## **HIGH PERFORMANCE RESEARCH COMPUTING** Primer: Running Jupyter Notebook on the ACES Portal

Presented by Keegan Smith September 3rd, 2024



High Performance Research Computing DIVISION OF RESEARCH



## Accessing the ACES Portal

- Dropdown menu on HPRC webpage: hprc.tamu.edu
- ACES portal shortcut portal-aces.hprc.tamu.edu
- FASTER portal shortcut portal-faster.hprc.tamu.edu
- Requires an ACCESS ID!



## Jupyter Notebook Interactive App

GUI
👿 VNC
z Nextsilicon VNC
Imaging
* CryoSPARC
ImageJ
🚵 cisTEM
Servers
📮 Jupyter Notebook
⊜ JupyterLab
RStudio

#### In the portal, go to:

#### **Interactive Apps**

-> Servers

A M

->Jupyter Notebooks

## Jupyter Notebook Interactive App

**ACES OnDemand Portal** Files - Jobs - Clusters -Interactive Apps -Dashboard -My Interactive Sessions

#### Home / My Interactive Sessions / Jupyter Notebook

Next we have to specify the environment and resources for our Jupyter session

Interactive Apps	Jupyter Notebook	
GUI	This app will launch a Jupyter Notebook server on the ACES cluster.	
🐹 VNC	Type of Environment	
🖉 Nextsilicon VNC	Python module + virtualenv	,
Imaging	Select the type of environment in which Jupyter is installed. Help me choose	
* CryoSPARC	Select Python Module Version	
ImageJ	Python/3.9.6	`
	Optional Python virtualenv to be activated	
m cisTEM		
Servers		
🗲 Jupyter Notebook	Enter the full path to a python virtual environment "activate" script to be activated. This field is optional. If used, it will override the default Jupyter.	ä
😅 JupyterLab	E.g. /scratch/user/netid/path-to-env/bin/activate	
e RStudio	Your optional environment must have been previously built with the Python module selected in the Module option above. It is expected to have a Jupyter package	3
	installed. Please see instructions.	



## Python Environments on ACES

#### **Supported Environment Types**

- Module + Virtualenv
- Anaconda + Conda Env
- Singularity Container

For now:

Choose these two:

Leave this blank:

#### Jupyter Notebook

This app will launch a Jupyter Notebook server on the ACES cluster.

Type of Environment

Python module + virtualenv

Select the type of environment in which Jupyter is installed. Help me choose

Select Python Module Version

Python/3.9.6

Optional Python virtualenv to be activated

Enter the full path to a python virtual environment "activate" script to be activated. This field is optional. If used, it will override the default Jupyter.

E.g. /scratch/user/netid/...path-to-env../bin/activate

Your optional environment must have been previously built with the Python module selected in the Module option above. It is expected to have a Jupyter package installed. Please see instructions.



## Session Options on ACES

Node type

- Provide **node type** (CPU only, H100 GPU or A30 GPU)
- Provide **number of hours** [1-168] for running notebook on the cluster
- Specify number of cores [1-96] allocated on node on the cluster
- Requested total memory [5-485 GB]
- Optionally:
  - Specify an account number to charge SUs
  - Provide an email to be notified when session is ready to launch

CPU only	~
Select a GPU node for software that supports GPU processing.	
Number of hours (max 168)	
1	
Number of cores (max 96)	
1	
Total GB memory (max 485)	
5	
Account	
This field is optional.	
Email	
This field is optional.	
Launch	

\* The Jupyter Notebook session data for this session can be accessed under the data root directory.



## **Interactive Sessions on ACES Portal**

After clicking Launch, you are sent to the My Interactive Sessions page.

