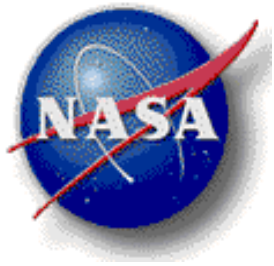




JupyterHub

at the
NCCS



Matt Stroud
Savannah Strong Finch



Agenda



Basic Information

- What is JupyterHub?
- Why is Jupyter useful?
- How have we implemented it in HPC environments?

Access & Usage

- Requesting Access
- Navigating the Interfaces
- Setting Up Custom Environments/Kernels

Demo

- Launching a Session
- NCCS Innovation Lab Demo Notebook
- eViz/iViz

Future & Questions

- Future Plans for JupyterHub Environments
- Additional Resources
- Questions & Open Discussion

What is Jupyter?

JupyterHub



The process responsible for launching and managing access to user's JupyterLab sessions.

“JupyterHub brings the power of notebooks to groups of users. It gives users access to computational environments and resources without burdening the users with installation and maintenance tasks. Users - including students, researchers, and data scientists - can get their work done in their own workspaces on shared resources which can be managed efficiently by system administrators.”

JupyterLab



The IDE served to users with support for the notebook format, over 40 programming languages, interactive data visualization tools, and a friendlier interface to our HPC systems.

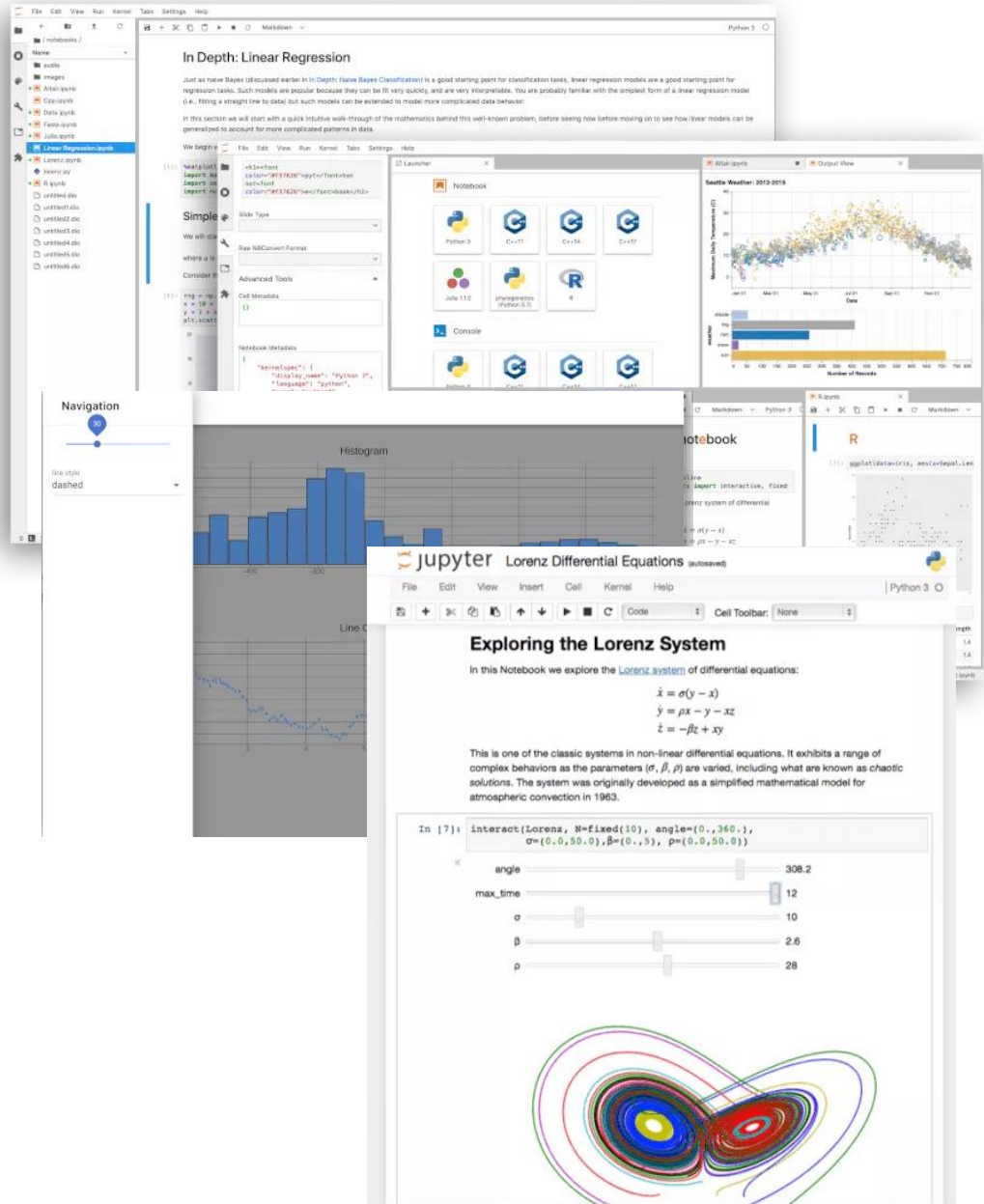
“JupyterLab is a highly extensible, feature-rich notebook authoring application and editing environment, and is a part of Project Jupyter, a large umbrella project centered around the goal of providing tools for interactive computing with computational notebooks.”



