

SDSC HPC-DSI 2023: Interactive Computing

August 7, 2023

Mary Thomas
SDSC

EXPANSE
COMPUTING WITHOUT BOUNDARIES

EXPANSE

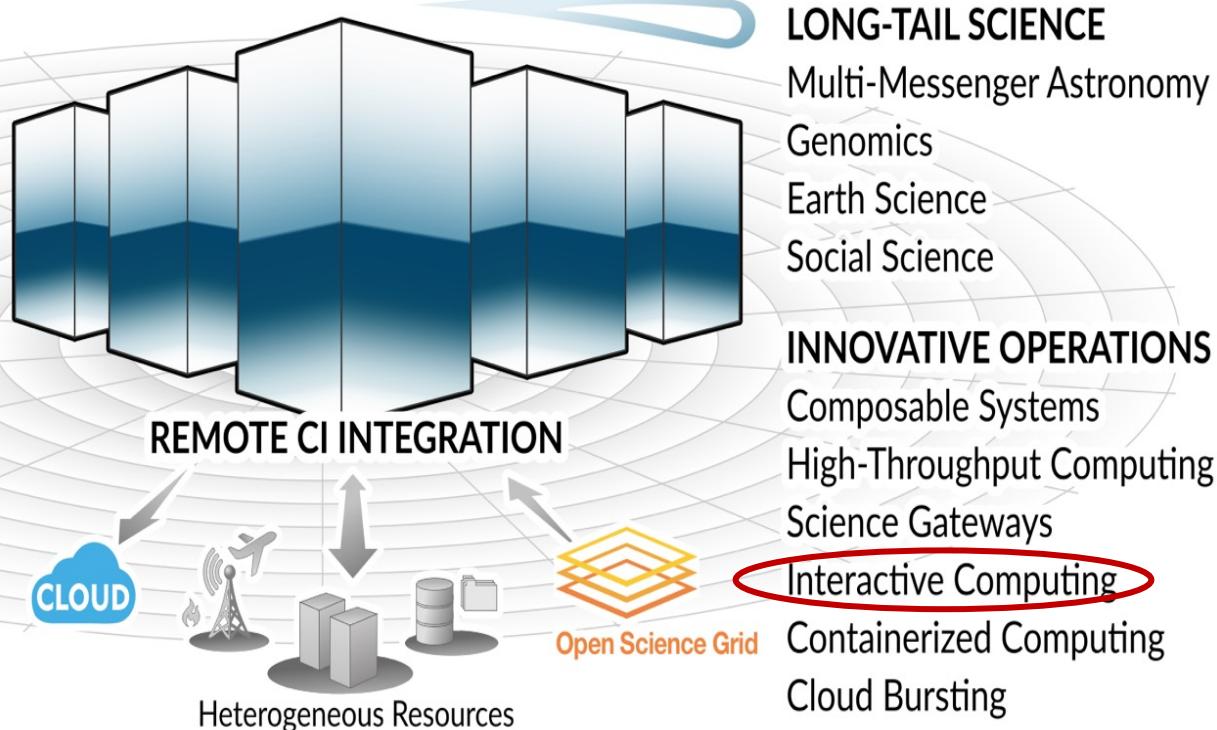
COMPUTING WITHOUT BOUNDARIES
5 PETAFLOP/S HPC and DATA RESOURCE

HPC RESOURCE

13 Scalable Compute Units
728 Standard Compute Nodes
52 GPU Nodes: 208 GPUs
4 Large Memory Nodes

DATA CENTRIC ARCHITECTURE

12PB Perf. Storage: 140GB/s, 200k IOPS
Fast I/O Node-Local NVMe Storage
7PB Ceph Object Storage
High-Performance R&E Networking



For more details see the Expanse user guide @ https://www.sdsc.edu/support/user_guides/expanse.html
and the "Introduction to Expanse" webinar @ https://www.sdsc.edu/event_items/202006_Introduction_to_Expanse.html

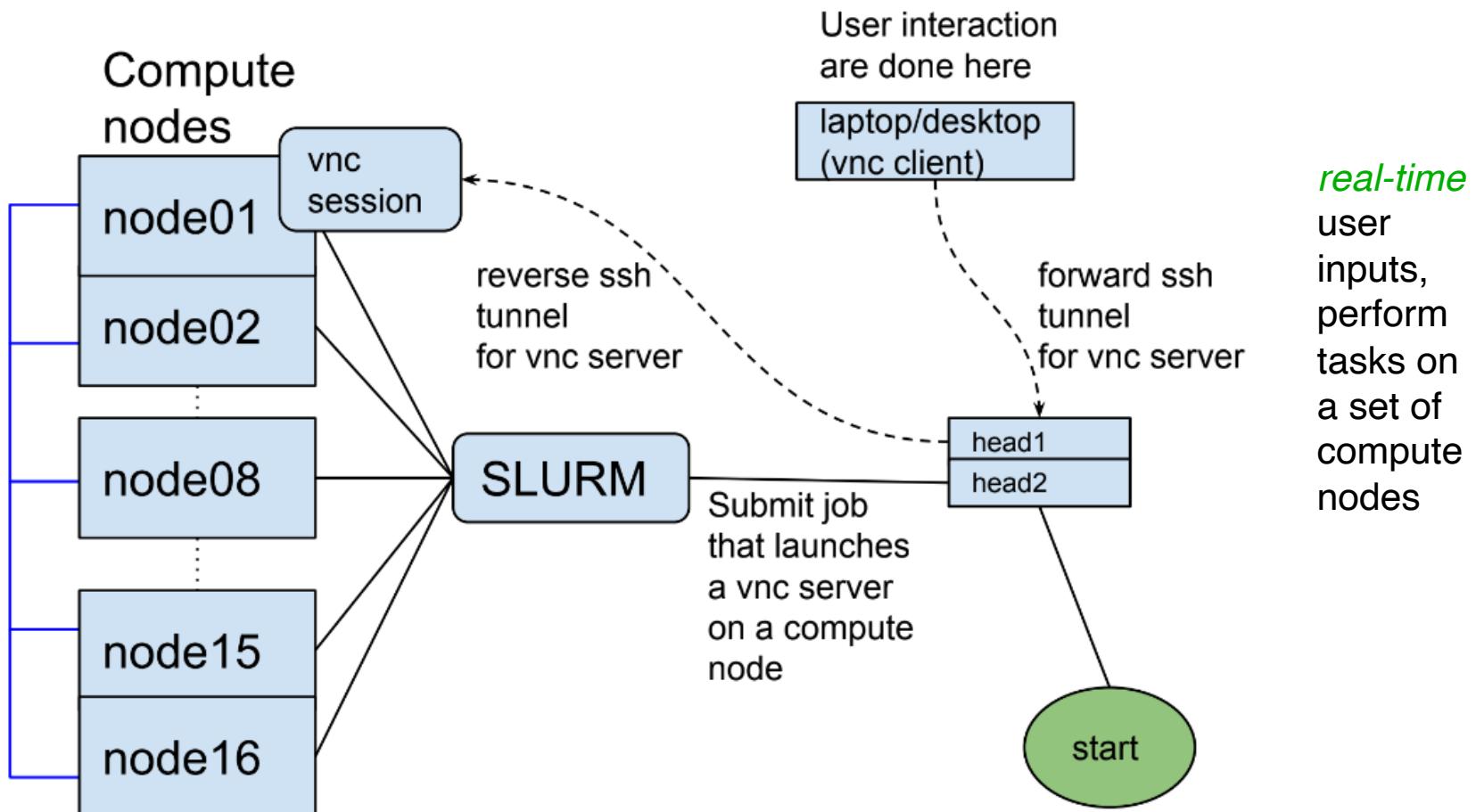
Outline

- What is Interactive High-Performance Computing?
- Accessing Interactive HPC Nodes
- Running Interactive Apps
- Running Interactive Apps Using the Expanse Portal
- Notebook Security

What is Interactive HPC Computing

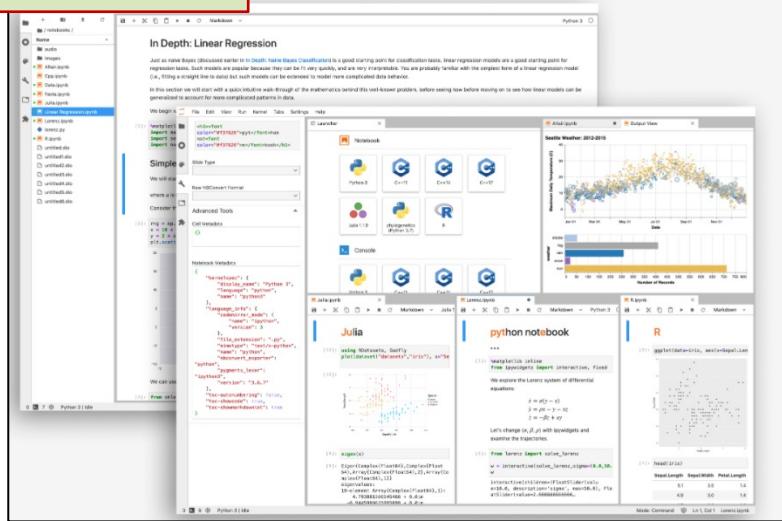
- In **computer science**, **interactive computing** refers to software which accepts input from the user as it runs.
 - **Interactive** software includes commonly used programs, such as word processors or spreadsheet applications.
- **Interactive HPC computing** involves *real-time* user inputs to perform tasks on a set of compute node(s) including:
 - Code development, real-time data exploration, and visualizations.
 - Used when applications have large data sets or are too large to download to local device, software is difficult install, etc.
 - User inputs come via command line interface or application GUI (Jupyter Notebooks, Matlab, R-studio).
 - Actions performed on remote compute nodes as a result of user input or program out.

