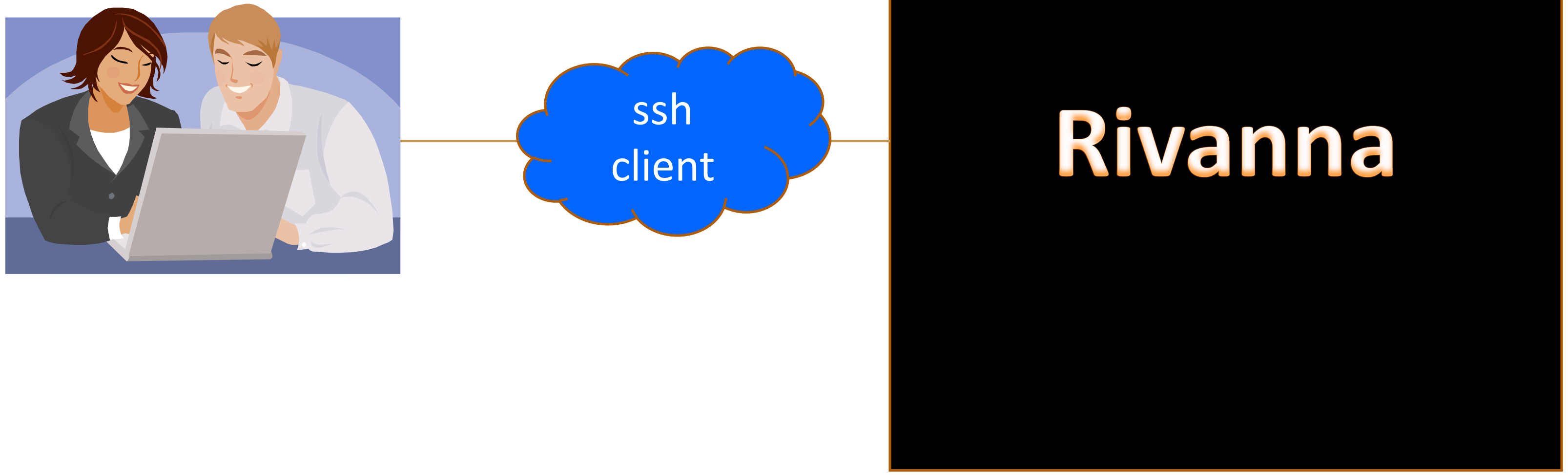
INTRODUCTION TO RIVANNA

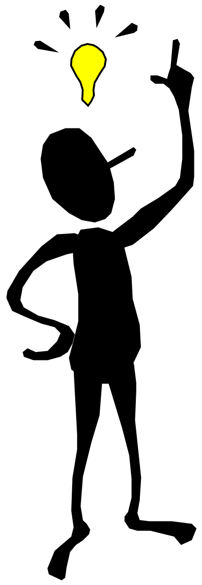
Last revised: 02/15/2019

University of Virginia
Advanced Research Computing Services

Your View of Rivanna

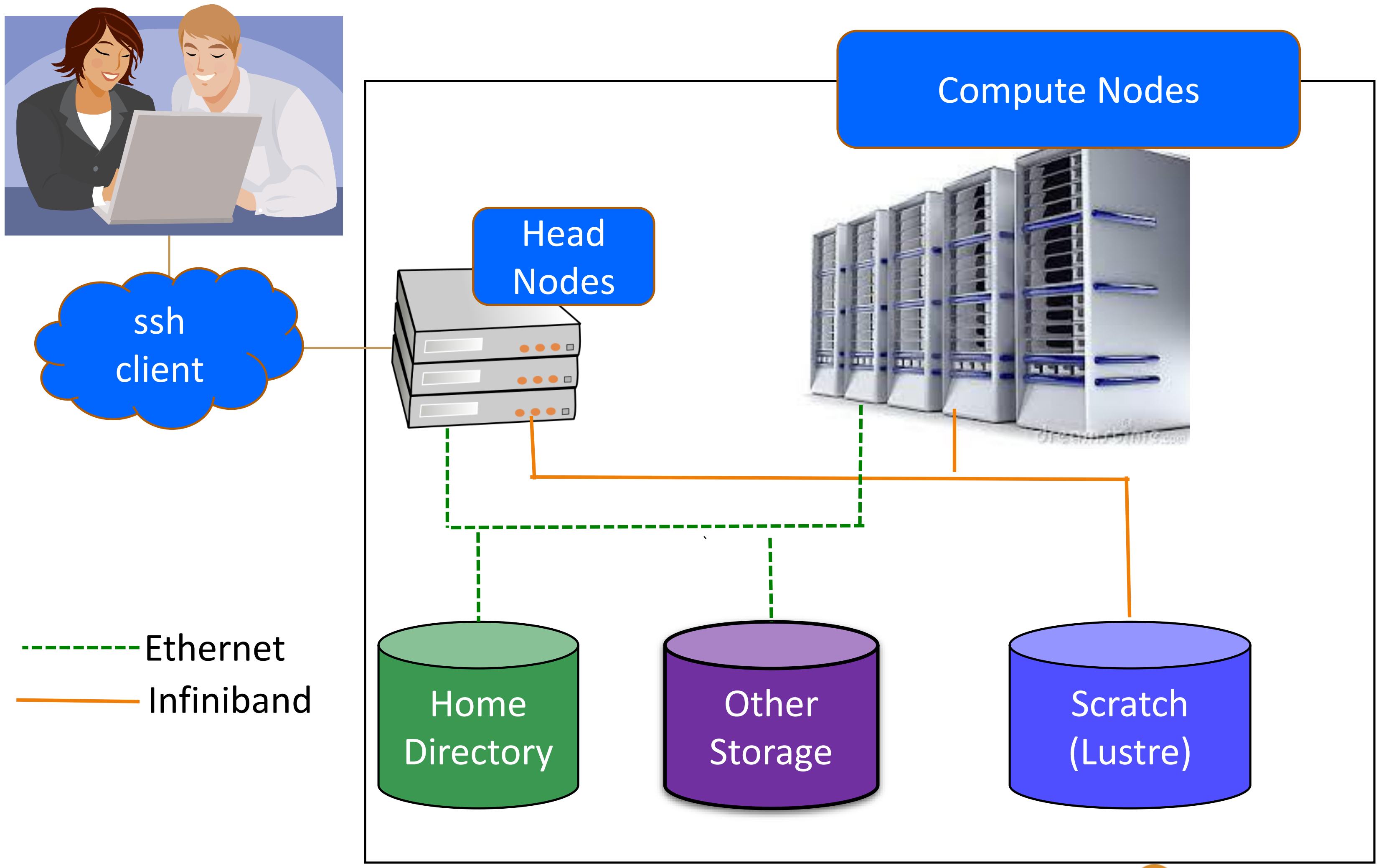


Terminology



- 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

Rivanna in More Detail

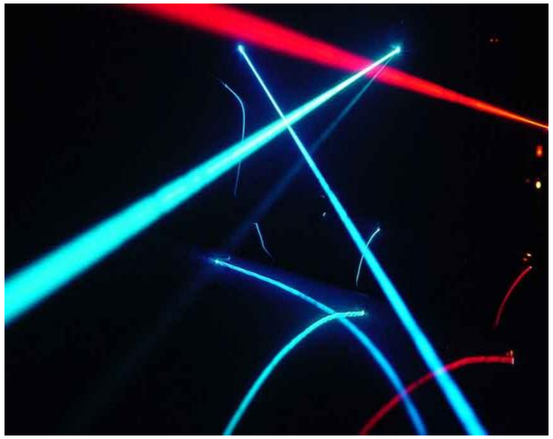


--------- Ethernet

——— Infiniband

Before we can use the Cluster . . .

- We need to know about:
  - Allocations & Accounts
  - Connections to the cluster
  - Cluster environment
  - Modules & Partitions
  - SLURM & Job Submissions



- The remaining slides will cover the basics of these topics.

ALLOCATIONS & ACCOUNTS

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) and staff are eligible for an allocations (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"

CONNECTING & LOGGING ONTO 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. Open-on-Demand -- Coming Soon!
    - Using your web browser, go to URL https://rivanna-portal.hpc.virginia.edu
    - You will need to "Netbadge" in.

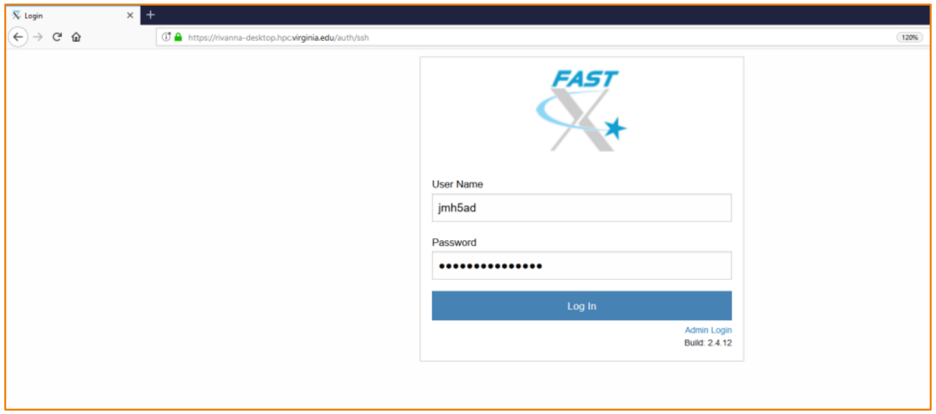
Regardless of how you connect, you must use the UVa Anywhere VPN when off-grounds.

