Interactive Computing and Running Jupyter Notebooks

Presented at the Using Python and Jupyter Notebooks on TSCC Workshop

September 2, 2021

Mary Thomas
(SDSC)

Outline

- Defining Interactive High-Performance Computing
- Running Interactive Jobs
- Application Examples

Resources

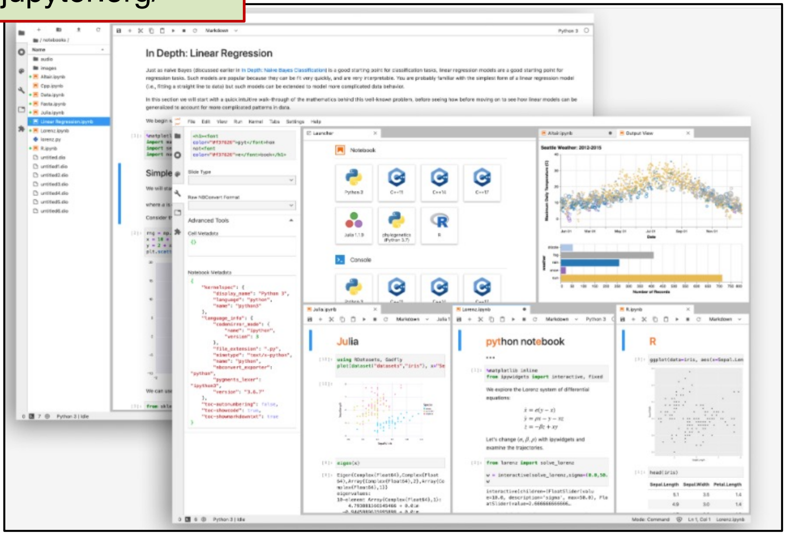
- TSCC User Guide
  - https://www.sdsc.edu/support/user_guides/tscc.html
- GitHub Repo for this workshop:
  - https://github.com/sdsc-hpc-training-org/Using-Python-and-Jupyter-Notebooks-on-TSCC/
- GitHub Repo for Example Notebooks:
  - https://github.com/sdsc-hpc-training-org/notebook-examples
- SDSC Training Resources
  - https://www.sdsc.edu/education_and_training/training

What is Interactive HPC-Computing

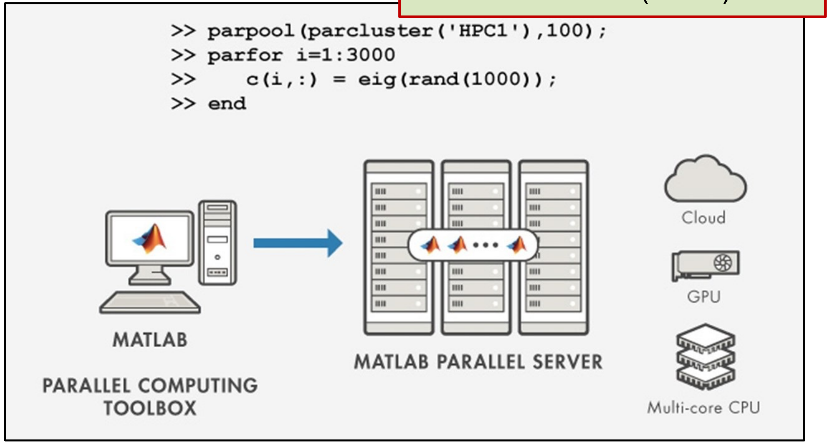
- In **computer** science, **interactive computing** refers to software which accepts input from the user as it runs.
  - commonly used programs, such as word processors or spreadsheet applications.
- **Interactive HPC computing** involves *real-time* user inputs to perform tasks on a set of compute 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.