- Jupyter notebook session with JobID (5129 in this case) and session is running.
- Session ID is link to a directory where you can find the jobs logs
- click **Connect to Jupyter** when it appears (depends on cluster usage)





## Welcome to Jupyter

#### File Navigator in Jupyter

- Default directory: user's \$SCRATCH directory or optionally-provided virtual environment directory
- Create new directories or Notebook files using New-> Python3
- If you are uploading existing .ipynb file, ensure Python version is compatible

💭 Jupyter		Quit	Logout
Files Running Clusters			+
Select items to perform actions on them.		Upload	New 🗸 🎗
	Name 🔶	Last Modified	File size
		2 months ago	
		22 days ago	
		3 months ago	

### Exercise 1

1. In Jupyter, create a new directory.

2. Create a new Jupyter Notebook file inside the directory. Save the file regularly to avoid losing changes.

## Jupyter Notebook Kernels

💭 jupyter U	Jntitled Last Checkpoi	nt: a minute ago (autosaved)			Logout		
File Edit Vie	ew Insert Cell	Kernel Help Interrupt I, I Restart 0, 0	+	Not Trusted	Python 3 (ipykernel) O	Python 3 ●	Thinking
In [2]:	print('Welcome t Welcome to HPRC	Restart & Clear Output Restart & Run All Reconnect Shutdown				Python 3 O	Idle
In [1]:	<pre>!pwd /scratch/user/pi.</pre>	Change kernel				Python 3 🐒	Broken!
In []:							1

The "kernel" is like the brain of your notebook. It determines what language you are thinking in–and if it's not working, you cannot process anything.

In the top-right of your window is the kernel status, showing what language you are using and what the kernel is doing.

When you first start a notebook, it will take a moment for the kernel to start up. If something goes wrong, you may have to restart it.

## Cells in Jupyter Notebooks

Notebooks are organized into Cells.

- These can be either code cells or markdown cells (we'll start with code).
- Click to the left of a cell to highlight it (blue, for "command mode"), [Double]click in the body of the cell to edit it (green, for "edit mode").
- Add new cells from the menu: Insert -> Insert Cell Below

💭 jupyter Unti	tled			Page 1
File Edit View	Insert Cell	Kernel	Help Trusted	Python 3 (ipykernel) O
B + % 4 B	Insert Cell Above Insert Cell Below	e A : v B	Code	¥ 🖃
In [ ]:				
In [ ]:				



## **Executing Code in Jupyter Notebooks**

**ctrl+enter** or click the **Run** button to run a cell.

The brackets to the left will show [\*] if the cell is running and change to a number (showing what order the cells were executed) once it's done.

File Edit	View Insert Cell Kernel Help
B + × 4	
In [2]	: print("Hello! Welcome to the course!!")
	Hello! Welcome to the course!!
In [3]	: print(1+1)
	2
In [4]	<pre>mylist = [1,2,3] print(mylist[0])</pre>
	1
In [10]	<pre>myint=1 myvar="Hello" print(type(myint)) print(type(myvar))</pre>
	<class 'int'=""> <class 'str'=""></class></class>



## Python in Jupyter Notebooks

Cells are just blocks of code you choose to execute all at once. You can execute them in any order, and any defined objects persist across them.

All that matters is what order you send code to the kernel.

In [2]:	<pre>#I'm going to execute this cell second! x = 5 * 20</pre>
In [3]:	<pre>#I'm going to execute this cell last! print(x)</pre>
	100
In [1]:	<pre>#I'm going to execute this cell first! print("Hello world!")</pre>
	Hello world!



## Plotting in Jupyter Notebooks

Making plots in Jupyter depends somewhat on your environment.

You may need a "notebook magic" to make images show up.

(magics are a big, kernel-dependent topic we won't cover here!)

In [1]:	# A simple plot with from matplotlib impo	<i>matplotlib</i> rt pyplot as plt
	x = [0,1,2,3,4,5,6,7 y = [2,4,7,11,6,2,0,	,8,9] -4,-1,2]
	<pre>plt.plot(x,y) plt.xlabel("x") plt.ylabel('y')</pre>	No magic, no plot!
Out[1]:	Text(0,0.5,'y')	







- 1. Enter some Python code in a cell. Print a message: "Welcome to HPRC".
- 2. Try to make a simple plot, like on the previous slide.



## Markdown in Jupyter Notebooks

Can use Markdown cells for annotation, documentation, organization, etc.

