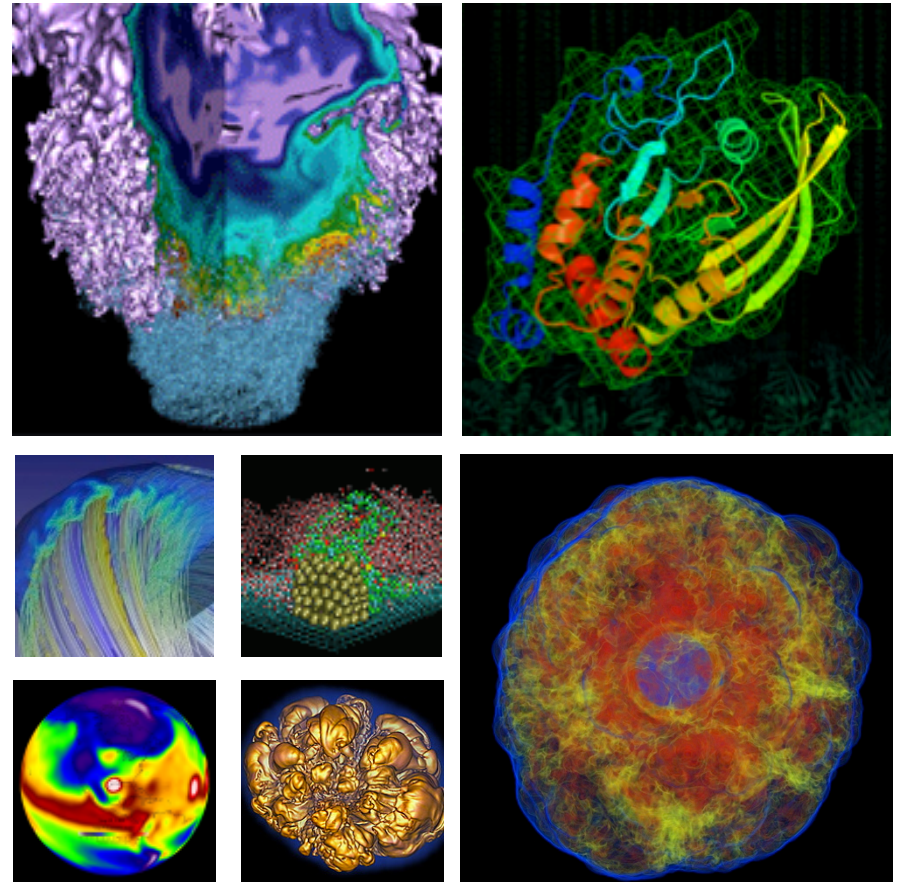


Enabling Interactive Notebooks on Supercomputers with Jupyterhub



**Shreyas Cholia, Rollin Thomas
and Shane Canon**

**NERSC, Lawrence Berkeley National Lab
Gateway 2016 San Diego CA**

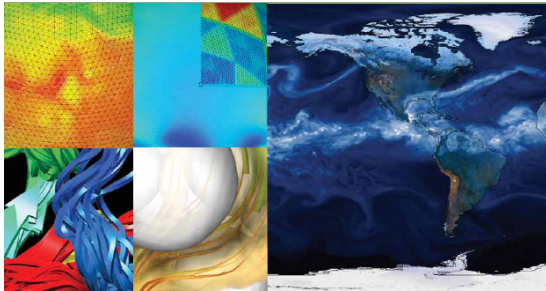
NERSC is the Production HPC & Data Facility for DOE Office of Science Research



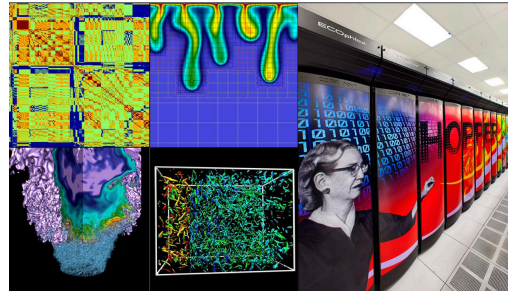
U.S. DEPARTMENT OF
ENERGY

Office of
Science

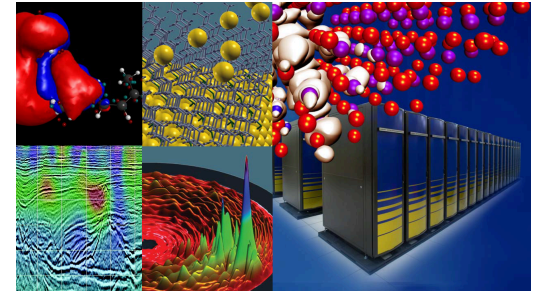
Largest funder of physical
science research in U.S.