Interactive HPC Scenarios

https://jupyter.org/



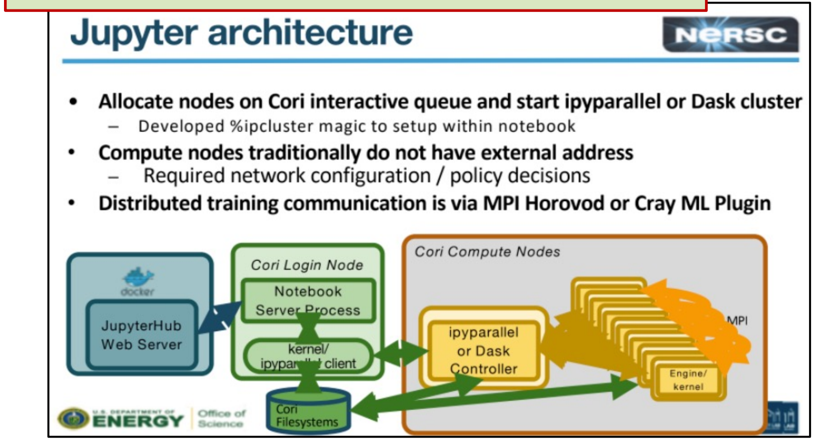
Parallel Matlab (AWS)



```
>> 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

Interactive Distributed Computing with Jupyter (NERSC)

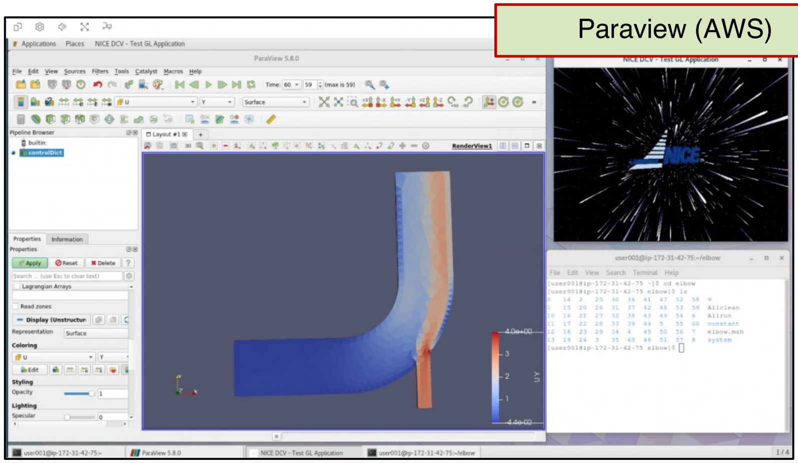


Jupyter architecture

- 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

Paraview (AWS)

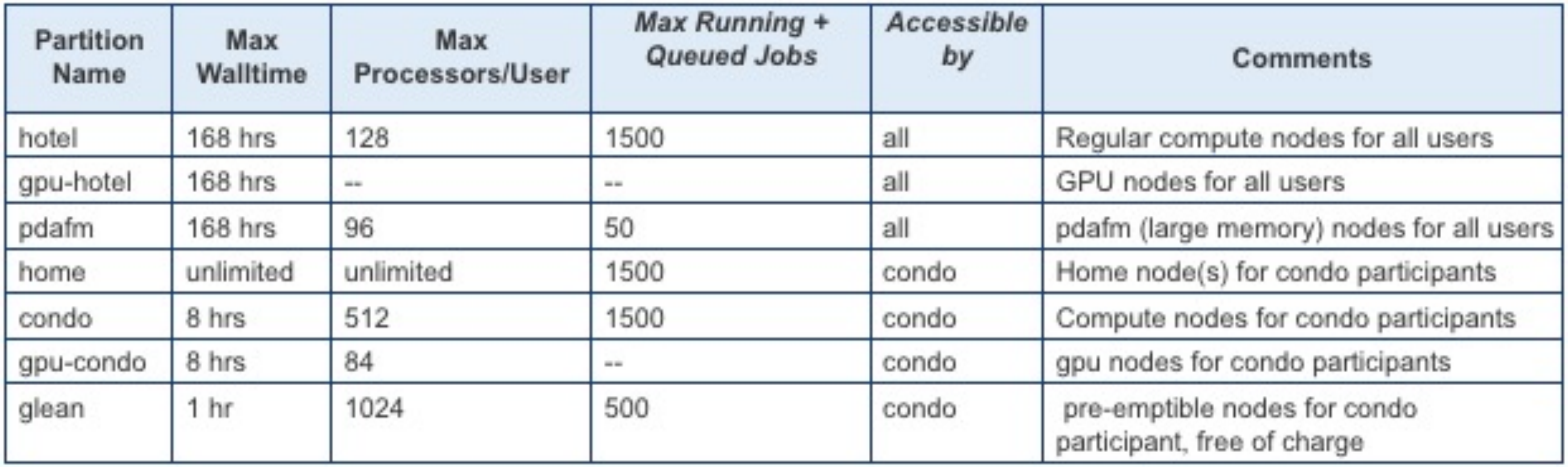


https://aws.amazon.com/blogs/compute/how-to-run-3d-interactive-applications-with-nice-dcv-in-aws-batch/

TSCC Interactive Jobs

- Must be run on compute nodes (CPU, GPU)
  - Depends on which partition you choose
- Submit a request to the queuing system
- Choose which queue you want to use
- Use the *qsub* command:

