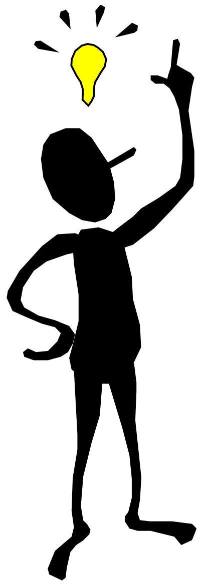
INTRODUCTION TO RIVANNA

Last revised: 02/15/2019

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Terminology



- Node
  - Basic building block of a cluster
  - Usually a specialized computer
- Two types of nodes:
  - Head Node – computer used for logging on and submitting jobs
  - Compute Node -- computer that does most of the work
- Core – an individual processor on a computer

ALLOCATIONS & ACCOUNTS

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Allocations

- Rivanna is allocated:

  At the most basic level, an allocation refers to a chunk of CPU time that you receive and can use too run your computation.

- Allocations are measured in service units (SUs), where

  1 SU = 1 core-hour

- All accounts on a given allocation share the service units.

Requesting an Allocation

- Faculty (including postdocs) are eligible for an allocation (see www.arcs.virginia.edu/allocations).
- Students must be sponsored by a PI (e.g., an advisor, a professor, a research mentor).
- The PI must complete the form at
  https://arcs.virginia.edu/allocation
  - To get to the form, scroll down and click on "Request a New or Renewal Standard Allocation"

University of Virginia
Advanced Research Computing Services

CONNECTING & LOGGING ON TO RIVANNA

How to connect to Rivanna

- There are three ways to connect to Rivanna:

1. ssh client
    - Instructions for installing and using an ssh client are provided in the appendix of these slides.

2. FastX
    - Using your web browser, go to URL https://rivanna-desktop.hpc.virginia.edu and log in.
    - Click on "Launch Session"; Select "MATE" and click on "Launch"

3. OpenonDemand https://rivanna-portal.hpc.virginia.edu
    - Authenticates with Netbadge.

Ssh and FastX require the UVa Anywhere VPN when off-grounds.

See http://its.virginia.edu/vpn/ for details.

Connecting to the Cluster

- The hostname for the Interactive frontends:
  rivanna.hpc.virginia.edu

  (does round-robin among the front-ends)

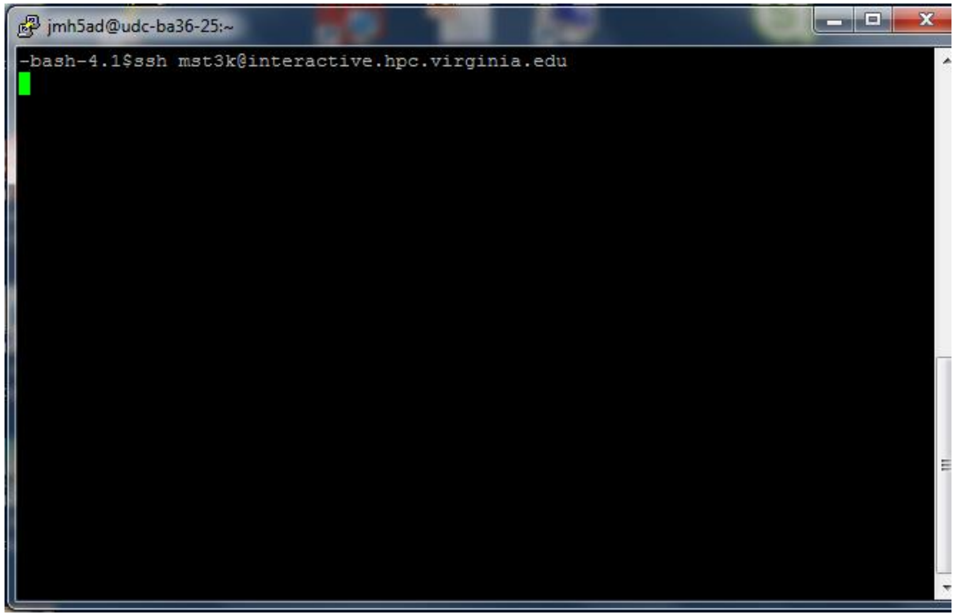
- However, you also can log onto a specific front-end:
  - rivanna1.hpc.virginia.edu
  - rivanna2.hpc.virginia.edu
  - rivanna3.hpc.virginia.edu

Connecting to the Cluster with ssh

- If you are on a Mac or Linux machine your can connect with ssh.
- Bring up a terminal window and type:

  ssh –Y *userID*@rivanna.hpc.virginia.edu

- When it prompts you for for a password, use your Eservices password.

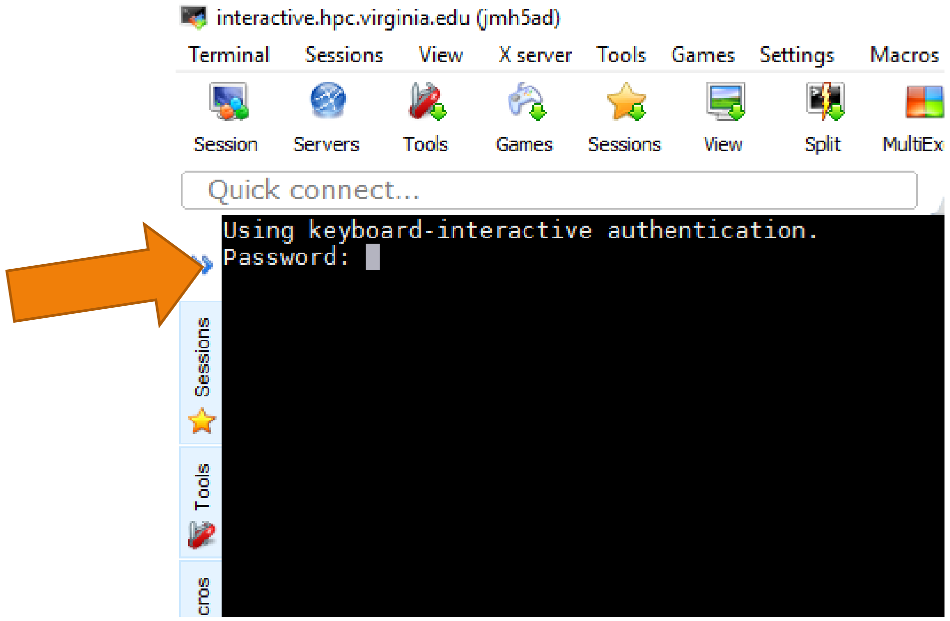

jmh5ad@udc-ba36-25:~

```
-bash-4.1$ssh mst3k@interactive.hpc.virginia.edu
```

University *of* Virginia
Advanced Research Computing Services

Connecting to the Cluster with MobaXterm

- We recommend MobaXterm for Windows users.
- It will prompt you for your password but will not echo asterisks. It can also remember your password.