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

We will use FastX today:

- In your web browser, go to URL:

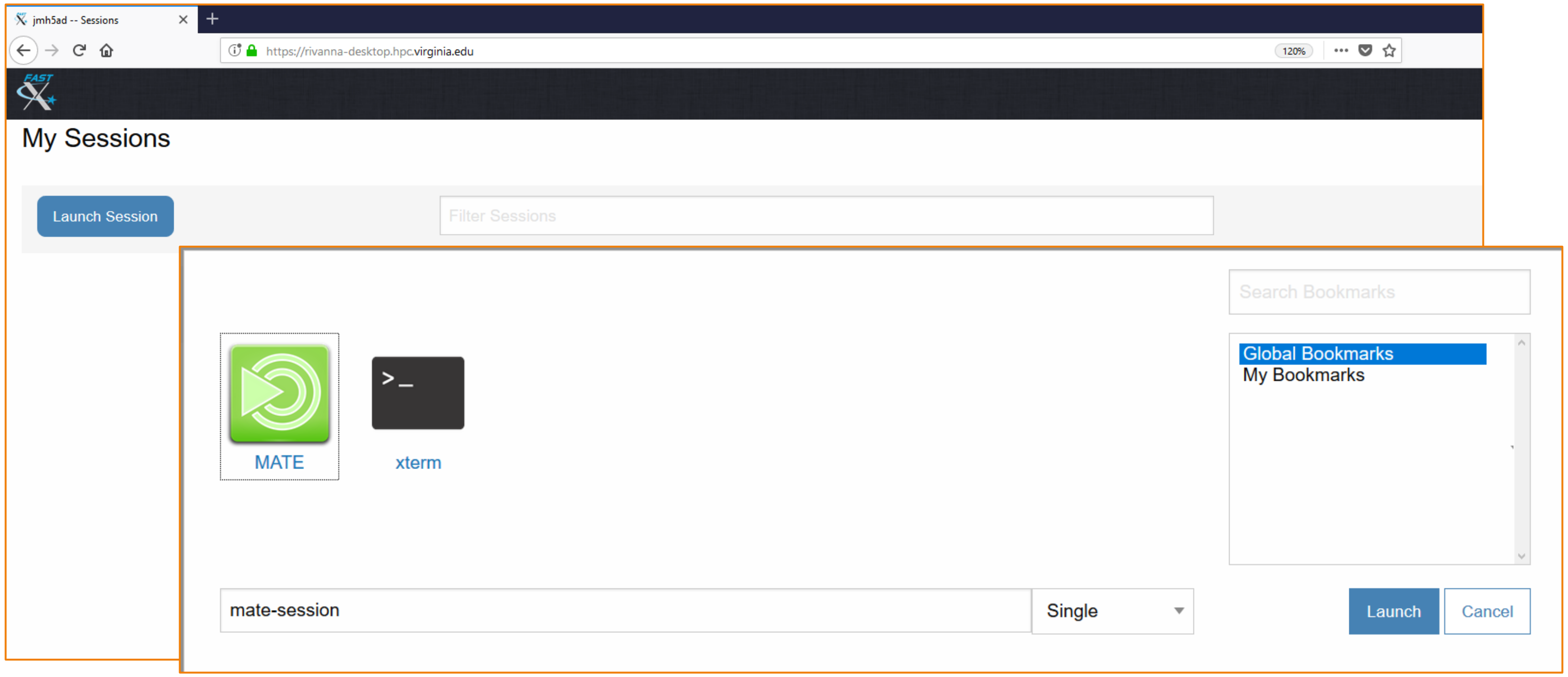
  https://rivanna-desktop.hpc.virginia.edu



User Name
jmh5ad
Password
••••••••••••••••
Build: 2.4.12

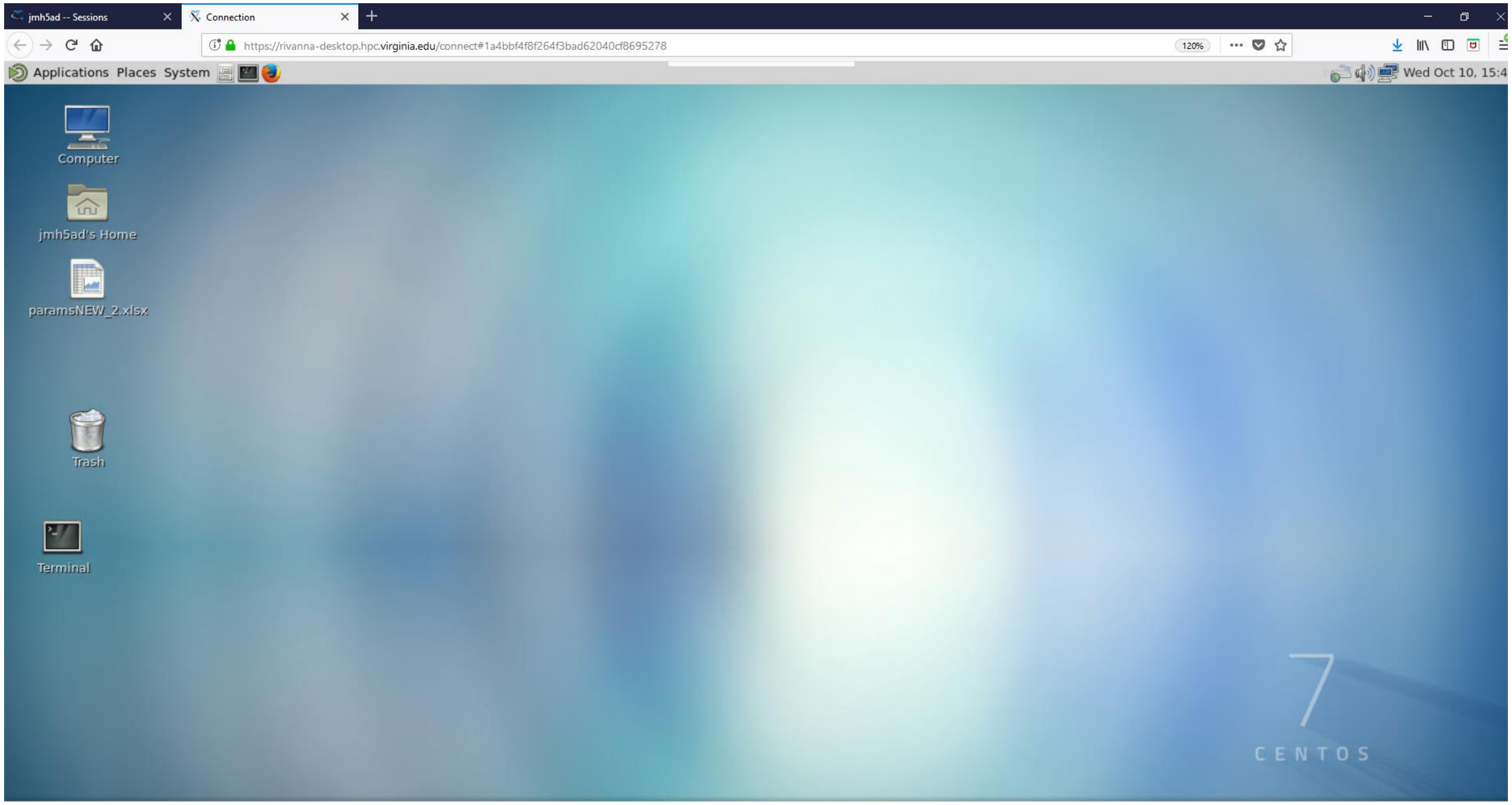
Starting up FastX

- Click “Launch Session”; Select MATE; Click Launch



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

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

CLUSTER ENVIRONMENT

After you have logged in . . .

- You will be in your home directory.
- How you navigate will depend on how you connected to Rivanna.
  - ssh client:
    - A terminal window will appear. To navigate within your directory, you will need to use Unix/Linux commands.
    - See https://arcs.virginia.edu/UNIX-tutorials-for-beginners to learn more about Unix/Linux commands
  - FastX:
    - A desktop environment will appear. You can use your mouse to navigate or open a terminal window to use Unix/Linux commands or start interacive applications.
  - Open-on-Demand:
    - A dashboard will appear. You can click on the menu items across the top to access different tools, like a file manager, a job composer, or interactive applications.

Rivanna is a Linux System

- Some useful Unix/Linux commands:

```
$ ls
$ pwd
$ cd folder_name
$ cp file_1 file_2
$ rm file_1
$ cd ..
```

https://arcs.virginia.edu/UNIX-tutorials-for-beginners

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

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
```

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.

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'
```

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).
  3. Use the web browser in the FastX desktop to download data from UVA Box.
  4. Set up a Globus endpoint on your laptop and use the Globus web interface to transfer files.

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

MODULES

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.
- `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 N.M of software mymod.
    - module load compiler mpi mymod – For compiler- and MPI- specific modules, loads the modules in the appropriate order and, optionally, the 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 bioinformatics software packages, try this:

```
$ module key bio
```

- 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
```

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

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	128 GB 256 GB	20 28	265 (20-core nodes shared w/ parallel queue)	1.0
gpu	For jobs that can use general purpose graphical processing units (GPGPUs) (K80 or P100)	3 days	256 GB	28	14 (max 4 nodes per job)	1.0
parallel	For large parallel jobs on up to 120 nodes (<= 2400 CPU cores)	3 days	128 GB 256 GB	20	240 (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 R                #load modules that my job needs
Rscript myProg.R                 #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-l
```

- 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
```

EXAMPLES

To follow along . . .

- Go ahead and log into Rivanna.
- If using FastX, open up a terminal window.

- First, we will copy a set of examples into your account. At the command line, type:

```
cd
cp -r /share/resources/source_code/R_examples/ .
```

Serial Job

- To see that the directory is there, type:

```
ls
```

- Move to the first folder (i.e., 01_serial) by typing:

```
cd R_examples/01_serial
ls
```

- You will see 3 files: hello.R, hello.slurm and run_cmds
- To view the contents of files, type `more` followed by the filename:

```
more hello.slurm
```

Serial Job

- If your program performs lots of computation, but uses only one processor, you should use the standard queue.

```
#!/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_ID #the account/allocation I am using

module purge
module load gcc R/3.4.0
Rscript hello.R                  #command-line execution of my job
```

Serial Job

- You will need to edit the slurm script to add your account information. For today, we will use pluma.