Why Jupyter?

The Jupyter Notebook and associated tools are especially useful for collaborative projects, scientific computing, rich and interactive visualizations, contextualizing large data sets, and analyzing output.

At the NCCS we have been working to improve accessibility to our HPC platforms, as well as to accelerate scientific workflows. We believe that interactive tools like JupyterHub are a step in this broader direction.



Source: jupyter.org

JupyterHub on NCCS Systems

We currently have JupyterHubs available on both Discover and ADAPT.

ADAPT has multiple hub instances implemented to support specific user groups. This includes:

- Prism GPU Cluster
- ICESat-2
- Roman Space Telescope
- ABoVE

Our systems have a couple of notable complications that are useful to understand.

Each of these hub instances is tied directly into a back-end Slurm cluster. This allows resources to be allocated directly to user sessions while also serving other jobs and users on a given pool of compute. This can pose unique architectural challenges for our various systems and can sometimes result in odd behaviors when using JupyterHub.

We also work with agency Launchpad integration and network access requirements to help ensure site and account security. This creates some additional steps for users trying to gain access to the hubs.



Requirements to Access

- 1. NAMS Request:** A request can be made for either 'NCCS Discover JupyterHub' to access the Discover hub, or 'NCCS Web services' at <https://idmax.nasa.gov> for any of the ADAPT hubs. (Note: You will only be able to submit NAMS requests from the NASA network/VPN.)
- 2. System Request:** Once you have submitted your NAMS request, please send a ticket to support@nccs.nasa.gov stating which specific JupyterHub you would like access to, and we will ensure that you have the necessary LDAP permissions to access the systems.
- 3. VPN Access:** Some of our hubs will only be accessible from the NASA network/VPN. This only applies to the RomanST, ABoVE, and the generic ADAPT hubs.

If your group would like a JupyterHub for your team's ADAPT VMs that aren't listed, please send a request to support@nccs.nasa.gov for further inquiry.



Navigate to desired JupyterHub

➤ You can access JupyterHubs by going to

<https://www.nccs.nasa.gov/nccs-users/instructional/adapt-instructional/jupyter>

And clicking on one of the desired connections:

A screenshot of the NASA Center for Climate Simulation (NCCS) website. The browser address bar shows the URL: https://www.nccs.nasa.gov/nccs-users/instructional/adapt-instructional/jupyter. The page header includes the NCCS logo, the text "NASA CENTER FOR CLIMATE SIMULATION" and "HIGH PERFORMANCE COMPUTING FOR SCIENCE", and the NASA logo. A navigation menu contains links for HOME, USERS, SERVICES, SYSTEMS, SCIENCE & TECHNOLOGY, NEWS & EVENTS, and ABOUT US, along with a search bar. Below the menu, there are additional links: USERS HOME, PORTALS, ACCOUNT HELP, LOGIN & PASSWORDS HELP, USER EVENTS, and INSTRUCTIONALS. The main content area is titled "Connection" and contains the text: "We currently have the following JupyterHubs deployed:" followed by a bulleted list of JupyterHub names: Discover, ADAPT, ABoVE Nodes, Prism GPU Cluster, ICESat-2, and RomanST.

