Outline

- Getting Started; Comet Overview
- What are Jupyter Notebooks?
- Security concerns
  - HTTP vs HTTPS
  - SSH vs SSH tunneling (HTTP)
- Software Requirements for Running Notebooks on Comet
  - Install conda, conda environments
- Methods for Running Notebooks on Comet
  - Running notebooks on the Login node or interactive node
- SDSC Reverse Proxy Service (HTTPS)
- Live Demo

- Key Goal: Learn how to run Jupyter Notebooks securely.

SDSC Summer Institute 2020: 5.3 Jupyter Notebooks, Reverse Proxy Server

Mary Thomas, Computational Scientist, SDSC

Outline

- Getting Started; Comet Overview
- What are Jupyter Notebooks?
- Security concerns
  - HTTP vs HTTPS
  - SSH vs SSH tunneling (HTTP)
- Software Requirements for Running Notebooks on Comet
  - Install conda, conda environments
- Methods for Running Notebooks on Comet
  - Running notebooks on the Login node or interactive node
- SDSC Reverse Proxy Service (HTTPS)
- Live Demo

- Key Goal: Learn how to run Jupyter Notebooks securely.

Basic Information

- This webinar location:
  - https://github.com/sdsc-hpc-training-org/notebooks-101
- Online repo for companion tutorial/webinar information:
  - https://github.com/sdsc-hpc-training-org/notebook_examples
  - https://github.com/sdsc-training-org/webinars
  - Access to the Jupyter Reverse Proxy Server:
  - https://github.com/sdsc-hpc-training-org/reverse-proxy
- Other training events and links to past events listed at SDSC:
  - https://www.sdsc.edu/education_and_training/training.html
- You must be familiar with running basic Unix commands, connecting to Comet via SSH, running notebooks, and other basic skills. Check out our basic skills repo:
    - https://github.com/sdsc-hpc-training-org/basic_skills
  - You must have a comet account in order to access the system. To obtain a trial account:
  - http://www.sdsc.edu/support/user_guides/comet.html#trial_accounts
- Comet User Guide:
  - https://www.sdsc.edu/support/user_guides/comet.html

REMINDER!!!!

Jupyter Notebooks should not be run on the login nodes. Those jobs will be deleted.

```
                    Last login: Thu May 21 05:15:32 2020 from 76.176.117.51
Rocks 7.0 (Manzanita)
Profile built 12:32 03-Dec-2019


Kickstarted 13:47 03-Dec-2019

 WELCOME TO
      ________________  __  _____________
        -----/ ____/ __ \/  |/  / ____/_  __/
          --/ /   / / / / /|_/ / __/   / /
           / /___/ /_/ / /  / / /___  / /
           \____/\____/_/  /_/_____/ /_/
################################################################################
NOTICE:
The Comet login nodes are not to be used for running processing tasks.
This includes running Jupyter notebooks and the like.  All processing
jobs should be submitted as jobs to the batch scheduler.  If you don't
know how to do that see the Comet user guide
https://www.sdsc.edu/support/user_guides/comet.html#running.
Any tasks found running on the login nodes in violation of this policy
 may be terminated immediately and the responsible user locked out of
the system until they contact user services.
################################################################################
```

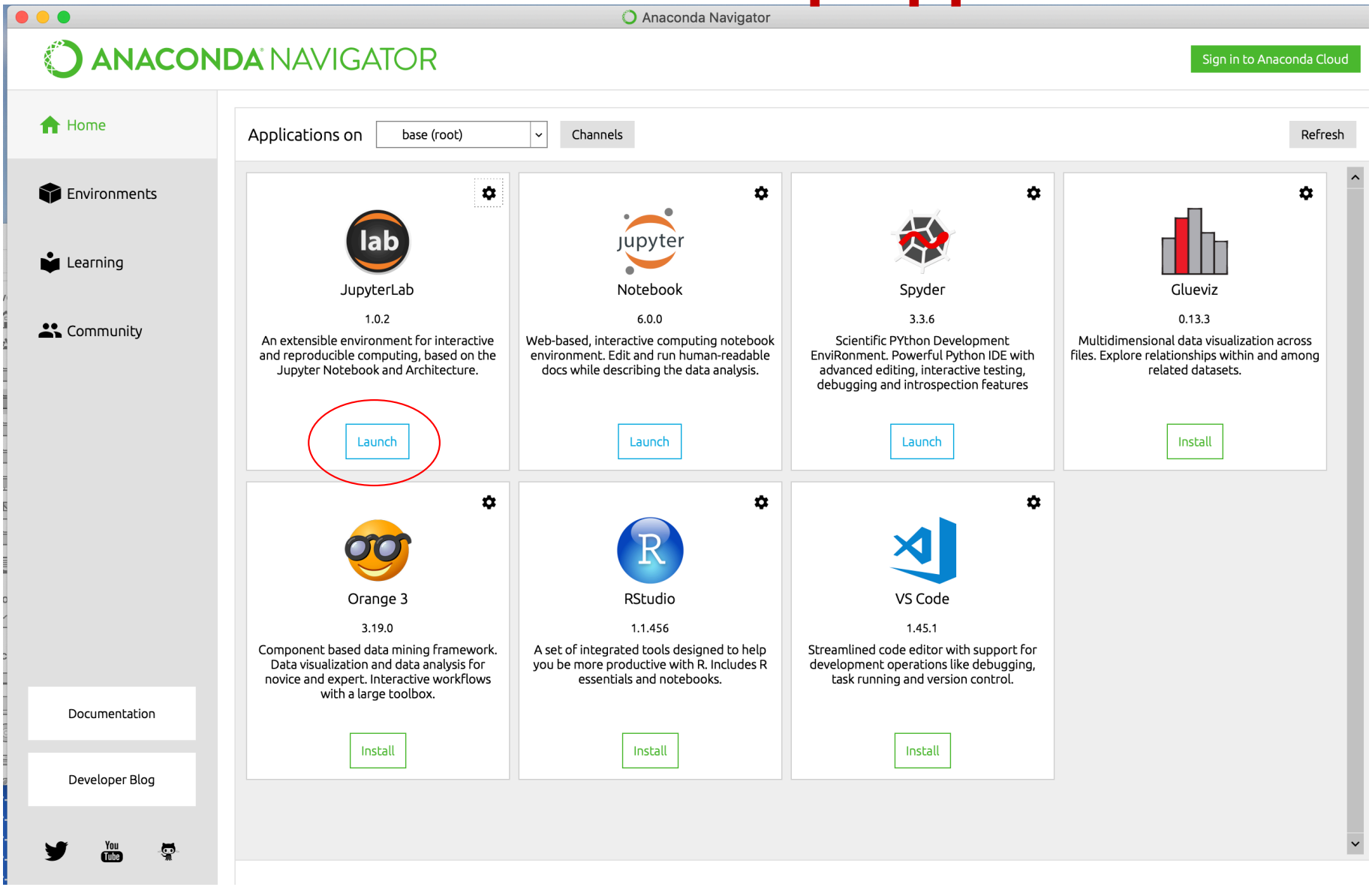
Obtaining Notebook Examples

```
(base) [username@comet-ln3:~] git clone https://github.com/sdsc-hpc-training-org/notebook_examples.git
Cloning into 'notebook_examples'...
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 55 (delta 6), reused 55 (delta 6), pack-reused 0
Unpacking objects: 100% (55/55), done.
(base) [username@comet-ln3:~] cd notebook_examples/
(base) [username@comet-ln3:~/notebook_examples] ll
total 609
drwxr-xr-x  7 username use300      9 May 20 12:38 .
drwxr-x--- 58 username use300     89 May 20 12:38 ..
drwxr-xr-x  3 username use300      8 May 20 12:38 Boring_Python
drwxr-xr-x  4 username use300      4 May 20 12:38 cuda
drwxr-xr-x  2 username use300      4 May 20 12:38 deep_learning
drwxr-xr-x  8 username use300     13 May 20 12:38 .git
-rw-r--r--  1 username use300 432678 May 20 12:38 gnuplot.ipynb
drwxr-xr-x  2 mthomas use300      6 May 21 07:34 hello-world
drwxr-xr-x  8 username use300   1060 May 20 12:45 hello_world.ipynb
drwxr-xr-x  2 username use300     10 May 20 12:38 Pandas
-rw-r--r--  1 username use300    322 May 20 12:38 README.md
(base) [username@comet-ln3:~/notebook_examples]
```