```
gedit hello.slurm
```

- Replace `Your_group_ID` with the your allocation name (i.e., the MyGroups name)

- After you have added you account information, save the edits, exit pluma, and submit the job:

```
sbatch hello.slurm
```

Serial Job

- Your results will be placed in a file with the name `slurm_12345678.out`, where 12345678 is replaced with the job ID number from your job submission.
  - Type **`ls`** to see if the output file exists in your directory.
  - You can look at the results by typing **`more`** following by the filename. For example:

```
more slurm_12345678.out
```

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.

High Throughput Job

- Move to the second folder (i.e., 02_jobArray) by typing:

```
cd ../02_jobArray
ls
```

- Again, you will see 3 files: hello.R, hello.slurm and run_cmds

- To view the contents of files, type `more` followed by the filename:

```
more hello.slurm
```

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
```

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.

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
- When the code reaches a place where it can do many computations simultaneously (e.g., a loop block that has no dependencies within iterations), the code can send the computations to run on various cores.
- 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
```

Multicore Job

- Move to the third folder (i.e., 03_multicore) by typing:

```
cd ../03_multicore
ls
```

- Again, you will see 3 files: hello_mc.R, hello_mc.slurm and run_cmds

- To view the contents of files, type `more` followed by the filename:

```
more hello_mc.slurm
```

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 openmpi R
Rscript hello_mc.R ${SLURM_CPUS_PER_TASK}
```

- And, submit by typing:

```
sbatch hello_mc.slurm
```

Multi-core Job: Bowtie

- Copy a set of examples into your account. At the command line, type:

```
cd
cp -r /share/resources/source_code/bio/ .
cd bio
ls
```

- You should see one file: `bowtie2.slurm`

- Update allocation group in SLURM script

```
pluma bowtie2.slurm
```

Multi-core Job: Bowtie

```
#!/bin/bash
#SBATCH --job-name="bowtie2”        # custom job name
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4          # request 4 cpu cores
#SBATCH --mem-per-cpu=2000         # request 2000 MB (2 GB) of memory for each cpu core → 8 GB total
#SBATCH --time=00:10:00
#SBATCH -o bowtie_test_%j.out      # custom SLURM output log file. %j is replaced with job id.
#SBATCH --partition=standard
#SBATCH --account=<YOUR_GROUP>

# set up environment
module purge
module load bowtie2

INPUT_PATH=$EBROOTBOWTIE2/example
OUTPUT_PATH=/scratch/$USER/bowtie_$SLURM_JOBID
mkdir -p $OUTPUT_PATH
cd $OUTPUT_PATH

#Indexing a reference genome
bowtie2-build $INPUT_PATH/reference/lambda_virus.fa lambda_virus --threads $SLURM_CPUS_PER_TASK
```

Multi-core Job: Bowtie

- Submit the job

```
sbatch bowtie2.slurm
```

- Check job status with sacct command

```
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
3116424         bowtie2   standard  hpc_build          4  COMPLETED      0:0
3116424.bat+      batch             hpc_build          4  COMPLETED      0:0
```

- View output file

```
more bowtie_test_3116424.out
```

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)