➤ Or by going directly to the URL. Example: <https://jh-discover.nccs.nasa.gov>



Logging in

After authentication..



**You will be presented with a Sign In page.
(Enter your LDAP username and password)**

Sign in

Username:

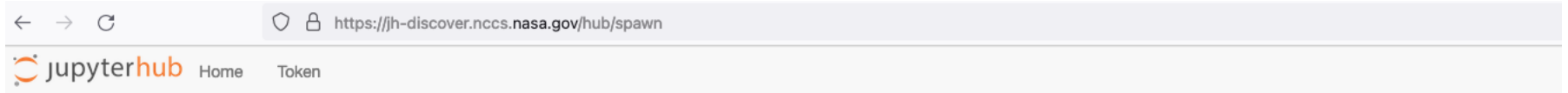
Password:

Sign in



Start Server

Select your desired Server Option



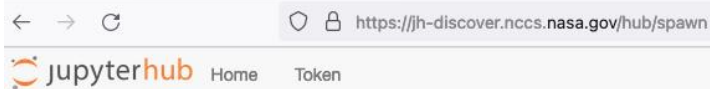
Server Options

Select a job profile:

Debug Session (1 CPU Core, 4GB Memory, 1 Hour)



Start



Server Options

Select a job profile:

- ✓ Debug Session (1 CPU Core, 4GB Memory, 1 Hour)
- Short Session (2 CPU Cores, 8GB Memory, 2 Hours)
- Long Session (4 CPU Cores, 16GB Memory, 12 Hours)
- Full Node Session (1 Node, 46 CPU Cores, 185GB Memory, 4 Hours)
- GPU Session (1 GPU Core, 12 CPU Cores, 120GB Memory, 8 Hours)
- GPU Session (2 GPU Core, 24 CPU Cores, 240GB Memory, 8 Hours)

Jupyter lab Console

Default directory is /home/<username>

JupyterLinks Dir will be visible if it doesn't exist user session will create it upon startup. User may add other symlinks there as well.

Multiple slurm output files contain log from the user session.

The screenshot shows the JupyterLab interface. On the left is a file browser with a search bar and a list of files and directories. The 'JupyterLinks' directory is highlighted. On the right is the 'Launcher' view, which displays various options for creating new files or notebooks, including 'Notebook', 'Console', and 'Other' (Terminal, Text File, Markdown File, Python File, Show Contextual Help). A text box on the right side of the launcher area states: 'All conda envs that NBs have access to will be listed here.'

Name	Last Modified
/	
css_dir	3 years ago
data	8 days ago
ilab_testing	8 months ago
JupyterLinks	2 minutes ago
slash_dir	3 years ago
2006-001-h09v05-MOD-RandomForest.tif	8 months ago
2006-002-h09v05-MOD-RandomForest.tif	8 months ago
2006-003-h09v05-MOD-RandomForest.tif	8 months ago
2006-004-h09v05-MOD-RandomForest.tif	8 months ago
2006-005-h09v05-MOD-RandomForest.tif	8 months ago
2006-h09v05-MOD-RandomForest-Mask.tif	8 months ago
2006-h09v05-MOD-RandomForest-ProbWater.tif	8 months ago
2006-h09v05-MOD-RandomForest-SumLand.tif	8 months ago
2006-h09v05-MOD-RandomForest-SumObs.tif	8 months ago
2006-h09v05-MOD-RandomForest-SumWater.tif	8 months ago
amfire_2.0.0.sif	9 days ago
centos.out	4 years ago
config_mk_embedPy	10 years ago
earthmlPlus3.out	3 years ago
eis-dev-1.0.3.sif	a month ago
et.tar.gz	9 months ago
floodmap.docx	8 months ago
Hello_World.py	2 years ago
ilab-kernel-pkgs.out	a year ago
IOC_init.cnfg	9 years ago
ls	a year ago
Makefile_embedPy	10 years ago
MCDWD_L3_F2_NRT.A2022188.h02v07.061.tif	10 months ago
mjstroud.sh	3 years ago
MOD.A2006.h09v05.RandomForest.AnnualBurnScar.20222651543.tif	8 months ago
mozilla.pdf	7 years ago
slurm-36501845.out	9 months ago
slurm-36502068.out	9 months ago