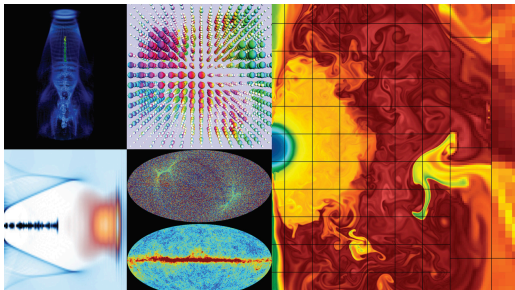
Bio Energy, Environment



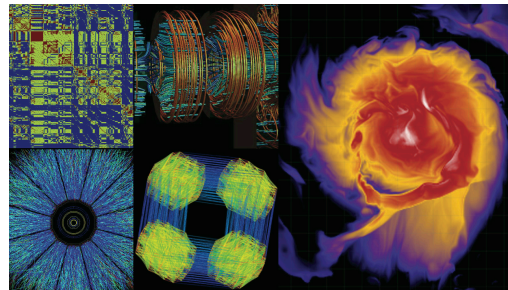
Computing



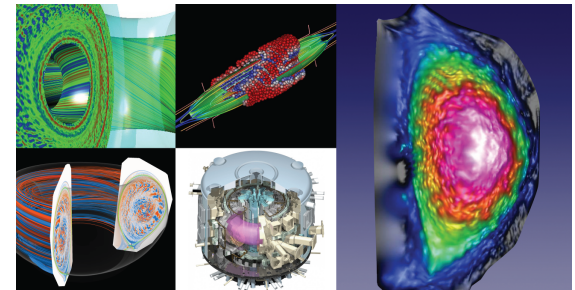
Materials, Chemistry,
Geophysics



Particle Physics,
Astrophysics



Nuclear Physics



Fusion Energy,
Plasma Physics

The Cori System

- Cori will transition HPC and data-centric workloads to energy efficient architectures
- Includes “data partition” with Haswell nodes, data friendly queues, pool of large memory interactive nodes
- Also has NVRAM layer, Software Defined Network



Image source: Wikipedia

System named after Gerty Cori, Biochemist and first American woman to receive the Nobel prize in science.



Interactive Notebook Computing



- **Combine an annotated notebook with live code execution and results**
 - “literate computing” environment, where narrative and computation go hand in hand
- **Jupyter emerged from iPython and is becoming a very popular web based framework for interactive notebook computing**
- **Not just python - has over 60 language kernels**
 - Julia, R, Bash, Root, Ruby ...
- **c/f Fernando Perez keynote**

Notebook examples

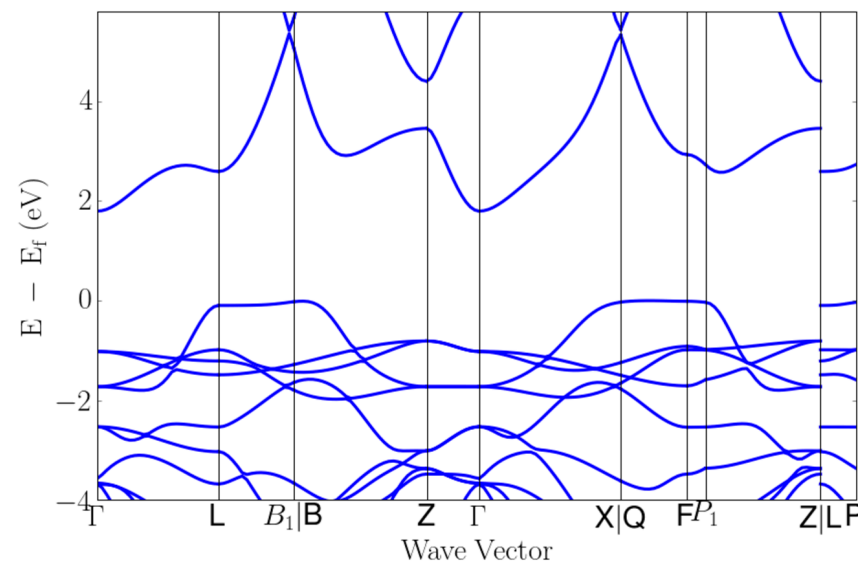
https://github.com/materialsproject/mapidoc/blob/master/example_notebooks/pymatgen_examples/Analyze%20and%20plot%20band%20structures.ipynb

```
In [9]: #is the material a metal (i.e., the fermi level cross a band)
print bs.is_metal()
#print information on the band gap
print bs.get_band_gap()
#print the energy of the 20th band and 10th kpoint
print bs.bands[Spin.up][20][10]
#print energy of direct band gap
print bs.get_direct_band_gap()
#print information on the vbm
print bs.get_vbm()

False
{'energy': 1.7978000000000005, 'transition': u'(0.591,0.409,0.000)-\\Gamma', 'direct': False}
18.0201
2.6904
{'kpoint': <pymatgen.electronic_structure.bandstructure.Kpoint object at 0x5192610>, u'kpoint_index': [123], u'energy': 6.1023, u'band_index': {1: [15]}, u'projections': {}}
```