- 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
```

NEED MORE HELP?

Office Hours

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

APPENDICES

A: Connecting to Rivanna with an ssh client

B: Using Jupyter Notebooks on Rivanna

C: Connecting to Rivanna with MobaXterm

APPENDIX A

Connecting to Rivanna with an ssh client

SSH Clients

- You will need an ssh (secure shell) client on your computer.
  - On a Mac or Linux system, use ssh (Terminal application on Macs)

    ```
    ssh –Y mst3k@rivanna.hpc.virginia.edu
    ```

  - On a Windows system, use `MobaXterm`
    - To install MobaXterm use the URL: **http://mobaxterm.mobatek.net**
    - The free "home" version is fine for our purpose.

When you are Off-Grounds, you must use the UVa Anywhere VPN client.

Connecting to the Cluster

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

  (does load-balancing among the three front-ends)

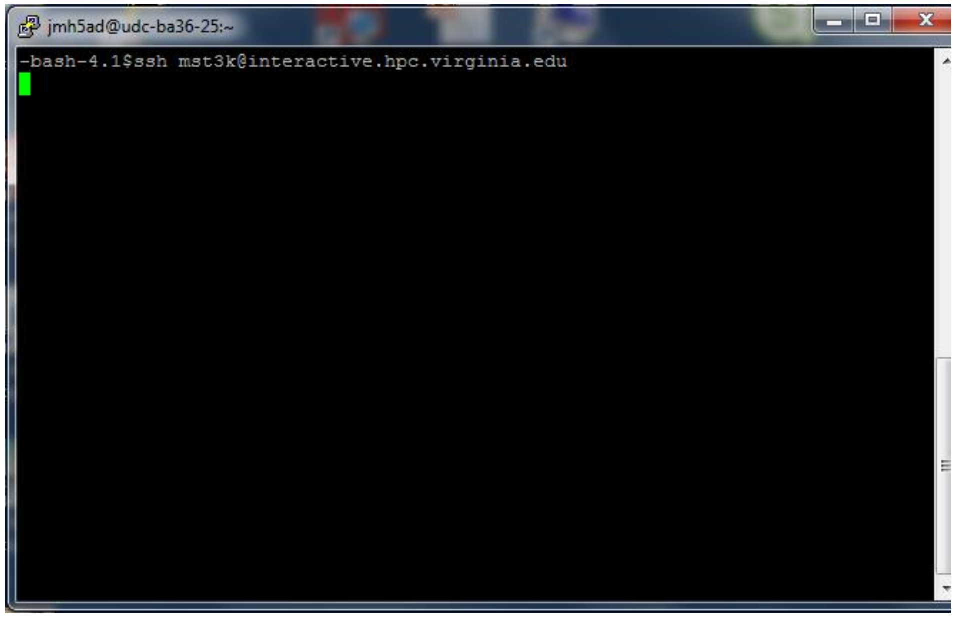
- However, you also can log onto a specific front-end:
  - rivanna1.hpc.virginia.edu
  - rivanna2.hpc.virginia.edu
  - rivanna3.hpc.virginia.edu
  - rivanna-viz.hpc.virginia.edu

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.



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

APPENDIX B

Using Jupyter Notebooks on Rivanna

JupyterLab

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

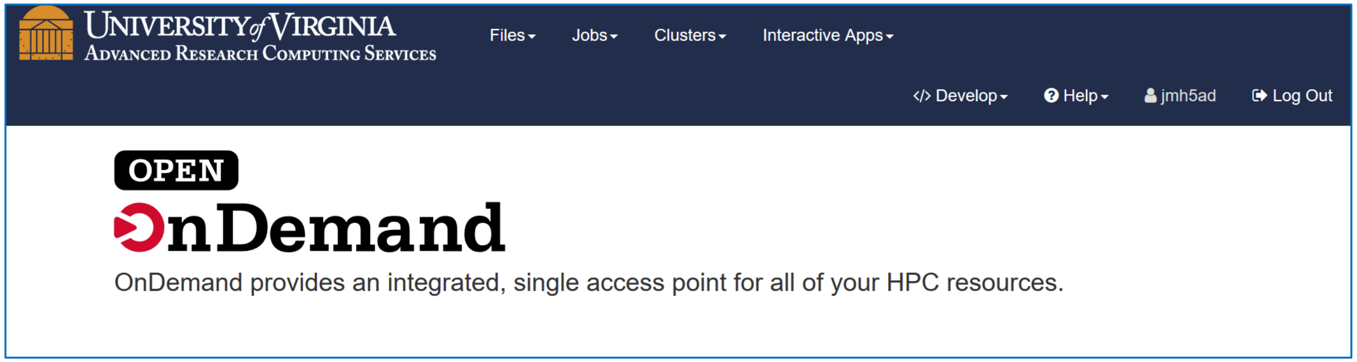
- ARCS now provides JupyterLab on Rivanna.

Accessing JupyterLab

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

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

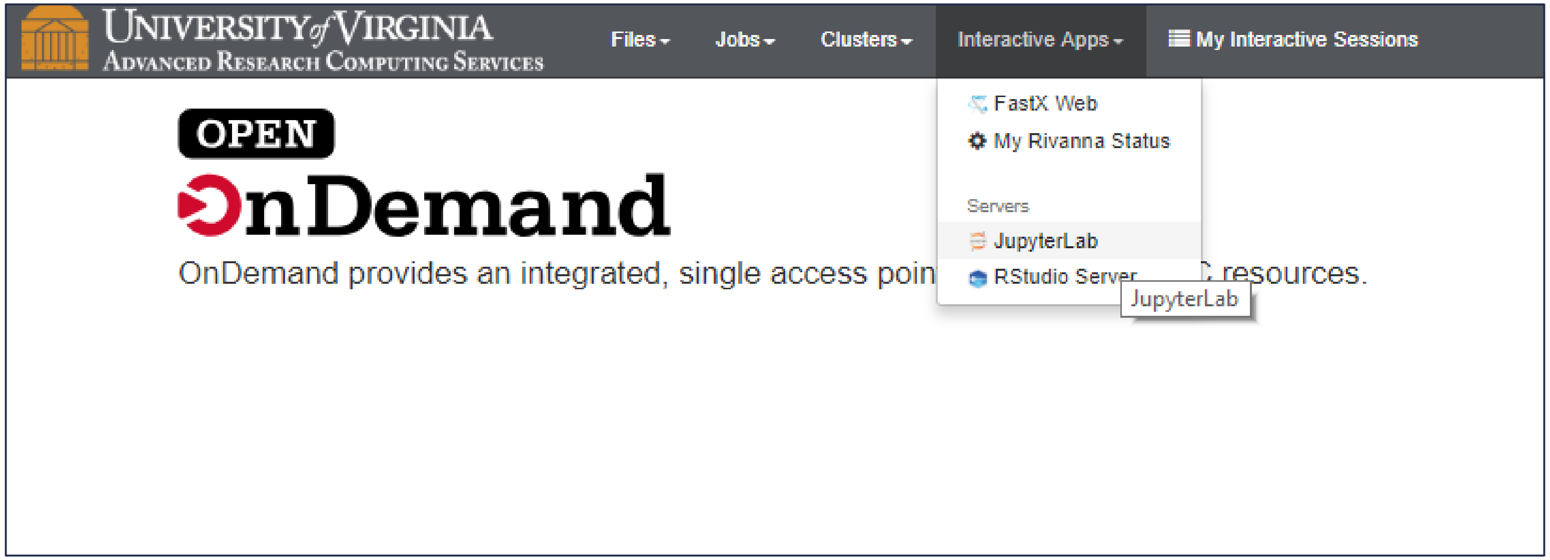
- After logging in via Netbadge in, you will be directed to the Open OnDemand main page.



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

Starting Jupyter Instance

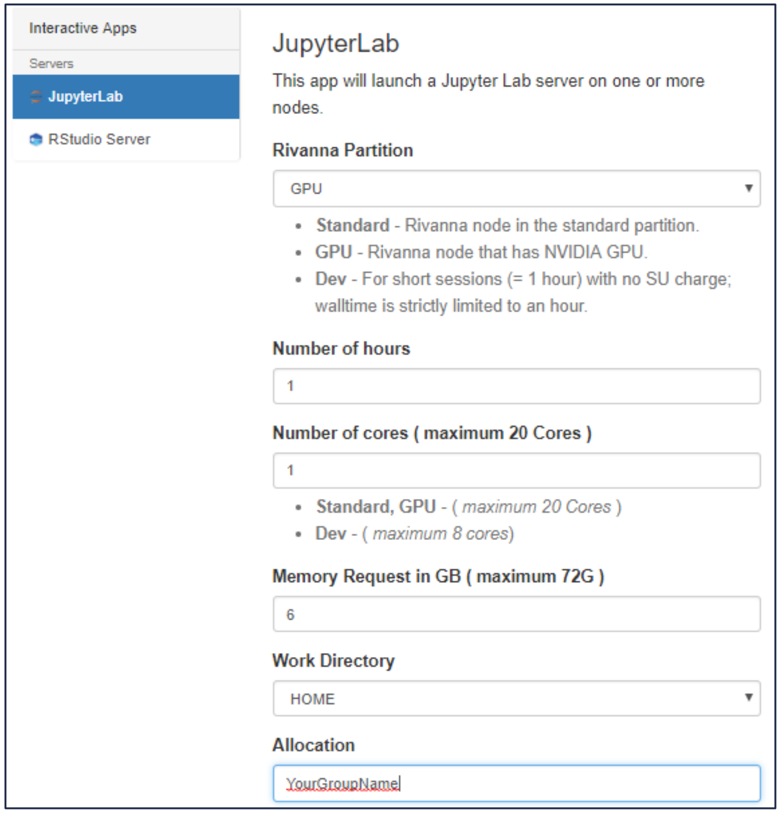
- In the top, click on “Interactive Apps” and in the drop-down box, click on “Jupyter Lab”.



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

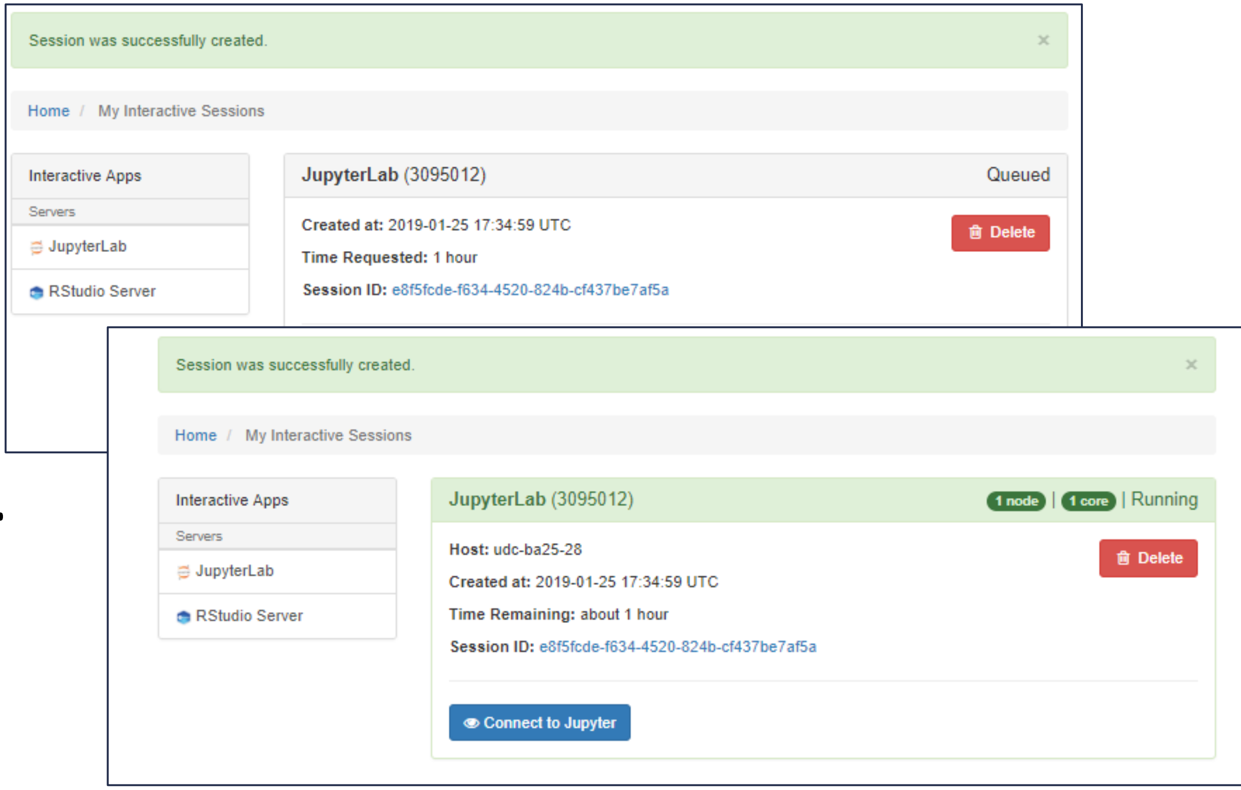
YourGroupName

Starting a Jupyter Instance

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