Real-time User Interactions



Interactive HPC Computing

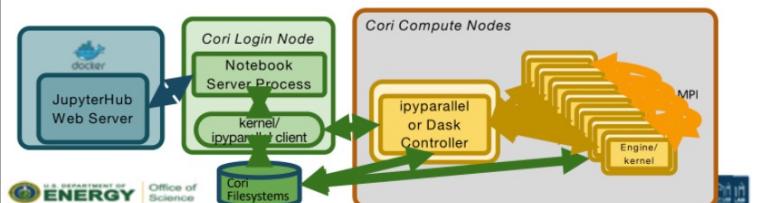
<https://jupyter.org/>



Interactive Distributed Computing with Jupyter (NERSC)

Jupyter architecture

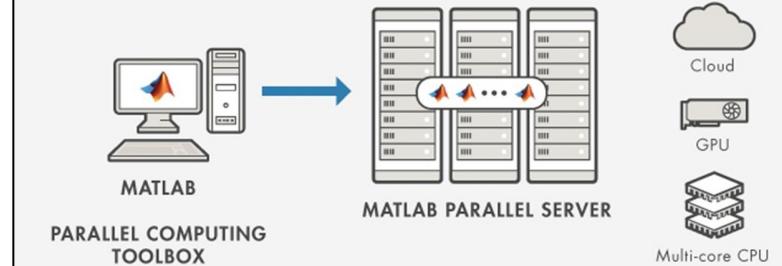
- Allocate nodes on Cori interactive queue and start ipyparallel or Dask cluster
 - Developed %ipcluster magic to setup within notebook
- Compute nodes traditionally do not have external address
 - Required network configuration / policy decisions
- Distributed training communication is via MPI Horovod or Cray ML Plugin



<https://drive.google.com/file/d/1-OFjrk1q3L1d3uakr2xkozrPn2c2VZpZ/view>

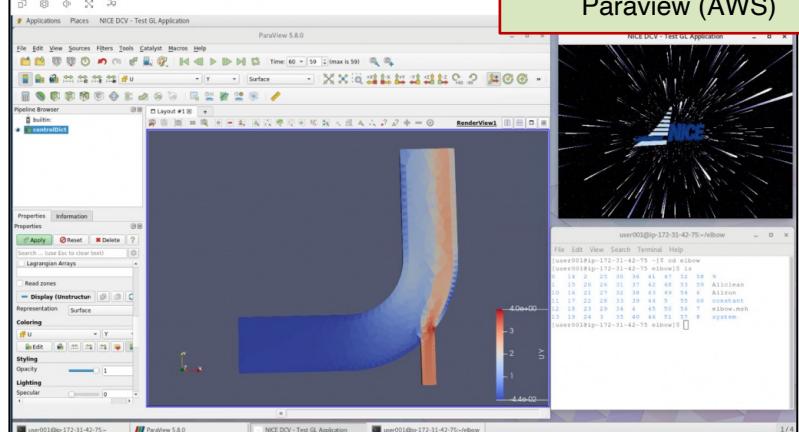
Parallel Matlab (AWS)

```
>> parpool(parcluster('HPC1'),100);
>> parfor i=1:3000
>> c(i,:) = eig(rand(1000));
>> end
```



<https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mathworks-inc.matlab-parallel-server-listing?tab=Overview>

Paraview (AWS)



<https://aws.amazon.com/blogs/compute/how-to-run-3d-interactive-applications-with-nice-dcv-in-aws-batch/>

Outline

- What is Interactive High-Performance Computing?
- Accessing Interactive HPC Nodes
- Running Interactive Apps
- Running Interactive Apps Using the Expanse Portal
- Notebook Security

Accessing Interactive Compute Nodes on Expanse

- Connect via terminal using SSH → secure connections
- Use the *srun* command to obtain nodes for ‘live,’ command line interactive access:

CPU	<code>srun --partition=debug --pty --account=use300 --nodes=1 --ntasks-per-node=24 --mem=8G -t 00:30:00 --wait=0 --export=ALL /bin/bash</code>
GPU	<code>srun --partition=gpu-debug --pty --account=use300 --ntasks-per-node=10 --nodes=1 --mem=96G --gpus=1 -t 00:30:00 --wait=0 --export=ALL /bin/bash</code>

(Tested 7/27/22)

Using An Interactive CPU node

```
[mthomas@login01 calc-prime]$ srun --partition=compute --pty -  
-account=use300 --nodes=1 --ntasks-per-node=24 --mem=8G -t  
00:30:00 --wait=0 --export=ALL /bin/bash  
srun: job 24457429 has been allocated resources  
[mthomas@exp-9-55 calc-prime]$ module purge  
[mthomas@exp-9-55 calc-prime]$ module load slurm  
[mthomas@exp-9-55 calc-prime]$ module load cpu  
[mthomas@exp-9-55 calc-prime]$ module load gcc/10.2.0  
[mthomas@exp-9-55 calc-prime]$ module load openmpi/4.1.1  
[mthomas@exp-9-55 calc-prime]$ mpirun -n 64 ./mpi_prime  
06 August 2023 11:10:26 PM  
PRIME_MPI n_hi= 5000000 C/MPI version  
An MPI example program to count the number of primes: #  
processes is 64  


| N       | Pi     | Time      |
|---------|--------|-----------|
| 1       | 0      | 0.013258  |
| 2       | 1      | 0.001058  |
| 4       | 2      | 0.000101  |
| 8       | 4      | 0.000101  |
| [SNIP]  |        |           |
| 131072  | 12251  | 0.110848  |
| 262144  | 23000  | 0.410792  |
| 524288  | 43390  | 1.527210  |
| 1048576 | 82025  | 5.733612  |
| 2097152 | 155611 | 21.725862 |

  
PRIME_MPI - Master process: Normal end of execution.  
06 August 2023 11:12:26 PM
```

Request an interactive node
for 30 minutes

- Exit interactive session when your work is done or you will be charged more CPU time.
- Beware of oversubscribing your job: don't ask for more cores than you have requested.
- Intel compiler allows this, but your performance will be degraded.

Using Interactive GPU nodes

```
[snip]
```

```
Last login: Fri Feb 18 12:58:32 2022 from 76.176.117.51
```

```
[username@login02 ~]$
```

```
[username@login02 ~]$ srun --partition=gpu-debug --pty --account=use300 --ntasks-per-node=10 --nodes=1 --mem=96G --gpus=1 -t 00:30:00 --wait=0 --export=ALL /bin/bash
```

```
srun: job 9794018 queued and waiting for resources
```

```
srun: job 9794018 has been allocated resources
```

```
[mthomas@exp-14-57 ~]$
```

```
[mthomas@exp-14-57 ~]$ nvidia-smi
```

```
Fri Feb 18 13:04:19 2022
```

```
+-----+-----+-----+
| NVIDIA-SMI 460.32.03     Driver Version: 460.32.03     CUDA Version: 11.2 |
+-----+-----+-----+
| GPU  Name      Persistence-M | Bus-Id      Disp.A  | Volatile Uncorr. ECC |
| Fan  Temp     Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M.  |
|                               |             |            MIG M.   |
+-----+-----+-----+
| 0  Tesla V100-SXM2...  On     | 00000000:86:00.0 Off    |           0 |
| N/A   34C     P0    41W / 300W |             0MiB / 32510MiB | 0%       Default |
|                               |                           N/A      |
+-----+-----+-----+
```

```
+-----+
| Processes:
| GPU  GI  CI          PID  Type  Process name          GPU Memory |
| ID   ID              ID   ID   |                    Usage  |
+-----+
| No running processes found |
+-----+
```

```
[username@login02 ~]$ exit
```

Request an interactive node
for 30 minutes

Verify you are on a GPU node

Exit when tasks are done

Outline

- What is Interactive High-Performance Computing?
- Accessing Interactive HPC Nodes
- Running Interactive Apps
- Running Interactive Apps Using the Expanse Portal
- Notebook Security

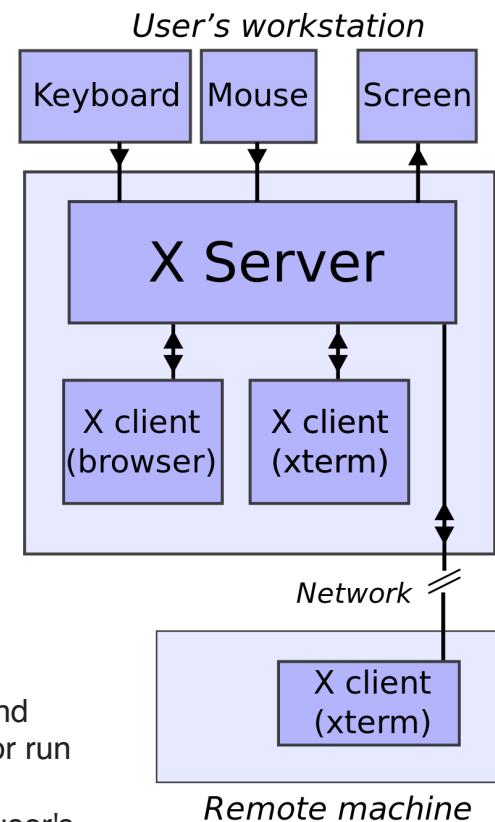
Viewing Data on HPC Cluster (Simple)

- Cat the file contents
- Plot data using gnuplot
- Run Matlab, R
- Run a data viewing app for the data type:
 - NetCDF, HPF, other
- Use a Jupyter Notebook

Visualization Apps: Use X11 Forwarding

- X Window System (X11): windowing system for bitmap displays, common on Unix-like operating systems.
 - MacOS uses Quartz
- X is an architecture-independent system for remote graphical user interfaces and input device capabilities

X server receives input from a local keyboard and mouse and displays to a screen. A web browser and a terminal emulator run on the user's workstation and a terminal emulator runs on a remote computer but IS controlled and monitored from the user's machine



SRC: https://en.wikipedia.org/wiki/X_Window_System

Visualization Apps: Use X11 Forwarding

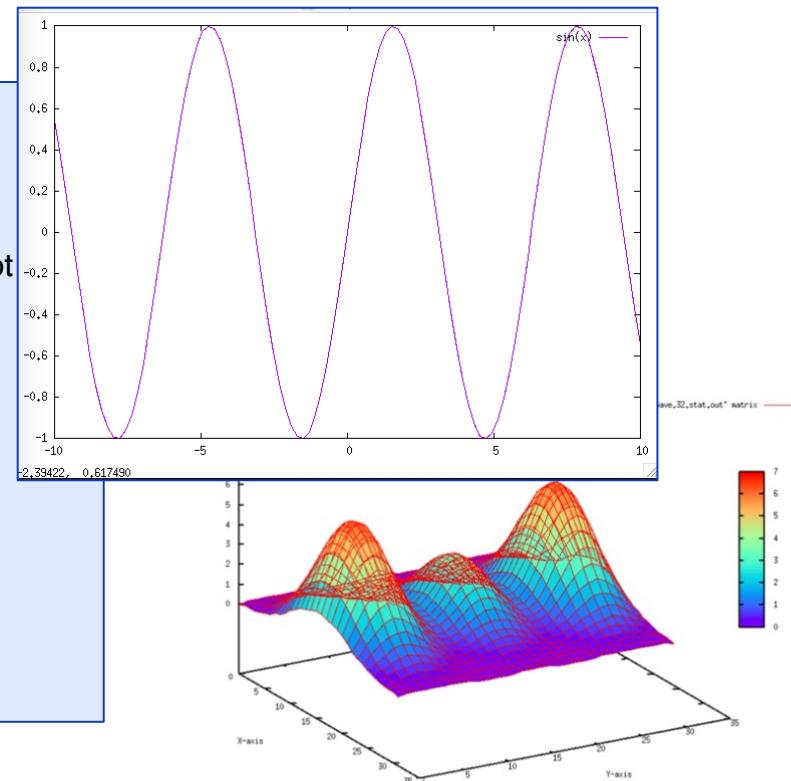
- On MacOS: Install X11 forwarding software:
 - MacOS: Xquartz
- Use the connection command:
`ssh -Y mthomas@login.expanse.sdsc.edu'`

```
[mthomas@home]$ brew install --cask xquartz
Updating Homebrew...
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
[SNIP]
installer: The upgrade was successful.
🍺 xquartz was successfully installed!
quantum:~ mthomas$ which xquartz
/opt/X11/bin/xquartz
```

gnuplot

- *gnuplot* is a command-driven interactive function plotting program. It can be used to plot functions and data points in both two- and three- dimensional plots in many different formats.
- It is designed primarily for the visual display of scientific data.
- Can be run from the gnuplot app or from within your program so you can save visualizations of results.

```
module load cpu/0.15.4 gcc/10.2.0
[mthomas@login01 ~]$ module load gnuplot/5.2.8
[mthomas@login01 ~]$ which gnuplot
/cm/shared/apps/spack/cpu/opt/spack/linux-centos8-zen2/gcc-
10.2.0/gnuplot-5.2.8-uwugzxg4dgxaiciheiepgol67cw7m6yg/bin/gnuplot
[mthomas@login01 gnuplot-ex]$ gnuplot
G N U P L O T
Version 5.2 patchlevel 8  last modified 2019-12-01
Copyright (C) 1986-1993, 1998, 2004, 2007-2019
Thomas Williams, Colin Kelley and many others
gnuplot home:  http://www.gnuplot.info
faq, bugs, etc: type "help FAQ"
immediate help: type "help" (plot window: hit 'h')
Terminal type is now 'x11'
gnuplot> plot sin(x)
gnuplot>
```