Here, we plot the bs object. By default for an insulator we have an energy limit of $\text{cbm}+4\text{eV}$ and $\text{vbm}-4\text{eV}$

```
In [7]: %matplotlib inline
from pymatgen.electronic_structure.plotter import BSPlotter
plotter=BSPlotter(bs)
plotter.get_plot().show()
```



**HPC workflows at scale for data-intensive science =>
asynchronous, batch execution.**

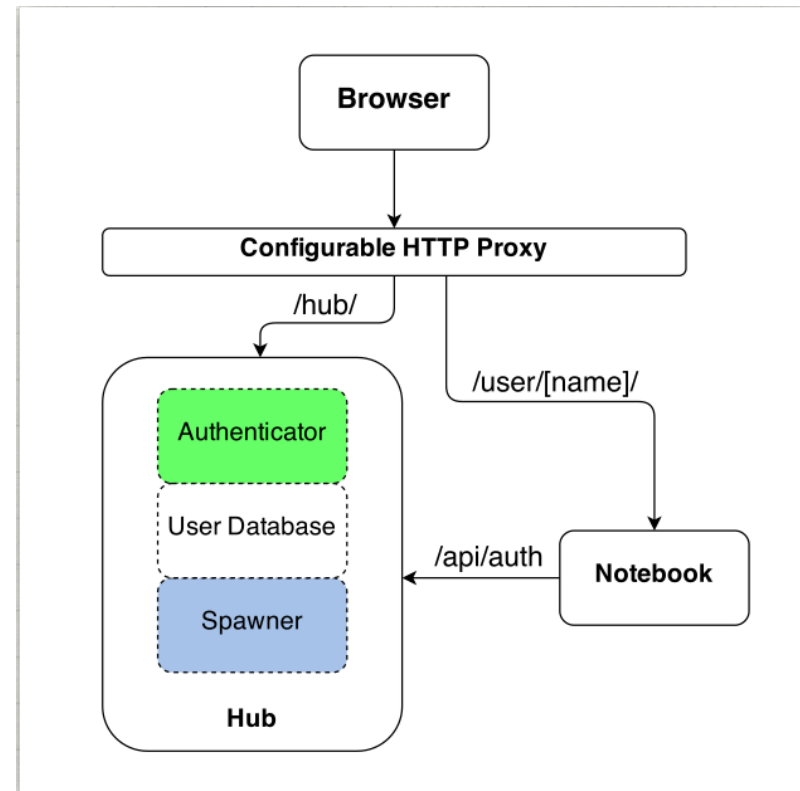
But ...

**Scientific insight => interactive, iterative exploration
and analysis**

How can Jupyter bridge this gap?

- **Python is the most popular language/tool at NERSC**
- **Users want to be able to use Jupyter notebooks to drive their jobs, run post-job analyses etc.**
- **Access to NERSC resources through these interfaces**
 - Filesystems
 - Batch Queue
 - Network, DBs
- **Support for custom libraries and kernels**
- **Centralized service to deploy notebooks in a standard authenticated manner**

- Service to deploy notebooks in a multi-user environment
- Manages user authentication, notebook deployment and web proxies
- Node.js + Python Tornado



Jupyterhub Example



Sign in

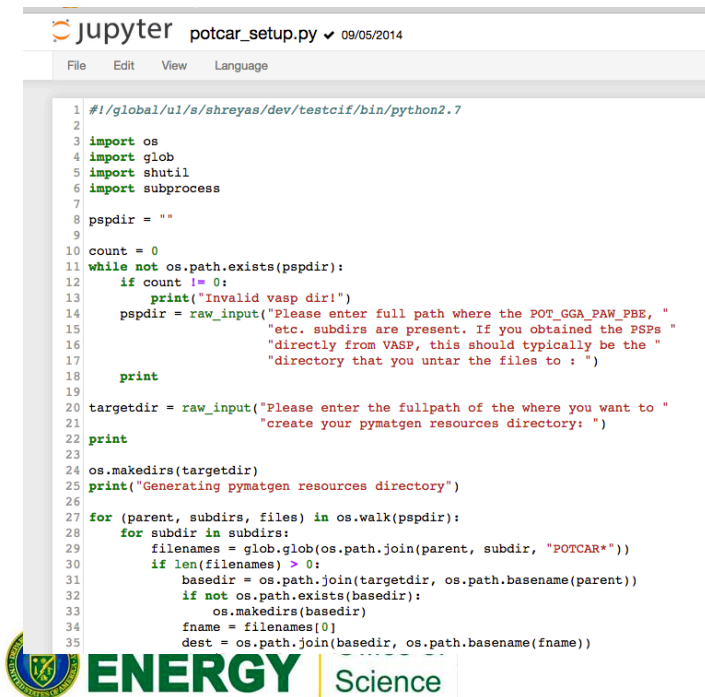
Username:

Password:

Sign In

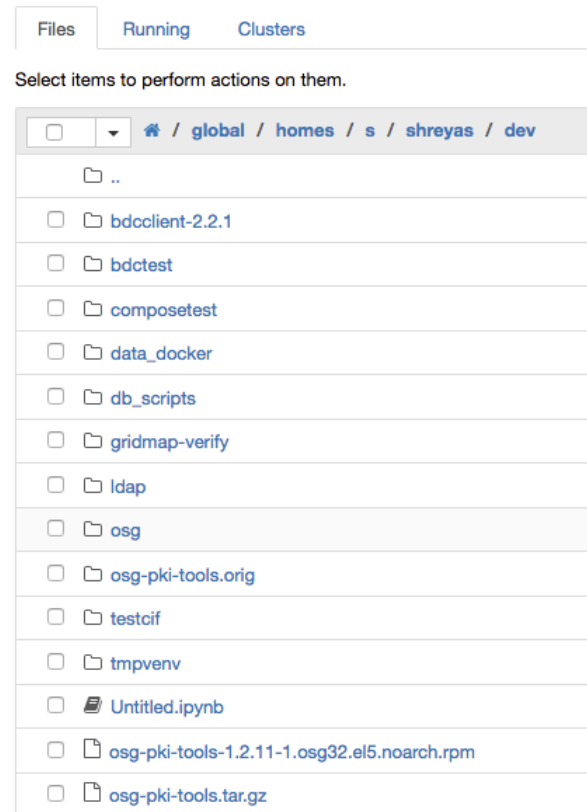
But Wait .. There's More

- Jupyter also includes
 - File Browser
 - Text Editor (with full syntax highlighting)
 - Terminal



```
jupyter potcar_setup.py ✓ 09/05/2014
File Edit View Language

1 #!/global/ul/s/shreyas/dev/testcif/bin/python2.7
2
3 import os
4 import glob
5 import shutil
6 import subprocess
7
8 pspdir = ""
9
10 count = 0
11 while not os.path.exists(pspdir):
12     if count != 0:
13         print("Invalid vasp dir!")
14     pspdir = raw_input("Please enter full path where the POT_GGA_PAW_PBE, "
15                       "etc. subdirs are present. If you obtained the PSPs "
16                       "directly from VASP, this should typically be the "
17                       "directory that you untar the files to : ")
18     print
19
20 targetdir = raw_input("Please enter the fullpath of the where you want to "
21                      "create your pymatgen resources directory: ")
22 print
23
24 os.makedirs(targetdir)
25 print("Generating pymatgen resources directory")
26
27 for (parent, subdirs, files) in os.walk(pspdir):
28     for subdir in subdirs:
29         filenames = glob.glob(os.path.join(parent, subdir, "POTCAR*"))
30         if len(filenames) > 0:
31             basedir = os.path.join(targetdir, os.path.basename(parent))
32             if not os.path.exists(basedir):
33                 os.makedirs(basedir)
34             fname = filenames[0]
35             dest = os.path.join(basedir, os.path.basename(fname))
```



Files Running Clusters

Select items to perform actions on them.

global / homes / s / shreyas / dev

- ..
- bdoclient-2.2.1
- bdctest
- composetest
- data_docker
- db_scripts
- gridmap-verify
- ldap
- osg
- osg-pki-tools.orig
- testcif
- tmpvenv
- Untitled.ipynb
- osg-pki-tools-1.2.11-1.osg32.el5.noarch.rpm
- osg-pki-tools.tar.gz