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

- Click on the blue button.



Session was successfully created.

Home / My Interactive Sessions

JupyterLab (3095012) Queued

Created at: 2019-01-25 17:34:59 UTC

Time Requested: 1 hour

Session ID: e8f5fcde-f634-4520-824b-cf437be7af5a

Session was successfully created.

Home / My Interactive Sessions

JupyterLab (3095012) 1 node | 1 core | Running

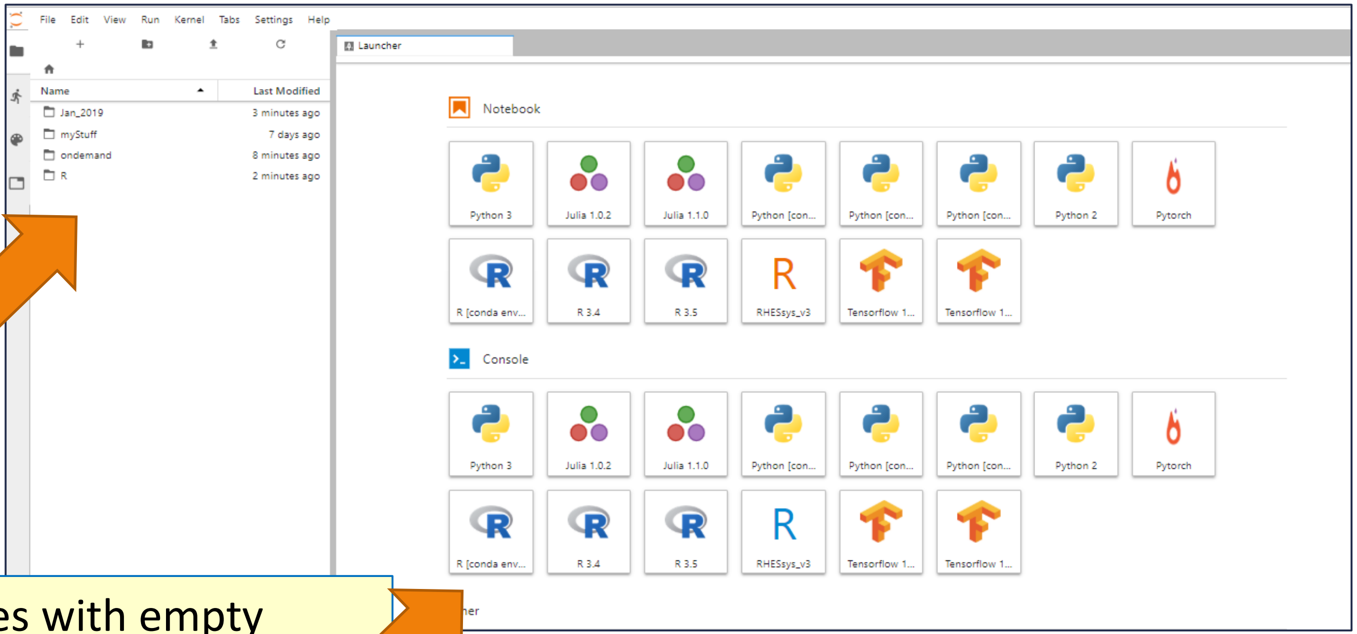
Host: udc-ba25-28

Created at: 2019-01-25 17:34:59 UTC

Time Remaining: about 1 hour

Session ID: e8f5fcde-f634-4520-824b-cf437be7af5a

JupyterLab Environment



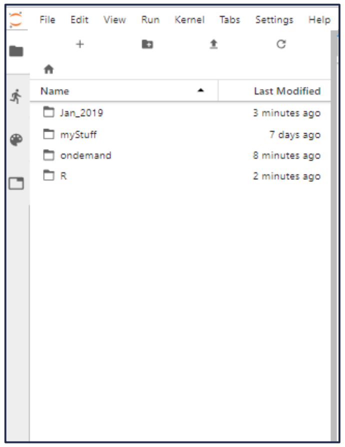
Name	Last Modified
Jan_2019	3 minutes ago
myStuff	7 days ago
ondemand	8 minutes ago
R	2 minutes ago

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

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

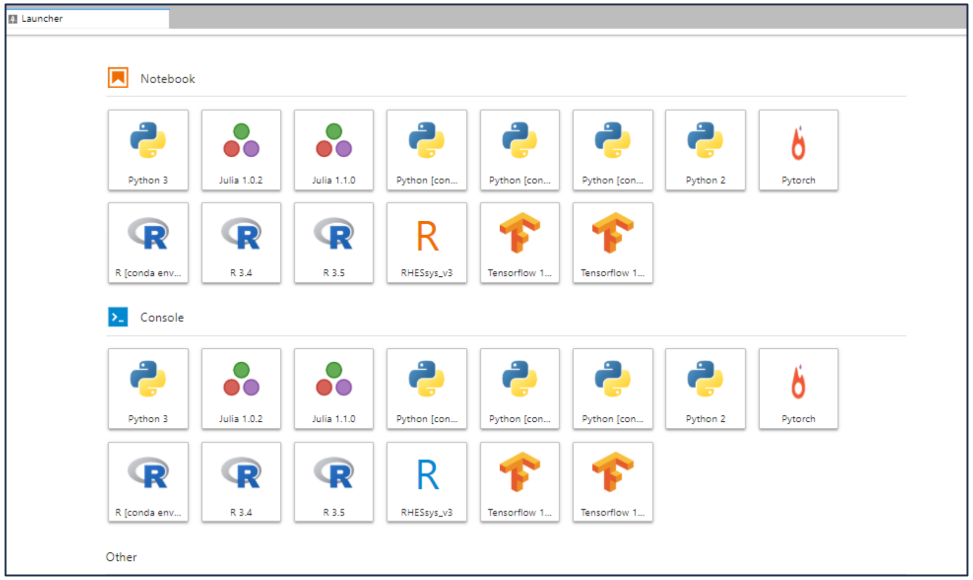
Opening a Notebook

- 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

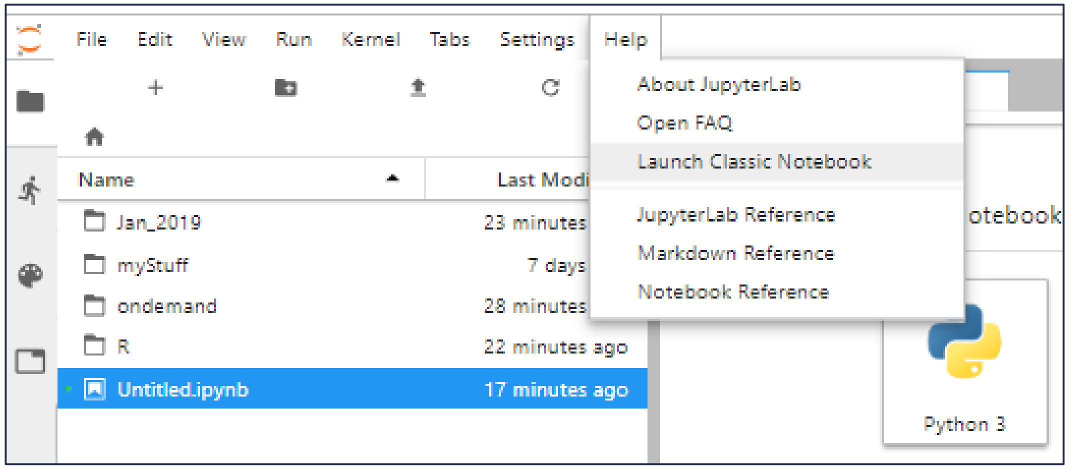
- Or, if you want to start a new notebook, you can click on the notebook tile, for the appropriate underlying system.



Classic Notebook

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

  Help> Launch Classic Notebook

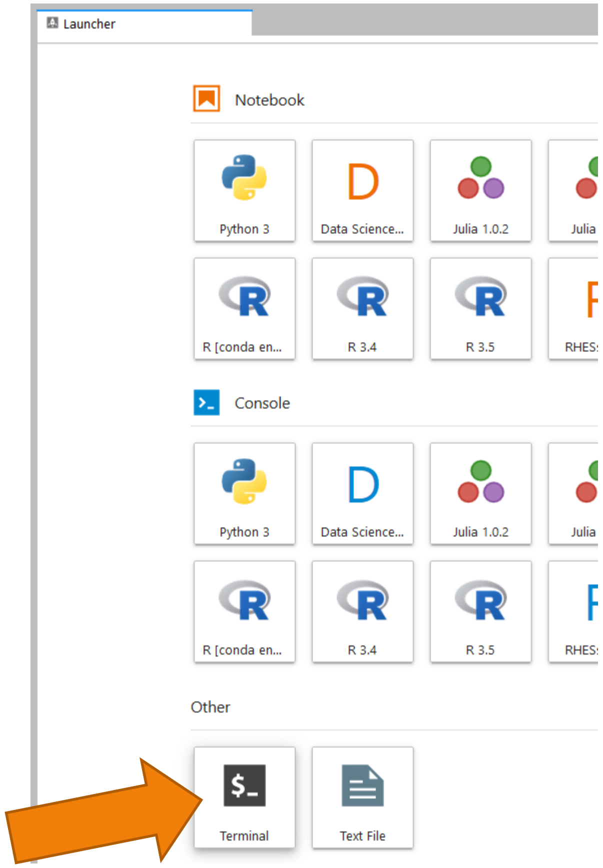


- But, for our example, we will stay with the Jupyter Lab format.

University of Virginia
Advanced Research Computing Services

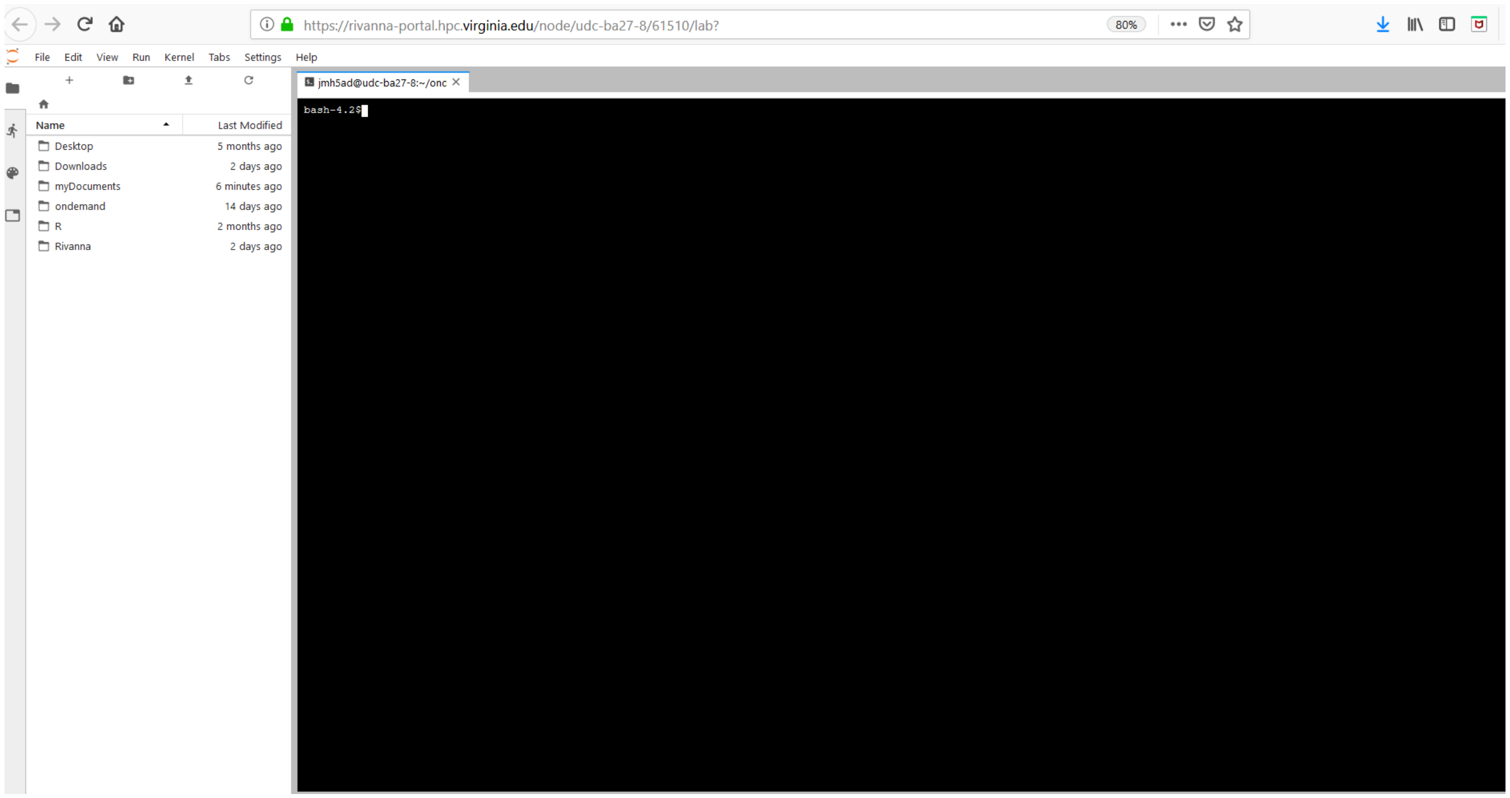
Copying our Notebook to your Directory

- We will open a terminal window to copy files into our home directory.
  - In the Launcher panel, scroll down until you see the “Other” category.
  - Click on the Terminal tile.



The Terminal Window

- A terminal window (or shell) will appear in a separate tab:


https://rivanna-portal.hpc.virginia.edu/node/udc-ba27-8/61510/lab?
80%
File
Edit
View
Run
Kernel
Tabs
Settings
Help
jmh5ad@udc-ba27-8:~/onc

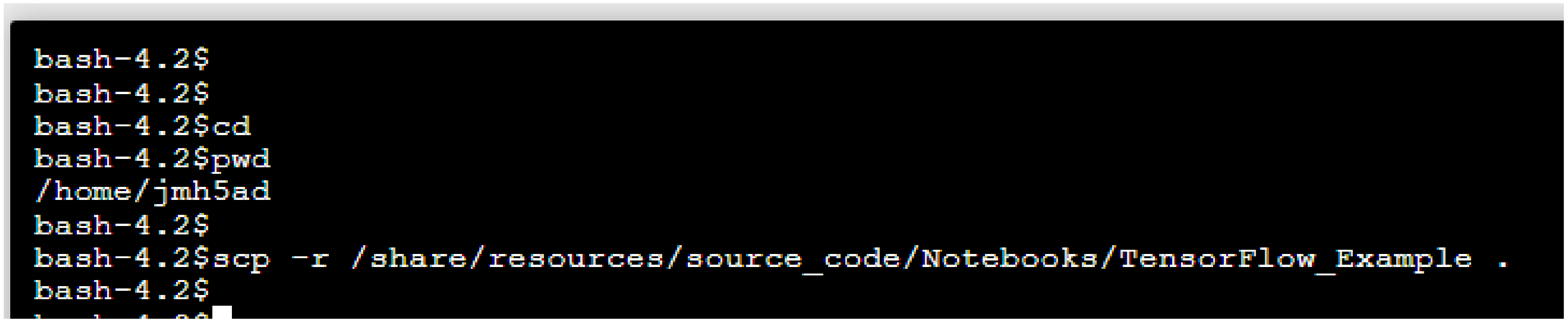
Name	Last Modified
Desktop	5 months ago
Downloads	2 days ago
myDocuments	6 minutes ago
ondemand	14 days ago
R	2 months ago
Rivanna	2 days ago