Expanse: Matlab from the command line

```
[mthomas@login01 ~]$ srun --partition=gpu-debug --pty --account=use300 --ntasks-per-node=10      --nodes=1 --mem=96G --gpus=1 -t 00:30:00 --wait=0 --export=ALL /bin/bash

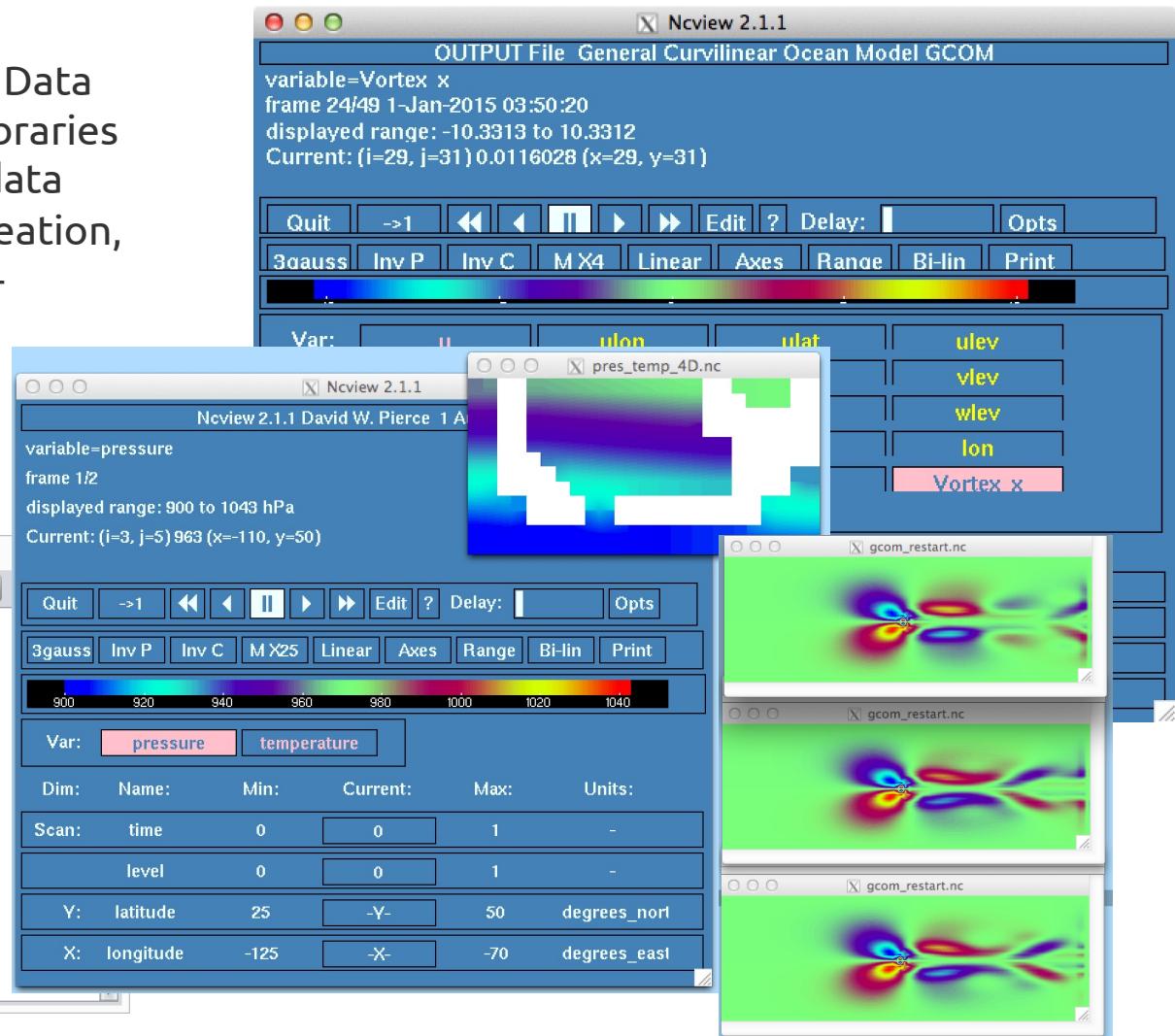
srun: job 14833549 queued and waiting for resources
srun: job 14833549 has been allocated resources
[mthomas@exp-9-55 ~]$ module list
Currently Loaded Modules:
  1) shared    2) slurm/expanse/21.08.8    3) sdsc/1.0    4) DefaultModules    5) cpu/0.15.4    6)
gcc/10.2.0    7) gnuplot/5.2.8
  1) shared    2) slurm/expanse/21.08.8    3) slurm/1.0    4) DefaultModules    5) cpu/0.15.4    6)
gcc/10.2.0    7) gnuplot/5.2.8
[mthomas@exp-9-55 ~]$ module load matlab/2022a
[mthomas@exp-9-55 ~]$ matlab
MATLAB is selecting SOFTWARE OPENGL
rendering.                                              < M A T L A B (R) >
Copyright 1984-2022 The MathWorks, Inc.
R2022a (9.12.0.1884302) 64-bit (glnxa64)
          February 16, 2022
>> a=[1 3 5; 2 4 6; 7 8 10]
a =
  1   3   5
  2   4   6
  7   8  10
>> b=sin(a)
b =
  0.8415  0.1411 -0.9589
  0.9093 -0.7568 -0.2794
  0.6570  0.9894 -0.5440
>> plot(b)
>>
```

Visual version requires OpenGL – run In portal

Viewing NetCDF Files

- NetCDF (Network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data
- NetCDF clients (ncview, ncdump) can be used to query and plot data in real-time

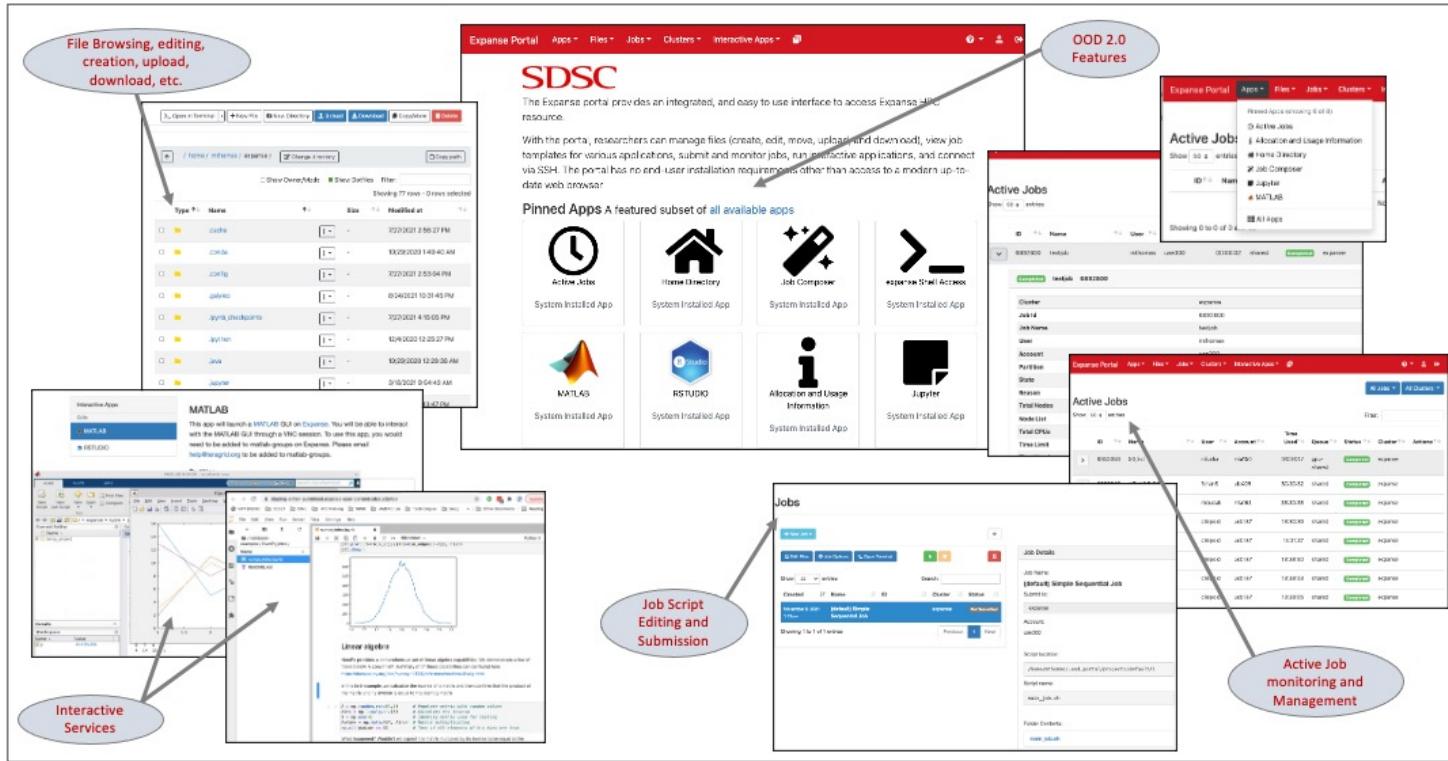
```
NCdump Variable Data  
Variable: humidity(0:9:1, 0:4:1)  
  
float humidity(station=10, time=5);  
:standard_name = "specific_humidity";  
:coordinates = "time lat lon alt";  
  
data:  
  
{  
    {1.0, 2.0, 3.0, 4.0, 5.0},  
    {6.0, 7.0, 8.0, 9.0, 10.0},  
    {11.0, 12.0, 13.0, 14.0, 15.0},  
    {16.0, 17.0, 18.0, 19.0, 20.0},  
    {21.0, 22.0, 23.0, 24.0, 25.0},  
    {26.0, 27.0, 28.0, 29.0, 30.0},  
    {31.0, 32.0, 33.0, 34.0, 35.0},  
    {36.0, 37.0, 38.0, 39.0, 40.0},  
    {41.0, 42.0, 43.0, 44.0, 45.0},  
    {46.0, 47.0, 48.0, 49.0, 50.0}  
}
```



Outline

- What is Interactive High-Performance Computing?
- Accessing Interactive HPC Nodes
- Running Interactive Apps
- Running Interactive Apps Using the Expanse Portal
- Notebook Security

Expanse User Portal



- <https://portal.expanse.sdsc.edu>; access using XSEDE credentials
- Securely hosts batch job submission & monitoring, and interactive applications
- Portal simplifies launching supported interactive applications → manages software dependencies

Expanse Portal: File Management

The image shows the Expanse Portal interface. At the top, there is a toolbar with buttons for 'Open in Terminal' (circled in red), 'New File', 'New Directory', 'Upload', 'Download', and 'Copy/Mov'. Below the toolbar is a breadcrumb navigation bar showing the path: '/ home / mthomas / expanse /' and a 'Change directory' button. There are also checkboxes for 'Show Owner/Mode', 'Show Dotfiles', and a 'Filter' input field. A message indicates 'Showing 28 of 62 rows'. The main area is a table listing files and directories:

Type	Name	Size	Modified at
Folder	classes	-	4/15/2022 3:37:19 PM
Folder	comet-files	-	2/16/2022 6:41:00 PM
Folder	conda-install-tmp	-	10/29/2020 2:27:20 AM
Folder	dev	-	2/16/2022 8:39:49 PM
Folder	galileo-examples	-	7/27/2022 2:27:08 AM
Folder	galileo-repo	-	2/16/2022 6:36:21 PM
Folder	gnuplot-ex	-	8/1/2022 8:50:45 PM
Folder	gpuhack22	-	5/11/2022 3:40:13 PM
Folder	hpctr-examples	-	6/27/2022 1:31:14 PM
Folder	hptrain	-	2/16/2022 8:47:23 PM

A red arrow points from the 'Open in Terminal' button in the toolbar to a terminal window on the right side of the interface. The terminal window shows a session on host 'login.expanse.sdsc.edu' with the initial directory '/home/mthomas/expanse'. The session lists numerous files and their details, such as owner, permissions, size, and modification date. The session ends with the prompt '[mthomas@login01 ~]\$'.

```
Host: login.expanse.sdsc.edu Initial directory: /home/mthomas/expanse
-rw----- 1 mthomas use300 17803 Aug 1 21:33 .viminfo
-rw-r--r-- 1 mthomas use300 36 Jan 27 2022 .vimrc
drwxr----- 2 mthomas use300 4 Aug 1 23:11 .vnc
-rw-r--r-- 1 mthomas use300 173 Oct 7 2020 .wget-hsts
-rw-r--r-- 1 mthomas use300 124 Oct 7 2020 README.txt
drwxr-xr-x 4 mthomas use300 4 Apr 15 15:37 classes
drwxr-xr-x 2 mthomas use300 6 Feb 16 18:41 comet-files
-rw-r--r-- 1 mthomas use300 116 Mar 4 2021 conda-activate.txt
drwxr-xr-x 2 mthomas use300 4 Oct 29 2020 conda-install-tmp
drwxr-xr-x 8 mthomas use300 8 Feb 16 20:39 dev
-rw-r--r-- 1 mthomas use300 12266 Nov 8 2021 ex.cl.cmds
drwxr-xr-x 3 mthomas use300 8 Jul 27 02:27 galileo-examples
drwxr-xr-x 5 mthomas use300 9 Feb 16 18:36 galileo-repo
drwxr-xr-x 2 mthomas use300 10 Aug 1 20:50 gnuplot-ex
drwxr-xr-x 3 mthomas use300 3 May 11 15:40 gpuhack22
drwxr-xr-x 13 mthomas use300 16 Jun 27 13:31 hpctr-examples
drwxr-xr-x 5 mthomas use300 6 Feb 16 20:47 hpctrain
drwxr-xr-x 2 mthomas use300 8 Jul 27 02:33 interactive.ex
drwxr-xr-x 2 mthomas use300 3 Aug 1 21:33 matlab-ex
drwxr-xr-x 24 mthomas use300 27 Jul 27 2021 miniconda3
drwxr-xr-x 3 mthomas use300 5 Jul 27 2021 ml-dev-mary
-rwx----- 1 mthomas use300 235 Jun 1 2021 modules.cpu.txt
-rwx----- 1 mthomas use300 84 Feb 8 2021 modules.gpu.txt
-rw-r--r-- 1 mthomas use300 6178 Mar 4 2021 modules.marty.ex.txt
drwxr-xr-x 3 mthomas use300 4 Jul 18 16:09 nn-pde-TEST
drwxr-xr-x 10 mthomas use300 13 Jul 28 2021 notebook-examples
drwxr-xr-x 22 mthomas use300 26 Jul 20 2021 notebook-examples-dev
-rwxr-xr-x 1 mthomas use300 234277552 Aug 1 20:04 ocean_his.nc.gz
drwxr-xr-x 9 mthomas use300 19 Jul 28 2021 reverse-proxy
drwxr-xr-x 2 mthomas use300 7 Feb 16 18:44 scc21
drwxr-xr-x 2 mthomas use300 5 Feb 16 20:37 tensorflow
drwxr-xr-x 2 mthomas use300 3 Jul 14 2021 tools
[mthomas@login01 ~]$
```

Expanse Portal: Running Matlab

The image displays a multi-step process for running Matlab on the Expanse Portal:

- Step 1: Session Launch Request**
A screenshot of the "Interactive Apps" section shows a modal for launching a "MATLAB" session. The modal contains instructions about interacting via VNC and provides a "Session ID: fee7f0fb-48df-46af-8f4".

This app will launch a MATLAB GUI on Expanse. You will be able to interact with the MATLAB GUI through a VNC session. Please email help@xsede.org to be added to matlab-groups.

Session ID: fee7f0fb-48df-46af-8f4
- Step 2: Session Confirmation**
A screenshot of the "My Interactive Sessions" page shows a message: "Session was successfully deleted." This indicates the initial session has been completed or removed.

Session was successfully deleted.
- Step 3: Session Creation**
A screenshot of the "Interactive Apps" section shows a new Matlab session being created. The session details include:
 - Host: <https://exp-1-18.expanse.sdsc.edu>
 - Created at: 2022-08-01 22:48:55 PDT
 - Time Remaining: 25 minutes
 - Session ID: fee7f0fb-48df-46af-8f40-ea7bf62c1bda
- Step 4: Matlab Interface**
A screenshot of the Matlab R2020b interface. The command window shows a plot of a piecewise linear function and some error messages related to array indexing. The workspace shows variables *a* and *b*.

Expanse Portal: Launching Notebooks

The screenshot illustrates the Expanse Portal's Jupyter Session feature. It consists of three main components:

- Left Panel (Top):** A "Jupyter Session" configuration form. Fields include:
 - Account: use300
 - Partition (Please choose the gpu, gpu-shared, or gpu-preempt as the partition if using gpus): shared
 - Time limit (min): 30
 - Number of cores: 1
 - Memory required per node (GB): 2
 - GPUs (optional): 0
- Middle Panel:** A file browser window showing files in the directory /notebook-examples/Hello_World/. The files listed are:
 - hello_world.ipynb (selected)
 - hello.py
 - README.md
- Right Panel:** A Jupyter Notebook interface with two tabs open:
 - `numpy_intro.ipynb`: Contains code related to numpy, such as `irperf x`, `lrv svm_l`, `ushbyasid`, `avic v_vm`, and `id overflo`.
 - `hello_world_gpu.ipynb`: Contains code related to GPU usage, including a command to check for NVIDIA-SMI availability: `[9]: # Check to see if system is GPU:
!nvidia-smi`. The output indicates that NVIDIA-SMI has failed because it couldn't communicate with the NVIDIA driver.

- PORTAL LIVE DEMO

Outline

- What is Interactive High-Performance Computing?
- Accessing Interactive HPC Nodes
- Running Interactive Apps
- Running Interactive Apps Using the Expanse Portal
- Notebook Security

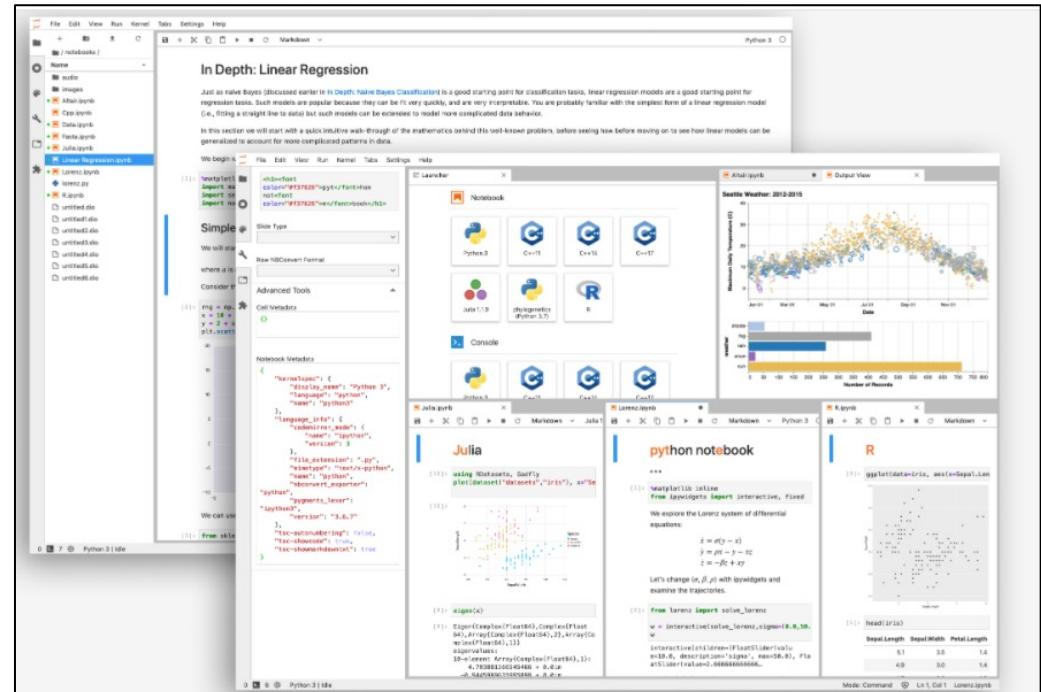
Jupyter Notebooks

What is Jupyter?

*Jupyter is a free, open-source, **interactive** web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. (J. Perkel, <https://www.nature.com/articles/d41586-018-07196-1>)*

Common Jupyter Services:

- Jupyter Notebooks (single user)
- JupyterLab: advanced version of notebook
- JupyterHub: multiuser Jupyter service



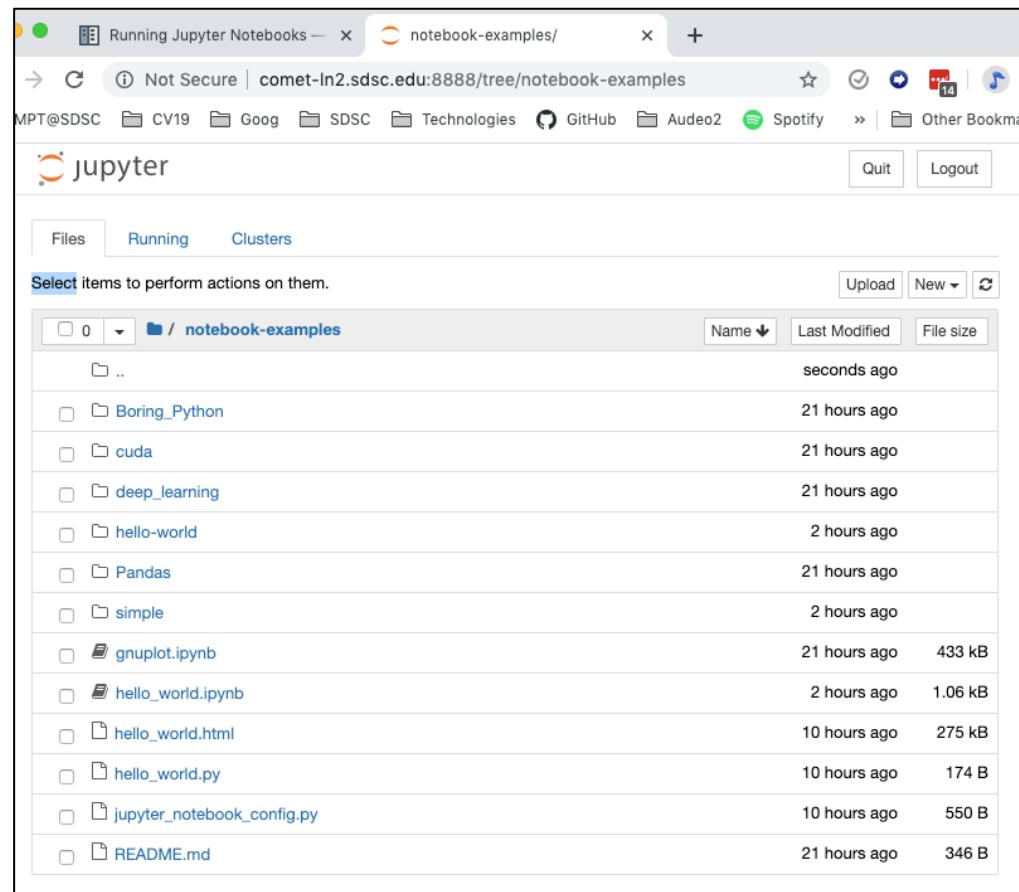
Interactive Services have a Key Vulnerability: They Provide Access to HPC File Systems, often over HTTP