jupyter

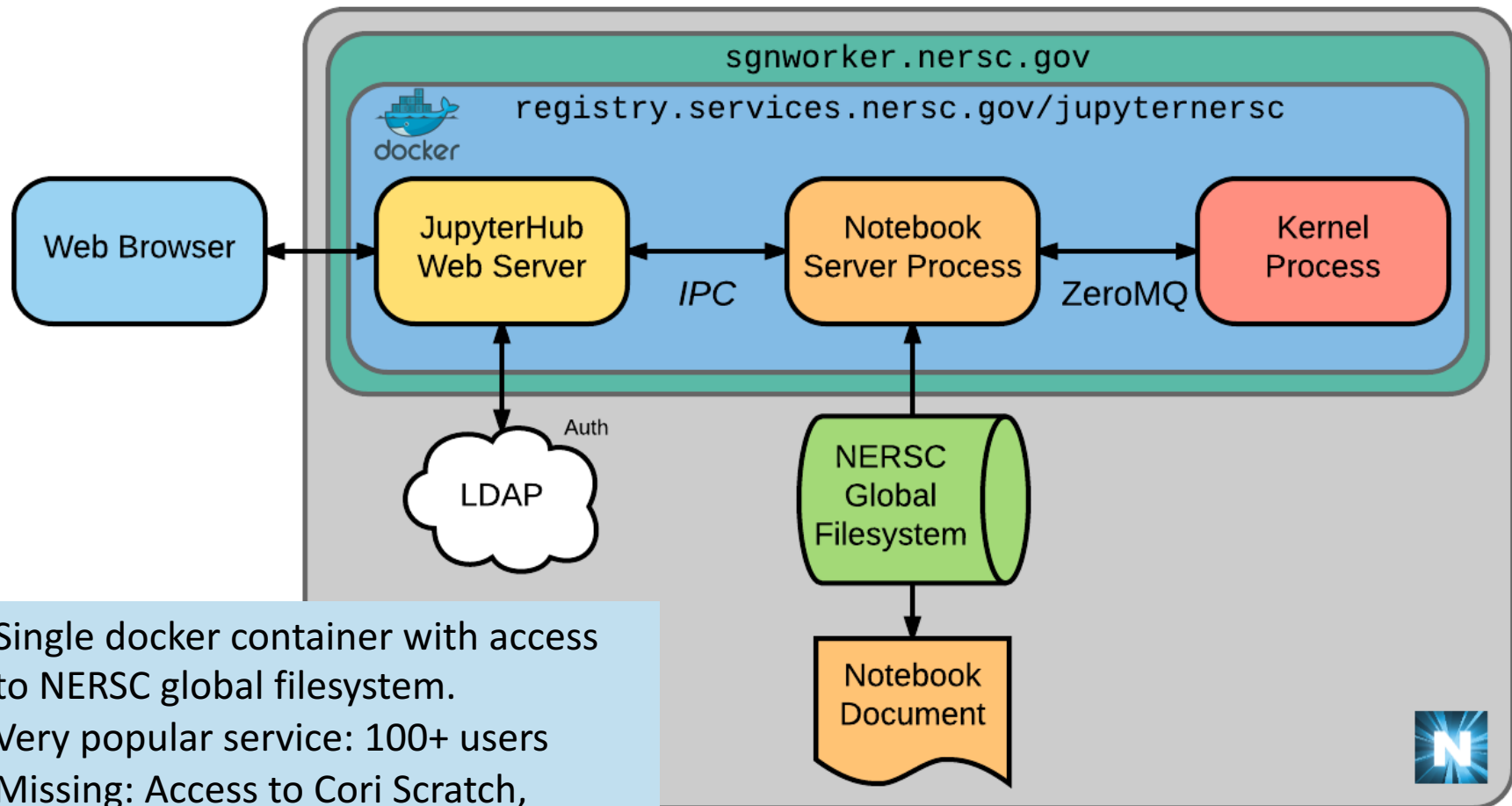
```
bash-4.1$ ls -l tmp
total 16416
drwxrwxr-x 2 shreyas shreyas 512 Jan 29 2016 build
drwxr-xr-x 2 shreyas shreyas 512 Oct 7 19:13 conf
drwxr-xr-x 3 shreyas shreyas 4096 Oct 7 19:14 conf.d
drwxrwxr-x 4 shreyas shreyas 4096 May 9 2013 dist.eugridpma.info
drwxrwxr-x 5 shreyas shreyas 512 Oct 21 16:51 foo
drwxrwxr-x 5 shreyas shreyas 512 Oct 21 23:01 foobar
-rw----- 1 shreyas shreyas 16384 Apr 30 2015 globus-user-env.sh.swp
drwxrwxr-x 5 shreyas shreyas 512 Oct 14 22:30 msq
drwxrwxr-x 5 shreyas shreyas 512 Jun 3 22:25 pmg
drwxrwxr-x 6 shreyas shreyas 512 Jan 29 2016 pydap-3.1.1
drwxrwxr-x 5 shreyas shreyas 512 Jan 29 2016 pydap-3.2
drwxr-x--- 18 shreyas shreyas 4096 May 26 2015 Python-2.7.10
-rw-r----- 1 shreyas shreyas 16768806 May 23 2015 Python-2.7.10.tgz
drwxr-x--- 5 shreyas shreyas 512 May 26 2015 req
bash-4.1$
```