```
bash-4.2$
```

Copying our Notebook to your Directory

- Make sure that you are in your home directory by typing `cd`.
- Type:

```
cd
scp -r /share/resources/source_code/Notebooks/TensorFlow_Example .
```



```
bash-4.2$
bash-4.2$
bash-4.2$cd
bash-4.2$pwd
/home/jmh5ad
bash-4.2$
bash-4.2$scp -r /share/resources/source_code/Notebooks/TensorFlow_Example .
bash-4.2$
```

UNIVERSITY of VIRGINIA
ADVANCED RESEARCH COMPUTING SERVICES

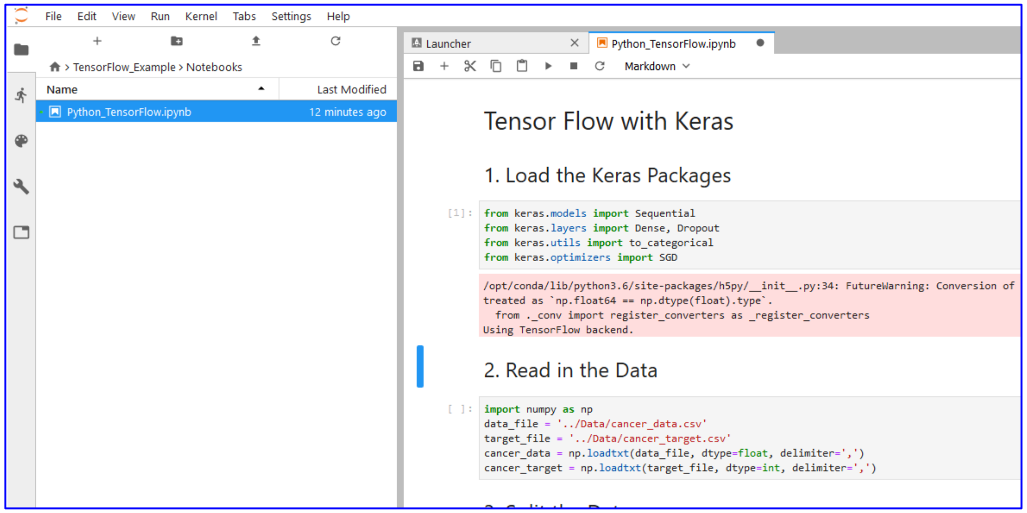
Opening the Notebook

- Close the browser tab for the Terminal Window.
- You should be back on the page that shows your Home directory in Jupyter. (If not, click on the browser tab to get back to the Jupyter Home page.)
- In the file browser pane, click on the folders TensorFlow_Example and Notebooks to get to the file: Python_TensorFlow.ipynb
- Double-click on Python_TensorFlow.ipynb to open the notebook.

Running the Notebook

- To run a particular cell, click inside the cell and press Shift & Enter or Ctrl & Enter.
  - Shift & Enter will advance to the next cell
  - Ctrl & Enter will stay in the same cell

- To run the entire notebook, select
  - Run > Run All Cells



Tensor Flow with Keras

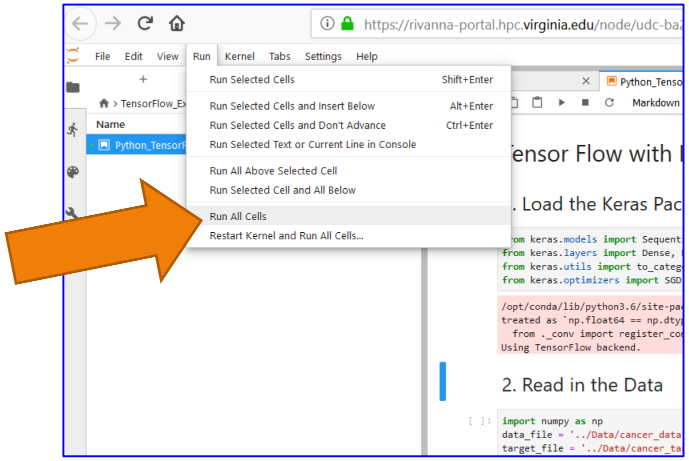
1. Load the Keras Packages

```
[1]: from keras.models import Sequential
     from keras.layers import Dense, Dropout
     from keras.utils import to_categorical
     from keras.optimizers import SGD
```

```
/opt/conda/lib/python3.6/site-packages/h5py/__init__.py:34: FutureWarning: Conversion of
treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

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=',')
```


https://rivanna-portal.hpc.virginia.edu/node/udc-ba
File Edit View Run Kernel Tabs Settings Help
Run Selected Cells Shift+Enter
Run Selected Cells and Insert Below Alt+Enter
Run Selected Cells and Don't Advance Ctrl+Enter
Run Selected Text or Current Line in Console
Run All Above Selected Cell
Run Selected Cell and All Below
Run All Cells
Restart Kernel and Run All Cells...
TensorFlow_Ex
Name
Python_TensorF
Python_Tenso
Markdown
ensor Flow with
. Load the Keras Pac
2. Read in the Data

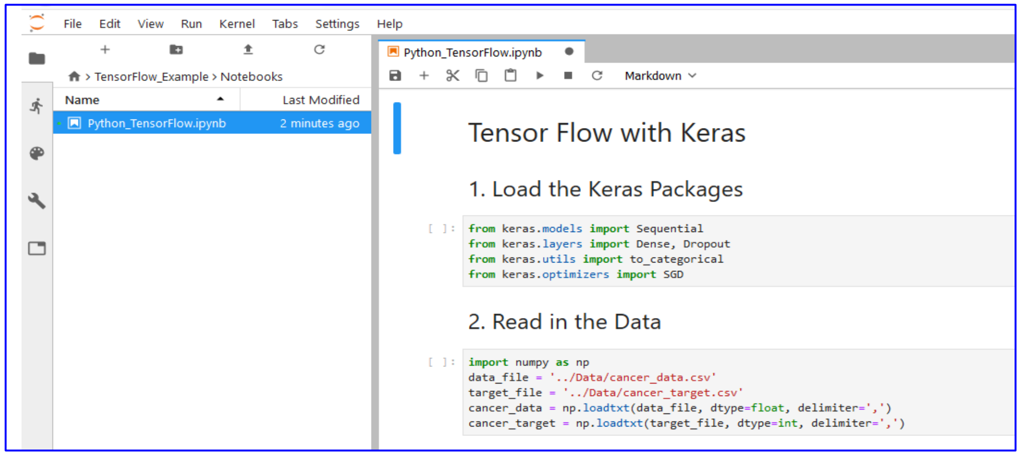
```
om keras.models import Sequent
from keras.layers import Dense,
from keras.utils import to_categ
from keras.optimizers import SGD
```

```
/opt/conda/lib/python3.6/site-pa
treated as `np.float64 == np.dty
  from ._conv import register_co
Using TensorFlow backend.
```