```
qsub –I -l nodes=2:ppn=10 -l walltime=0:50:00
```



Partition Name	Max Walltime	Max Processors/User	*Max Running + Queued Jobs*	*Accessible by*	Comments
hotel	168 hrs	128	1500	all	Regular compute nodes for all users
gpu-hotel	168 hrs	--	--	all	GPU nodes for all users
pdafm	168 hrs	96	50	all	pdafm (large memory) nodes for all users
home	unlimited	unlimited	1500	condo	Home node(s) for condo participants
condo	8 hrs	512	1500	condo	Compute nodes for condo participants
gpu-condo	8 hrs	84	--	condo	gpu nodes for condo participants
glean	1 hr	1024	500	condo	pre-emptible nodes for condo participant, free of charge

HPC Basics - Running jobs on TSCC

- All jobs submitted via TORQUE resource manager (pbs) with Maui scheduler.
- Batch jobs : submit batch scripts from login nodes.
  - qsub test.sh
    - Submit script to include PBS directives such as
      - number of nodes, cores, estimated time
      - right queue
      - job name, email address
      - account
- Interactive jobs:
  - qsub –I -l nodes=1:ppn=1 –t 00:30:00 –q condo –A account

Some useful commands…

- #PBS –k oe => writes directly to destination dir
- **qalter** – change parameters of a queued job (e.g. email, account)
- **qdel** <jobnumber> – delete job

Monitoring Job Submission

- qstat - current status of jobs and queues
  - qstat –u username
  - qstat –f jobnumber

- yqd – prints out why the job hasn’t started

```
$ yqd 13977625
13977625 (xxxxxx home-YXX 1x24 ['haswell'] 0:12:23): 0 nodes free
```

- lsjobs - useful to see the node availability

```
$ lsjobs --property=condo-node
tscc-4-69: FREEx28
tscc-4-70: FREEx28
```

Job Monitoring contd…

- checkjob

```
$ checkjob 13976617
.....
Reservation '13976617' (9:19:34:24 -> 12:19:34:24  Duration:
3:00:00:00)
PE:  3.00  StartPriority:  203174
job cannot run in partition DEFAULT (idle procs do not meet
requirements
: 0 of 3 procs found)
idle procs: 3995  feasible procs:   0
....
```

-
  - showq –i   (position in queue)
  - showstart <jobid> (estimated start time)

Account Monitoring

- Users can belong to multiple groups

```
groups <username>
```

  - belonging to multiple groups

- Accounts are used to keep track of allocation usage

```
gbalance –u <username>
```

  - Specified by #PBS –A
- Check usage with

```
gstatement –u <username>
```

TSCC Interactive Jobs: Application Examples

- MPI ”Hello World”
- Matlab
- R console
- Jupyter Notebooks → Covered in next two talks

TSCC Interactive Job: Hello-MPI

```
qsub –I -l nodes=2:ppn=10 -l walltime=0:50:00
```

Interactive nodes can be used for running parallel jobs (MPI, OpenMP, CUDA code if GPU), compiling large codes, etc.

```
[mthomas@tscc-gpu-9-4 hello-mpi]$
[mthomas@tscc-gpu-9-4 hello-mpi]$ qsub -I -l nodes=1:ppn=16 -l walltime=0:30:00 -q glean
qsub: waiting for job 26842841.tscc-mgr7.local to start
qsub: job 26842841.tscc-mgr7.local ready

[mthomas@tscc-gpu-9-4 ~]$ pwd
[mthomas@tscc-gpu-9-4 hello-mpi]$ mpirun -np 8 ./hello-mpi
 node          0 : Hello and Welcome to TSCC!
 node          6 : Hello and Welcome to TSCC!
 node          1 : Hello and Welcome to TSCC!
 node          3 : Hello and Welcome to TSCC!
 node          4 : Hello and Welcome to TSCC!
 node          5 : Hello and Welcome to TSCC!
 node          7 : Hello and Welcome to TSCC!
 node          2 : Hello and Welcome to TSCC!
```

Request interactive node using glean queue node

TSCC Interactive Jobs: Applications
Running Matlab without GUI

qsub –I –l nodes=1:ppn=16 walltime=0:30:00 –q glean