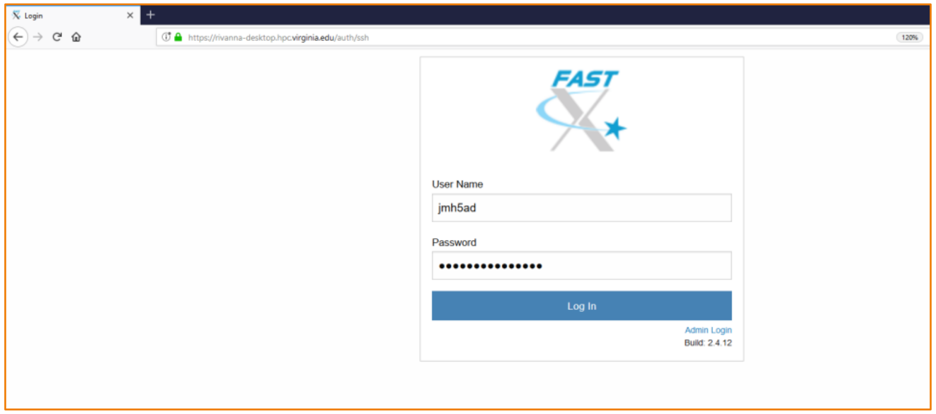

interactive.hpc.virginia.edu (jmh5ad)
Terminal Sessions View X server Tools Games Settings Macros
Session Servers Tools Games Sessions View Split MultiEx
Quick connect...
Sessions
Tools
cros

```
Using keyboard-interactive authentication.
Password:
```

FastX

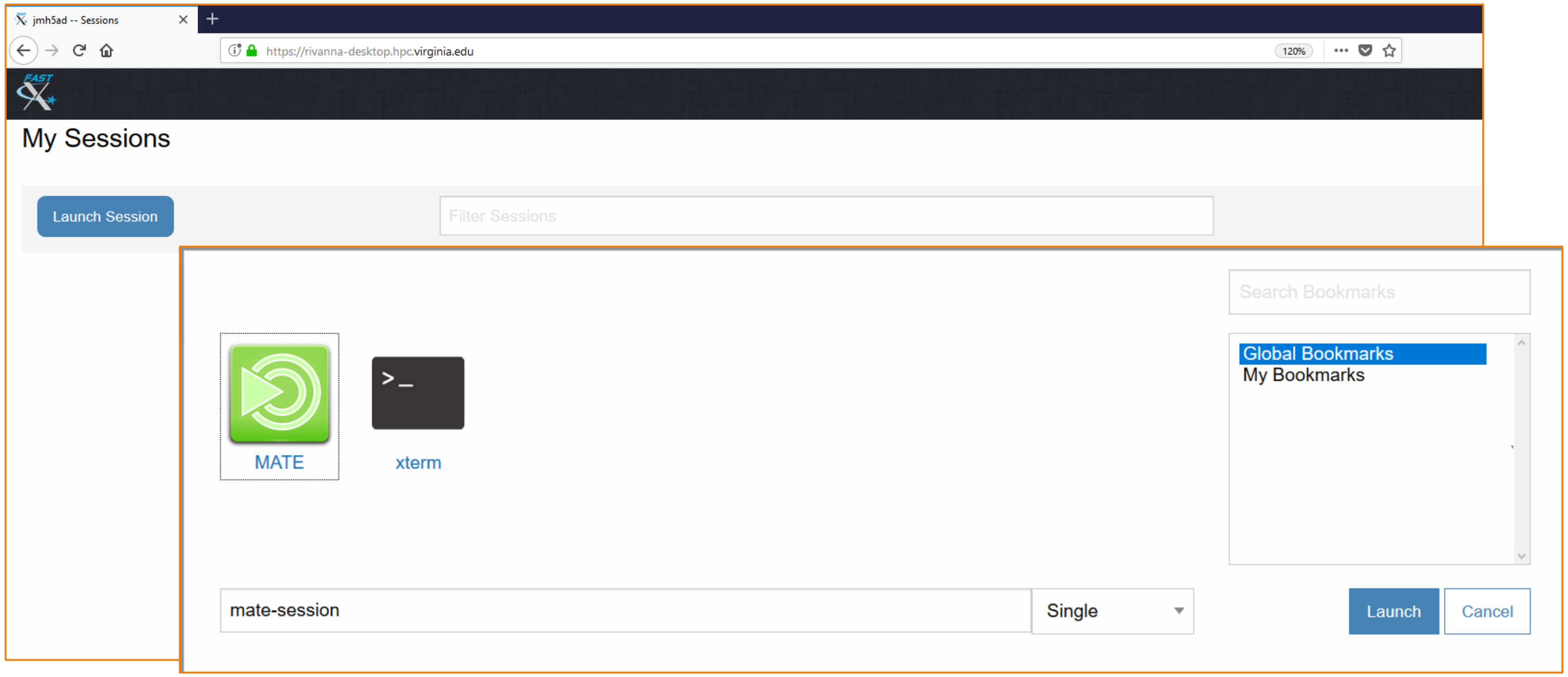
- In your web browser, go to URL:

  https://rivanna-desktop.hpc.virginia.edu



Starting up FastX

- Click "Launch Session"; Select MATE; Click Launch



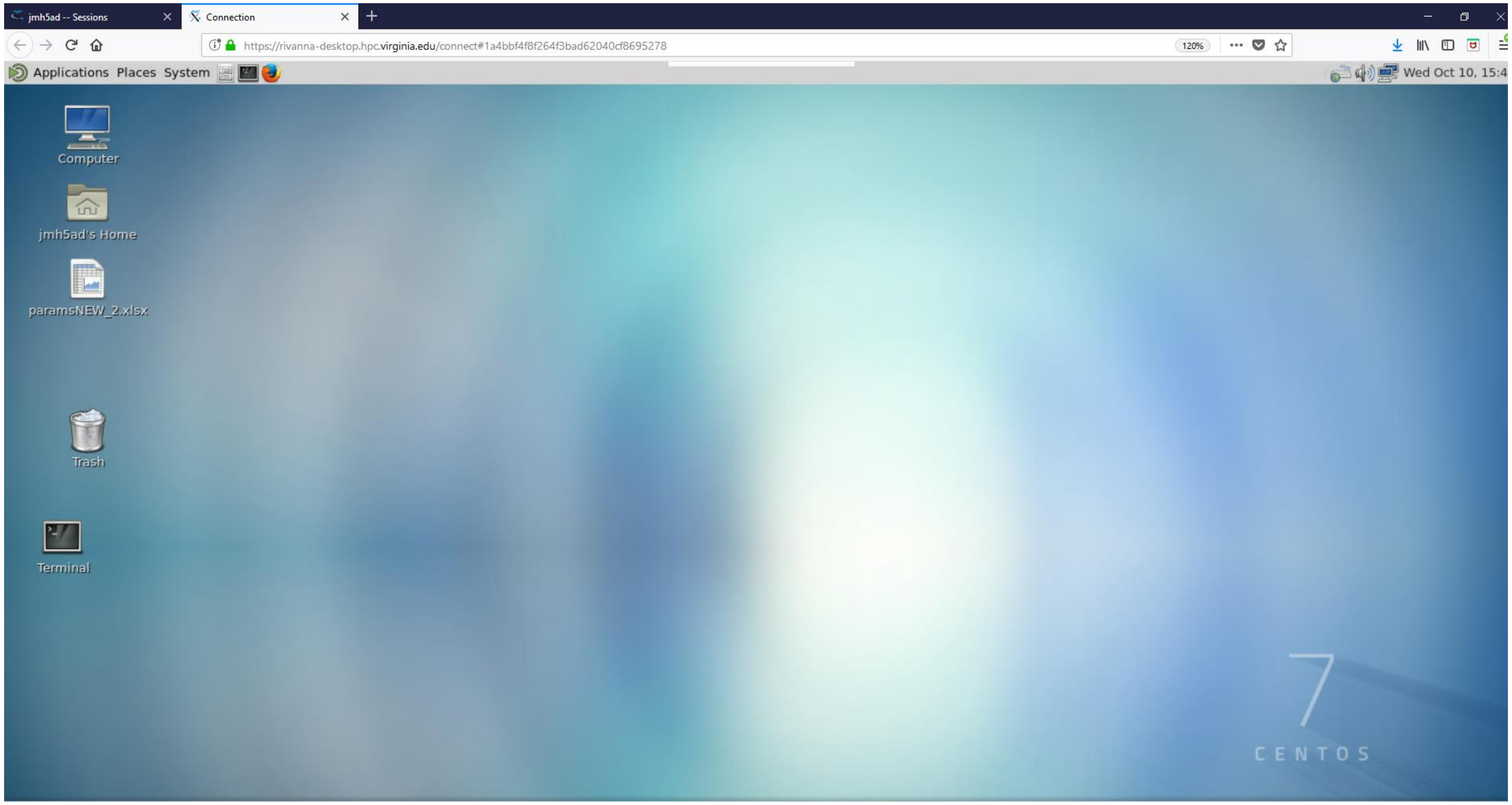
mate-session

Single

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

FastX Environment

- A desktop for working on Rivanna


jmh5ad -- Sessions
Connection
https://rivanna-desktop.hpc.virginia.edu/connect#1a4bbf4f8f264f3bad62040cf8695278
120%
Applications Places System
Wed Oct 10, 15:4
Computer
jmh5ad's Home
paramsNEW_2.xlsx
Trash
Terminal
7
CENTOS

CLUSTER ENVIRONMENT

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Your Home Directory

- The default home directory on Rivanna has 50GB of storage capacity
  - This directory is distinct from the 4GB home directory provided by ITS.
  - The ITS home directory is available as `/tiny/$USER`

Checking your Home Storage

- To see how much disk space you have used in your home directory, open a Terminal window and type **`hdquota`** at the command-line prompt:

```
$ hdquota


Filesystem  |  Used  |  Avail  |  Limit  |  Percent Used
  qhome        39G       12G       51G          77%
```

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Leased (Group) Storage

- Groups can lease space for longer-term storage
  - Project: has snapshots.
  - Value: no snapshots
  - Overview at
    - https://arcs.virginia.edu/storage

Your /scratch Directory

- Each user will have access to 10 TB of **temporary** storage.
  - It is located in a subdirectory under /scratch, and named with your userID
  - e.g., `/scratch/mst3k`
  - You are limited to 350,000 files in your scratch directory.

Important:
/scratch is **NOT permanent** storage and files older than **90 days** will be marked for deletion.

UNIVERSITY *of* VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Running Jobs from Scratch

- We recommend that you run your jobs out of your /scratch directory for two reasons:
    - /scratch is on a Lustre filesystem (a storage system designed specifically for parallel access).
    - /scratch is connected to the compute nodes with Infiniband (a very fast network connection).

We also recommend that

- You keep copies of your programs and data in more permanent locations (e.g., your home directory or leased storage).
- After your jobs finish, you copy the results to more permanent storage.

Checking your /scratch Storage

- To see the amount of scratch space that is available to you, type **sfsq** at the command line prompt.

```
$ sfsq

'scratch' usage status for 'mst3k', last
updated: 2016-09-08 16:26:12

 - ~28/10,000 GBs allocated disk space
 - 153/350,000 files created
 - 151/153 files marked for deletion due to
age limits

To view a list of all files marked for
deletion, please run 'sfsq -l'
```

Checking your Allocation

- To see how many SUs you have available for running jobs, type **`allocations`** at the command-line prompt:

```
$ allocations

Allocations available to Misty S. Theatre(mst3k):

 * robot_build: less than 6,917 service-units remaining.
 * gizmonic-testing: less than 5,000 service-units remaining.
 * servo: less than 59,759 service-units remaining, allocation will expire on
2017-01-01.
 * crow-lab: less than 2,978 service-units remaining.
 * gypsy: no service-units remaining
```

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Moving data onto Rivanna

- You have several options for transferring data onto your home or /scratch directories.
  1. Use the scp command in a terminal window.
  2. Use a drag-and-drop option with MobaXterm (Windows) or Fugu (Mac OS). Cyberduck and Filezilla are cross-platform (but always use ssh/scp protocol)
  3. Use OpenOnDemand for small files.
  4. Set up a Globus endpoint on your local computer and use the Globus web interface to transfer files.

     (See https://arcs.virginia.edu/globus for details)

MODULES

University of Virginia
Advanced Research Computing Services

Modules

- Any application software that you want to use will need to be loaded with the `module load` command.
- For example:
    - module load matlab
    - module load anaconda/5.2. 0-py3.6
    - module load gcc R/3.5.1
- You will need to load the module any time that you create a new shell
    - Every time that you log out and back in
    - Every time that you run a batch job on a compute node

Module Details

- `module avail` – Lists all available modules and versions for a given hierarchy (compiler or compiler+MPI).
- `module spider` – Shows all available modules
- `module key <keyword>` – Shows modules with the keyword in the description
- `module list` – Lists modules loaded in your environment.
- `module load mymod` – Loads the default module to set up the environment for some software.
  - module load mymod/N.M – Loads a specific version
- `module purge` – Clears all modules.