Jupyter lab Console

tutorialNotebooks is a system generated symlink to publicly available Notebooks.

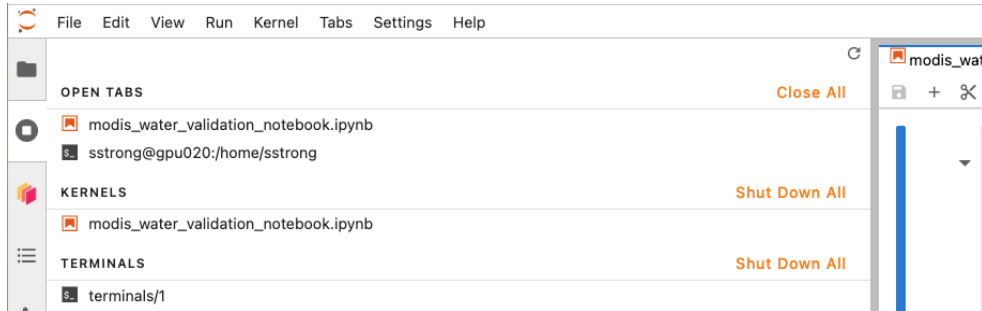
```
File Edit View Run Kernel Git Tabs Settings Help
+ + + + +
Filter files by name
/ JupyterLinks /
Name Last Modified
css 3 years ago
icesat-2 3 years ago
ilab_projects 8 months ago
nobackup 7 minutes ago
test_tut_note... 3 years ago
top 3 years ago
tutorialNotebo... 7 minutes ago
```

```
(base) [sstrong@gpu022 ~]$ pwd
/home/sstrong
(base) [sstrong@gpu022 ~]$ cd JupyterLinks/
(base) [sstrong@gpu022 JupyterLinks]$ pwd
/home/sstrong/JupyterLinks
(base) [sstrong@gpu022 JupyterLinks]$ ls
total 416
lrwxrwxrwx 1 sstrong ilab 45 May 12 09:33 tutorialNotebooks -> /explore/nobackup/projects/tutorial_notebooks
drwxr-xr-x. 4 sstrong k3000 4096 May 12 09:33 .
lrwxrwxrwx 1 sstrong ilab 32 May 12 09:33 nobackup -> /explore/nobackup/people/sstrong
drwx----- 45 sstrong 40002 40960 May 12 09:33 ..
lrwxrwxrwx. 1 sstrong ilab 32 Sep 22 2022 ilab_projects -> /explore/nobackup/projects/ilab/
drwxr-xr-x. 3 sstrong k3000 4096 Sep 15 2020 test_tut_notebooks
lrwxrwxrwx. 1 sstrong k3000 14 Feb 20 2020 icesat-2 -> /css/icesat-2/
drwxr-xr-x. 2 sstrong k3000 4096 Feb 20 2020 .ipynb_checkpoints
lrwxrwxrwx. 1 sstrong k3000 4 Feb 18 2020 css -> /css
lrwxrwxrwx. 1 sstrong k3000 1 Feb 18 2020 top -> /
(base) [sstrong@gpu022 JupyterLinks]$
```