Jupyterhub Prototype



- Jupyterhub on a single node
- Service runs as root and spins up notebooks locally
- Jupyter has access to global filesystems and lives within the NERSC network

Jupyterhub in a Single Container



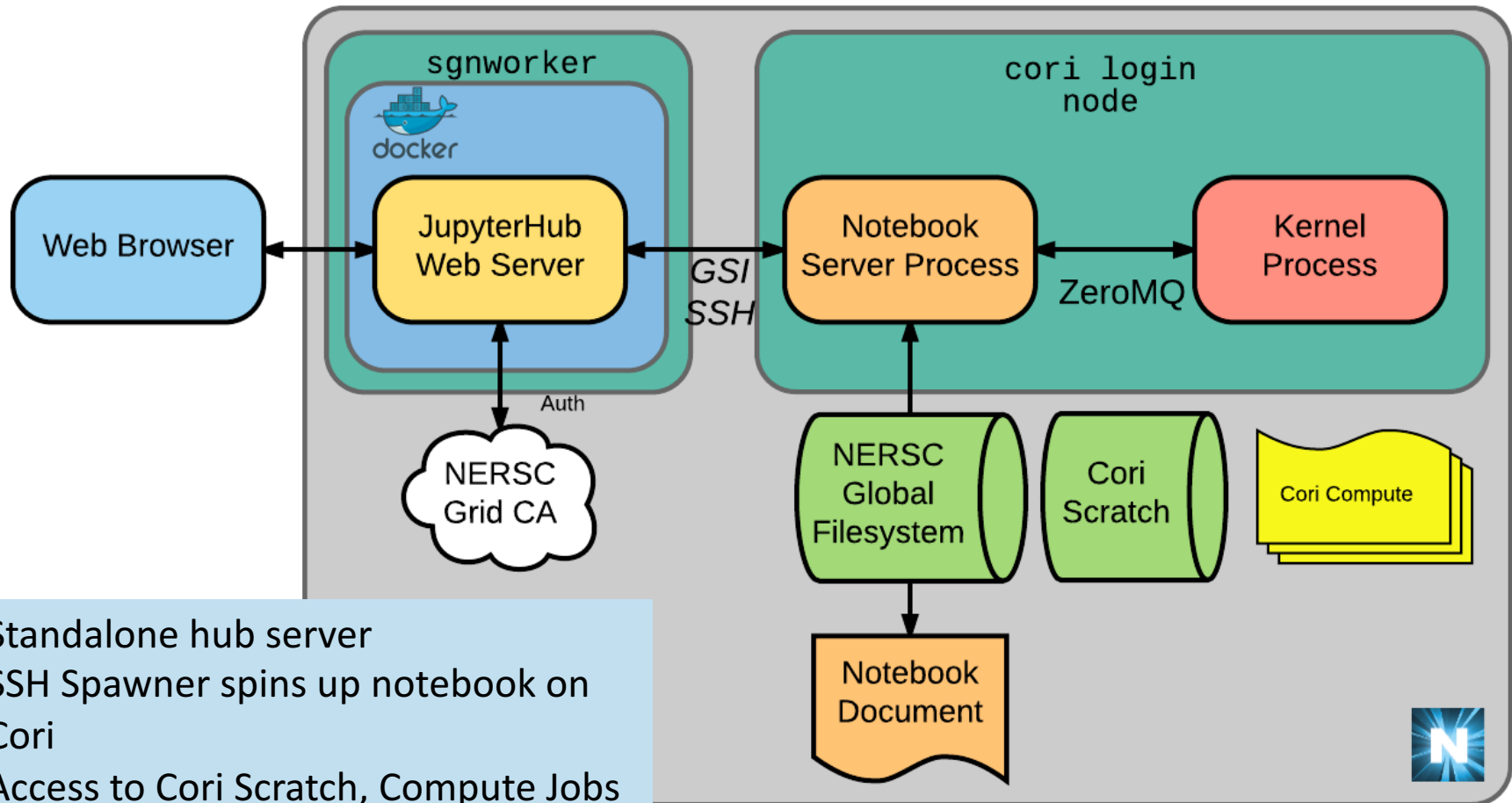
- Single docker container with access to NERSC global filesystem.
- Very popular service: 100+ users
- Missing: Access to Cori Scratch, Compute Jobs

Jupyterhub Cori Service



- **Users would like access to big compute jobs!!!**
- **Access to Cori Scratch Filesystem**
- **Existing Software on Cori (including kernels)**
- **Take advantage of large memory interactive nodes**
- **Security considerations**
 - Don't want to run as root
 - Don't want to run public facing web service inside a Cori node
- **New service uses a distributed model where Jupyterhub runs on a science gateway node but spins up notebooks remotely**