Learning more about a Module

- To locate a python module, try the following:

```
  $ module avail python

  $ module spider python

  $ module key python
```

- To find chemistry/materials software packages, try this:

```
  $ module key chem
```

- The available software is also listed on our website:
  https://arcs.virginia.edu/software-list

PARTITIONS (QUEUES)

Partitions (Queues)

- Rivanna has several partitions (or queues) for job submissions.
  - You will need to specify a partition when you submit a job.
  - To see the partitions that are available to you, type `queues` at the command-line prompt.

```
$ queues

Queue        Availability   Time       Queue    Maximum    Maximum    Idle    SU     Usable
(partition)  (idle%)        Limit      Limit    Cores/Job  Mem/Core   Nodes   Rate   Accounts

standard     43 13(72.2%)   7-days     none     20         64-GB      195     1.00   robot-build, gypsy
dev          1833(65.2%)    1 hours    none     4          254GB      59      0.00   robot-build, gypsy
parallel     3528(73.5%)    3-days     none     240        64-GB      176     1.00   robot-build, gypsy
largemem     48(60.0%)      7-days     none     16         500-GB     3       1.00   robot-build, gypsy
gpu          334(85.0%)     3-days     none     8          128-GB     10      1.00   robot-build, gypsy
knl          2048(100.0%)   3-days     none     2048       1-GB       8       1.00   robot-build, gypsy
```

Compute Node Partitions (aka Queues)

Queue Name	Purpose	Job Time Limit	Memory / Node	Cores / Node	# of Available Nodes	SU / Core Hour
standard	For jobs on a single compute node	7 days	256 GB 384 GB	28 40	26+108	1.0
gpu	For jobs that can use general purpose graphical processing units (GPGPUs) (K80 or P100)	3 days	256 GB	28	13 (max 4 nodes per job)	1.0 (may go up to 2.0)
parallel	For large parallel jobs on up to 120 nodes (<= 2400 CPU cores)	3 days	128 GB	20	220 (shared w/ standard queue)	1.0
largemem	For memory intensive jobs (<= 16 cores/node)	7 days	1 TB	16	5 (max 2 per user)	1.0
dev	To run jobs that are quick tests of code	1 hour	128 GB	4	2	0.0

SLURM SCRIPTS

SLURM

- SLURM is the Simple Linux Utility for Resource Management.
  - It manages the hardware resources on the cluster (e.g. compute nodes/cpu cores, compute memory, etc.).
- SLURM allows you to request resources within the cluster to run your code.
  - It is used for submitting jobs to compute nodes from an access point (generally called a *frontend*).
  - Frontends are intended for editing, compiling, and very short test runs.
  - Production jobs go to the compute nodes through the resources manager.
- SLURM documentation:

  https://arcs.virginia.edu/slurm

  http://slurm.schedmd.com/documentation.html

SLURM Script

- A SLURM script is a bash script with SLURM directives (#SBATCH) and command-line instructions for running your program.

```
#!/bin/bash
#SBATCH --nodes=1                #total number of nodes for the job
#SBATCH --ntasks=1               #how many copies of code to run
#SBATCH --time=1-12:00:00        #amount of time for the whole job
#SBATCH --partition=standard     #the queue/partition to run on
#SBATCH --account=myGroupName    #the account/allocation to use

module purge
module load gcc/7.1.0            #load modules that my job needs
./mycode                         #command-line execution of my job
```

Submitting a SLURM Job

- To submit the SLURM command file to the queue, use the **`sbatch`** command at the command line prompt.

- For example, if the script on the previous slide is in a file named job_script.slurm, we can submit it as follows:

```
-bash-4.1$ sbatch job_script.slurm
Submitted batch job 18316
```

Checking Job Status

- To display the status of only your ***active*** jobs, type:

  `squeue –u <your_user_id>`

```
-bash-4.1$ squeue –u mst3k

JOBID    PARTITION    NAME      USER    ST    TIME    NODES   NODELIST(REASON)
18316    standard     job_sci   mst3k   R     1:45     1      udc-aw38-34-1
```

- The `squeue` command will show pending jobs and running jobs, but not failed, canceled or completed job.

Checking Job Status

- To display the status of all jobs, type:

  sacct -S <start_date>

```
-bash-4.1$ sacct -S 2019-01-29

3104009       RAxML_NoC+   standard  hpc_build          20   COMPLETED       0:0
3104009.bat+       batch             hpc_build          20   COMPLETED       0:0
3104009.0     raxmlHPC-+             hpc_build          20   COMPLETED       0:0
3108537       sys/dashb+        gpu  hpc_build           1  CANCELLED+       0:0
3108537.bat+       batch             hpc_build           1   CANCELLED      0:15
3108562       sys/dashb+        gpu  hpc_build           1     TIMEOUT       0:0
3108562.bat+       batch             hpc_build           1   CANCELLED      0:15
3109392       sys/dashb+        gpu  hpc_build           1     TIMEOUT       0:0
3109392.bat+       batch             hpc_build           1   CANCELLED      0:15
3112064             srun        gpu  hpc_build           1      FAILED       1:0
3112064.0           bash             hpc_build           1      FAILED       1:0
```

- The `sacct` command lists all jobs (pending, running, completed, canceled, failed, etc.) since the specified date.

University *of* Virginia Advanced Research Computing Services

Deleting a Job

- To delete a job from the queue, use the **scancel** command with the job ID number at the command line prompt:

```
-bash-4.1$ scancel 18316
```

MORE ADVANCED JOBS

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

High Throughput Job

- High throughput computing (HTC) runs a large number of serial jobs (or sometimes minimally parallel jobs).
- Usually the computations are identical but may use different input files and should produce different output files.
- Job arrays are usually the best way to handle HTC.
- You also can use job arrays to organize the input and output.

Job Arrays

- Create a batch script describing how to do *one* job.

```
#!/bin/bash
#SBATCH --nodes=1                  #total number of nodes for the job
#SBATCH --ntasks=1                 #how many processes I will run
#SBATCH --time=00:05:00            #amount of time for the whole job
#SBATCH --partition=standard       #the queue/partition I will run on
#SBATCH --account=Your_group_name #the account/allocation

module purge
module load gcc R/3.4.0
#command-line execution of my job with command-line arguments

Rscript hello.R ${SLURM_ARRAY_TASK_ID} `pwd`
```

- And, submit by typing:

```
sbatch --array=1-30 hello.slurm
```

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Job Array Numbering

- An increment can be provided

```
sbatch --array=1-7:2 myjob.sh
```

  - This will number them 1, 3, 5, 7

- Or provide a list

```
sbatch --array=1,3,4,5,7,9 myjobs.sh
```

Job Array Environment Variables

- Each job will be provided an environment variable

  `SLURM_ARRAY_JOB_ID`

- And each task will be assigned

  `SLURM_ARRAY_TASK_ID`

  based on the numbers in the range or list specified with --array.