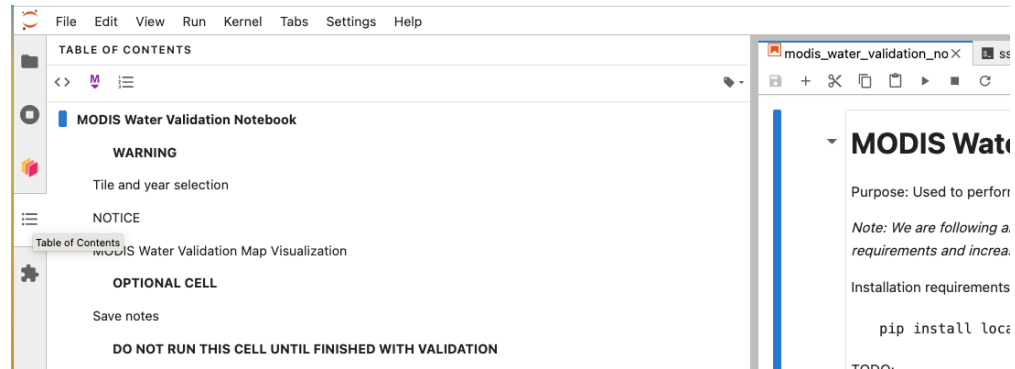
Jupyter lab Console

Explore left side menu

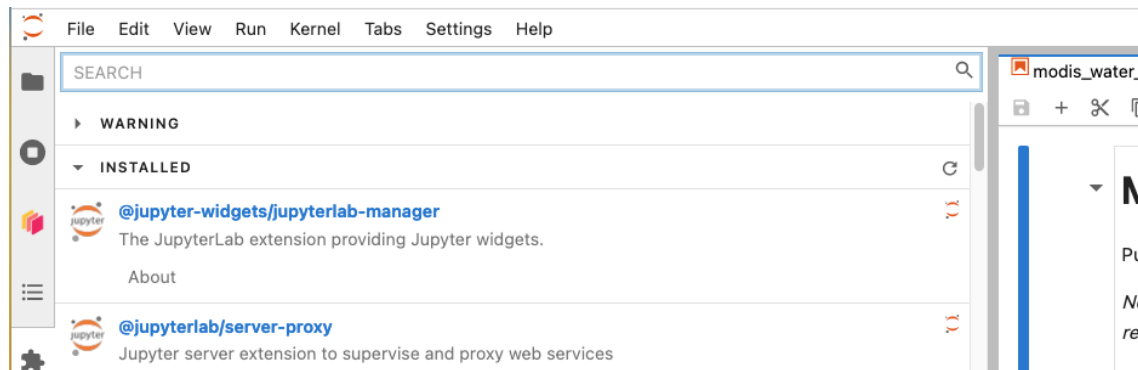
