Introduction to Jupyter & Virtual Environments

Tutorial 2

Agenda

- What is Jupyter?
  - Jupyter Notebooks or Jupyter Lab?
- Jupyter on your local machine!
- How do I access Jupyter on the clusters?
- Sorcery . . . witchcraft? . . . No, just magics

- What are virtual environments?
- Why should I care?
- Setting one up locally and on the cluster.

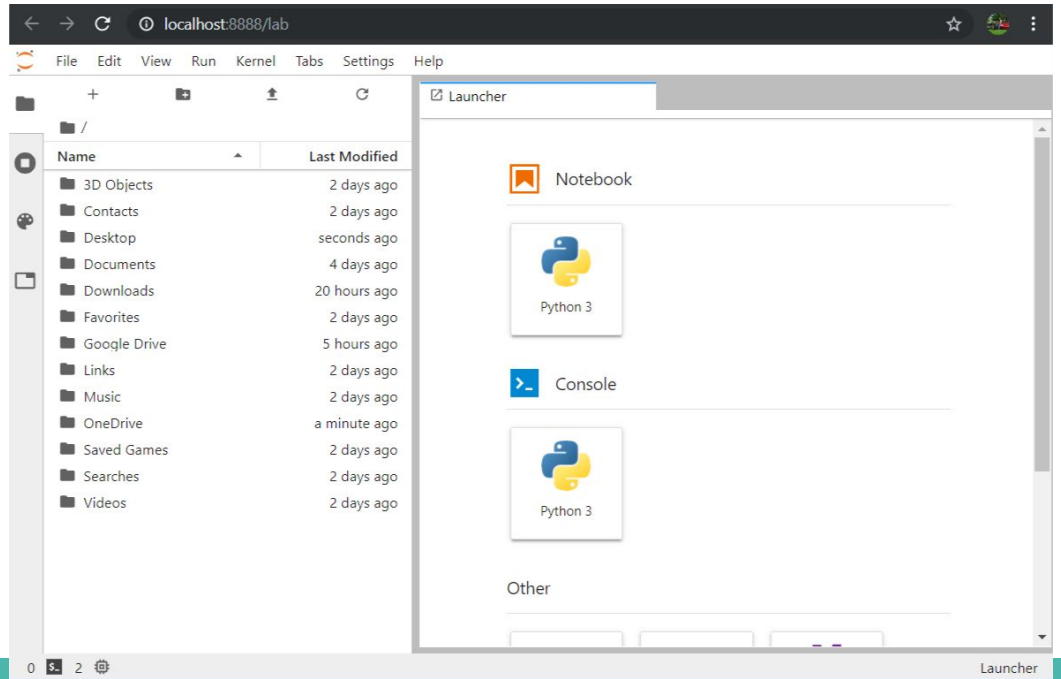
- Where does this all fit into my workflow?

What is Jupyter?

- Jupyter is short for **Ju**lia, **Py**thon, and **R**. Jupyter provides an interface to code in and quickly get results.
- Many other languages can be used in Jupyter.
- Great for rapid prototyping and documenting your work.
- Two Options
  - Jupyter Notebooks
  - **Jupyter Lab - More full featured, this is where we'll focus our time but they're very similar. If you've used one, you can use the other with no problems.**