- You can use these environment variables as labels for input/output files, directories, etc.
  - In the SLURM script, a variable

    `%A` represents the overall `SLURM_ARRAY_JOB_ID` and
    `%a` represents `SLURM_ARRAY_TASK_ID`

  - These variables can be used with output and input file names.

Array Script

- Job arrays *should* be named (most jobs don't have to be named).

```
#SBATCH --job-name=<name>
```

or

```
#SBATCH -J <name>
```

- All subjobs will use the same global resource requests.

UNIVERSITY *of* VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Output File Specifications

- It would be prudent to separate stdout and stderror in this case, and give them names corresponding to job and task IDs, such as:

```
#SBATCH -o myjobs.%A_%a.out
#SBATCH -e myjobs.%A_%a.err
```

Hands-on Activity:

Modify the file 02_jobArray/hello.slurm to create separate files for output and error.

Multicore in SLURM

- Multicore programs run on a single node
- Different libraries, example will be for OpenMP
- SLURM scripts for multicore programs should use the following combination of directives:

```
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=M
#where M is replaced
#with the actual number
#of cores that you want
```

Requesting Cores for Threads

- Update SLURM script

```
pluma hello_mc.slurm
```

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=10      #number of cores requested
#SBATCH --time=00:10:00
#SBATCH --partition=standard
#SBATCH --account=<Your_group_name>

module purge
module load gcc
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}
./myexec
```

- And, submit by typing:

```
sbatch hello_mc.slurm
```

Multinode in SLURM

- Multinode programs run across nodes. Nearly all use MPI.
- SLURM scripts for multinode programs should use the following combination of directives:

```
#SBATCH --nodes=N
#SBATCH --ntasks-per-node=M
#SBATCH --partition=parallel
```

- Try to fill nodes when possible (20 cores per node on parallel)

Requesting Cores for MPI

```
#!/bin/bash
#SBATCH --nodes=5
#SBATCH –ntasks-per-node=20
#SBATCH --time=00:10:00
#SBATCH --partition=parallel
#SBATCH --account=<Your_group_name>

module purge
module load intel
module load intelmpi
srun myexec
```

- And, submit by typing:

```
sbatch hello_mc.slurm
```

ACCESSING GPU NODES

Using GPUs

- Certain applications can utilize for general purpose graphics processing units (GPGPUs) to accelerate computations.
- GPGPUs on Rivanna:
  - K80: dual GPUs per board, can do double precision
  - P100: single GPUs per board, double precision is software (slow), better for machine learning
  - More on the way (1 V100, 2 RTX2080)
  - Build with gcc/5.4.0
- You must first request the `gpu` queue. Then with the `gres` option, type the architecture (if you care) and the number of GPUs.