```
[mthomas@tscc-login1 hello-mpi]$ qsub -I -l nodes=1:ppn=16 -l
walltime=0:30:00 -q glean
qsub: waiting for job 26842936.tscc-mgr7.local to start
qsub: job 26842936.tscc-mgr7.local ready


[mthomas@tscc-13-12 ~]$ module load matlab
[mthomas@tscc-13-12 ~]$ matlab -nodisplay
               < M A T L A B (R) >
      Copyright 1984-2016 The MathWorks, Inc.
      R2016b (9.1.0.441655) 64-bit (glnxa64)
                September 7, 2016

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.
```

```
>>  A = [1 3 0; 2 4 -1; 4 9 -1]
A =
     1     3     0
     2     4    -1
     4     9    -1
>> B=A'
B =
     1     2     4
     3     4     9
     0    -1    -1
>> A*B
ans =
    10    14    31
    14    21    45
    31    45    98
>> exit
[mthomas@tscc-13-12 ~]$
```

TSCC Interactive Jobs: Running Matlab with GUI

To use a Graphical User Interface (GUI) as part of your interactive job, you will need to set up Xforwarding. Example below is for using XQuartz on a MAC. For examples for MacOS and Windows, see: http://systems.eecs.tufts.edu/x11-forwarding/

***Step 1:** Set X11 forwarding on the computer that you are connecting from: Update or install Xquartz and **restart** your system*

```
[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
```

***Step 2:** Log on to TSCC, using **–Y** option (trusted)*

```
[mthomas@home]$ ssh –Y tscc-login.sdsc.edu -lmthomas
```

***Step 3:** Request an interactive node, using **–X** option*

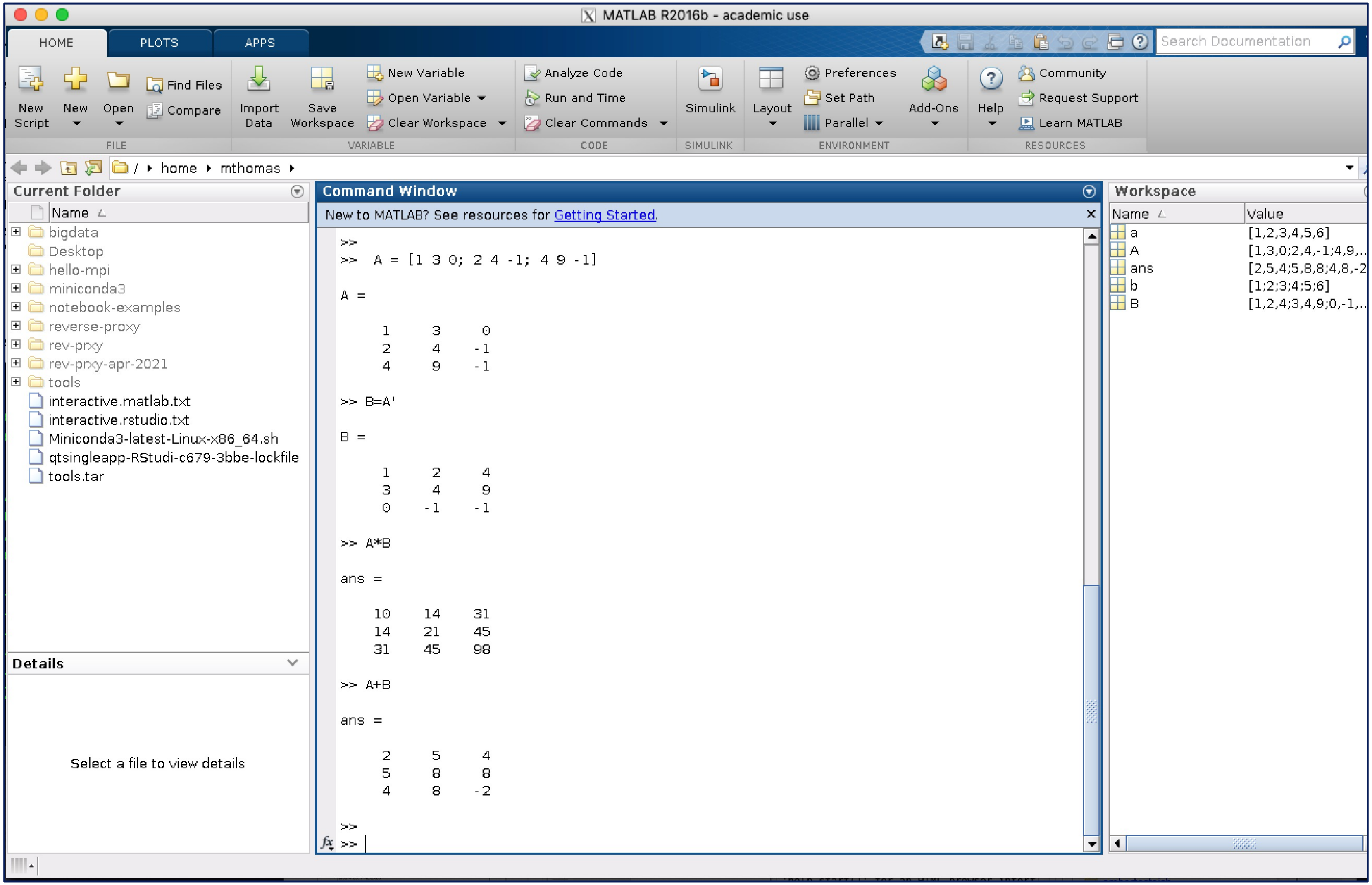
***Step 3:** Request an interactive node, using **–X** option*