```
[ ]: import numpy as np
     data_file = '../Data/cancer_data
     target_file = '../Data/cancer_ta
```

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.



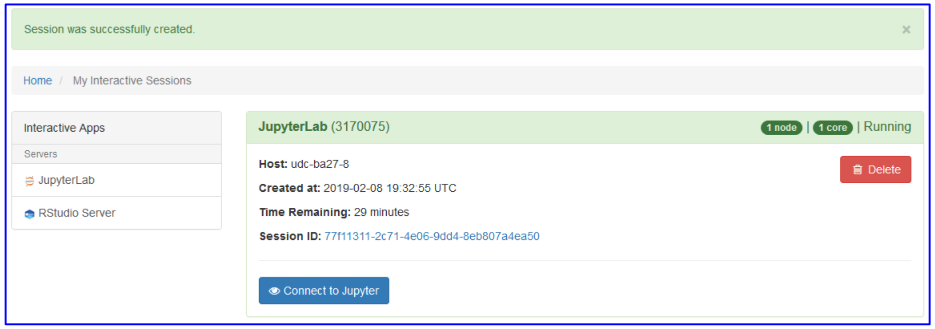
Tensor Flow with Keras

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) 1 node | 1 core | Running

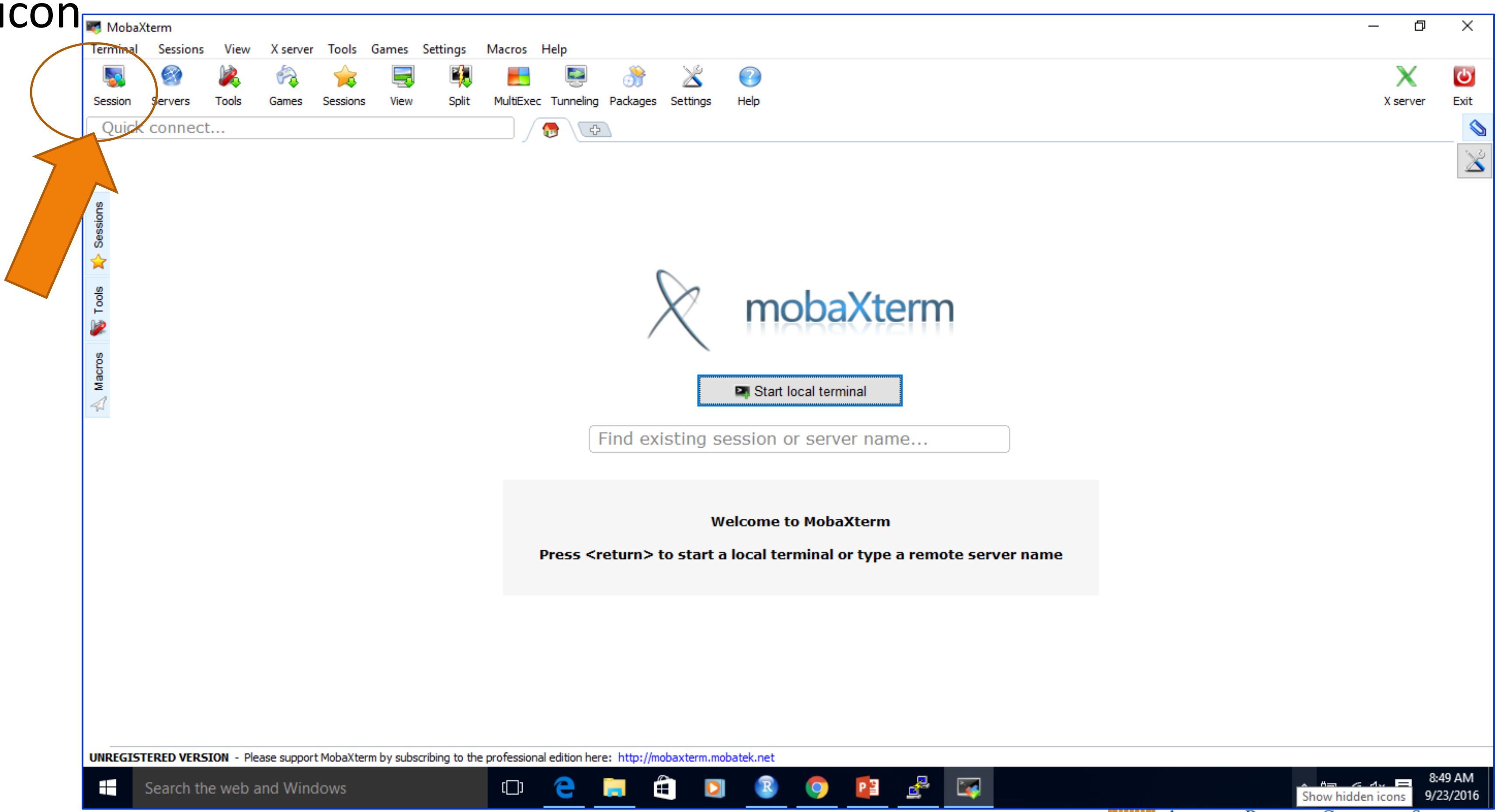
Host: udc-ba27-8
Created at: 2019-02-08 19:32:55 UTC
Time Remaining: 29 minutes
Session ID: 77f11311-2c71-4e06-9dd4-8eb807a4ea50

APPENDIX C

Connecting to Rivanna with MobaXterm (Windows)

Connecting to the Cluster with MobaXterm

- The first time that you start up MobaXterm, click on the Session icon


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
Sessions Tools Macros
mobaXterm
Start local terminal
Find existing session or server name...

Welcome to MobaXterm

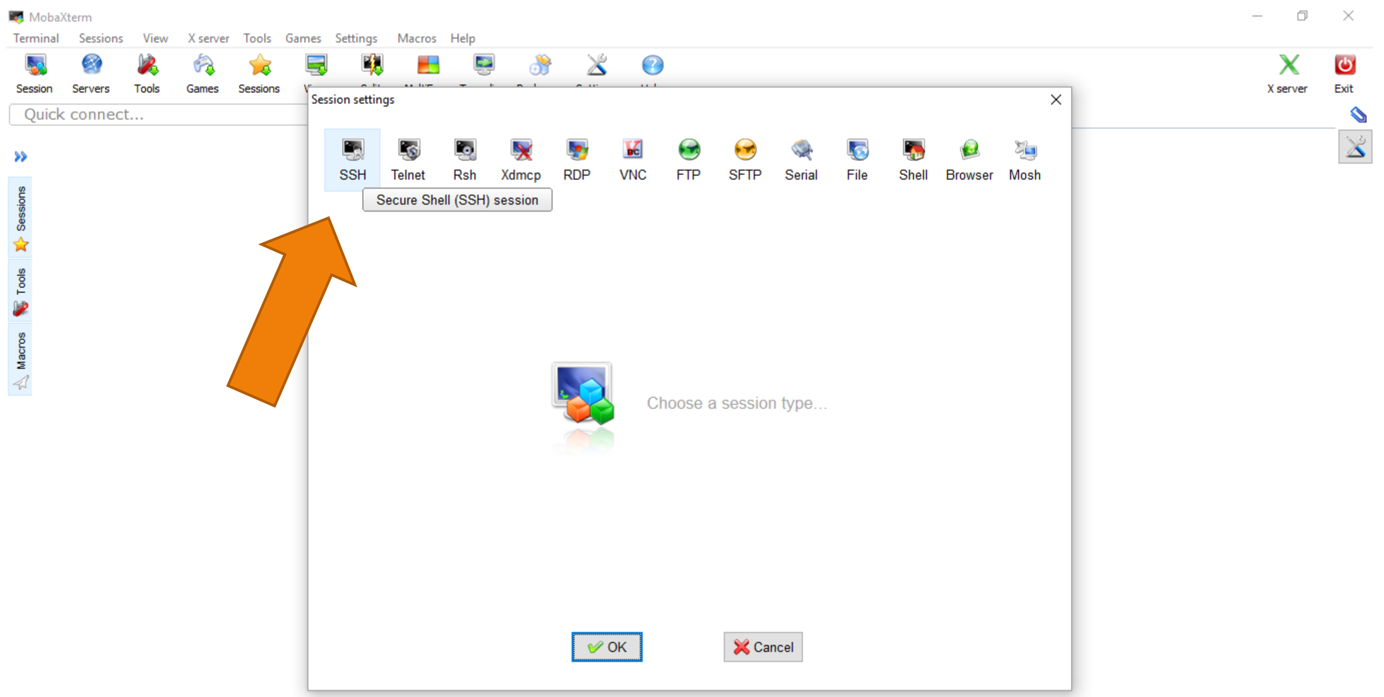
Press <return> to start a local terminal or type a remote server name

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

Search the web and Windows
Show hidden icons
8:49 AM
9/23/2016

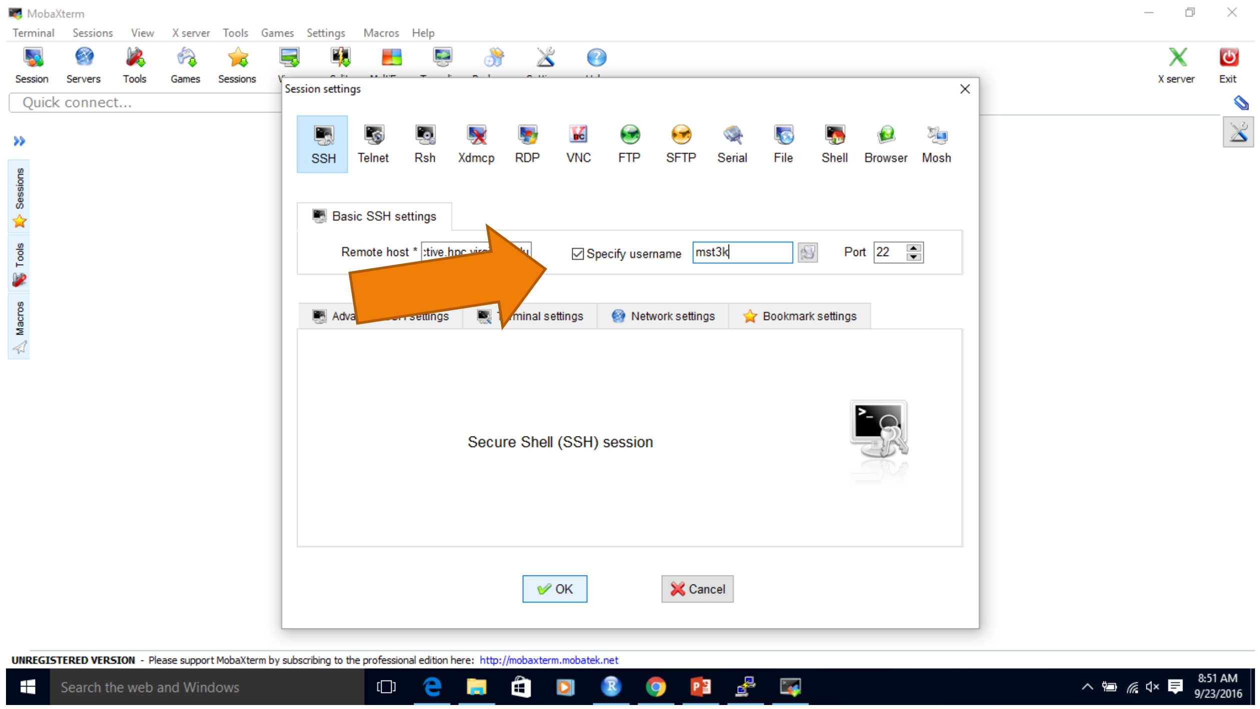
Connecting to the Cluster with MobaXterm

- It will bring up a window asking for the type of session.
- Select SSH and click Okay.


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions
X server Exit
Quick connect...
Sessions Tools Macros
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh
Secure Shell (SSH) session
Choose a session type...
OK
Cancel

Connecting to the Cluster with MobaXterm

- It will prompt you for remote host and username.
- You will have to click on the box next to "Specify username" before you can type in your username.



MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions
X server Exit
Quick connect...
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh
Basic SSH settings
Advanced SSH settings
Terminal settings
Network settings
Bookmark settings
Secure Shell (SSH) session
OK
Cancel
Sessions
Tools
Macros
Search the web and Windows

Remote host * :tive.hpc.virginia.edu
☑ Specify username mst3k
Port 22

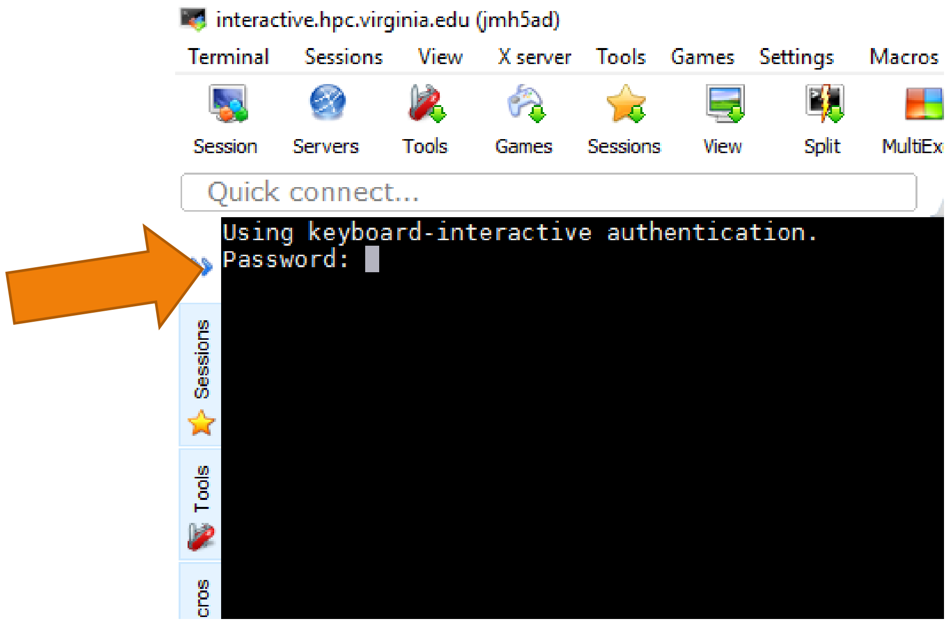
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

8:51 AM
9/23/2016

University of Virginia
Advanced Research Computing Services

Connecting to the Cluster with MobaXterm

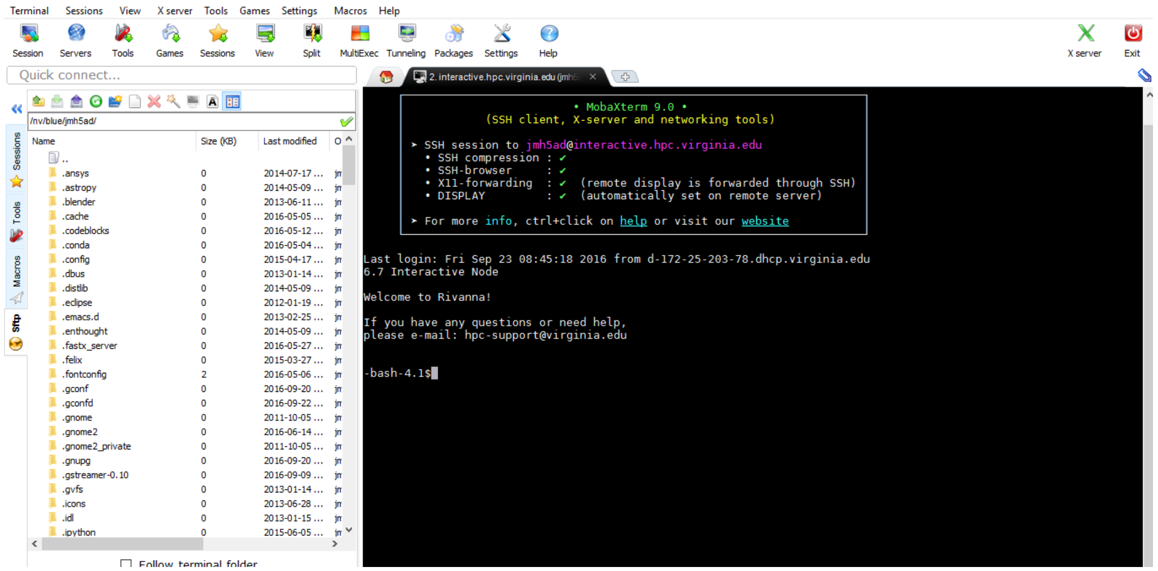
- It will prompt you for your password.
- Note: It will appear as if nothing is happening when you type in your password. It will not display circles or asterisks in place of the characters that you type.