```
#SBATCH -p gpu
#SBATCH --gres=gpu:k80:2
```

JupyterLab

- JupyterLab is a web-based tool that allows multiple users to run Jupyter notebooks on a remote system.

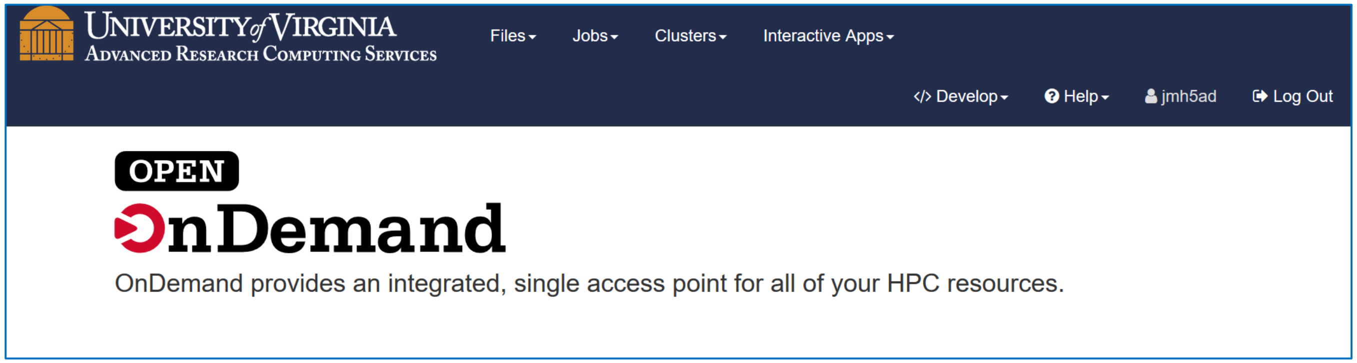
- We now provide JupyterLab on Rivanna.

Accessing JupyterLab

- To access JupyterLab, type the following in your web browser:

  https://rivanna-portal.hpc.virginia.edu/

- After logging in via Netbadge in, you will be directed to the Open OnDemand main page.

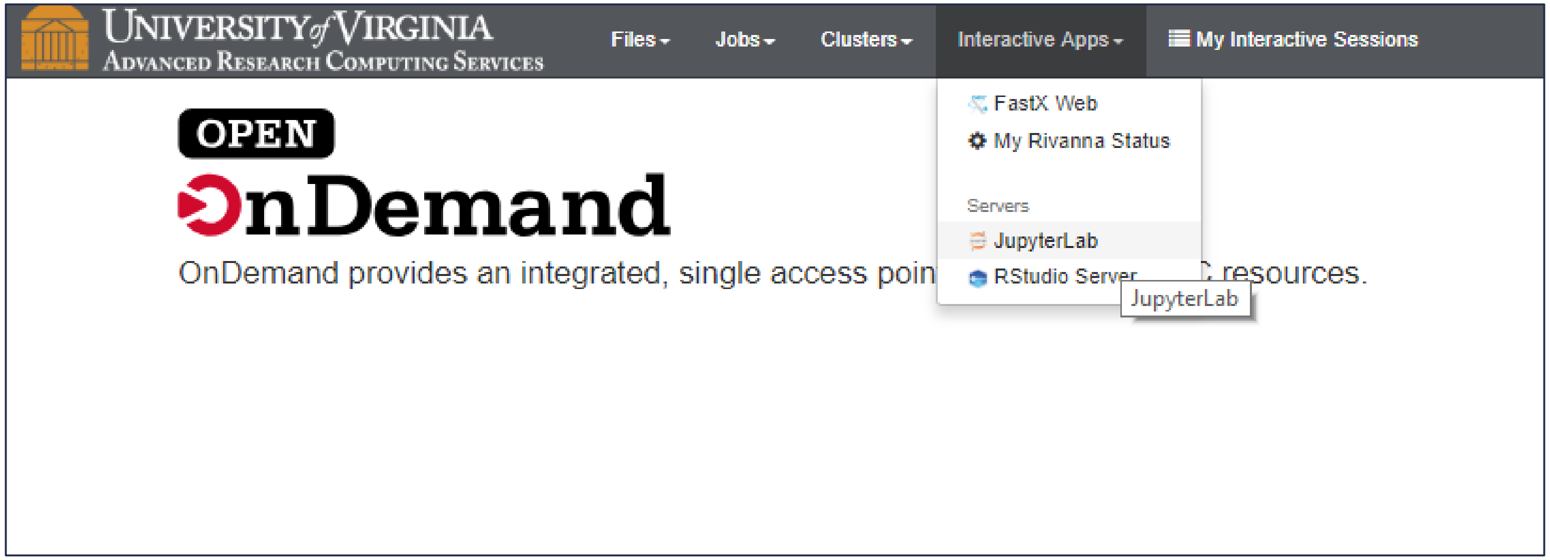


OnDemand provides an integrated, single access point for all of your HPC resources.

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Starting Jupyter Instance

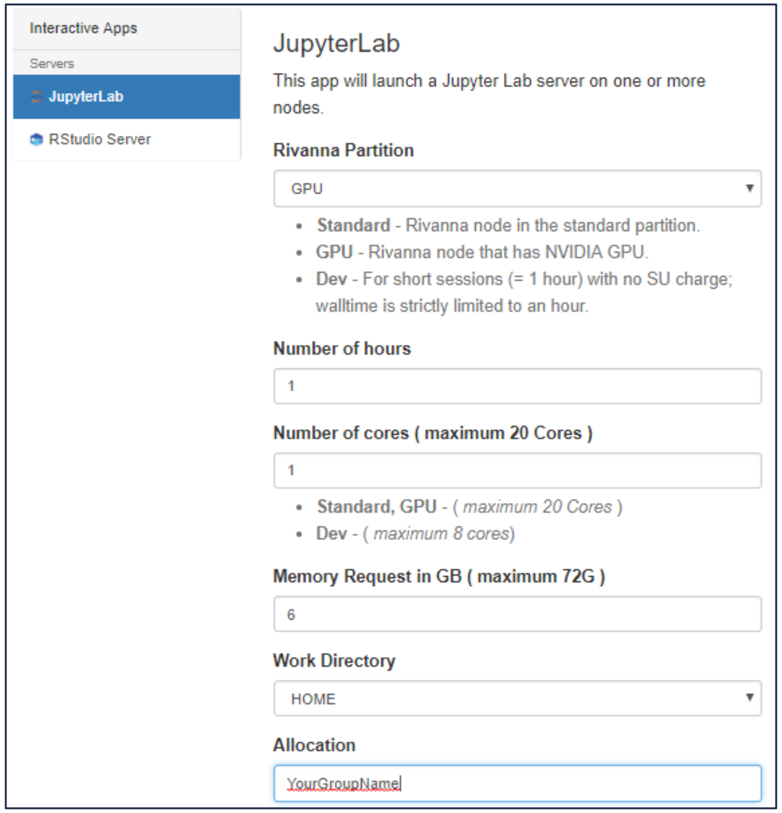
- In the top, click on “Interactive Apps” and in the drop-down box, click on “Jupyter Lab”.



OnDemand provides an integrated, single access poin… resources.

Starting a Jupyter Instance

- A form will appear that allows you to specify the resources for your Notebook.
  - Our example will be using TensorFlow; so, we need to make sure that we select the Rivanna Partition called "GPU".
  - Also, don't forget to put in your "MyGroup" name for the Allocation
  - Finally, click the blue "Launch" button at the bottom of the form (not shown here).



JupyterLab

This app will launch a Jupyter Lab server on one or more nodes.

Rivanna Partition

GPU

- **Standard** - Rivanna node in the standard partition.
- **GPU** - Rivanna node that has NVIDIA GPU.
- **Dev** - For short sessions (= 1 hour) with no SU charge; walltime is strictly limited to an hour.

Number of hours

1

Number of cores (maximum 20 Cores)

1

- **Standard, GPU** - (*maximum 20 Cores*)
- **Dev** - (*maximum 8 cores*)