```
[mthomas@tscc-login1 hello-mpi]$ qsub -I -X -l
nodes=1:ppn=1 -q glean
qsub: waiting for job 26843028.tscc-mgr7.local to start
qsub: job 26843028.tscc-mgr7.local ready
[mthomas@tscc-4-46 ~]$
```

***Step 4:** Setup your module environment and run Matlab*

```
[mthomas@tscc-4-46 ~]$ module load matlab
[mthomas@tscc-4-46 ~]$ matlab
MATLAB is selecting SOFTWARE OPENGL rendering.
```

TSCC Interactive Jobs: Running Matlab with GUI



Name

- ⊞ bigdata
- Desktop
- ⊞ hello-mpi
- ⊞ miniconda3
- ⊞ notebook-examples
- ⊞ reverse-proxy
- ⊞ rev-prxy
- ⊞ rev-prxy-apr-2021
- ⊞ tools
- interactive.matlab.txt
- interactive.rstudio.txt
- Miniconda3-latest-Linux-x86_64.sh
- qtsingleapp-RStudi-c679-3bbe-lockfile
- tools.tar

Select a file to view details

New to MATLAB? See resources for Getting Started.

```
>>
>>  A = [1 3 0; 2 4 -1; 4 9 -1]

A =

     1     3     0
     2     4    -1
     4     9    -1

>> B=A'

B =

     1     2     4
     3     4     9
     0    -1    -1

>> A*B

ans =

    10    14    31
    14    21    45
    31    45    98

>> A+B

ans =

     2     5     4
     5     8     8
     4     8    -2

>>
>> 
```

Name	Value
a	[1,2,3,4,5,6]
A	[1,3,0;2,4,-1;4,9,..
ans	[2,5,4;5,8,8;4,8,-2
b	[1;2;3;4;5;6]
B	[1,2,4;3,4,9;0,-1,..

TSCC Interactive Jobs: Running R console (no GUI)

***Step 3:** Request an interactive node, using **–X** option*