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

Connecting to the Cluster with MobaXterm

- Finally, a split screen will appear.
  - The right pane is a terminal window.
  - The left pane is a list of files in your remote folder that you can click, drag, and drop onto your local desktop.


Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. interactive.hpc.virginia.edu (jmh5
Sessions Tools Macros Sftp
Follow terminal folder

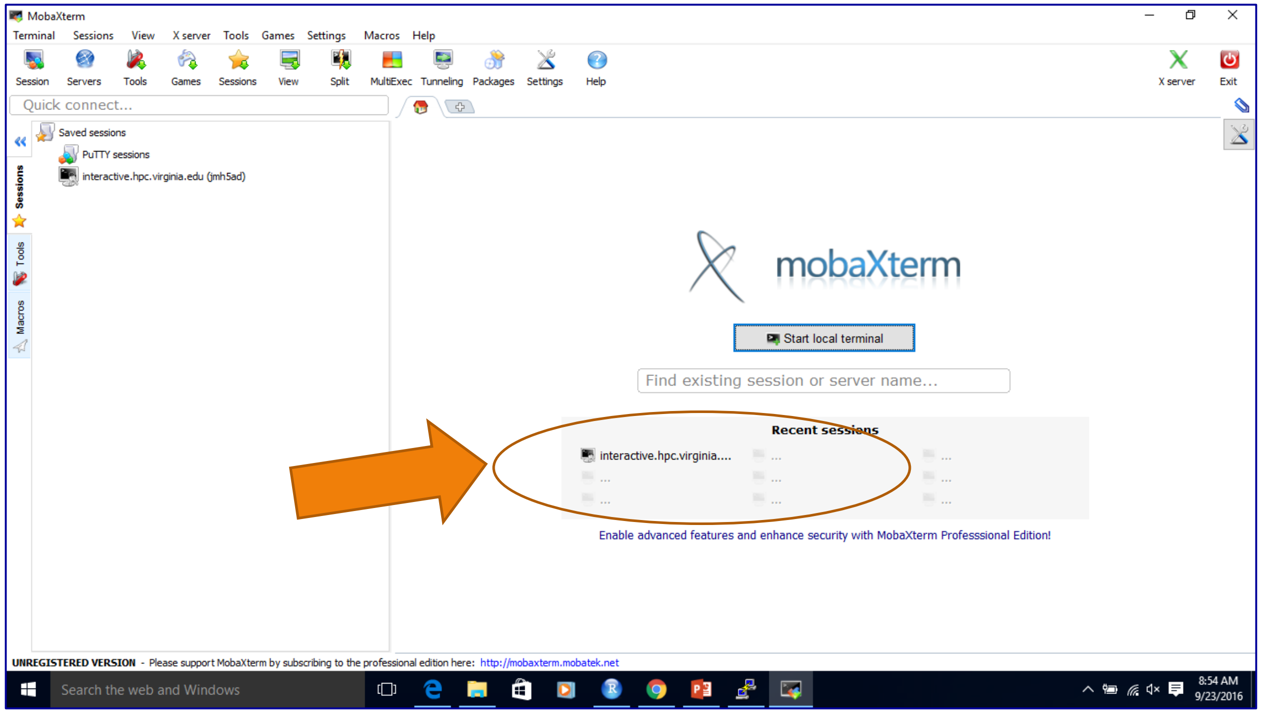
/nv/blue/jmh5ad/

Name	Size (KB)	Last modified	O
..			
.ansys	0	2014-07-17 ...	jn
.astropy	0	2014-05-09 ...	jn
.blender	0	2013-06-11 ...	jn
.cache	0	2016-05-05 ...	jn
.codeblocks	0	2016-05-12 ...	jn
.conda	0	2016-05-04 ...	jn
.config	0	2015-04-17 ...	jn
.dbus	0	2013-01-14 ...	jn
.distlib	0	2014-05-09 ...	jn
.eclipse	0	2012-01-19 ...	jn
.emacs.d	0	2013-02-25 ...	jn
.enthought	0	2014-05-09 ...	jn
.fastx_server	0	2016-05-27 ...	jn
.felix	0	2015-03-27 ...	jn
.fontconfig	2	2016-05-06 ...	jn
.gconf	0	2016-09-20 ...	jn
.gconfd	0	2016-09-22 ...	jn
.gnome	0	2011-10-05 ...	jn
.gnome2	0	2016-06-14 ...	jn
.gnome2_private	0	2011-10-05 ...	jn
.gnupg	0	2016-09-20 ...	jn
.gstreamer-0.10	0	2016-09-09 ...	jn
.gvfs	0	2013-01-14 ...	jn
.icons	0	2013-06-28 ...	jn
.idl	0	2013-01-15 ...	jn
.ipython	0	2015-06-05 ...	jn

```
                    • MobaXterm 9.0 •
          (SSH client, X-server and networking tools)

 ➤ SSH session to jmh5ad@interactive.hpc.virginia.edu
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote display is forwarded through SSH)
   • DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Last login: Fri Sep 23 08:45:18 2016 from d-172-25-203-78.dhcp.virginia.edu
6.7 Interactive Node

Welcome to Rivanna!

If you have any questions or need help,
please e-mail: hpc-support@virginia.edu


-bash-4.1$
```

Connecting to the Cluster with MobaXterm

- MobaXterm will save your session information.
- The next time that you open MobaXterm, you can double-click on the Session that you want.


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
Saved sessions
PuTTY sessions
interactive.hpc.virginia.edu (jmh5ad)
Sessions Tools Macros
mobaXterm
Start local terminal
Find existing session or server name...
Recent sessions
interactive.hpc.virginia....

Enable advanced features and enhance security with MobaXterm Professsional Edition!

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

Search the web and Windows
8:54 AM
9/23/2016