

# Unidata Science Gateway JupyterHub

ESIP Summer 2018 Meeting

---

Julien Chastang

Friday, 20 July 2018

# Outline

Background

Jupyter

JupyterHub as a Science Gateway Companion

Unidata Science Gateway JupyterHub

Implementation Details, Software Carpentry

Future Plans

# Background

---

# Unidata Five-Year Plan

*Transition to a remote-access "cloud" environment for geoscience data is vital to our community's ability to pursue research and education in the 21st century.*

-Unidata Five-year plan 2013-2018

# Forces at Play

- Data are becoming unwieldy and expensive to move
- Cloud-computing and technologies like Docker, Kubernetes etc. allow easy deployments of complex environments at scale
- Web browser is the de facto cross-platform "app".
- Economic advantages to running VMs in the cloud

# Consequences of These Forces

- Computation, visualization, analysis should be done where data live
- Workflows should be cloud-based and accessible through a web browser
- Opportunity to unify these workflows under the concept of **science gateway**

# Target Audience

- Core Unidata Community
- Atmospheric science departments and their students
- Professors and students get to science faster

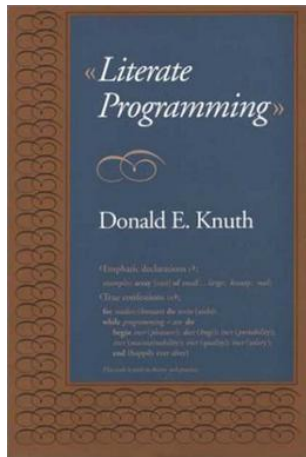
# Jupyter

---



# What is Literate Programming?

- Prose, diagrams, equations first, code second.
- Excels at teaching and expository objectives and "executable" scientific papers.
- Despite long roots in programming, LP is finally popular thanks to **Jupyter Notebooks**.



# What is a Jupyter Notebook?

Cells of:

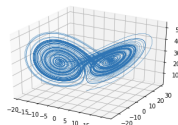
- Markdown
- Equations (w/ MathJax,  $\text{\LaTeX}$ )
- Executable code (Python, R etc.)
- Figures

## Lorenz System

The Lorenz system is a series of Ordinary Differential equation studied by Edward Lorenz.

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

```
In [10]: def lorenz(x, y, z, s=10, r=28, b=2.667):  
    x_dot = s*(y - x)  
    y_dot = r*x - y - x*z  
    z_dot = x*y - b*z  
    return x_dot, y_dot, z_dot  
  
dt = 0.01; stepCnt = 10000  
xs = np.empty((stepCnt + 1,))  
ys = np.empty((stepCnt + 1,))  
zs = np.empty((stepCnt + 1,))  
xs[0], ys[0], zs[0] = (0, 1., 1.05)  
  
for i in range(stepCnt):  
    x_dot, y_dot, z_dot = lorenz(xs[i], ys[i], zs[i])  
    xs[i + 1] = xs[i] + (x_dot * dt)  
    ys[i + 1] = ys[i] + (y_dot * dt)  
    zs[i + 1] = zs[i] + (z_dot * dt)  
  
fig = plt.figure()  
ax = fig.gca(projection='3d')  
ax.plot(xs, ys, zs, lw=0.5)  
plt.show()
```



# A Tool of Thought

- Concept by Ken Iverson and APL programming language.
- Foreshadowed interactive, exploratory, scientific programming
- Instead of APL we have Python

APL:

```
1,↑¨+.×\N2 21 1 1 0
```

Python:

```
def fb(n):  
    if n < 2:  
        return n  
    else:  
        return fb(n-1) + fb(n-2)
```

# Success of Jupyter Notebooks

1.4K



## Jupyter receives the ACM Software System Award

It is our pleasure to announce that Project Jupyter has been awarded the 2017 ACM Software System Award, a significant honor for the project. We are humbled to join an illustrious list of projects that contains major highlights of computing history, including Unix, TeX, S (R's predecessor), the Web, Mosaic, Java, INGRES (modern databases) and more.

Officially, the recipients of the award are the fifteen members of the Jupyter steering council as of November 2016, the date of nomination (listed in chronological order of joining the project): Fernando Pérez, Brian Granger, Min Ragan-Kelley, Paul Ivanov, Thomas Kluyver, Jason Grout, Matthias Bussonnier, Damián Avila, Steven Silvester, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Carol Willing, Sylvain Corlay and Peter Parente.