SDSC Interactive Services Policy:

- Portals, JupyterHub, and other services cannot be mounted directly to disk (must be on VM or external)
 - Many use root in vulnerable ways
 - If a user launches Jupyter Lab or Notebooks, the jobs will be killed.
- Applications cannot run on login nodes

SDSC recommendation:

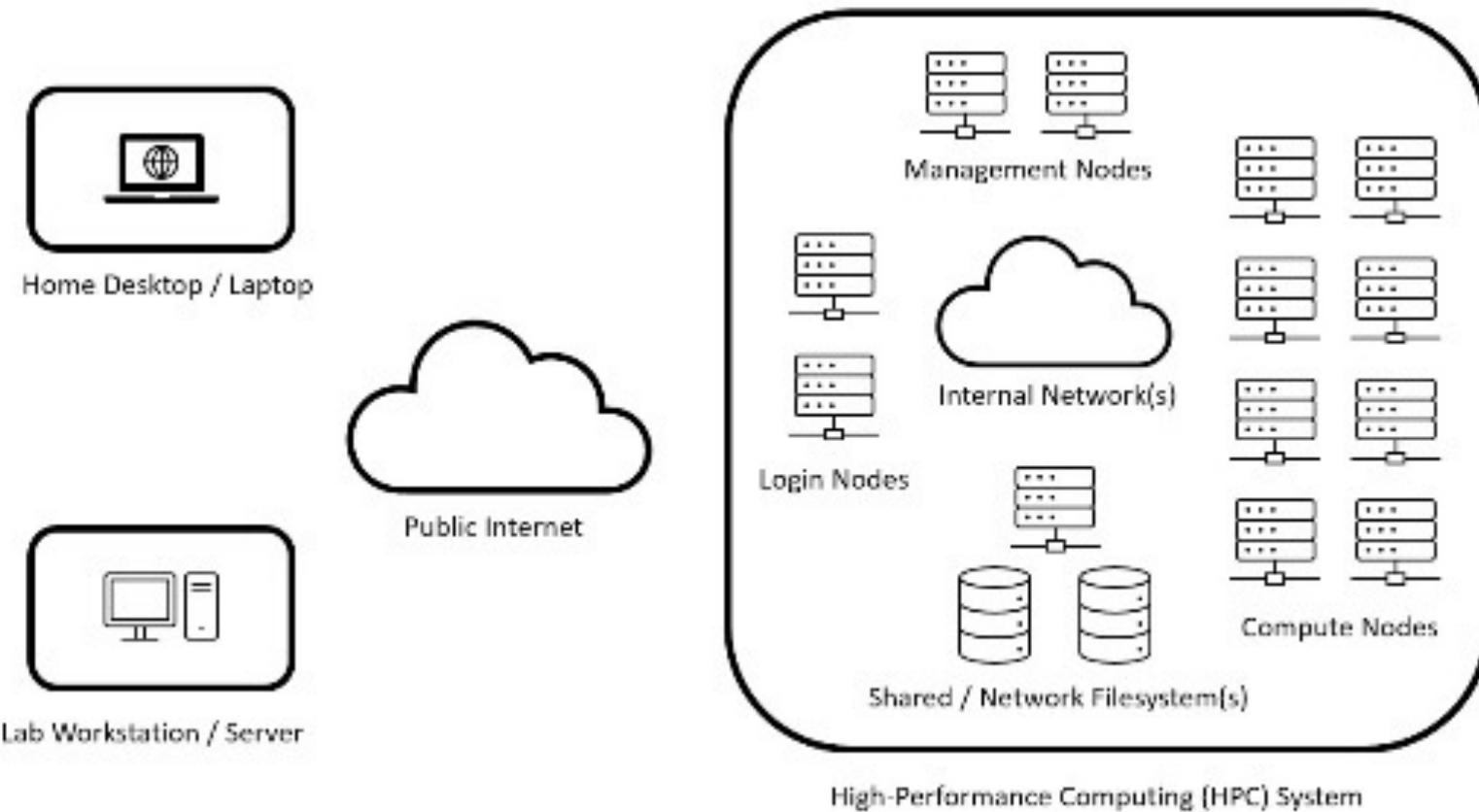
- Use secure connections: when you choose insecure connections your account is vulnerable to hacking
- Use portal.expanse.sdsc.edu



Accessing and Running Secure Notebooks on HPC Systems

- Install notebook application:
 - Locally: install Anaconda on your laptop
 - Remotely:
 - Install Anaconda/conda on the remote machine (default is HTTP) – **not recommended**
 - Launch securely (HTTPS) using SRPS/*galyleo* -- **recommended**
- Running remotely:
 - Connect over HTTP (default, insecure)
 - Connect over HTTP + SSH tunneling (secure, but inconvenient)
 - **Connect over HTTPS + using the *Satellite Reverse Proxy Service* (SRPS) and *galyleo client* (secure, convenient)**
- You can launch Jupyter services on SDSC:
 - CPU and GPUs
 - Interactive nodes: command line
 - Slurm batch script
- **Treat the Notebook URL like a Password!**

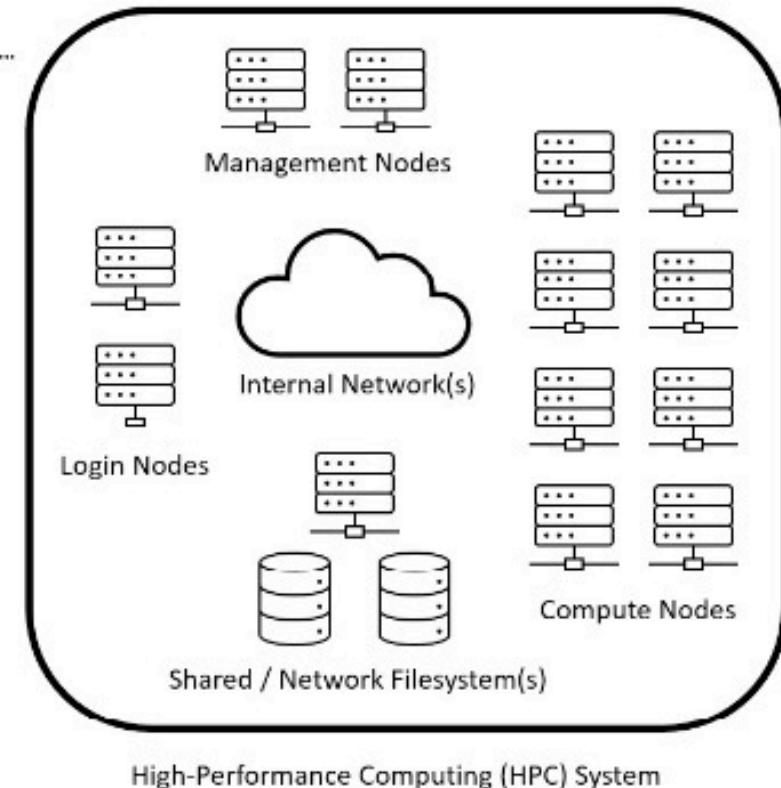
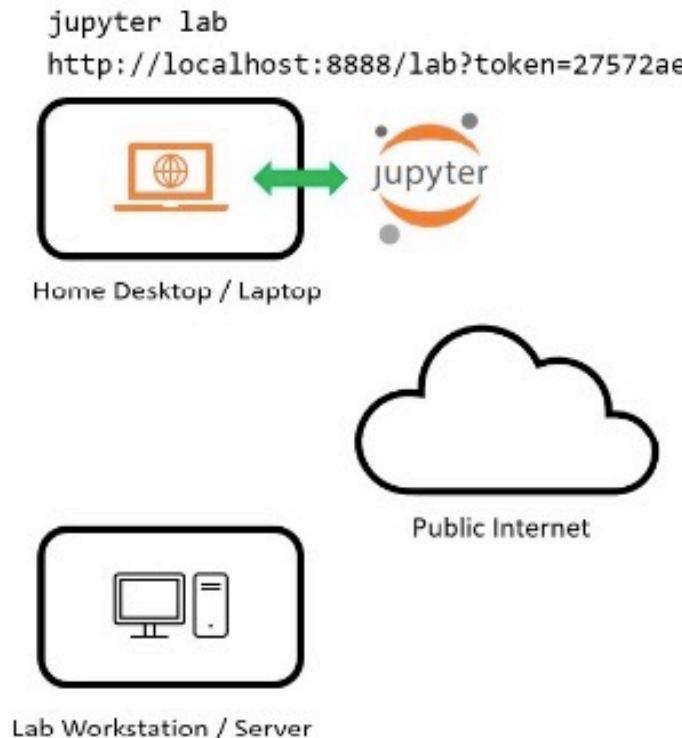
Standard HPC Ecosystem



Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/

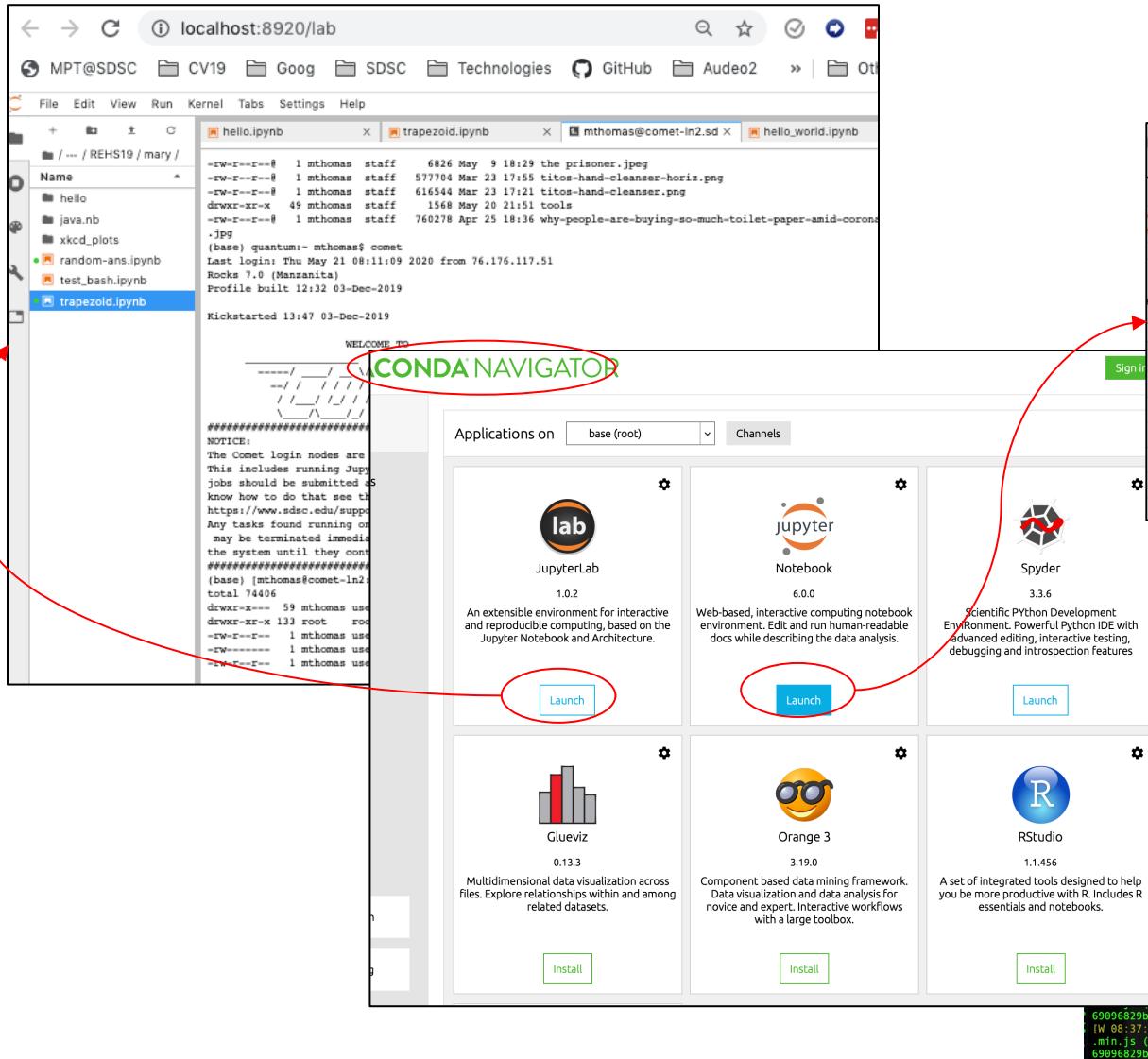
Running Notebooks Locally Over HTTP

Install the Anaconda application and use it to launch your notebooks



Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/

Anaconda on Local Host or Laptop



The screenshot shows a Jupyter notebook interface at `localhost:8807/tree/dev/sdsc....`. The top navigation bar includes "jupyter", "Files", "Running", and "IPython Clusters". The left sidebar shows a file tree with files like `hello.ipynb`, `hello_world.ipynb`, `hello.py`, and `hello.rb`. The right pane shows a terminal session with the following output:

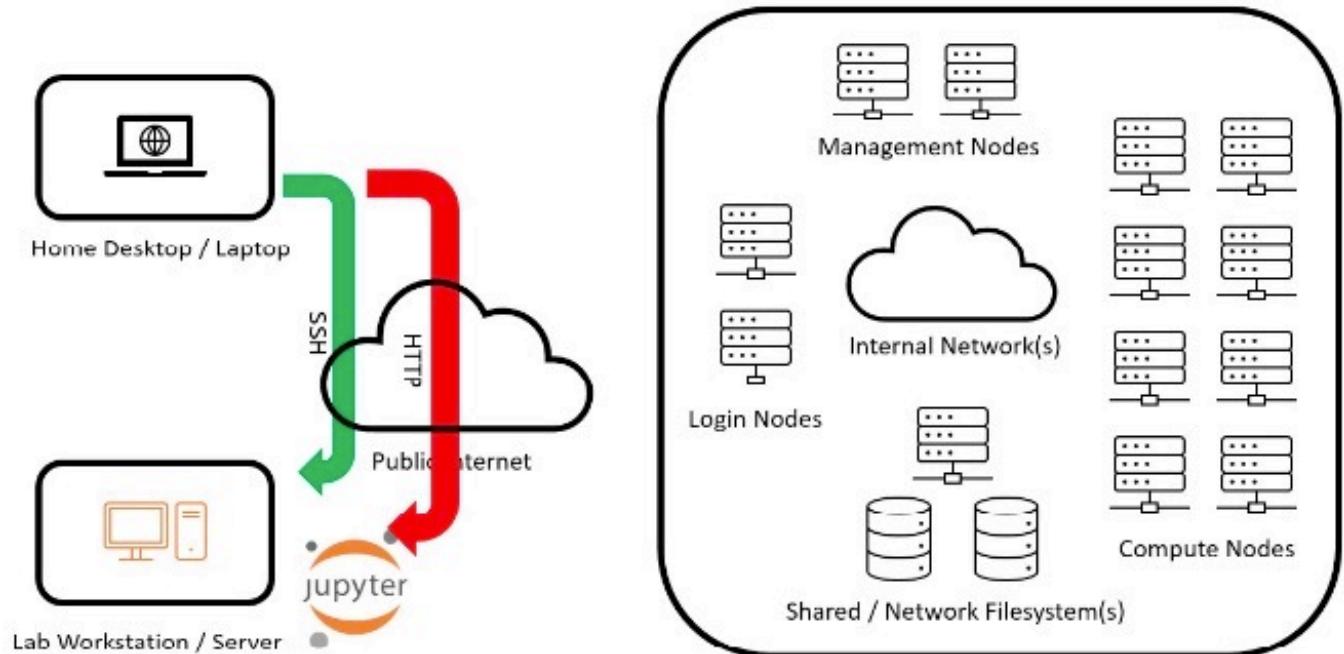
```
mthomas — jupyter_mac.command — python -bash — 82x43
: May 20 23:47:17 on ttys003
:jupyter_mac.command : exit;
: mthomas$ /anaconda3/bin/jupyter_mac.command : exit;
: NotebookApp] The port 8888 is already in use, trying another port.
: NotebookApp] The port 8889 is already in use, trying another port.
: NotebookApp] The port 8890 is already in use, trying another port.
: NotebookApp] The port 8891 is already in use, trying another port.
: NotebookApp] Loading IPython parallel extension
: NotebookApp] JupyterLab extension loaded from /anaconda3/lib/python3.7/jupyterlab
: NotebookApp] JupyterLab application directory is /anaconda3/share/jupyterlab
: NotebookApp] Serving notebooks from local directory: /Users/mthoma...
: NotebookApp] The Jupyter Notebook is running at:
: NotebookApp] http://localhost:8798/?token=681f898da657418d5cce6909...
: NotebookApp] or http://127.0.0.1:8798/?token=681f898da657418d5cce6909...
: NotebookApp] Use Control-C to stop this server and shut down all kernels without confirmation.
: NotebookApp]

the notebook, open this file in a browser:
'/Users/mthomas/Library/Jupyter/runtime/nbserver-55091-open.html'
I paste one of these URLs:
'localhost:8798/?token=681f898da657418d5cce69096829b6064b3313c100fdc...
/127.0.0.1:8798/?token=681f898da657418d5cce69096829b6064b3313c100fdc...

NotebookApp] Could not open static file ''
NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (:::1) 1 0@ms referer=http://localhost:8798/tree?token=681f898da657418d5cce69096829b6064b3313c100fdc...
```

Running Notebooks Remotely Over HTTP

- Messages over HTTP
- Not encrypted
- Vulnerable to hacking



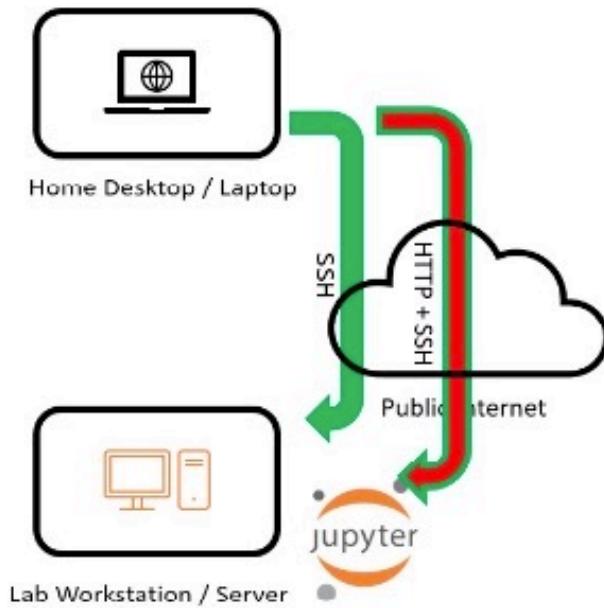
```
jupyter lab --ip="$(hostname)" --no-browser      High-Performance Computing (HPC) System  
http://remote.jupyter.lab:8888/lab?token=27572ae ...
```

Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/

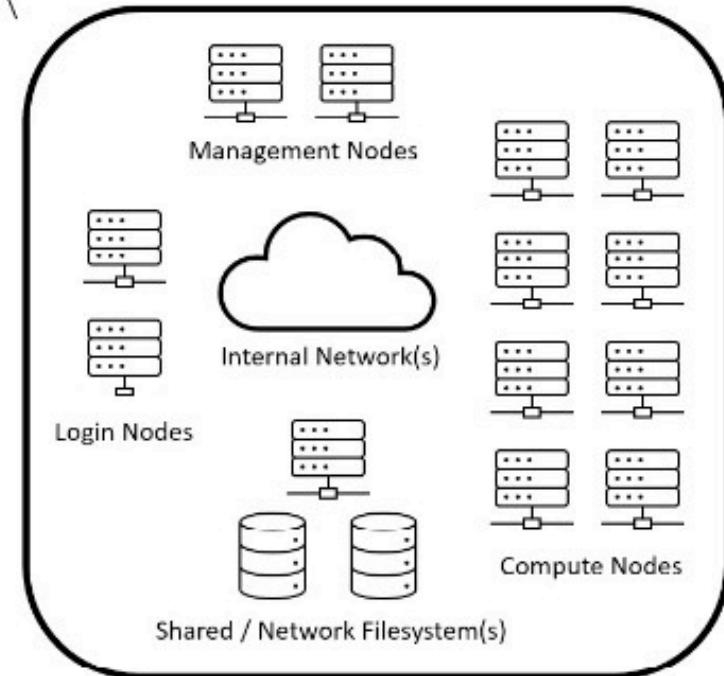
Running Notebooks Remotely Over HTTP Using SSH Tunneling

- Notebooks running on Remote host
- Messages over HTTP, but sent via secure tunnel

```
ssh -N -L localhost:8888:localhost:8889 \
user@remote.jupyter.lab
```



```
jupyter lab --port=8889 --no-browser
http://localhost:8889/lab?token=27572ae ...
```