Memory Request in GB (maximum 72G)

6

Work Directory

HOME

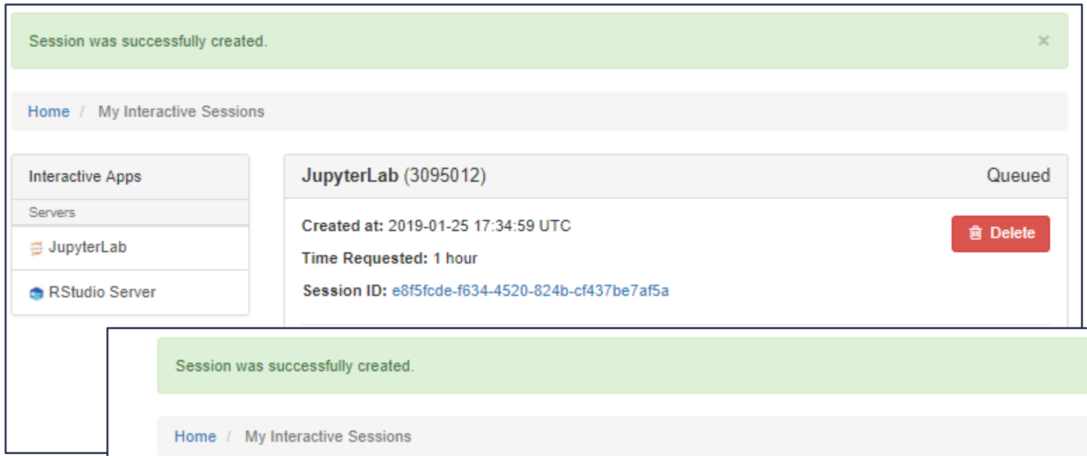
Allocation

YourGroupName

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Starting a Jupyter Instance

- It may take a little bit of time for the resources to be allocated.



Session was successfully created.

Home / My Interactive Sessions

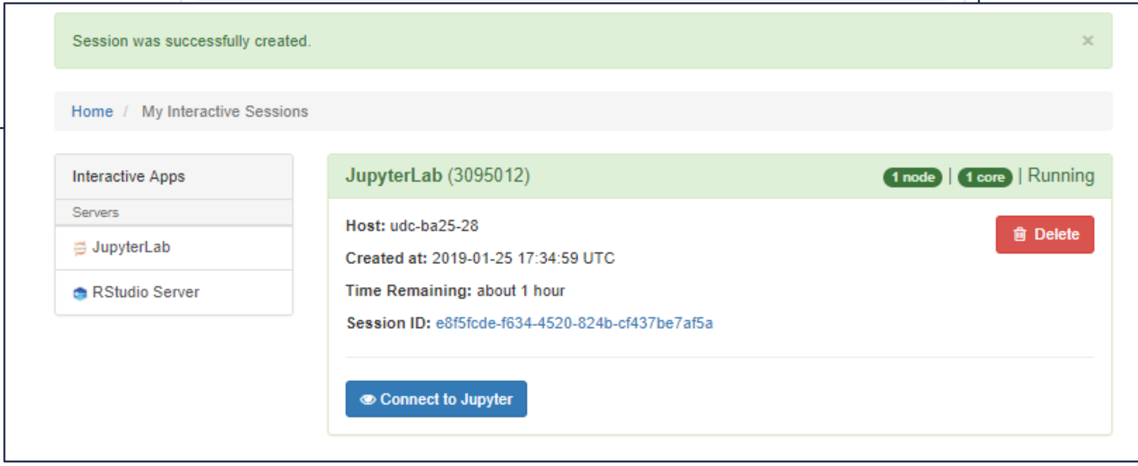
Interactive Apps
Servers
JupyterLab
RStudio Server

JupyterLab (3095012) Queued

Delete

Created at: 2019-01-25 17:34:59 UTC
Time Requested: 1 hour
Session ID: e8f5fcde-f634-4520-824b-cf437be7af5a

- Wait until a blue button with "Connect to Jupyter" appears.



Session was successfully created.

Home / My Interactive Sessions

Interactive Apps
Servers
JupyterLab
RStudio Server

JupyterLab (3095012) 1 node | 1 core | Running

Delete

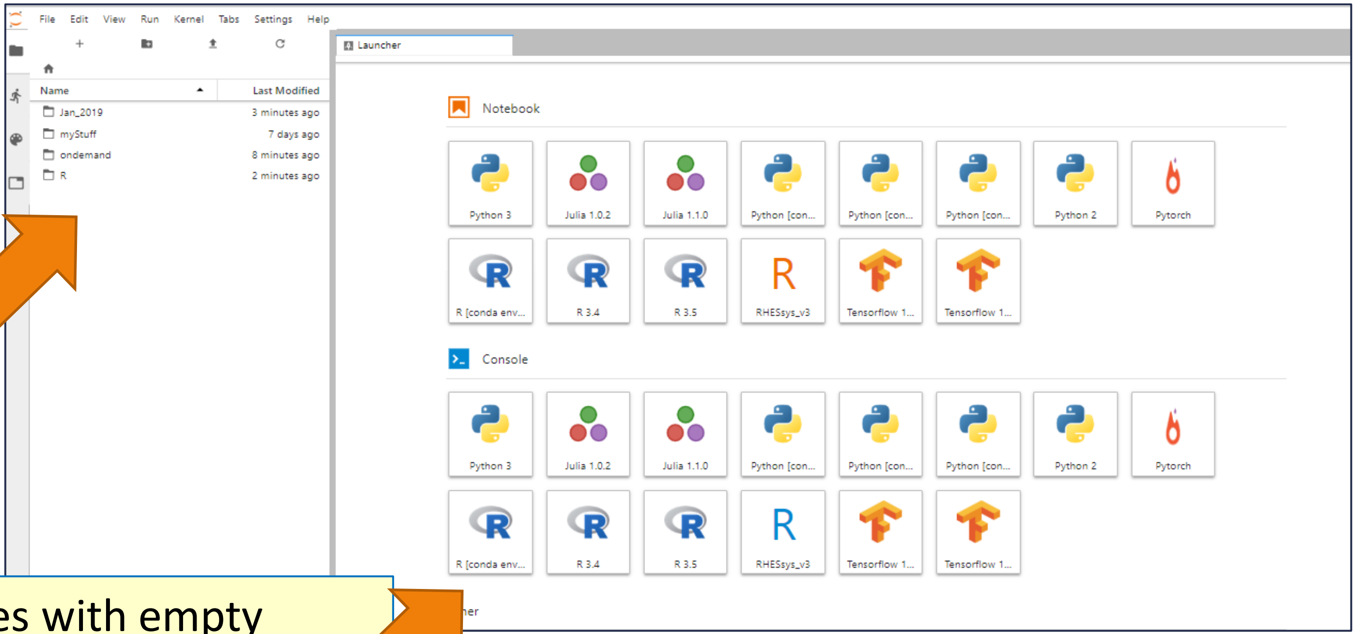
Host: udc-ba25-28
Created at: 2019-01-25 17:34:59 UTC
Time Remaining: about 1 hour
Session ID: e8f5fcde-f634-4520-824b-cf437be7af5a

Connect to Jupyter

- Click on the blue button.

University of Virginia
Advanced Research Computing Services

JupyterLab Environment

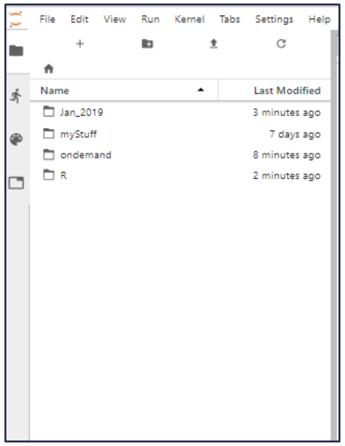


You should see a list of folders and files in your home directory.

And, a set of tiles with empty notebooks or consoles.

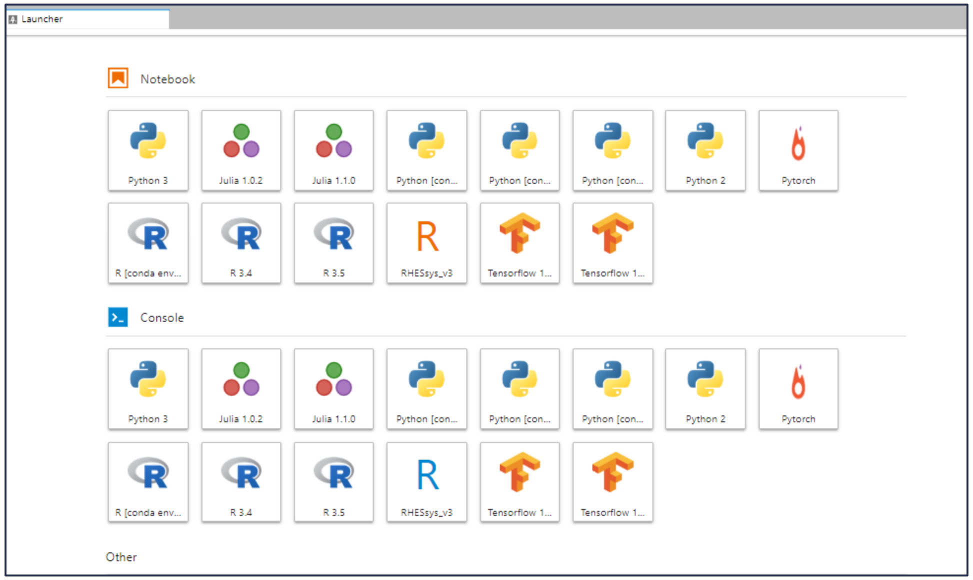
Opening a Notebook

- If you have an existing notebook, you can use the left-pane to maneuver to the file and click on it to open it.



Name	Last Modified
Jan_2019	3 minutes ago
myStuff	7 days ago
ondemand	8 minutes ago
R	2 minutes ago

- Or, if you want to start a new notebook, you can click on the notebook tile, for the appropriate underlying system.