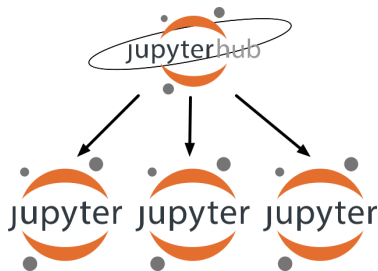
This is the largest team ever to receive this award, and we are delighted that the ACM was willing to recognize that modern collaborative projects are

# JupyterHub as a Science Gateway Companion

---

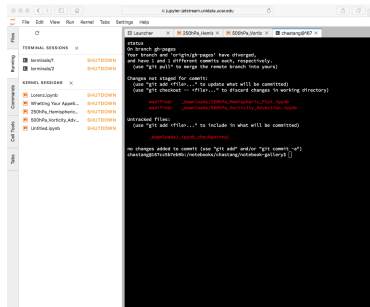
# JupyterHub: Multi-user Jupyter Notebooks

- Users log in to a JupyterHub server
- Users have their own work space
- Excellent for workshops or in the classroom
- Administrator can configure ahead of time on behalf of user



# More Than Just Notebooks

- git utilities
- software carpentry tools
- Data transfer utilities  
(e.g., Globus personal connect)
- Powerful b/c command lines are powerful.



The screenshot shows a JupyterLab window titled "jupyterlab:untitled.untitled". The interface includes a sidebar with "Files", "Environments", "Commands", "Cell Tools", and "Tabs". The "Environments" panel shows several environments, including "terminal1", "terminal2", "kernel.conda", "conda.conda", "conda.conda", "conda.conda", and "conda.conda", all with a "SHUTDOWN" status. The main area displays a terminal window with the following output:

```
status
On branch @pages
Your branch and 'origin/gh-pages' have diverged,
and have 1 and 0 different commits each, respectively.
(Use "git pull" to merge the remote branch into yours.)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   _downloads/200Pa_Therapeutic_Full_Tools
        modified:   _downloads/200Pa_Therapeutic_Advitation_Tools

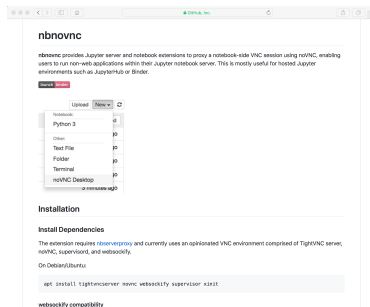
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        _downloads/100Pa_Therapeutic

no changes added to commit (use "git add" and/or "git commit -a")
(jupyterlab:untitled.untitled) [1]
```

# Jupyter Extensions

- Jupyter and JupyterHub can be extended by software developers
- Dozens of extensions have been written, e.g.,
- code folding
- code formatting
- Web VNC client





# Unidata Science Gateway JupyterHub

---

# Unidata Science Gateway JupyterHub

`https://jupyter-jetstream.unidata.ucar.edu`

Three PyAOS notebook projects w/ pre-configured environments:

1. Unidata Python Workshop
2. Unidata Notebook Gallery
3. Unidata Online Python Training

67 PyAOS notebooks total

# Pre-configured Environments

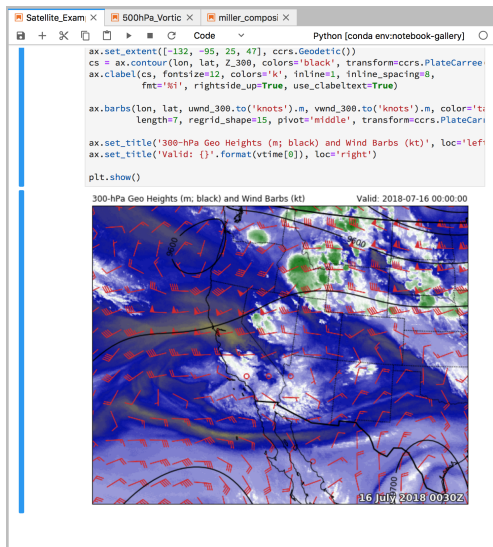
Complex environments already built w/ conda package manager

- conda environments e.g.,
  - numpy
  - netCDF4-Python
  - cartopy
  - metpy
  - xarray
  - etc.
- Select appropriate kernel from kernel menu

