Primer: Running Jupyter Notebook on the ACES Portal

Presented by Victoria Chen
February 6th, 2024
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!
In the portal, go to:

Interactive Apps

-> Servers

-> Jupyter Notebooks
Next we have to specify the environment and resources for our Jupyter session.
Python Environments on ACES

Supported Environment Types
- Module + Virtualenv
- Anaconda + Conda Env
- Singularity Container

For now:
Choose these two:
Leave this blank:
Session Options on ACES

- 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

*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
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.
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**
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.

```python
In [2]: print("Hello! Welcome to the course!!")
Hello! Welcome to the course!!

In [3]: print(1+1)
2

In [4]: mylist = [1,2,3]
   print(mylist[0])
1

In [10]: myint=1
   myvar="Hello"
   print(type(myint))
   print(type(myvar))
<class 'int'>
<class 'str'>
```
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.
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 matplotlib
   from matplotlib import pyplot as plt

   x = [0,1,2,3,4,5,6,7,8,9]
   y = [2,4,7,11,6,2,0,-4,-1,2]

   plt.plot(x,y)
   plt.xlabel("x")
   plt.ylabel("y")

Out[1]: Text(0.05,'y')
```

No magic, no plot!

```
In [2]: # A simple plot with matplotlib
   from matplotlib import pyplot as plt

   x = [0,1,2,3,4,5,6,7,8,9]
   y = [2,4,7,11,6,2,0,-4,-1,2]

   plt.plot(x,y)
   plt.xlabel("x")
   plt.ylabel("y")

Out[2]: Text(0,0.5,'y')
```

Magic first, then all your cells can make plots!
Exercise 2

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.
# 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>`
Exercise 3: Markdown

1. Create a Markdown Cell and add some text.
2. Change the formatting as indicated below:
   - Bold
   - Italics
   - Header1 format
   - Include a link ([https://hprc.tamu.edu/](https://hprc.tamu.edu/)) next to the text
More Text Formats in Jupyter Notebooks

LaTeX: for mathematical operations
- In a Markdown cell:
- $ math expression $ 

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)
```
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.)
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.

You can click on the Session ID to go to the log files if you want to debug!
Exercise 4: Shutting Down

1. Shut down your Jupyter session

2. 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.)
Thank you.
Any questions?
Need Help?

First check the FAQ [https://hprc.tamu.edu/kb/FAQ/Accounts/](https://hprc.tamu.edu/kb/FAQ/Accounts/)
- 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