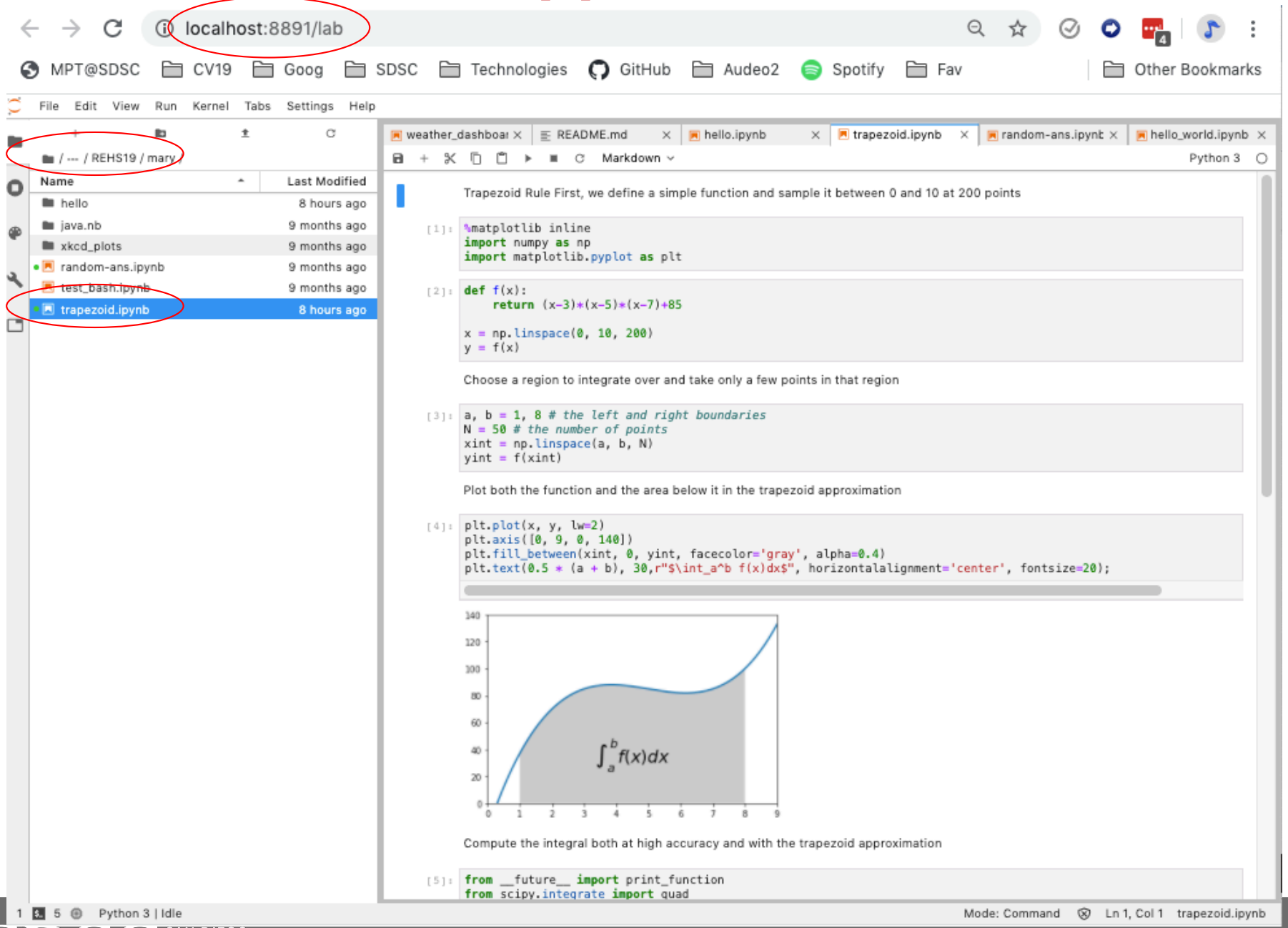
Software Requirements for Running Notebooks on Comet

https://comet-notebooks-101.readthedocs.io/en/latest/prerequisites.html

Anaconda: desktop application



OS X – Launch Apps with click of a Button



Software Requirements on HPC Systems

- Not so easy to run notebooks on HPC system/Unix
- Important and convenient to have customized, virtual Python environments,
  - install packages that aren't installed with the system's Python installation
  - You need different sets of Python packages for different purposes.
- We recommend that you setup your own local environment:
  - This gives you control over libraries used by your notebooks
  - You can install either Anaconda or just conda
    - Anaconda includes the conda command (which can be used to create, use, and manage virtual Python environments).
    - Use system Python
- Optionally: use singularity
  - Install locally using anaconda/etc.
  - Advantage of using containers: everything is built for you to use
  - Disadvantage: not easy to modify

Conda

- https://docs.conda.io/projects/conda/en/latest/
- Conda is an open-source package management system and environment management system (like pip)
- Created for Python programs
  - can package and distribute software for any language.
- Conda Cheat Sheet:
  - https://kapeli.com/cheat_sheets/Conda.docset/Contents/Resources/Documents/index

Create a virtual environment

- Use conda to create a virtual environment
  - Choose whatever name you want
  - $ conda create --name example_env
- To see which virtual environments you've created:
  - $ conda env list
- To use a particular virtual environment (e.g., one named 'example_env'):
- $ source activate example_env # Note: don't use 'conda activate'
- Install JupyterLab and JupyterNotebooks