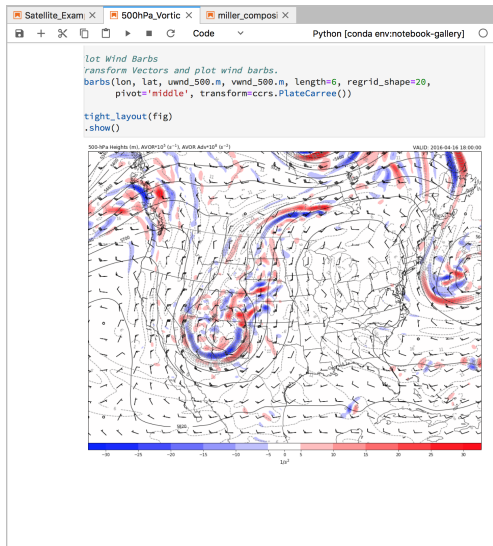
# JupyterHub Workflows

- Data-proximate analysis and viz:
  - LDM/IDD → THREDDS Data Server → Jupyter Notebooks
  - XSEDE HPC → Terminal Globus Tools → Jupyter Notebooks
- Remote access:
  - JupyterHub and Notebooks hitting (e.g., NOMADS)

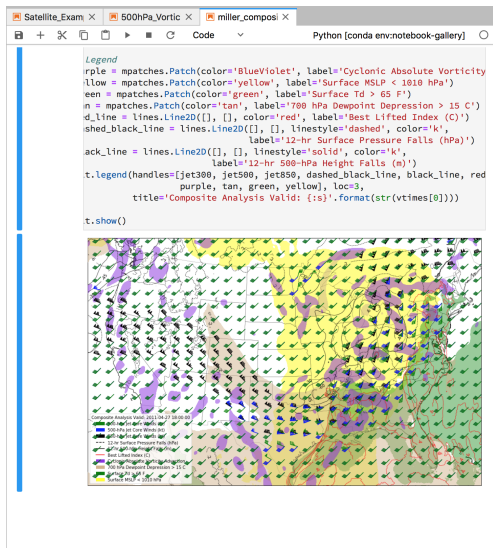
# Example 1: Satellite + GFS



# Example 2: Vorticity Advection



# Example 3: Miller Composite



# Implementation Details, Software Carpentry

---



# GitHub Repository

- Collaborative development on GitHub
- <https://github.com/Unidata/xsede-jetstream>
- <https://github.com/Unidata/xsede-jetstream/tree/master/vms/jupyter>

# Unidata JupyterHub Docker Container

- Based on official `jupyterhub/jupyterhub:master`
- Pre-built conda/pip environments (available via Kernel menu in Jupyter)
- Globus Personal Connect for moving data locally
- Environments are complex, take a long time to build
- Best if we do the work so you get to science faster
- Using `docker-compose` w/ stock `nginx` container

# OpenStack VM on Jetstream Cloud

- JupyterHub Deployed On Jetstream Cloud "XLarge VM"  
24 CPUs 60 GB of RAM

## Caveat emptor:

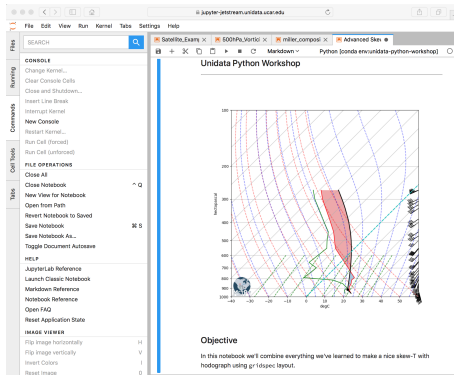
- No Kubernetes *yet* so will not scale to an influx of concurrent users
- OK for a number of occasional users

# OAuth

- Leveraging other authentication systems
- Globus OAuth via XSEDE
- Need login at <https://portal.xsede.org>
- Configured in `jupyterhub_config.py`

# JupyterLab

- Next generation Jupyter interface
- Easier to navigate and more functionality
- Classic view still available



# Upon First Login

- User directory space created with three notebook projects
- Responsibility then falls on the user for maintaining the projects up to date with `git` via the terminal
- User data (e.g., notebooks) is persisted and backed up

# Best Practices for Sharing Notebook Projects

- Ensure you have a `conda environment.yml` at the root of the notebook project
- Remember that notebooks are more than a series of code cells
- It is an expository exercise

# Future Plans

---



# Scaling for Cloud

- Kubernetes on Jetstream
- Continue collaboration w/ XSEDE ECSS (Semir Sarajlic and Suresh Marru)
- Andrea Zonca's Zero to JupyterHub solution (once ready)

# nbnovnc Jupyter Extension

- May solve "last mile" problem to CloudIDV
- AWIPS CAVE client also a possibility
- "Hub" in JupyterHub leveraged to solve problems beyond Jupyter

# Final Thoughts

- The technology hurdles surround JupyterHub will be solved
- Content creation will be the rate limiting step
- Domain expertise and effort required to create quality, literate programming notebooks

# Unidata JupyterHub

- <http://science-gateway.unidata.ucar.edu>
- <https://jupyter-jetstream.unidata.ucar.edu>
- <https://github.com/Unidata/xsede-jetstream>