- Convert a cell from drop down menu in the toolbar (can also be found in the Cell menu)
- As with code cells, Ctrl+enter or Run to execute and format Text.

Font changes from monospace to something more **beautiful**.

File	Edit	V	/iew	Ins	ert	Cell	Ke	rnel	V	Vidgets H	elp	/
-	*	ළු	•	1	≁	N Run		C	₩	Code	~	
										Code	0	
									_	_ Markdown		<b>P</b>
	In [	]:								Raw NBConv	ert	٢
1										Heading		5
	Tn [	1.										1





## Markdown in Jupyter Notebooks

- Headers

   # (Header 1, title) stands for html code
   <h1>Header 1, title</h1>

   Add more '#' for smaller headers
- Line Break Insert line breaks using <br>
- Inline Formatting Bold: \*\*HPRC\*\* Italics: \*HPRC\* Horizontal Line: \*\*\*
- Embed external link
   Works like html:
   <a href="url" >link text</a>

#### # Look! Markdown!

```
Normal text<br>
Normal text on a new line
```

\*\*Bold text!\*\*
\*Italic text\*
\*\*\*

<a href="<u>https://hprc.tamu.edu/</u>">Link to HPRC</a>



#### Look! Markdown!

Normal text Normal text on a new line

Bold text! Italic text

Link to HPRC

## Exercise 3: Markdown

- 1. Create a Markdown Cell and add some text.
- 2. Change the formatting as indicated below:

Bold

Italics

Headerl format

Include a link (<u>https://hprc.tamu.edu/</u>) next to the text



## More Text Formats in Jupyter Notebooks

#### LaTeX: for mathematical operations

- In a Markdown cell:
- \$ math expression \$

\$\sqrt{k}\$	
$\sqrt{k}$	

#### Embed Code in Text:

- In a Markdown cell:
- use the triple backticks ```
   followed by a language name.

```
```Python
str = "This is block level code"
print(str)
```

str = "This is block level code"
print(str)



### Shutting Down



- Notebooks (i.e. their kernels) keep running even if you exit their browser tab
- To shut one down, find it in the Jupyter file browser, where its icon will be green...
- Select its checkbox and then shut it down at the top
- (Can also find them in the "Running" tab at the top and shut them down there.)

Files	Running	Clusters
Duplic	Shutdown V	/iew Edit 💼
<b>1</b>	- Inotel	books
C	J	
	CDMSliteR2_	cuts
	□ g4dmc_samp	le_analysis
	old C	
	Simulation_and	alysis_tutorial
	□ temp_data	
	animatedPlotT	ests.ipynb
	checkspectra.	ipynb
	test_Notebook	ipynb
	my_py3_funct	ions.py
	rotPlot.gif	

### Ending the Jupyter Session

As long as your Jupyter session is running (not just notebooks!) you're taking up compute resources!

To quit before the time you specified at the beginning, select 'Quit' on the file browser.

Back in "My Interactive Session" in the HPRC Portal, the box for your session will turn grey to indicate it is done.





## **Exercise 4: Shutting Down**

- 1. Shut down your Jupyter session
- Find the Log Files for your Recent Session.
   (Hint: the Session ID link from the My Interactive Sessions is a shortcut; you should end up in:

\$SCRATCH/ondemand/data/sys/dashboard/batch\_connect/sys/jupyter/output/

(You may never need these, but be aware of them.)





High Performance Research Computing DIVISION OF RESEARCH

## Thank you.

## Any questions?

Please let us know what you thought of this course by filling out this survey: https://u.tamu.edu/hprc\_shortcourse\_survey



## Need Help?

First check the FAQ <u>https://hprc.tamu.edu/kb/FAQ/Accounts/</u>

- ACES User Guide <a href="https://hprc.tamu.edu/kb/User-Guides/ACES/">https://hprc.tamu.edu/kb/User-Guides/ACES/</a>
- Email your questions to help@hprc.tamu.edu

Help us help you -- when you contact us, tell us:

- Which Cluster you're using
- Your Username
- Job id(s) if any
- Location of your jobfile, input/output files
- Application used if any
- Module(s) loaded if any
- Error messages
- Steps you have taken, so we can reproduce the problem

## **Continued Learning**

#### Intro to HPRC Video Tutorial Series

HPRC's Knowledge Base





High Performance Research Computing DIVISION OF RESEARCH

# Thank you. Any questions?



## Notebook Files In Jupyter

#### Jupyter Notebook- Familiarize the setup

- Jupyter server is up and running.
  - To create a notebook:

#### New -> Python 3 (ipykernel)

- Upload notebook (Python version must be the same/compatible):
   Upload -> Select notebook
- File-naming convention:

All lowercase module names. Long module names can have words separated by underscores (really\_long\_module\_name.py). CamelCase for class names.

## Jupyter Notebook Sessions

#### Jupyter Notebook- Familiarize the setup

#### **Kernel Options:**

- Interrupt
- Restart
- Restart & Clear Output
- Restart & Run All
- Reconnect
- Shutdown

- Jupyter Untitled Last Checkpoint: a minute ago (autosaved) Logout Kernel Help Not Trusted Python 3 (ipykernel) O Interrupt \$ 0,0 Restart Restart & Clear Output In [2]: print('Welcome t Restart & Run All Reconnect Welcome to HPRC Shutdown In [1]: !pwd Change kerne In [ ]:
- Change kernel (switch between various python versions available)

## Navigating Files in the ACES Portal

#### File Explorer

- Can view a file explorer for the users home or scratch directory
- View, delete, edit and create new files or directories on clusters
- Upload or download files

ACES OnDemand Portal	Files • Jobs • Clusters •	Interactive Apps 🝷 Dashboar	rd • 🗐 😡 • 💄 G
>_ Open in Termin	al ) C <sup>H</sup> Refresh + New File	🗅 New Directory 主 Upload 🛃 D	ownload 🕒 Copy/Move
Home Directory /scratch/user/u.ac122202	↑ / scratch / user / u.ac batch_connect / sys / jupyte Change directory	122202 / ondemand / data / sys / r / output / 9457ec3b-ff17-49fe-	/ dashboard / a4ab-8de2f53f27f4 / ① Copy path
	□ Show Own	er/Mode □ Show Dotfiles Filt Showir	er: ng 9 rows - 0 rows selected
	Type Name	♦ Size ♦	Modified at

## **Exercise 3: Log Files**

#### Log files can be found on disk at

\$SCRATCH/ondemand/data/sys/dashboard/batch\_connect/sys/jupyter/output/

Find the Log Files for your Recent Session.
 (Hint: the Session ID link from the My Interactive Sessions is a shortcut)

What log files are available?



## Wrapping Up on the ACES Portal

#### **Final Steps**

#### • End Interactive Session:

In your Jupyter File Navigator, click 'Quit' in the top navigation bar. This will stop the running job and prevent wasting your SU's. (Logout does not do this)

#### • Saving the Notebook files:

Notebooks will be saved in the default directory (\$SCRATCH or virtual environment) This space persists between sessions.

#### • Logging Out of the Portal:

To properly log out the portal, follow the below mentioned steps:

- log out the portal by clicking 'Log out' from the top navigation bar
- close the browser to completely terminate the session

## Plots in Jupyter Notebooks

If you don't have matplotlib installed on your virtual environment, you need to install it

In [1]: !pip install matplotlib

Magic function that makes plots appear in your notebook

```
In [2]: %matplotlib inline
```

```
Let's make an example plot of Y=X<sup>2</sup>
```

```
In [3]: import matplotlib.pyplot as plt
x = [0,1,2,3,4,5]
y = [0,1,4,9,16,25]
plt.plot(x,y)
```

More documentation on plotting: <u>https://matplotlib.org/stable/users/index</u>



## Code Cells in Jupyter Notebooks

Variables can persist between code cells

In [1]:	$\mathbf{x} = 5$
In [2]:	print(x)
	5