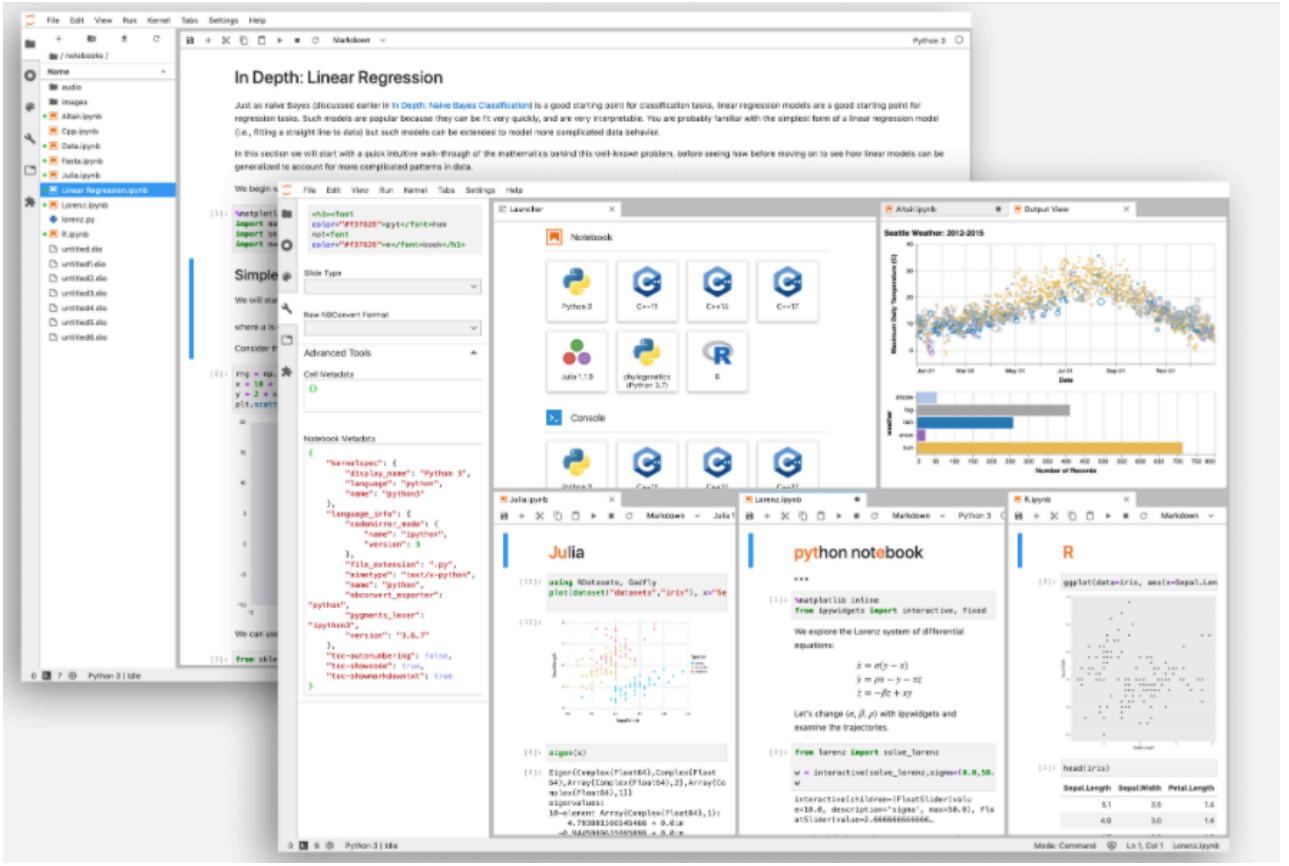
A caveat about file systems

- Be aware of where you launch your notebook service:
- Login nodes and the nodes that run batch scripts have access to the user's home directory, but the compute nodes do not.
- The home directory is where the files that make up the virtual environment are stored by default.
- So if you want to use the virtual environment from a batch script, it either has to run on the batch node (e.g., don't try to run it via a jsrun command) or you will have to figure out how to force conda to store virtual environments in your $MEMBERWORK directory.
- If you launch the notebook from your home dir, you will not be able to run notebooks from your projects directory

Overview of Jupyter Notebooks

What are Jupyter Notebooks?

- Why do we use them?



https://jupyter.org/

Overview of Jupyter Notebooks

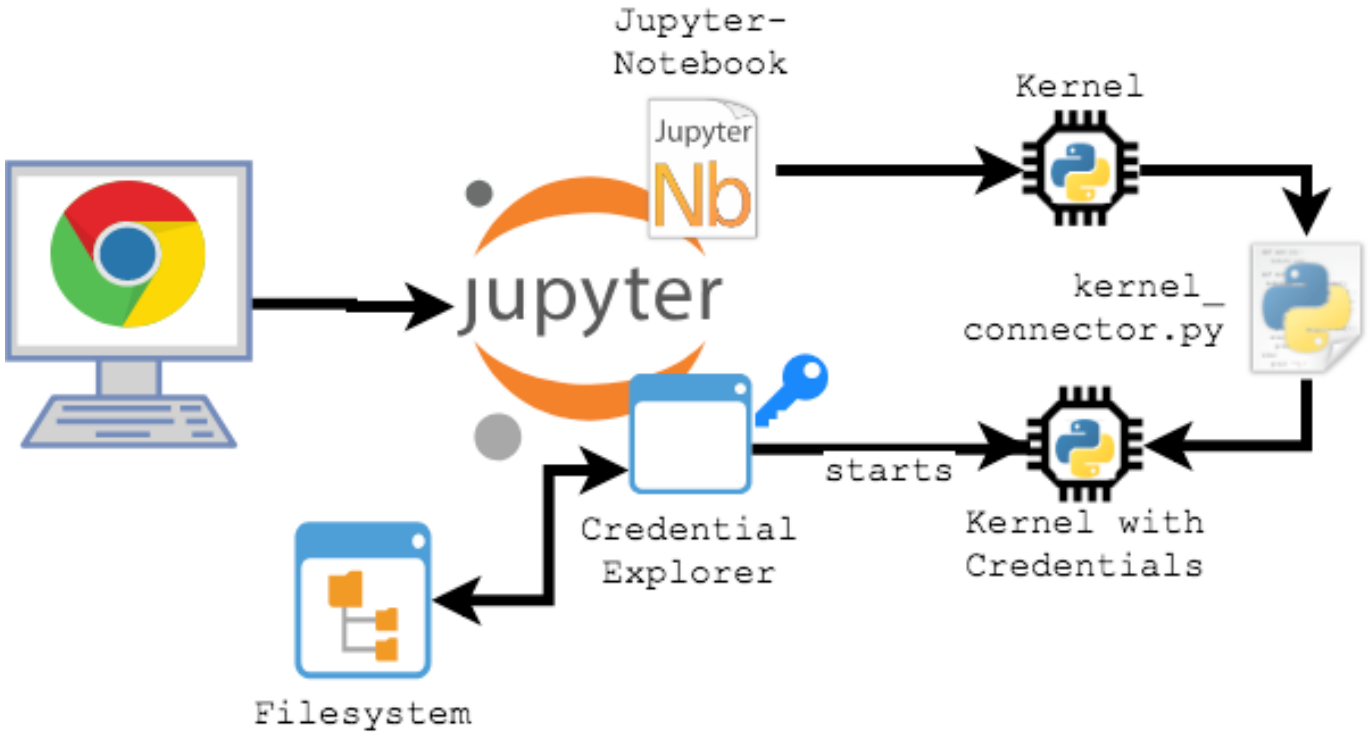
- Community of open-source developers, scientists, educators, and data scientists.
- Goal: build open-source tools and create community that facilitates scientific research, reproducible and open workflows, education, computational narratives, and data analytics.
- Jupyter supports over 100 programming languages, and connects data analytics tools across a range of disciplines and communities.

