopulate with \$HOME and \$SCRATCH**  
**Add the current directory by clicking the ★ icon**

The code editor shows a Python 3 kernel with the following code:

```
[1]: import os
      %cd
      workdir = os.path.join(os.environ["SCRATCH"], "dask")
      mkdir -p $workdir
      %cd $workdir
      %rm -rf *
```

```
/global/u1/r/rthomas
/global/gscratch1/sd/rthomas/dask
```

```
[2]: %load_ext slurm_magic
```

```
[3]: %%sbatch
      #!/bin/bash
      #SBATCH --constraint=haswell
      #SBATCH --nodes=16
      #SBATCH --qos=regular
      #SBATCH --sdn
      #SBATCH --time=30

      echo $SDN_IP_ADDR
      export PATH=/global/common/gerty/software/python/3.6-anaconda-5.2/bin:$PATH

      python -u $(which dask-scheduler) --scheduler-file scheduler.json --local-directory local-scheduler-dir &
      sleep 5

      srun -u -n 512 python -u $(which dask-worker) --scheduler-file scheduler.json --local-directory local-worker-dir --nthreads 1 --reconnect
```

```
[3]: 'Submitted batch job 1001599\n'
```

```
[4]: job_id = _.split()[-1]
      job_id
```

```
[4]: '1001599'
```

```
[5]: %squeue -u rthomas
```

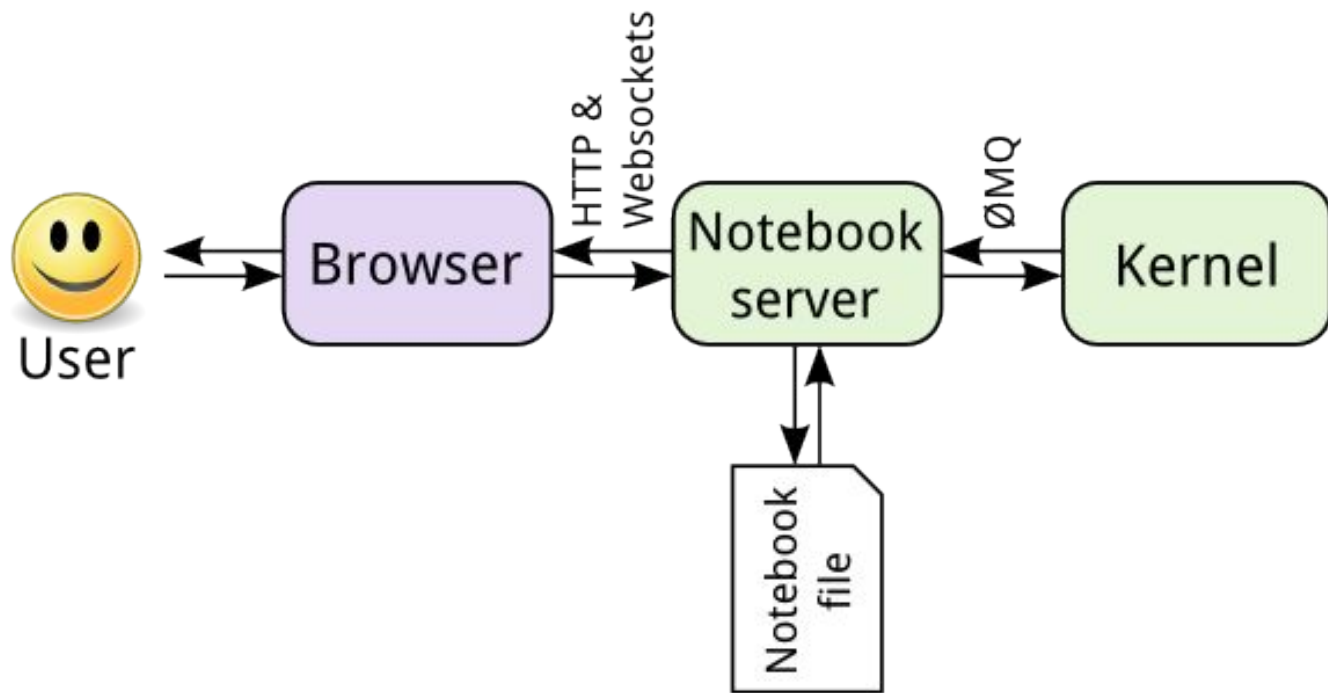
```
[5]:
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
0	1001599	regular sbatch	rthomas	R	0:04	16	nid00[384-391,440-447]

```
[6]: from dask.distributed import Client
```



# 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=3.7
$ 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 modules.**

Meanwhile, in jupyter-helper.sh:

```
#!/bin/bash  
export SOMETHING=123  
module load texlive  
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

(myenv) rthomas@cori01:~> cat ~/.jupyter.log ← **YOUR FRIEND!!!**

```
[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
```

# Near Future Jupyter Support

## Working on:

Expanding resources to support Jupyter

Preparing the strategy for Jupyter with Perlmutter

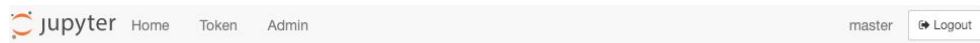
**Streamlining Dask, IPyParallel cluster launch and management**

Custom JupyterLab notebook servers

Expanding JupyterLab interface to:

Track and monitor batch jobs

New viewers



## NERSC JupyterHub Console

	Shared CPU Node	Exclusive CPU Node	Exclusive GPU Node	Configurable
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>	<a href="#">start</a>
Spin	<a href="#">start</a>			
Resources	On a node shared with other users' notebooks but outside the batch queues.	On a node by itself within an interactive job allocation using your default repo.	One or more nodes within an interactive job allocation.	
Use Cases	Visualization and analytics that are not memory intensive and can run on just a few cores.	Visualization, analytics, machine learning that is compute or memory intensive but can be done on a single node.	Large-scale data analytics, visualization, and machine learning; reservation or non-default repository.	

# 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!