Show what kernels are running →



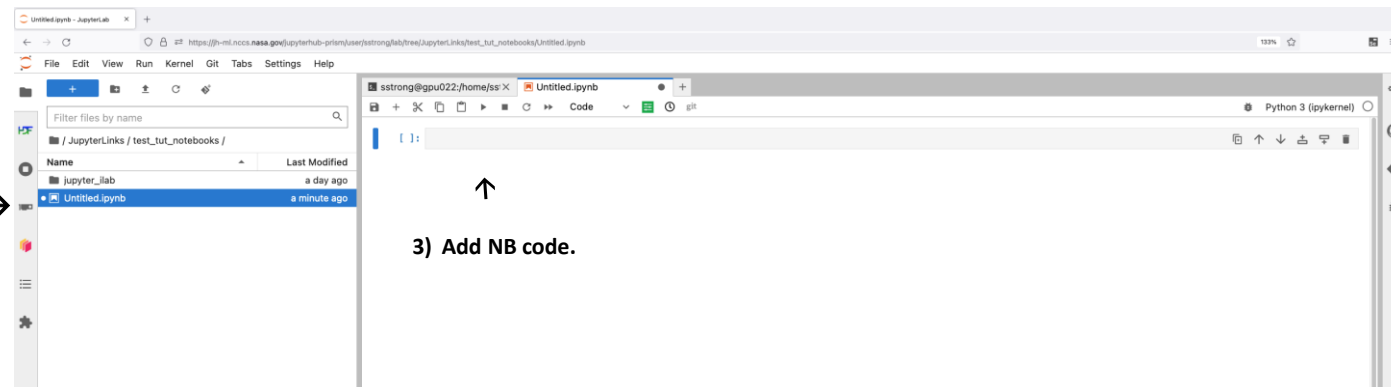
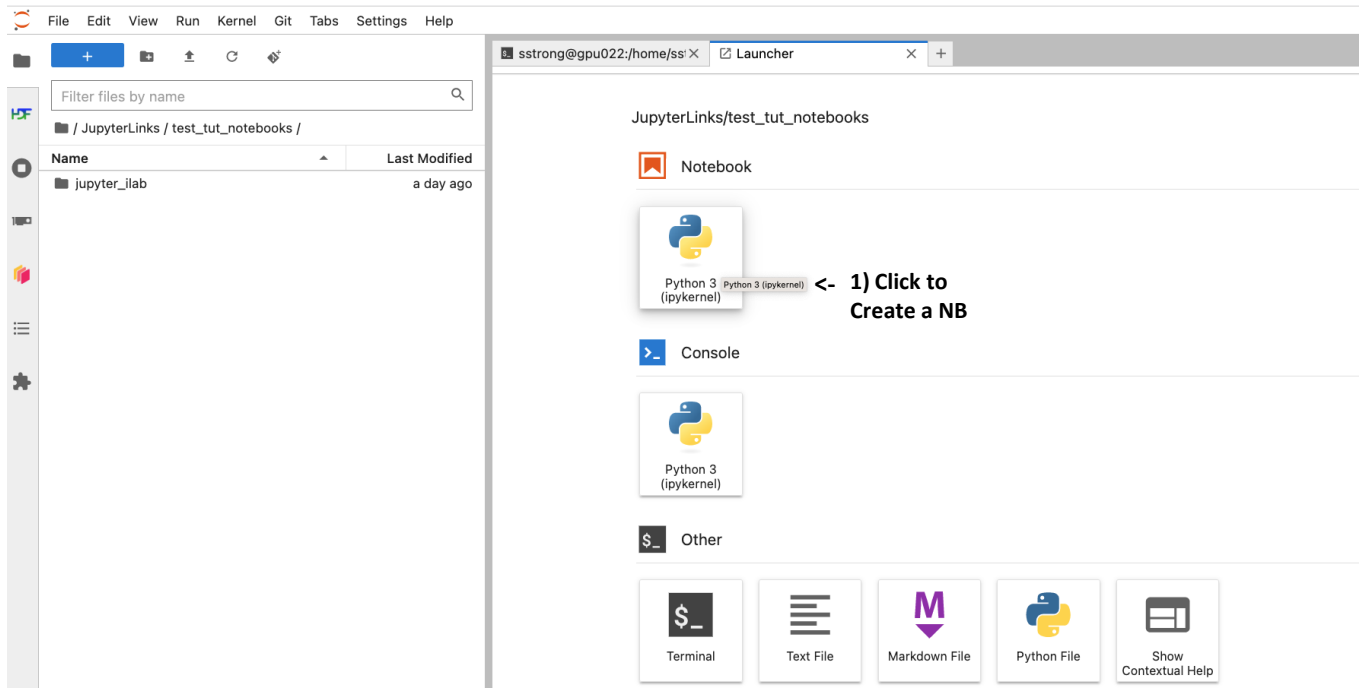
Markup →