Source: **https://bids.berkeley.edu/research/project-jupyter**

Jupyter Notebooks

- Web-based interactive computing platform
- Allows users to author computational apps
- code, equations, narrative text, interactive user interfaces, and other rich media.

- Enables collaborative creation of notebooks
- Can be used across a wide range of discipines



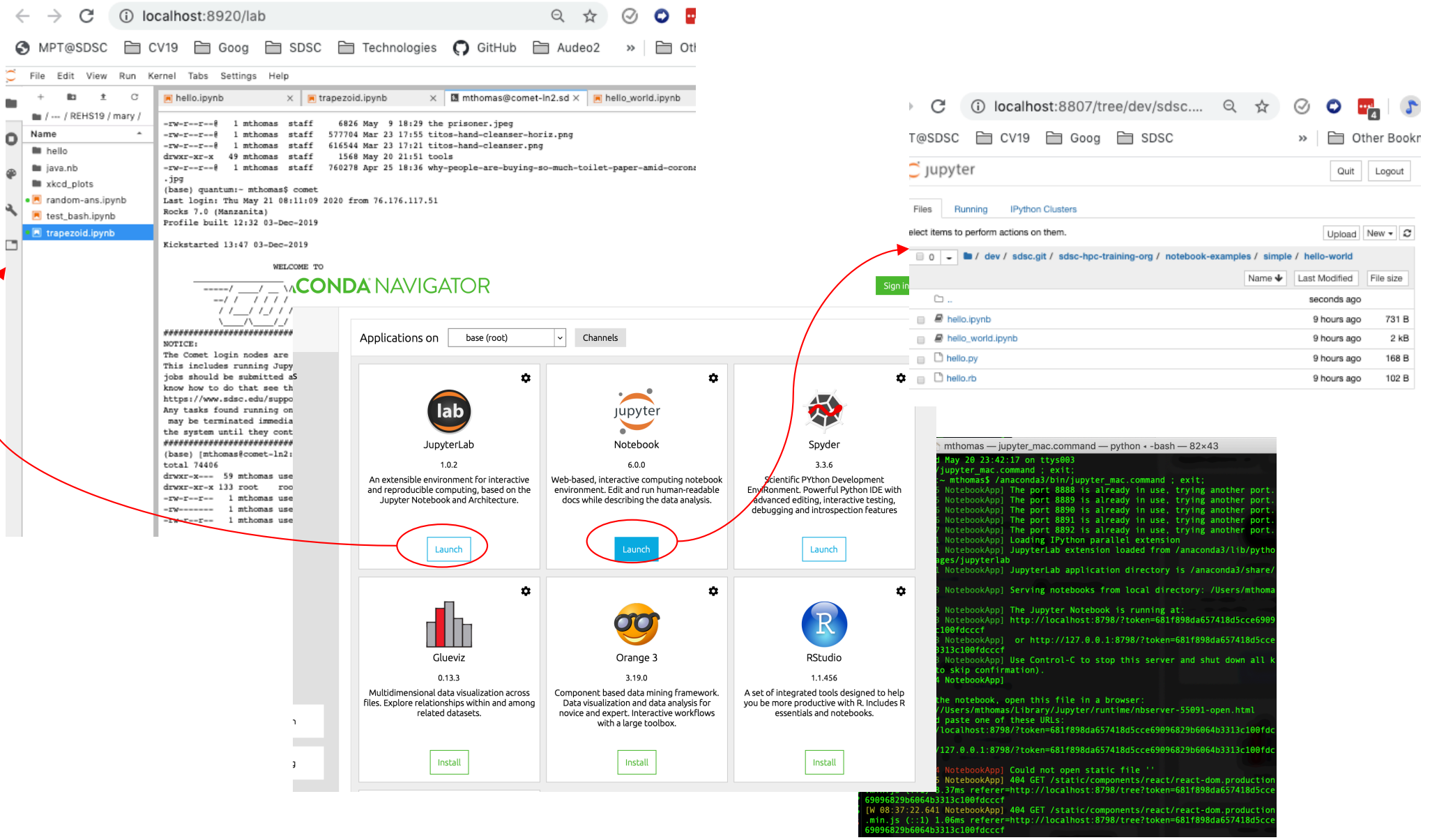
https://towardsdatascience.com/the-jupyterlab-credential-store-9cc3a0b9356

JupyterLab

- Jupyter's next-generation interface, JupyterLab facilitates data scientists to compose the interface that suits their needs.
- Flexible, extensible user interface - supports diversity of workflows in data science.
- Runs using same Jupyter server as Notebook interface → allows it to be accessed remotely on shared infrastructure (for example, via a JupyterHub)

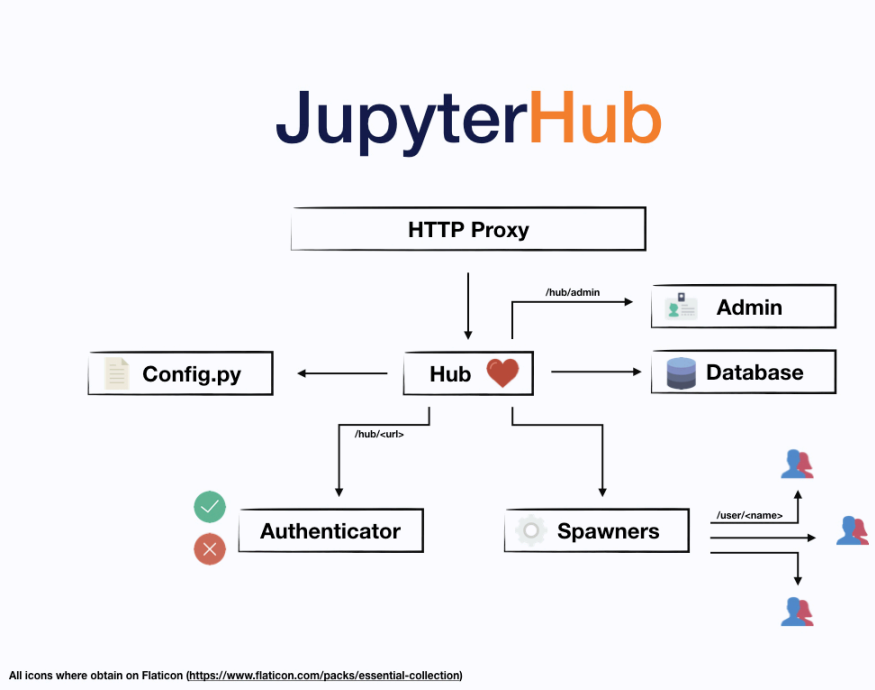
Source: https://bids.berkeley.edu/research/project-jupyter

Jupyter Env - Desktop



JupyterHub

- Provides remote access to Jupyter servers via Web browser.
- Make high-powered computational environments and resources more accessible to students, researchers, and collaborators.
- Runs in the cloud or on your own hardware
- Makes it possible to serve a pre-configured data science environment to any user in the world.
- Used in education and large-scale courses as well as in collaborative and massively-open data analytics projects.