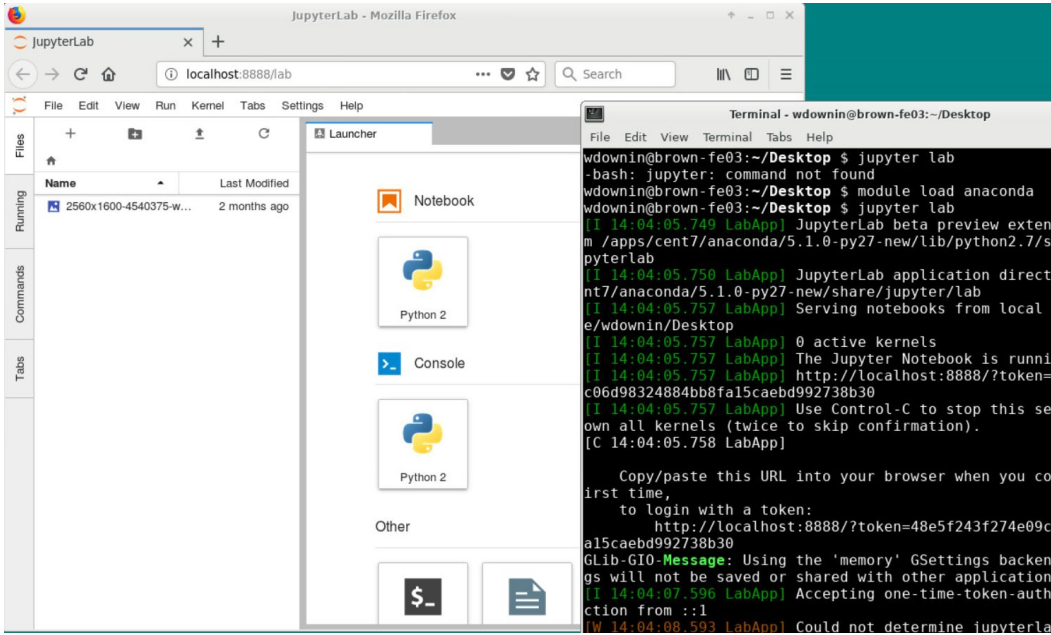
Jupyter on your desktop!

- On your desktop, it's as simple as opening Anaconda Navigator
  - Or opening an anaconda shell and typing jupyter lab (or jupyter notebook) and hitting enter.
- By default you can launch Python notebooks but you can also set Jupyter up to run other languages. (other languages are called Kernels in Jupyter)



Jupyter on the cluster.

- Suppose you want to move your workflow to the cluster, how do we interact with Jupyter there?
  - Desktop.scholar.rcac.purdue.edu
- You can also download the thinlinc client and skip the browser altogether.



Magics in Jupyter?

- Jupyter provides some really neat functionality through something called magics!
  - With magics we can run bash commands (even us windows users :D, provided we have Windows Subsystem for Linux), R commands, SQL Queries, and quite a bit more!

  - Line magic - Denoted by %
    - Indicates a magic function that works on a single line
  - Cell magic - Denoted by %%
    - Indicates a magic function that works on the entire cell

What are virtual environments and why do I care?

- Virtual environments are one of the keys to reproducible work (not the only key but they are important).
- What happens when you try to run your code a year later and some of the libraries you used before have been updated?
    - Some functions may have been deprecated or updated and now no longer work the way they did when you worked on your code last, leading to bugs.
- Virtual environment provide a version control system for your packages that you use!

Steps to setting up a virtual environment!

1. Open a terminal (anaconda shell if you're on windows, a regular terminal will suffice on MacOS, Linux, or the Cluster).
2. Type in the following . . .
   Conda create --name CHOOSE PACKAGES
   a. Enter the name you've chosen for your virtual environment (be descriptive) and the packages you want installed.
   b. NOTE: On the cluster you first need to enter
      module load anaconda
      i. You can see what versions of anaconda and python are available by typing module spider
3. Now you can activate that environment!
   a. conda activate CHOSEN_NAME

What if I forgot what environments I've made?

- Simply type in the following command
  conda env list
  - This will list out all virtual environments and their corresponding names.
  - This is the same on the cluster and on the local machine.

The importance of keeping your workflow clean.

- Some great blog posts have been included in the repository to showcase what a clean workflow might look like.
  - **These are suggestions not rules, ultimately you have to decide what makes sense for your project. The most important thing is that you cook up some way of staying organized and making your work reproducible (most likely by you six months later).**