Installed plugins →



Jupyter lab Create a NB

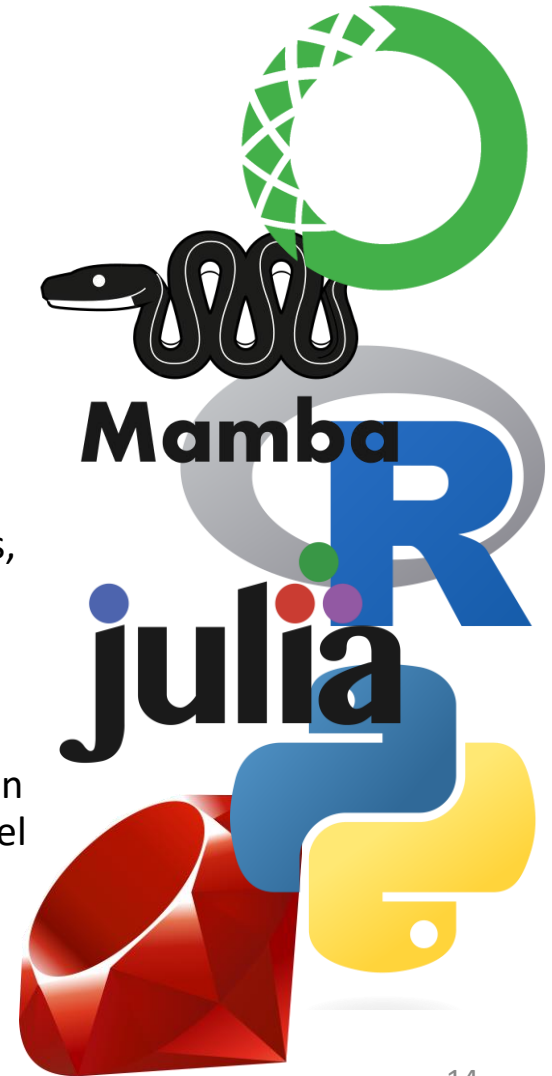


2) Creates an Untitled NB in current dir.



Environments & Kernels

- Many projects will require a distinct set of libraries and software for running code against. Package management tools such as Anaconda are available on our systems to help users build custom environments to support their various projects. While we do have various global environments available, it is highly recommended that users learn how to create and customize their own environments.
- Alternative tools exist for creating environments such as Mamba, or language native tools such as Python's venv, Julia's Pkg environments, Renv, to name a few. While Anaconda tends to be the easiest for integrating new environments into Jupyter sessions, these other tools can also be made to work with some additional configuration steps.
- A kernel, in the context of Jupyter, is simply an environment that has been made available for use in Jupyter. The IPython kernel for example is the Python execution backend for Jupyter. While python is the most common, other languages also use an interactive kernel to bridge a software environment to your Jupyter Notebook (IRKernel, IJulia, Iruby, etc).



Simple Python Environment Example

We use Lmod modules to provide Anaconda for users without dependency on installing the package management pieces in user space.

```
$ ml anaconda
```

With Anaconda loaded, we can then create our environment and we include ipykernel so that the environment will be usable as a kernel in your JupyterLab session.

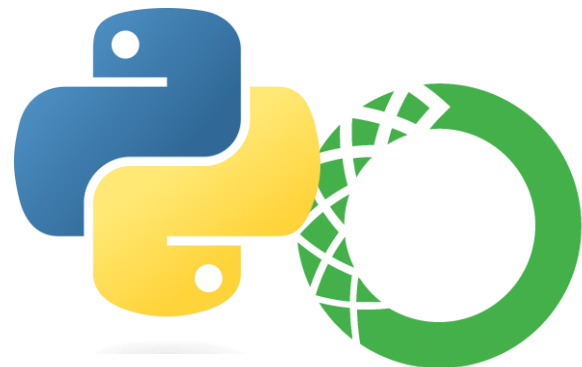
```
$ conda create -n MyEnvironment -c conda-forge python ipykernel -y
```

We can then activate our environment and run python.

```
$ conda activate MyEnvironment  
(MyEnvironment) $ python
```

We can also use Anaconda to install or remove different packages, or manage environments as a whole.

```
(MyEnvironment) $ conda install -c conda-forge numpy -y  
(MyEnvironment) $ conda remove numpy -y  
(MyEnvironment) $ conda deactivate  
$ conda env list  
...  
$ conda env remove -n MyEnvironment -y
```