New Jupyterhub Implementation For Cori



- Standalone hub server
- SSH Spawner spins up notebook on Cori
- Access to Cori Scratch, Compute Jobs via SLURM magic

What we built

Jupyter architecture is completely pluggable

So we developed

- **Custom Authenticator that allows user to login and get a GSI X509 certificate**
- **Custom Spawner that creates a remote notebook via SSH (using public key or GSI auth)**
- **Ipython Magic Commands to launch and query jobs via SLURM**

GSI Authenticator



- User logs in with username and password. Authenticator uses myproxy to login to NERSC CA server with username/password and retrieves credentials (X509 certificate)
- Jupyterhub runs as a standalone service and doesn't need root access. In fact, no root access needed across this architecture.
- <https://github.com/NERSC/gsiauthenticator>

Spawner: SSH

- **We wrote an SSH Spawner that will will SSH into the Cori node with users credential**
 - Supports GSISSH (use with certificates from GSI authenticator)
 - Supports SSH key based auth
- **SSH Spawner starts up notebook server process and goes away; Notebook server communicates directly with hub**
 - No tunnels or persistent connections needed
- **Keep track of the PID for poll and shutdown functions (also via SSH)**
- **Inspired by Andrea Zonca's RemoteSpawner (SDSC)**
- **<https://github.com/NERSC/SSHSpawner>**

Ipython Magic for SLURM

- Created some simple magics to facilitate interaction with SLURM batch queuing system
- Implemented %%sbatch and %squeue
- Pandas mode to slurp the results
- <https://github.com/NERSC/slurm-magic>

```
In [2]: %%sbatch -p debug -t 10 -N 1
#!/bin/bash
srun -n 32 hostname
...:
Out[2]: u'Submitted batch job 2754280\n'

In [3]: !cat slurm-2754280.out
nid00044
nid00044
...
```

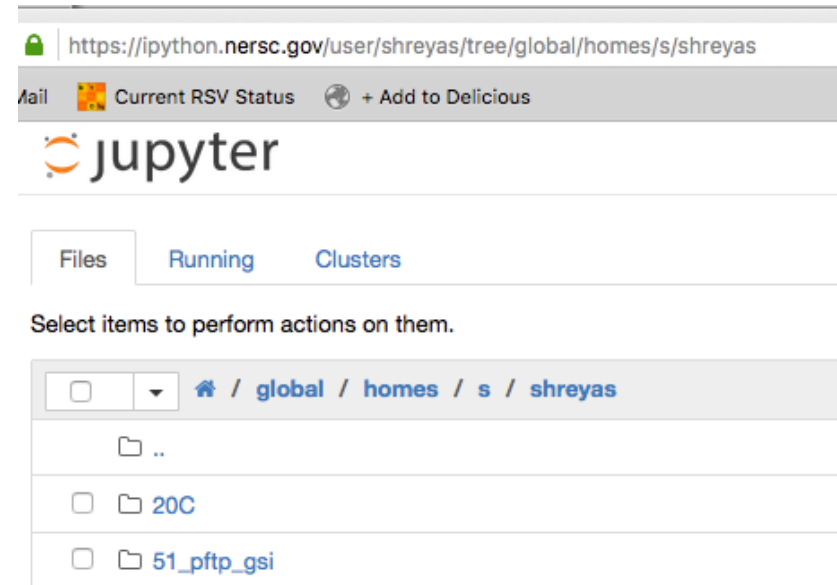
```
In [3]: %squeue -u rthomas
Out[3]:
```

	JOBID	USER	ACCOUNT	NAME	PARTITION	QOS	NODES	TIME_LIMIT	TIME	ST	\
0	2764292	rthomas	mpccc	sh	debug	debug	1	10:00	0:11	R	

	PRIORITY	SUBMIT_TIME	START_TIME
0	69060	2016-07-21T21:12:34	2016-07-21T21:13:21

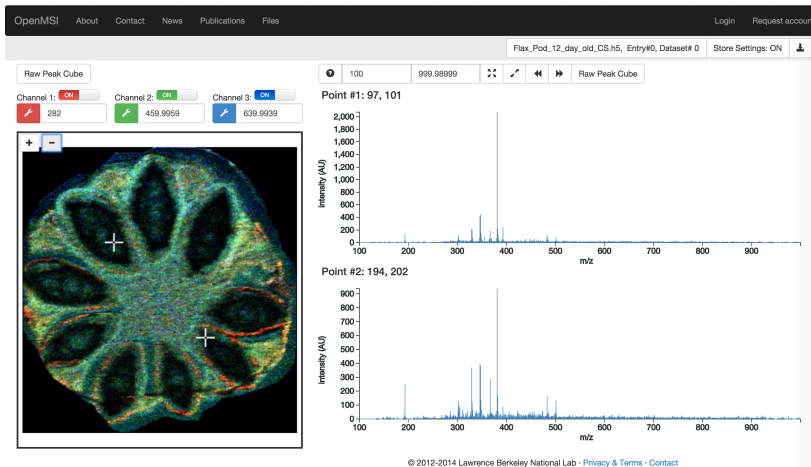
Other Features

- Allow users to browse other filesystems while defaulting to homedir
- Users can drop in custom kernels in their .ipython dir