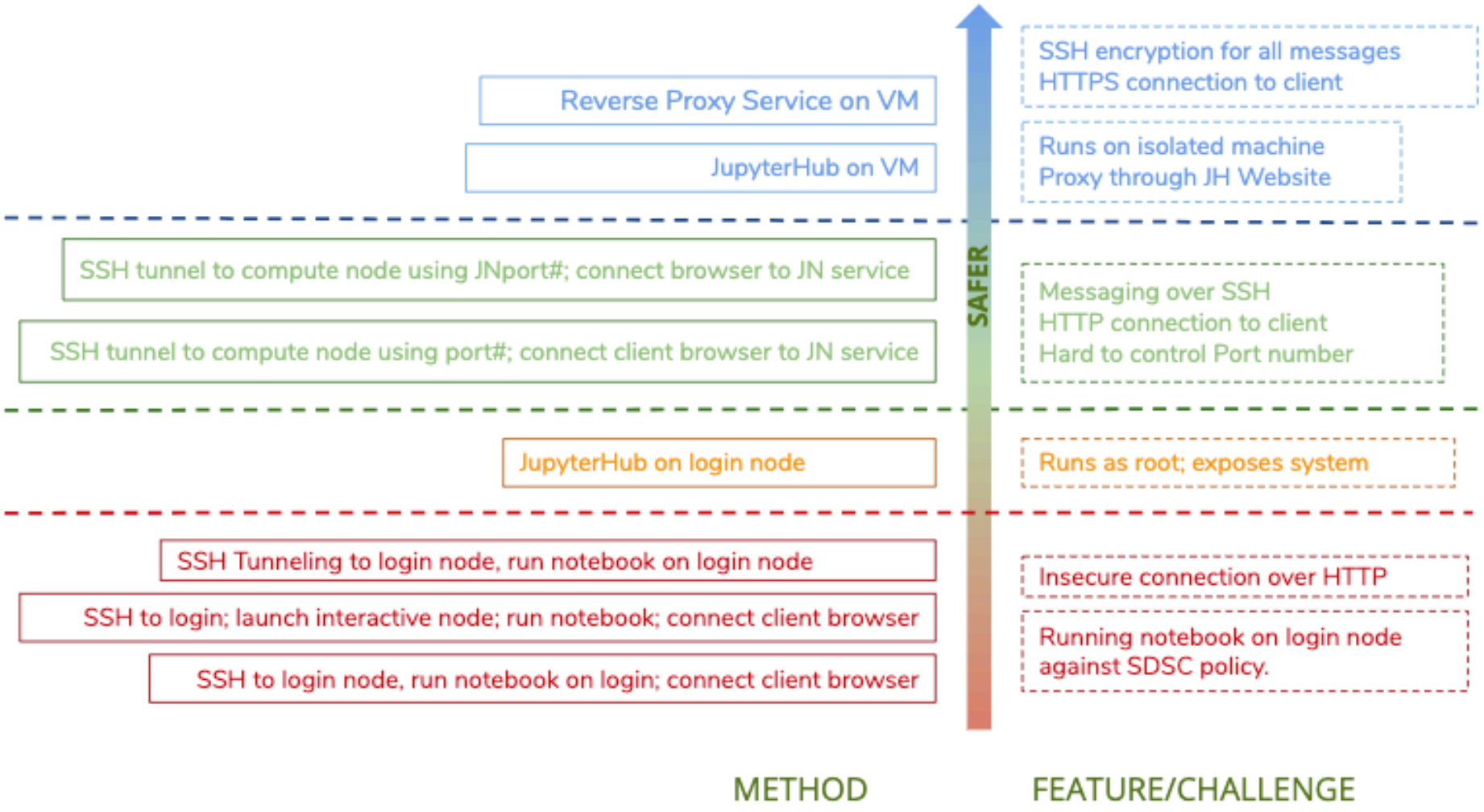
https://jupyterhub.readthedocs.io/en/stable/

Jupyter Notebook Security

Not All Methods are Secure

- Notebooks on Comet/Level of security
- Security concerns
  - HTTP vs HTTPS
  - SSH vs SSH tunneling (HTTP)
- Most insecure method: HTTP (public IP)
  - Next levels of security: tunneling
  - Mention Jupyter Hub - somewhat more secure (out of the box)???
  - More secure - Reverse Proxy

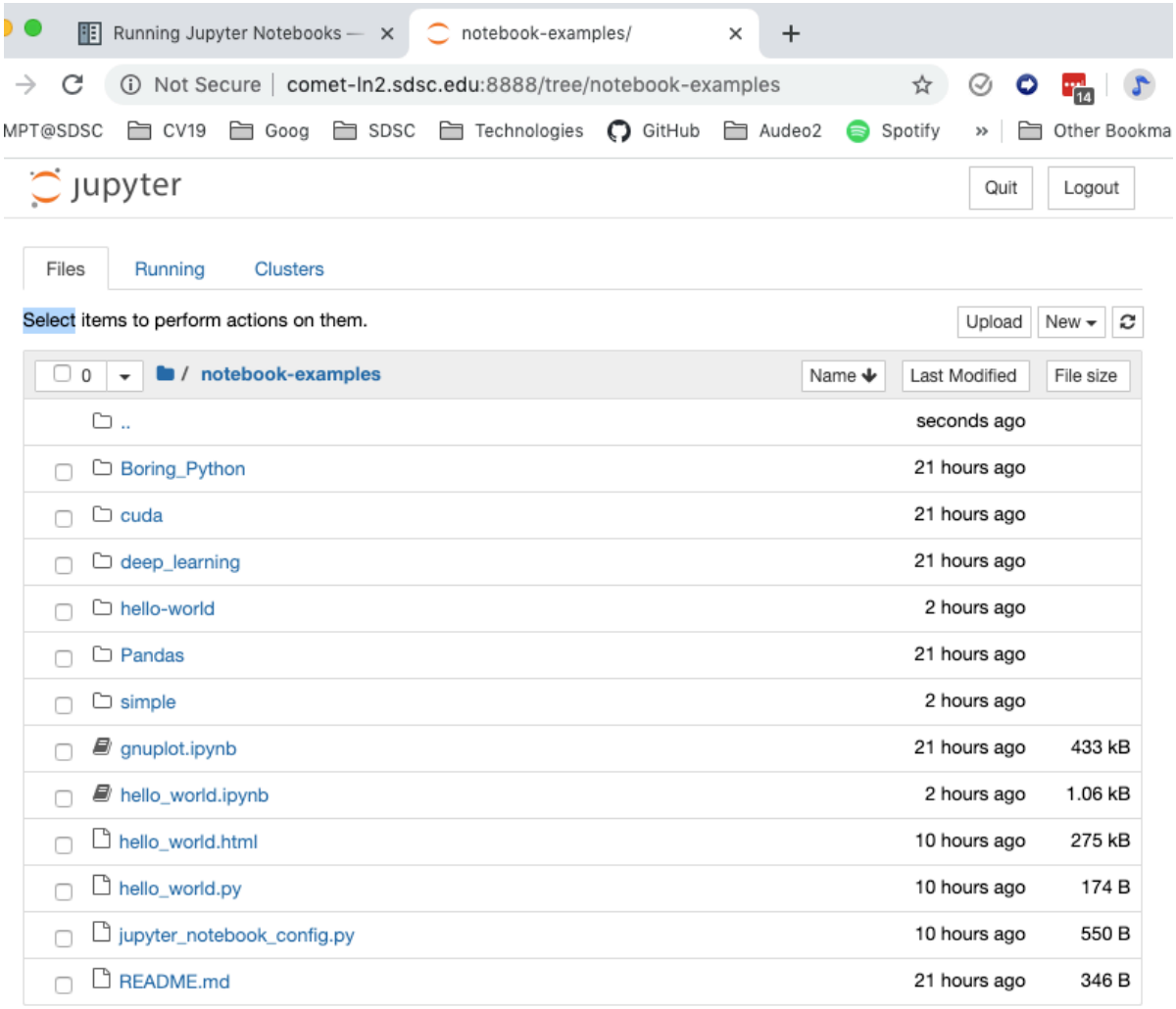
Methods for Running Notebooks



Key Vulnerability:
Notebooks Provide Access to HPC File Systems

SDSC Jupyter Services Policy:

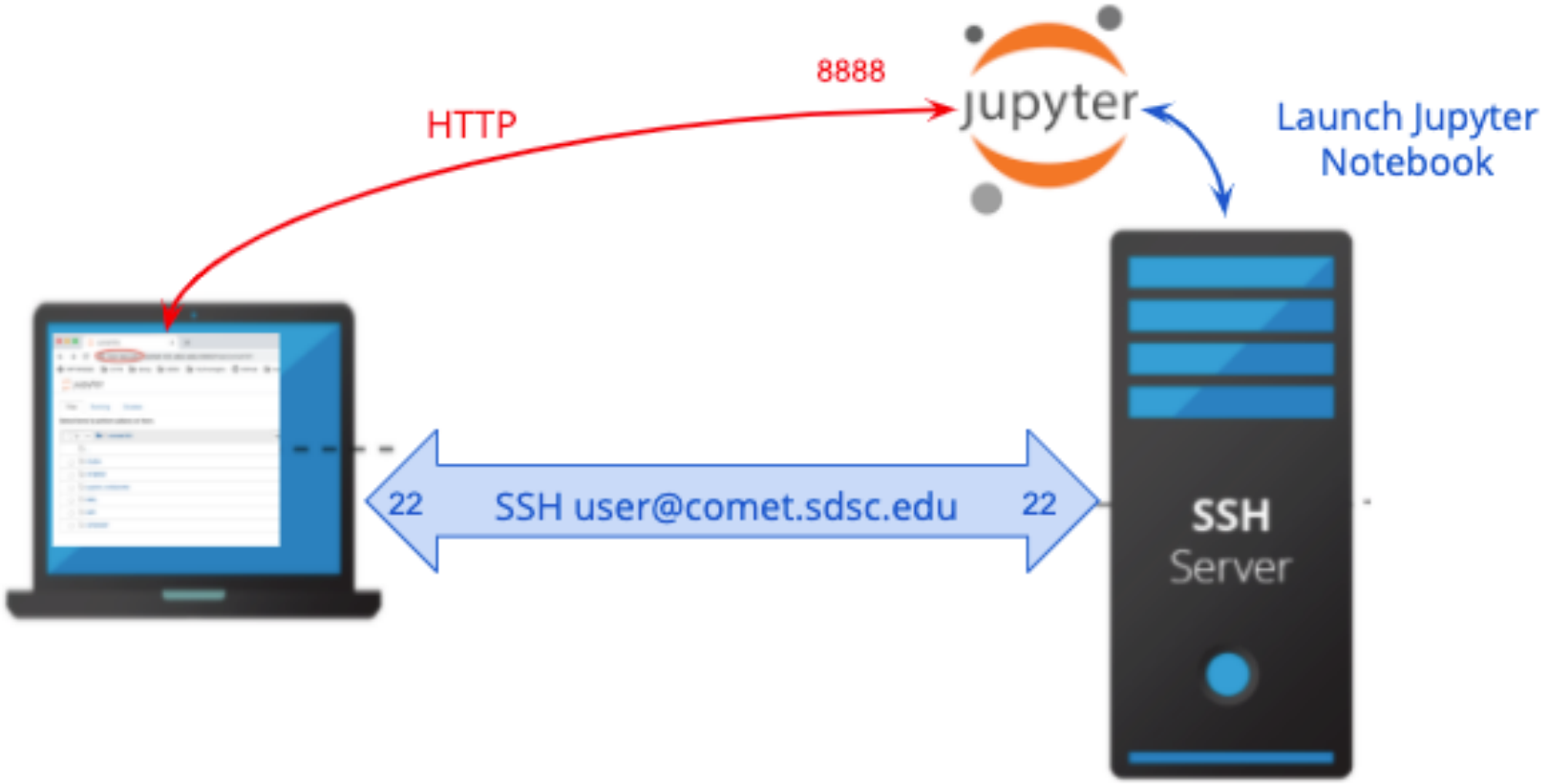
- Portals, JupyterHub, and other services cannot be mounted directly to disk (must be on VM)
  - Many use root in vulnerable ways
  - If a user launches Jupyter Lab or Notebooks, the jobs will be killed.
- No applications can run on login nodes
- SDSC recommendation:
  - use secure connections: when you choose unsecure connections your account is vulnerable to hacking



Methods for Running Notebooks on Comet

- Connection scenarios:
  - Connection to Notebook over HTTP (very insecure)
  - Connection to Notebook over SSH tunneling (secure)
  - Connection to Notebook over HTTPS using the Jupyter Reverse Proxy Service (very secure)
  - Coming Soon: Galyleo remote notebook launcher
- Notebooks can be run on the following nodes:
  - Login node
  - Interactive node
  - Compute node
  - GPU node

Why Connection over HTTP (unsecure)