```
[mthomas@tscc-login2 ~]$ qsub -I -q glean -l nodes=1:ppn=1
qsub: waiting for job 26844488.tscc-mgr7.local to start
qsub: job 26844488.tscc-mgr7.local ready
[mthomas@tscc-4-46 ~]$
```

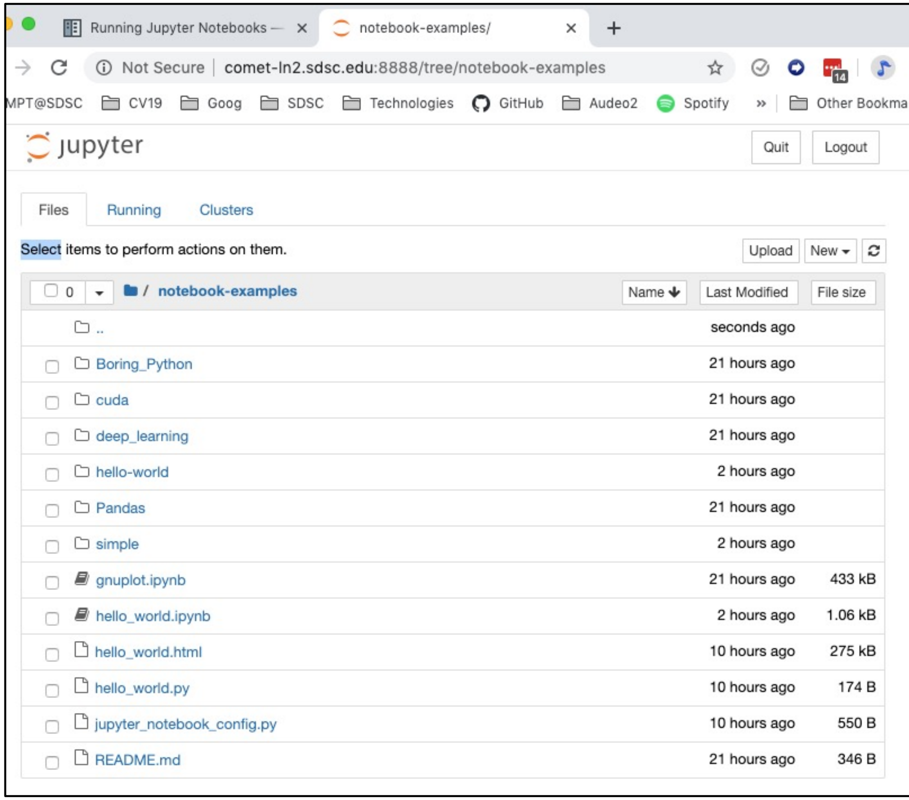
***Step 4:** Setup your module environment and run Matlab*

```
[mthomas@tscc-4-46 ~]$ module load R
Unloading compiler-dependent module openmpi_ib/3.1.4
[mthomas@tscc-4-46 ~]$ R
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
[SNIP]
R is a collaborative project with many contributors.
[SNIP]
> myString <- "Hello, World!"
> print ( myString )
[1] "Hello, World!"
>
```

TSCC Interactive Jobs: Juptyer Notebooks

- What is Jupyter?
  - Free, open-source, interactive web tool known as a computational notebook
- Jupyter Services:
  - Notebooks (single user)
  - JupyterLab: advanced version of notebook
  - JupyterHub: multiuser.
- Jupyter Has a Key Vulnerability:
  - Jupyter Servers Provide Access to HPC File Systems
  - By default serve up as HTTP
- SDSC has created a solution:
  - Satellite Reverse Proxy Service
  - See presentation by Marty Kandes in this workshop



Select items to perform actions on them.

/ notebook-examples

Name	Last Modified	File size
..	seconds ago	
Boring_Python	21 hours ago	
cuda	21 hours ago	
deep_learning	21 hours ago	
hello-world	2 hours ago	
Pandas	21 hours ago	
simple	2 hours ago	
gnuplot.ipynb	21 hours ago	433 kB
hello_world.ipynb	2 hours ago	1.06 kB
hello_world.html	10 hours ago	275 kB
hello_world.py	10 hours ago	174 B
jupyter_notebook_config.py	10 hours ago	550 B
README.md	21 hours ago	346 B

Resources

- TSCC User Guide
  - https://www.sdsc.edu/support/user_guides/tscc.html
- GitHub Repo for this workshop:
  - https://github.com/sdsc-hpc-training-org/Using-Python-and-Jupyter-Notebooks-on-TSCC/
- GitHub Repo for Example Notebooks:
  - https://github.com/sdsc-hpc-training-org/notebook-examples
- TSCC Support email: tscc-support@ucsd.edu
- TSCC Forum –
  - https://mailman.ucsd.edu/mailman/listinfo/tscc-l
  - Email: tscc-l@mailman.ucsd.edu
- SDSC Training Resources
  - https://www.sdsc.edu/education_and_training/training

Thank You