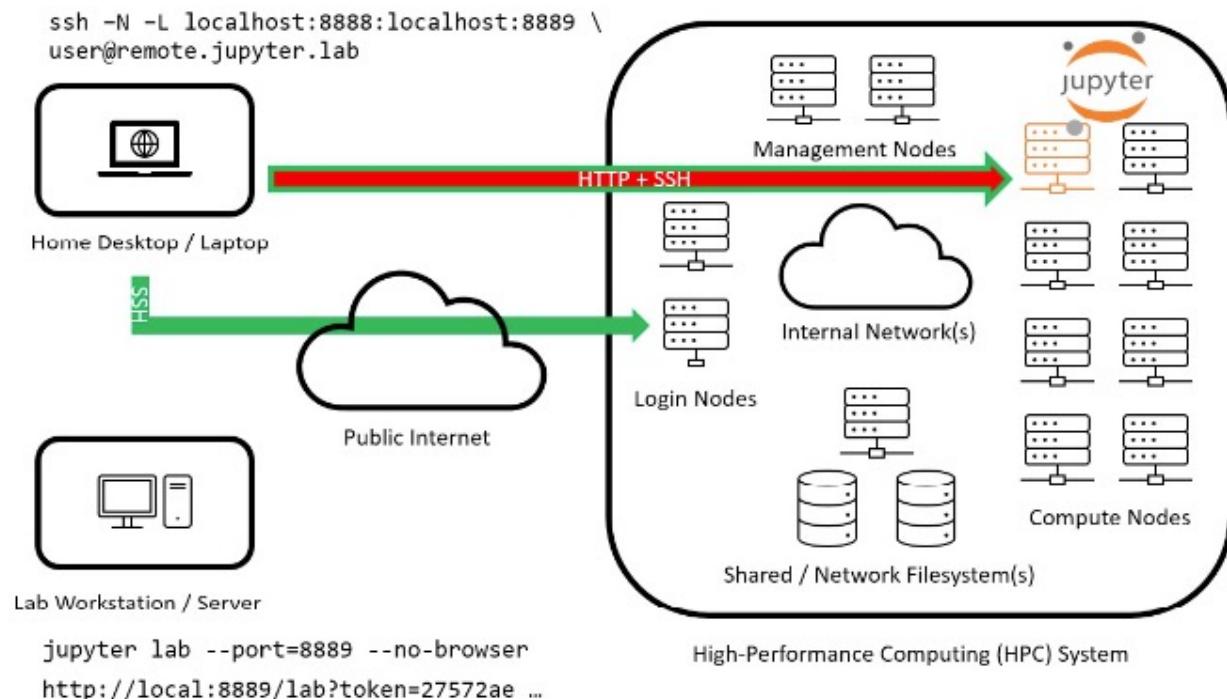


High-Performance Computing (HPC) System

Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/

Running Notebooks Remotely Over HTTP Using SSH Tunneling on HPC System

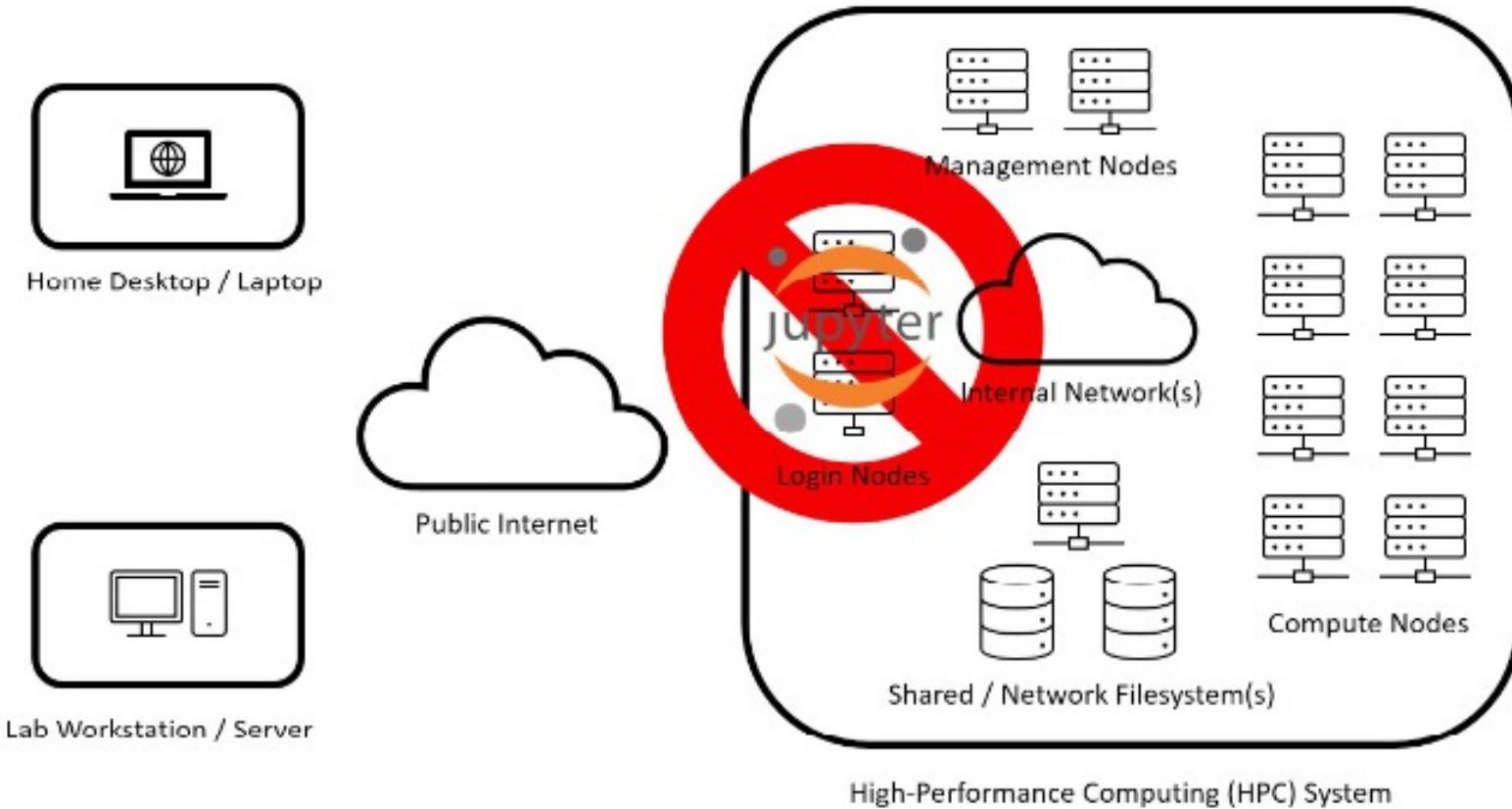
- Notebook running on HPC System
- Messages over HTTP, but passing via secure tunnel
- But do not run on compute nodes!



Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/

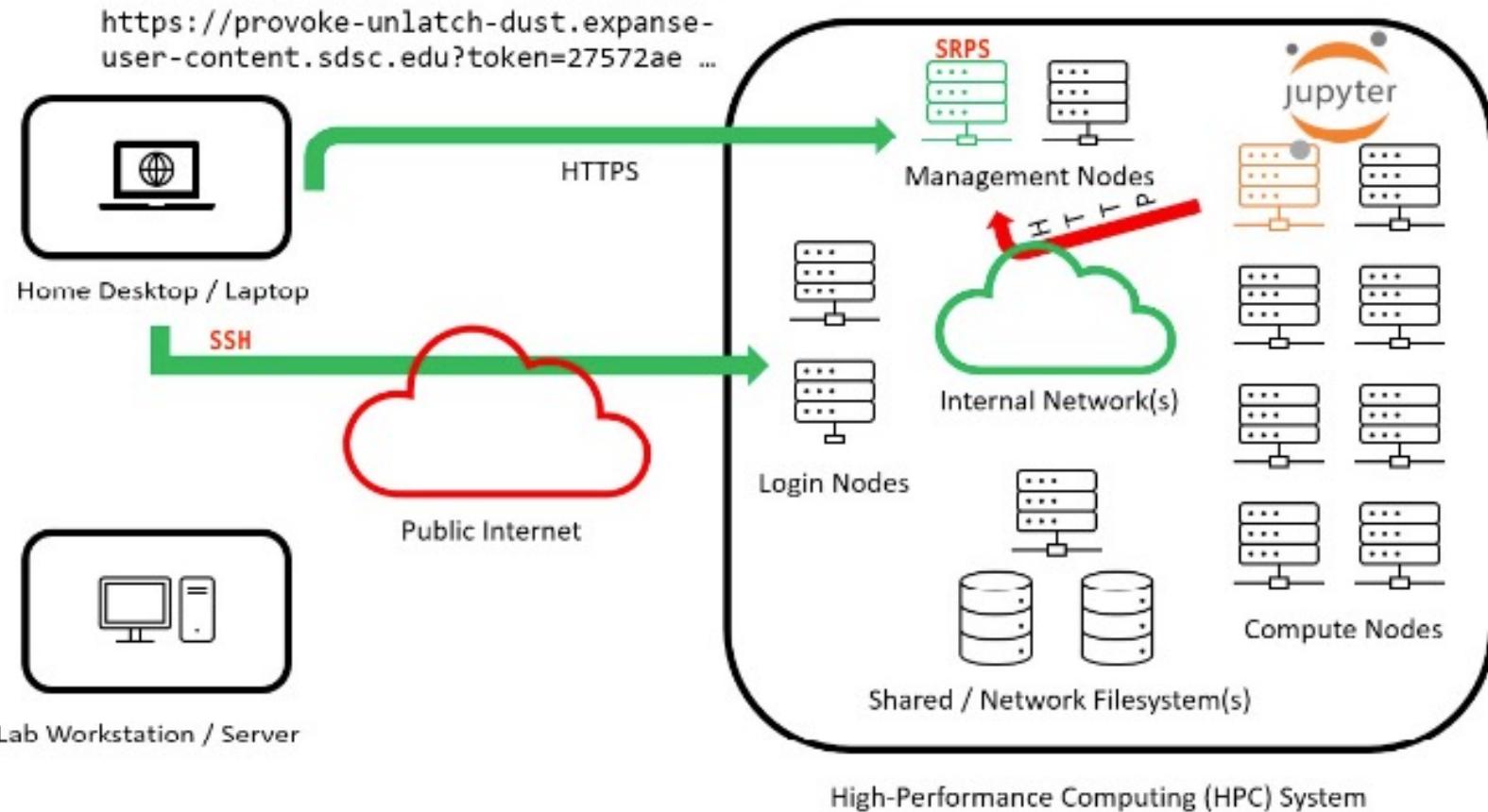
Warning

Do not run notebooks/interactive services on the login nodes!



Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/

Running Notebooks Remotely Using SRPS



Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/

SDSC Satellite Reverse Proxy Service

- SRPS: prototype system that allows users to launch secure standard Jupyter Notebooks on any Expanse compute node using a reverse proxy server.
 - Notebooks will be hosted on the internal cluster network as an HTTP service using standard Jupyter commands.
 - Service available to the user outside of the cluster firewall over HTTPS connection between the external users web browser and the reverse proxy server.
- Goal: minimize software changes for users, improve security of user notebooks running on SDSC systems.
- SRPS can run on any HPC system capable of supporting Apache on internal network.

galyleo

- 2nd generation shell utility developed to orchestrate a user's interaction with both Satellite and Slurm to start a Jupyter session within a batch job.
- Developed while reviewing start-jupyter (prototype client) codebase to sort out how best to support Expanse (OOD) Portal and HPC User Services Group long-term; integrated into an existing SSH tunneling orchestration utility to use Satellite proxy service instead
- Key features in design:
 - No need to install conda environment or update packages
 - Increases flexibility for users to configure software environment; but also try to makes it simpler for them to do this themselves
 - Batch job script is generated completely on-the-fly.
 - Command-line argument driven.
 - Quiet mode for OOD portal

<https://github.com/mkandes/galyleo>

Satellite Client: galyleo

Key features in design:

- User calls galyleo.sh launch script, which requests token from Satellite, passes token to batch job script and submits the job to Slurm; token redeemed from batch job once it runs
- Increase flexibility for users to configure software environment; but also try to make it simpler for them to do themselves
- Batch job script is generated completely on-the-fly.
- Command-line argument driven.
- Quiet mode for OOD portal

```
[username@login01 ~]export  
PATH="/cm/shared/apps/sdsc/galyleo:${PATH}"  
  
[username@login01 ~]$ galyleo.sh --help  
USAGE: galyleo.sh launch [command-line  
option] {value}  
  
command-line option : value  
  
-A | --account :  
-R | --reservation :  
-p | --partition :  
-q | --qos :  
-N | --nodes :  
-n | --ntasks-per-node :  
-c | --cpus-per-task :  
-M | --memory-per-node : GB  
-m | --memory-per-cpu : GB  
-G | --gpus :  
--gres :  
-t | --time-limit :  
-j | --jupyter :  
-d | --notebook-dir :  
-r | --reverse-proxy :  
-D | --dns-domain :  
-s | --sif :  
-B | --bind :  
-e https://github.com/mkandes/galyleo  
| --env-modules :  
| --conda-env :  
-Q | --quiet : 1
```

Launching CPU notebooks using galileo

Step 1:
Login and setup user environment

Step 2:
Run command to launch secure notebook

Step 3:
Copy URL; paste into browser on local system

Step 4:
Monitor Slurm queue; wait for job to start

```
[username@login01 ~]$ which galileo.sh  
/usr/bin/which: no galileo.sh in (/cm/shared/apps/spack/cpu/opt/spack/linux-  
centos8-zen/gcc-8.3.1  
[SNIP]  
home/username/.local/bin:/home/username/bin)  
[username@login01 ~]$ export PATH="/cm/shared/apps/sdsc/galileo:${PATH}"  
[username@login01 ~]$ which galileo  
/cm/shared/apps/sdsc/galileo/galileo  
  
[username@login01 ~]$ galileo launch --account abc123 --partition shared --cpus 2 --memory 4 --time-limit  
00:30:00 --env-modules cpu/0.17.3b,anaconda3/2021.05  
[snip]  
Submitted Jupyter launch script to Slurm. Your SLURM_JOB_ID is 9773665  
[snip]  
Your Jupyter notebook session will begin once compute resources are allocated to  
your job by the scheduler.  
  
https://carload-spray-koala.expanse-user-content.sdsc.edu/lab  
  
[username@login01 ~]$ squeue -u username  
JOBID PARTITION NAME USER ST TIME NODES  
NODELIST(REASON)  
9773665 gpu-debug galileo- username PD 0:00 1 (None)  
[username@login01 ~]$ squeue -u username  
JOBID PARTITION NAME USER ST TIME NODES  
NODELIST(REASON)  
9773665 gpu-debug galileo- username R 0:20 1 exp-7-59
```

Satellite Server Pending Page

- Load notebook URL in browser; wait for it to launch
- Monitor pending page
- Run the “squeue” command on the HPC system to check job status
- If the job queue is busy, it may take a while to launch the notebook
- **Treat Jupyter Notebook URL as a password!**

The screenshot shows the "Satellite Reverse Proxy Service" interface for the SDSC Expanse system. At the top, it displays the job state as "Mapped". Below this, there is a horizontal timeline with four circular nodes: "In Queue" (green), "Running" (green), "Mapped" (green), and "Proxied" (white). A yellow oval highlights the first three nodes. To the right of the timeline is a legend explaining the job states:

In Queue	Job has not yet started.
Running	Job has started, but has not redeemed Satellite Token.
Mapped	Job has redeemed Satellite Token, but no proxy entry exists yet.
Proxied	Proxy entry created, ready to go!
Dead	Job died or exited, no further progress will occur.

At the bottom of the page, a terminal window shows the output of the "squeue -u mthomas" command:

```
[mthomas@login02 ~]$ [mthomas@login02 ~]$ squeue -u mthomas
      JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
      9774239    shared galileo-  mthomas R      3:49      1 exp-1-13
      9774274    debug galileo-  mthomas R      0:12      1 exp-9-55
[mthomas@login02 ~]$
```

Satellite Pending Page

The screenshot shows a web browser window with three tabs open, each displaying the "Satellite Reverse Proxy Service" status for "SDSC Expanse".

- Tab 1:** Job State: Unknown. Status bar: In Queue → Running → Mapped → Proxied.
- Tab 2:** Job State: Mapped. Status bar: In Queue → Running → Mapped → Proxied.
- Tab 3:** Job State: Proxied. Status bar: In Queue → Running → Mapped → Proxied. Below the status bar, a legend defines the states:
 - In Queue:** Job has not yet started.
 - Running:** Job has started, but has not redeemed Satellite Token.
 - Mapped:** Job has redeemed Satellite Token, but no proxy entry exists yet.
 - Proxied:** Proxy entry created, ready to go!
 - Dead:** Job died or exited, no further progress will occur.

Jupyter Notebook Interface:

- File Explorer:** Shows a directory structure under "/notebook-examples / Hello_World /".
 - hello_world_cpu.ipynb (selected)
 - hello_world_gpu.ipynb
 - hello_world.py
 - README.md
- Code Editor:** Displays the content of the selected file, "hello_world_cpu.ipynb".

```
[8]: print('Hello world!!!!')
Hello world!!!!

[9]: # Import hello module
import hello

# Define a local function
def world2(name):
    print(name)

[10]: world2("mary")
mary

[11]: hello.greeting("good times")
Greetings, good times

[12]: hello.world("World.")
Hello, World.
```

Launching GPU notebooks using galyleo

- GPU Notebooks run better when using containers. SDSC maintains several containers on Expanse
- See: /cm/shared/apps/containers/singularity

Step 1:
Login and setup user environment

```
[username@login01 ~]$ which galyleo.sh  
/usr/bin/which: no galyleo.sh in (/cm/shared/apps/spack/cpu/opt/spack/linux-centos8-  
zen/gcc-8.3.1  
[SNIP]  
home/username/.local/bin:/home/username/bin)
```

Step 2:
Run command to
launch secure
notebook

```
[username@login01 ~]$ export PATH="/cm/shared/apps/sdsc/galyleo:${PATH}"  
[username@login01 ~]$ which galyleo  
/cm/shared/apps/sdsc/galyleo/galyleo  
[username@login01 ~]$ galyleo launch --account use300 --partition gpu-debug --cpus  
10 --memory 93 --gpus 1 --time-limit 00:5:00 --env-modules singularitypro --sif  
/cm/shared/apps/containers/singularity/pytorch/pytorch-latest.sif --bind  
/expanse,/scratch --nv  
[snip]
```

Step 3:
Copy URL; paste into
browser on local
system

```
Submitted Jupyter launch script to Slurm. Your SLURM_JOB_ID is 9773912  
[snip]  
Your Jupyter notebook session will begin once compute resources are allocated to  
your job by the scheduler.  
https://grief-fantastic-given.expanse-user-content.sdsc.edu?token=5097acb6f32ab82dd51b4524c267d2fd
```

Step 4:
Monitor Slurm queue;
wait for job to start

```
[username@login01 ~]$ squeue -u username  
JOBID PARTITION      NAME      USER ST          TIME   NODES NODELIST(REASON)  
9773912 gpu-debug  galyleo-  username PD          0:00      1 (None)  
[username@login01 ~]$ squeue -u username  
JOBID PARTITION      NAME      USER ST          TIME   NODES NODELIST(REASON)  
9773912 gpu-debug  galyleo-  username R          0:20      1 exp-7-59
```

Notebook is launched on GPU device

MPT@SDSC File Edit View Run Kernel Tabs Settings Help

numpy_intro.ipynb X hello_world_gpu.ipynb ● Python 3

```
sc art arch_perfmon pebs rep_good nopl xtopology nonstop_tsc cpuid aperfmpf perfmon_pni pclm ulldq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xptr pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowp refetch cpuid_fault epb cat_l3 cdp_l3 invpcid_single intel_ppin ssbd mba ibrs ibpb stibp i brs_enhanced tpr_shadow vnmi flexpriority ept vpid fsqsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm mpq rdt_a avx512f avx512dq rdseed adx smap clflushopt clwb intel_pt avx512cd avx512bw avx512vl xsaveopt xsaveavx xgetbv1 xsaves cqmm_llc cqmm_occu_llc cqmm_mb m_total cqmm_mb_local dtherm ida arat pln pts pku ospke avx512_vnni md_clear flush_l1d arc_h_capabilities
```

[9]: # Check to see if system is GPU:
!nvidia-smi

```
Fri Feb 18 00:15:50 2022
```

NVIDIA-SMI 460.32.03		Driver Version: 460.32.03	CUDA Version: 11.2
GPU Name	Persistence-M	Bus Id	Disp.A Volatile Uncorr. ECC
Fan Temp	Perf	Pwr:Usage/Cap	Memory-Usage GPU-Util Compute M.
			MIG M.
0 Tesla V100-SXM2...	On	00000000:18:00.0	Off 0
N/A 37C	P0	68W / 300W	0MiB / 32510MiB 0% Default N/A

[10]: # if you see: /bin/bash: nvidia-smi: command not found
the system is not GPU

Notebook Examples

<https://github.com/sdsc-hpc-training-org/notebook-examples>

- Collection of tested, working notebooks tested on Expanse and other SDSC HPC systems
- Includes range of materials from “hello world” to Spark ML notebooks.
- Note: collection changes often, based on testing and contributions

The screenshot shows a GitHub repository page for 'sdsc-hpc-training-org / notebook-examples'. The repository is public, as indicated by the 'Public' button. The main navigation bar includes 'Code', 'Issues 1', 'Pull requests 1', 'Actions', 'Projects', and 'Wiki'. Below the navigation, there's a dropdown for the 'master' branch, showing '3 branches' and '0 tags'. Buttons for 'Go to file', 'Add file -', and 'Code -' are available. The commit history lists 60 commits from 'marypthomas' (32163d3) at 'now' to 'README.md' (now). Other commits include updates to training material for various topics like Python, CUDA, Data Analysis, and Matplotlib, as well as merges and additions related to HPC notebooks and basic development. The most recent commit is 'merge hpc notebooks material' (16 minutes ago).

Commit	Message	Time
marypthomas Update README.md	32163d3 now	60 commits
Boring_Python_Book	updating training material	7 months ago
CUDA_GPU_NVIDIA	updating training material	7 months ago
Data_Analysis	updating training material	7 months ago
Hello_World	updating training material	7 months ago
Matplotlib_Intro	updating training material	7 months ago
Notebook_Dev_Basics	updating training material	7 months ago
NumPy_Intro	updating training material	7 months ago
hpc-notebooks	merge hpc notebooks material	16 minutes ago
.DS_Store	add hello world	10 months ago
.gitignore	update material	2 years ago
README.md	Update README.md	now

Resources

- Expanse :
 - Landing page: expanse.sdsc.edu
 - User Guide: https://expanse.sdsc.edu/support/user_guides/expanse.html
- GitHub Repo for Satellite and Galileo:
 - <https://github.com/sdsc-hpc-training-org/satellite>
 - <https://github.com/mkandes/galileo>
- SDSC Training Resources
 - https://www.sdsc.edu/education_and_training/training
 - <https://github.com/sdsc-hpc-training-org/notebook-examples>
- XSEDE Training Resources
 - <https://www.xsede.org/for-users/training>
 - <https://cvw.cac.cornell.edu/expanse/>
- Problems? Contact consult@sdsc.edu

Thank You!

if you have problems, please contact consult@sdsc.edu