Improve Security: SSH Tunneling

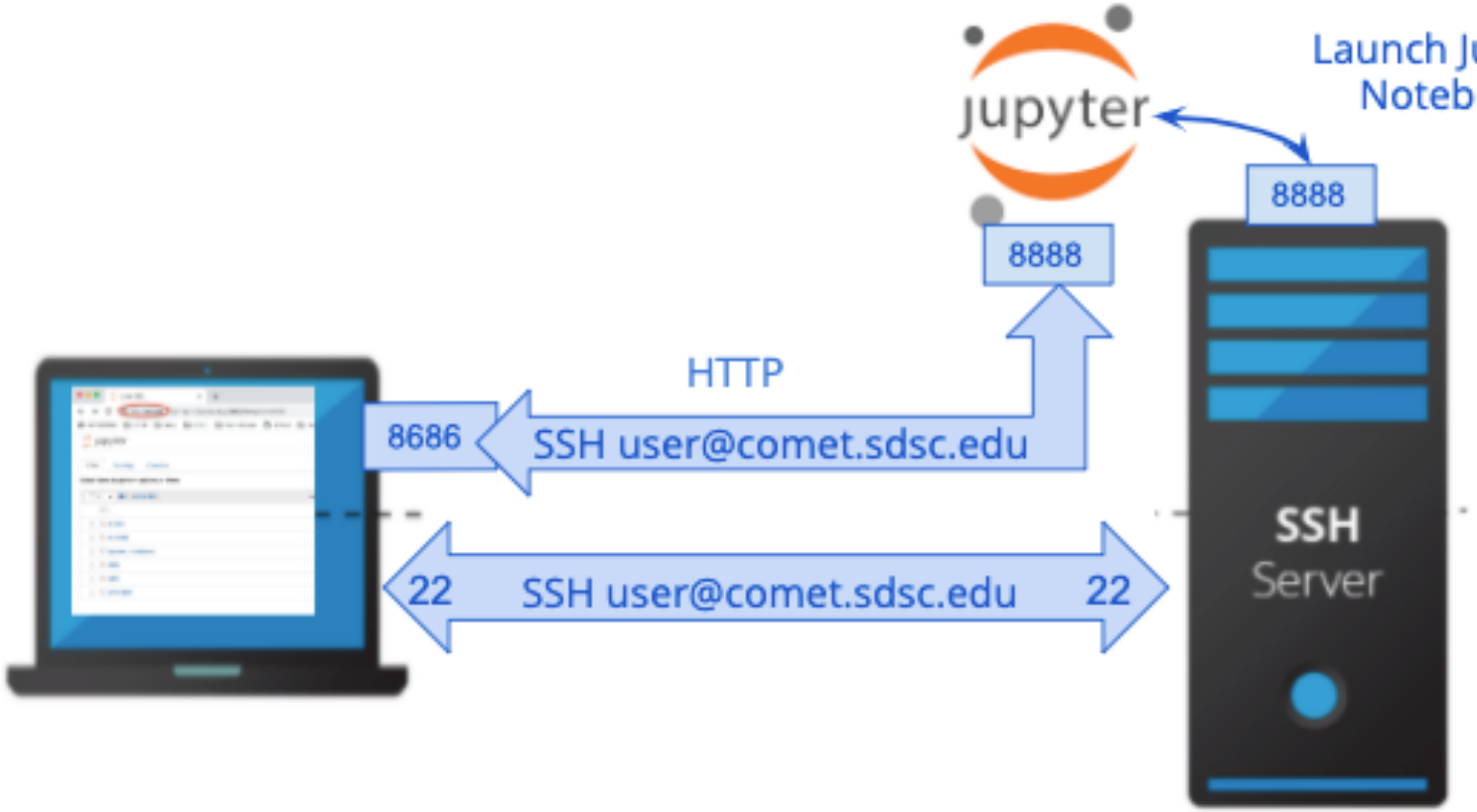
See: https://comet-notebooks-101.readthedocs.io/en/latest/methods/tunneling.html

- Port forwarding via **SSH tunneling** creates a secure connection between a local computer and a remote machine through which services can be relayed.
- Connections are encrypted
- Useful for transmitting information that uses an unencrypted protocol (IMAP, VNC, HTTP server).
- 3 Types:
  - **Local port forwarding (will use for notebook servers)**: connections *from SSH client* are forwarded *via the SSH server*, then *to a destination server.*
  - **Remote port forwarding**: connections *from the SSH server* are forwarded *via the SSH client*, then *to a destination server*
  - **Dynamic port forwarding**: connections from *programs* forwarded *via the SSH client*, then *via the SSH server*, and finally *to destination servers.*

Source: https://help.ubuntu.com/community/SSH/OpenSSH/PortForwarding

Secure Connection over SSH Tunneling

Uses Local Port Forwarding to connect to a Jupyter Notebook Server



Very secure but somewhat complicated and hard to keep running

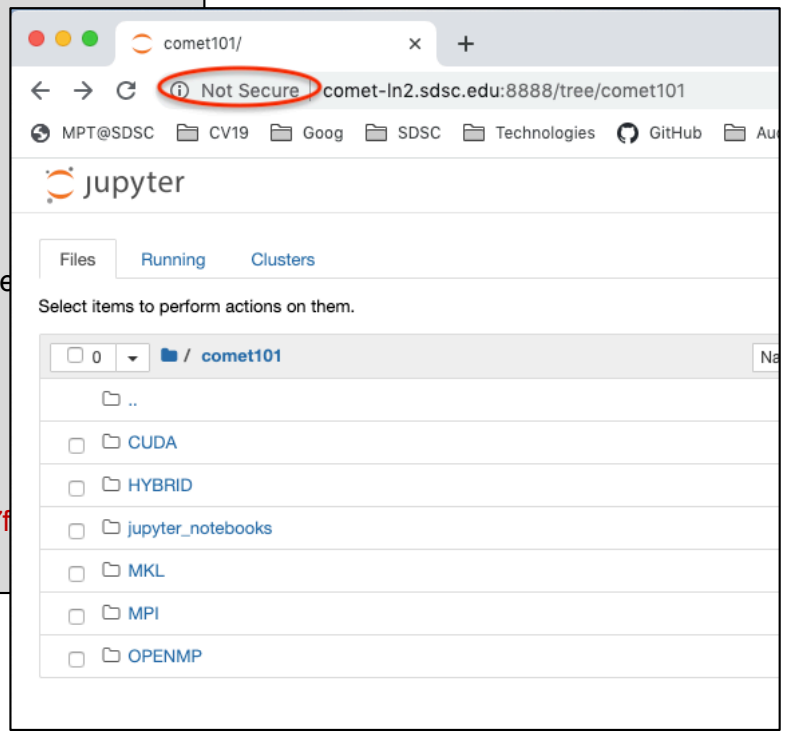
SSH Tunneling @ Work:

Uses Local Port Forwarding to connect to a Jupyter Notebook Server

```
(base) quantum:Docs username$ ssh -L 8888:127.0.0.1:8888 username@comet.sdsc.edu
```

```
(base) [username@comet-ln2:~] jupyter notebook --no-browser --ip=`/bin/hostname`
[I 12:03:54.005 NotebookApp] JupyterLab extension loaded from
/home/username/miniconda3/lib/python3.7/site-packages/jupyterlab
[I 12:03:54.005 NotebookApp] JupyterLab application directory is
/home/username/miniconda3/share/jupyter/lab
[I 12:03:54.497 NotebookApp] Serving notebooks from local directory: /home/username
[I 12:03:54.497 NotebookApp] The Jupyter Notebook is running at:
[I 12:03:54.498 NotebookApp] http://comet-
ln2.sdsc.edu:8888/?token=bc1a7238d7dd6d401cd099a7e863d5bfb6db8a6a7f19a243
[I 12:03:54.498 NotebookApp]  or
http://127.0.0.1:8888/?token=bc1a7238d7dd6d401cd099a7e863d5bfb6db8a6a7f19a243
[I 12:03:54.498 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice
confirmation).
[C 12:03:54.505 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/username/.local/share/jupyter/runtime/nbserver-650-open.html
    Or copy and paste one of these URLs:
        http://comet-ln2.sdsc.edu:8888/?token=bc1a7238d7dd6d401cd099a7e863d5bfb6db8a6a7f
     or http://127.0.0.1:8888/?token=bc1a7238d7dd6d401cd099a7e863d5bfb6db8a6a7f19a243
```