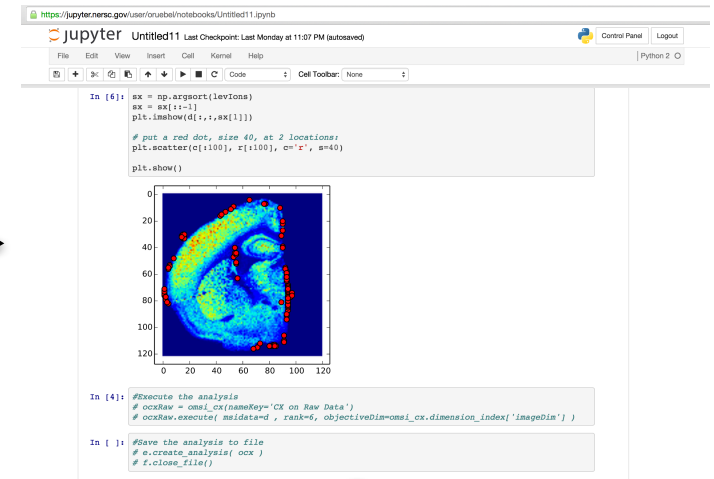
Use Case: OpenMSI + Jupyterhub

openmsi.nerisc.gov



- Manage data, analyses, users and resources
- Customize templates

jupyter.nerisc.gov

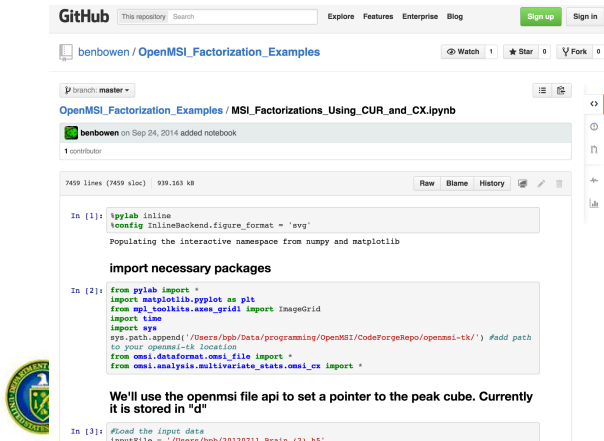


Launch, run,
and integrate
notebooks

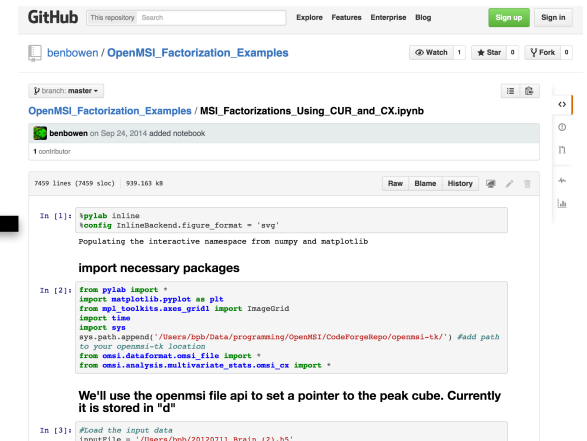
Share analysis
results

Template analysis notebook repository

User notebook repository



Contribute new
analysis templates



Future Directions



- **Driving a Spark/Dask job from a notebook on Cori**
- **Spawning notebook kernels on Cori compute nodes**
- **Tighter interaction with MPI jobs**
- **Interactive HPC workflows (LDRD)**

What Our Users Say



**“I’ll never have to leave a notebook again
that’s like the ultimate dream”**

We're Hiring



- Postdoc to work on Jupyter for human-in-the-loop interactive HPC workflows
- Various CSE positions
- <http://jobs.lbl.gov>

Contact: scholia@lbl.gov

More info

Contact us:

- scholia@lbl.gov
- scanon@lbl.gov
- rcthomas@lbl.gov

All our code is on github

<https://github.com/NERSC/gsiauthenticator>

<https://github.com/NERSC/SSHSpawner>

<https://github.com/NERSC/slurm-magic>