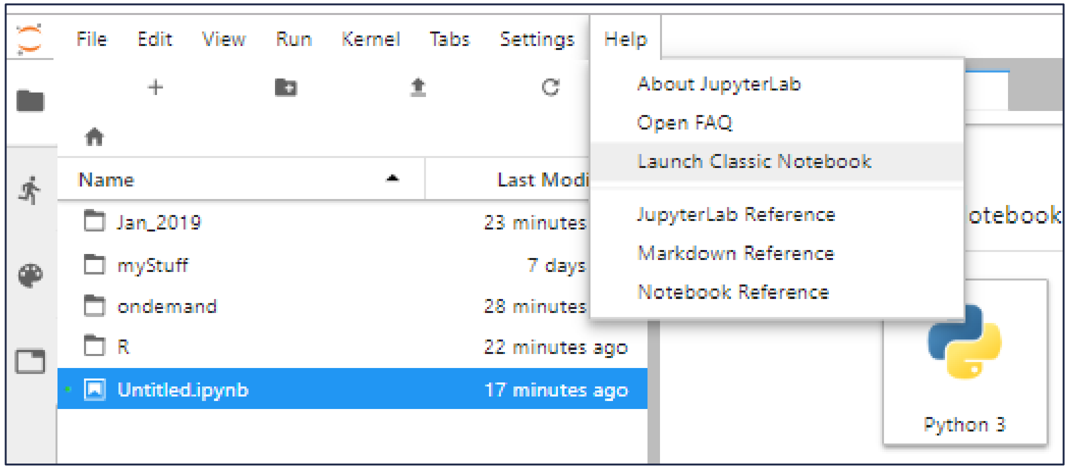


UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

Classic Notebook

- If you feel more comfortable working with the former Jupyter interface, you can select:

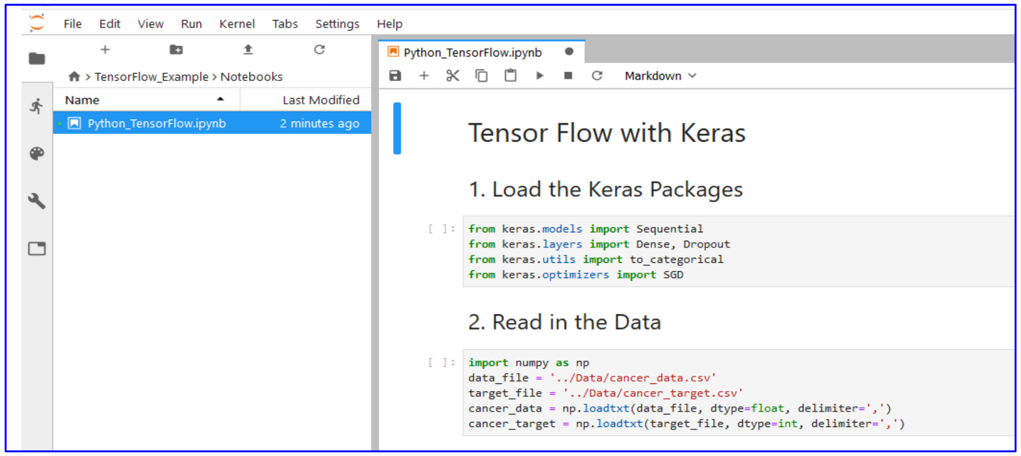
  Help> Launch Classic Notebook



- But, for our example, we will stay with the Jupyter Lab format.

Cautions

- Any changes that you make to the notebook may be saved automatically.
- When the time for your session expires, the session will end without warning.
- Your Jupyter session will continue running until you delete it.
  - Go back to the "Interactive Sessions" tab.
  - Click on the red Delete button.



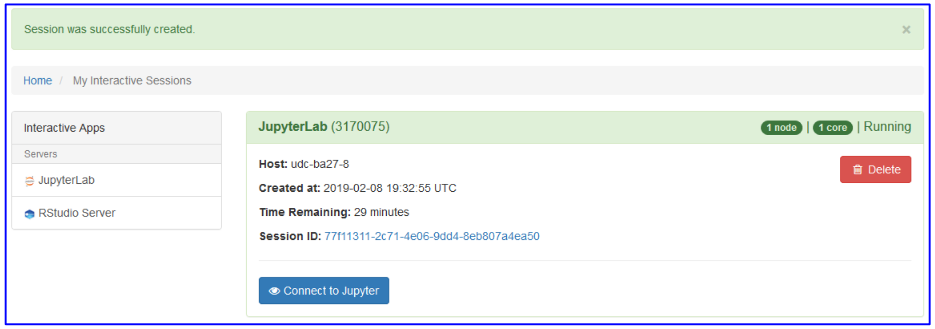
Tensor Flow with Keras

1. Load the Keras Packages

```
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.utils import to_categorical
from keras.optimizers import SGD
```

2. Read in the Data

```
import numpy as np
data_file = '../Data/cancer_data.csv'
target_file = '../Data/cancer_target.csv'
cancer_data = np.loadtxt(data_file, dtype=float, delimiter=',')
cancer_target = np.loadtxt(target_file, dtype=int, delimiter=',')
```



Session was successfully created.

JupyterLab (3170075) | Running

Host: udc-ba27-8
Created at: 2019-02-08 19:32:55 UTC
Time Remaining: 29 minutes
Session ID: 77f11311-2c71-4e06-9dd4-8eb807a4ea50

University of Virginia
Advanced Research Computing Services

NEED MORE HELP?

<u>Office Hours</u>

Tuesdays:	3 pm - 5 pm, PLSB 430
Thursdays:	10 am - noon, HSL, downstairs
Thursdays:	3 pm - 5 pm, PLSB 430

Website:

arcs.Virginia.edu

Or, for immediate help:

hpc-support@virginia.edu

University of Virginia
Advanced Research Computing Services