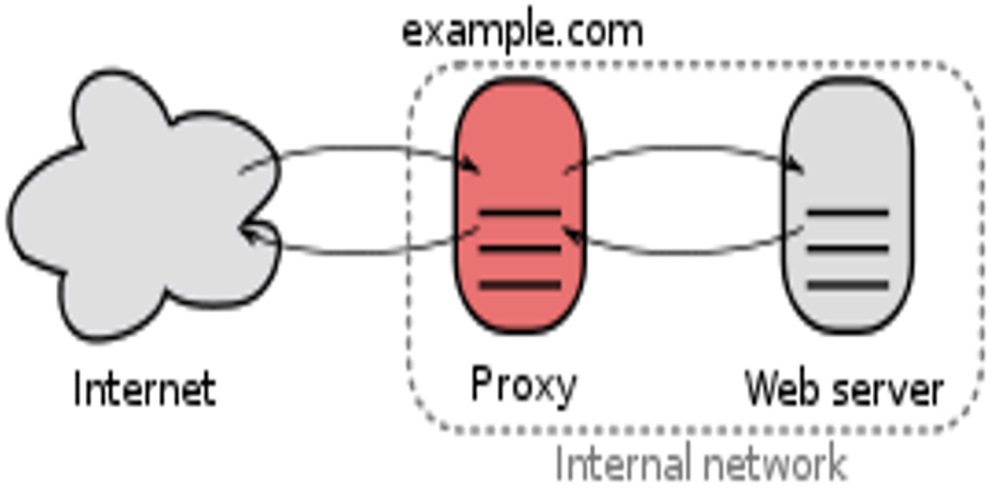
SDSC Reverse Proxy Service (RPS) (beta testing)

https://comet-notebooks-101.readthedocs.io/en/latest/methods/reverseProxy.html

https://github.com/sdsc-hpc-training-org/reverse-proxy

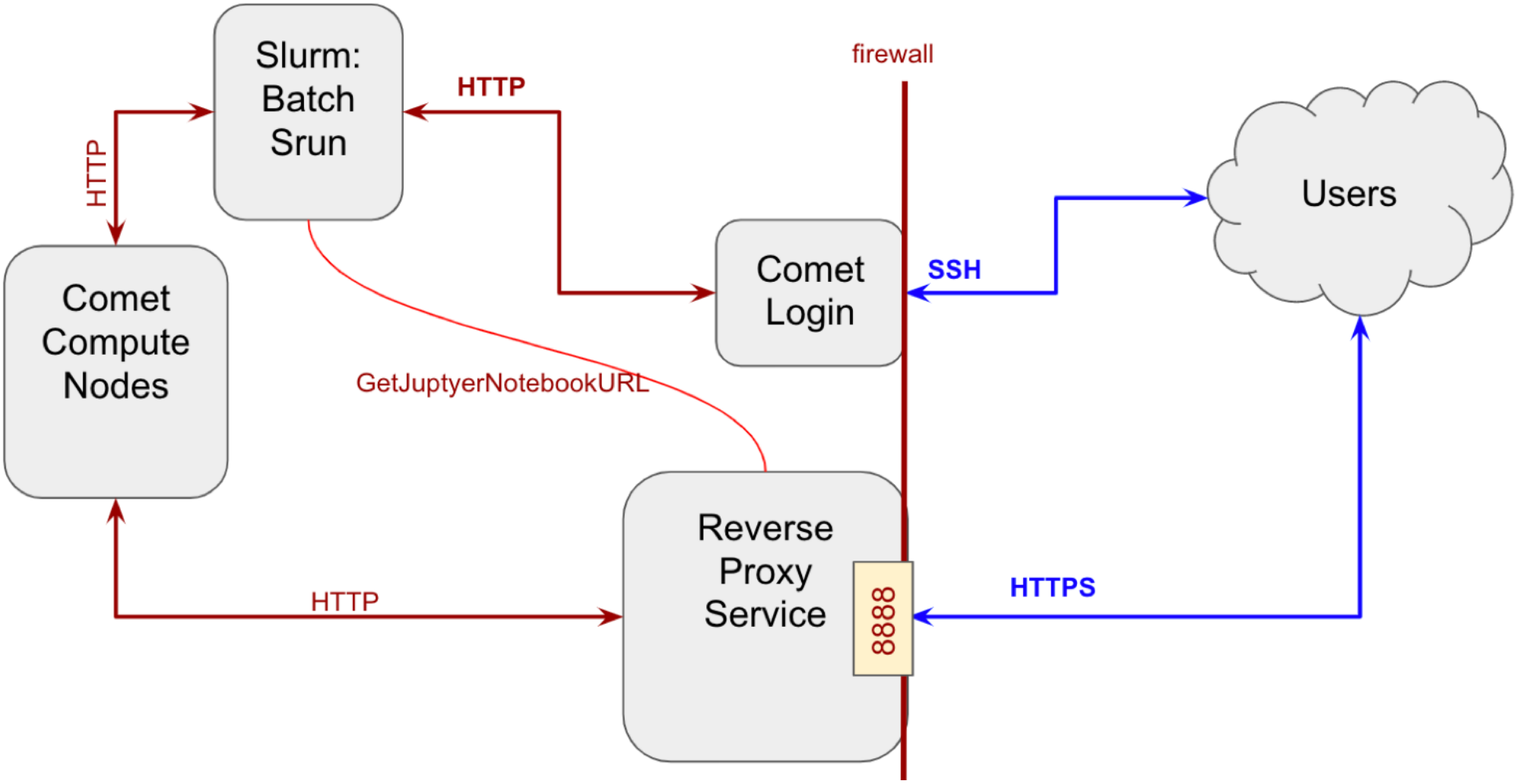
What is a Reverse Proxy?

- A reverse proxy takes requests from the Internet and forwards them to servers in an internal network. Those making requests to the proxy may not be aware of the internal network.



Img Source: [Wikipedia reverse proxy]

RPS Architecture



SDSC Reverse Proxy Service Overview

- RPS is a prototype system that will allow users to launch standard Jupyter Notebooks on on any Comet compute node using a reverse proxy server.
- The notebooks will be hosted on the internal cluster network as an HTTP service using standard jupyter commands.
- The service will then be made available to the user outside of the cluster firewall as an HTTPS connection between the external users web browser and the reverse proxy server.
- The goal is to minimize software changes for our users while improving the security of user notebooks running on our HPC systems.
- The RP service is capable of running on any HPC system capable of supporting the RP server (needs Apache)

SDSC Reverse Proxy Service Overview

- Using RPS is very simple and requires no tunneling and is secure (produces HTTPS URLs).
- To use RPS:
  - SSH to a comet login node.
  - Clone the Repo:
    git clone https://github.com/sdsc-hpc-training-org/reverse-proxy.git
  - Check your software environment on the login node: Anaconda, conda, Jupyter (notebooks, lab), and other Python packages needed for you application.
    - See: https://comet-notebooks-101.readthedocs.io/en/latest/prerequisites.html

SDSC Reverse Proxy Service

- Project Team:
  - Scott Sakai (SDSC)
  - Marty Kandes (SDSC)
  - Mary Thomas (SDSC)
  - James McDougall (UCSD Undergraduate)

- Project Status:
  - RPS is in beta testing.
  - Please give it a try. If you have trouble, help@xsede.org
  - Send feedback to mthomas at ucsd dot edu. to

Live Demo

by

James McDougall

https://comet-notebooks-101.readthedocs.io/en/latest/methods/reverseProxy.html

https://github.com/sdsc-hpc-training-org/reverse-proxy