Jupyter lab Conda Envs



The screenshot displays the Jupyter Lab interface. On the left is a file browser with a search bar and a list of files and folders. The main area is titled 'Terminal 1' and 'Launcher', showing a grid of kernel options under 'JupyterLinks'. The 'Notebook' section contains two rows of kernels: Python 3 (ipykernel), DEV Kernel, ILAB Kernel, ILAB Kernel (Pytorch), ILAB Kernel (TensorFlow), Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:earthml], and Python [conda env:ilab-]; and Python [conda env:ilab-], Python [conda env:ml], Python [conda env:r-], Python [conda env:root], R [conda env:.conda-], R [conda env:.conda-], R [conda env:.conda-], and R [conda env:environment]. The 'Console' section contains two rows of kernels: Python 3 (ipykernel), DEV Kernel, ILAB Kernel, ILAB Kernel (Pytorch), ILAB Kernel (TensorFlow), Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:.conda-], Python [conda env:earthml], and Python [conda env:ilab-]; and Python [conda env:ilab-], Python [conda env:ml], Python [conda env:r-], Python [conda env:root], R [conda env:.conda-], R [conda env:.conda-], R [conda env:.conda-], and R [conda env:environment]. The 'Other' section contains icons for Terminal, Text File, Markdown File, Python File, R File, and Show Contextual Help.

Name	Last Modified
css	3 years ago
icesat-2	3 years ago
ilab_projects	8 months ago
nobackup	4 hours ago
test_tut_notebooks	10 days ago
top	3 years ago
tutorialNotebooks	4 hours ago

Common Environment/Kernel Issues

- Anaconda environments are installed into user home directories by default. Home quotas are small and environments can become large quickly. My preferred method for solving this issue is to relocate `~/.conda` to `$NOBACKUP` and to create a symlink in your home directory, but other methods can be used to get around this (custom prefix on install, `.condarc` config, environment variables).
- On occasion Anaconda will prompt a user to run 'conda init' which will modify their `~/.bashrc` (or `cshrc`, etc.) to preload Anaconda. This process should be handled by loading the module, and if done through login scripts instead can result in kernels not showing up in your Jupyter session.
- As mentioned, environments require an interactive kernel package to be used within Jupyter. An environment will appear to exist from the command line, but will not be visible in your available kernel list until such a package is installed.

Demo

*Innovation Lab
Tutorial Notebook*



eViz

“An easy-to-use visualization framework for earth system models.”

EViz consists of two Python-based tools, eViz and iViz, that provide a simple way to visualize a wide variety of data sources. eViz is a CLI-driven static plotting tool configurable with YAML files. iViz is a Jupyter Notebook-based tool with a Bokeh backend for providing an interactive exploration of data sources.

https://www.nccs.nasa.gov/sites/default/files/eviz_tech_talk.pdf

<https://www.nccs.nasa.gov/nccs-users/instructional/instructional-videos>

Resources & Documentation

- Jupyter: <https://jupyter.org/>
- JupyterLab: <https://jupyterlab.readthedocs.io/en/latest/>
- Anaconda: <https://www.anaconda.com/>
- NCCS Documentation:
 - <https://www.nccs.nasa.gov/nccs-users/instructional/adapt-instructional/jupyter>
 - <https://www.nccs.nasa.gov/nccs-users/instructional/instructional-videos>

For any further assistance using JupyterHub on NCCS systems, or for any software related requests contact us at:

support@nccs.nasa.gov



Future Plans

- JupyterLab 4.0
- Allocation customization
- Anaconda/Environment management improvements
- Access and reliability for hubs behind www-proxy-dev

Questions & Open Discussion

Thanks for Attending!

- Recording will be made available after the conclusion of the tech talk.
- Slides will be made available on our website at: <https://www.nccs.nasa.gov/nccs-users/user-events/tech-talks>
- Suggestions for future tech talk topics or feedback regarding today's discussion can be put in the chat of this meeting
- Join us again next week on 6/29 at 12PM for another tech talk discussing explainable AI/ML.