# Introduction to Linux/HPC

# and

# Command-line Computing

Ivan Chang

06/27/2022

Genomics Highthoughput Facility & Research Cyberinfrastructure Center

Workshop Goal:

# Quickstart to navigating UCI HPC3

Workshop Agenda:

- Introduction to Linux and High Performance Computing (HPC)
  - Exercise #1: Connecting to UCI HPC3
- Basic Linux commands for the command line interface (CLI)
  - Exercise #2: Navigating file systems
  - Exercise #3: Working with files (creating, viewing, copying, and editing)
- HPC3 specific commands
  - Exercise #4: Managing your storage
  - Exercise #5: Data transfer from your computer to HPC3
  - Exercise #6: Selecting software modules
  - Exercise #7: Sample job submission

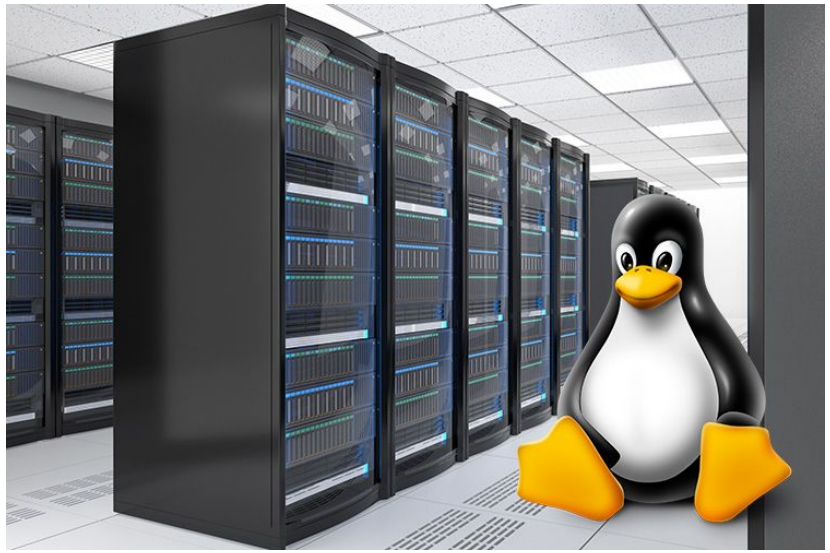
# Introduction

to Linux and High Performance Computing (HPC)

# Operating Systems

Microsoft Windows based PCs

Apple MacOS based Macbook



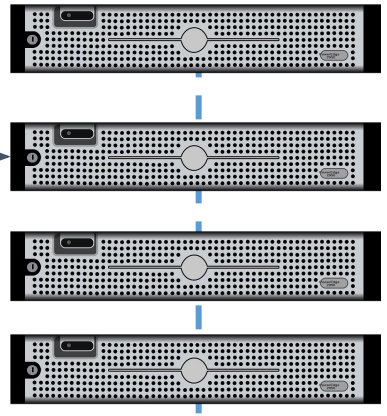
Linux OS based servers

# Basic HPC Architecture

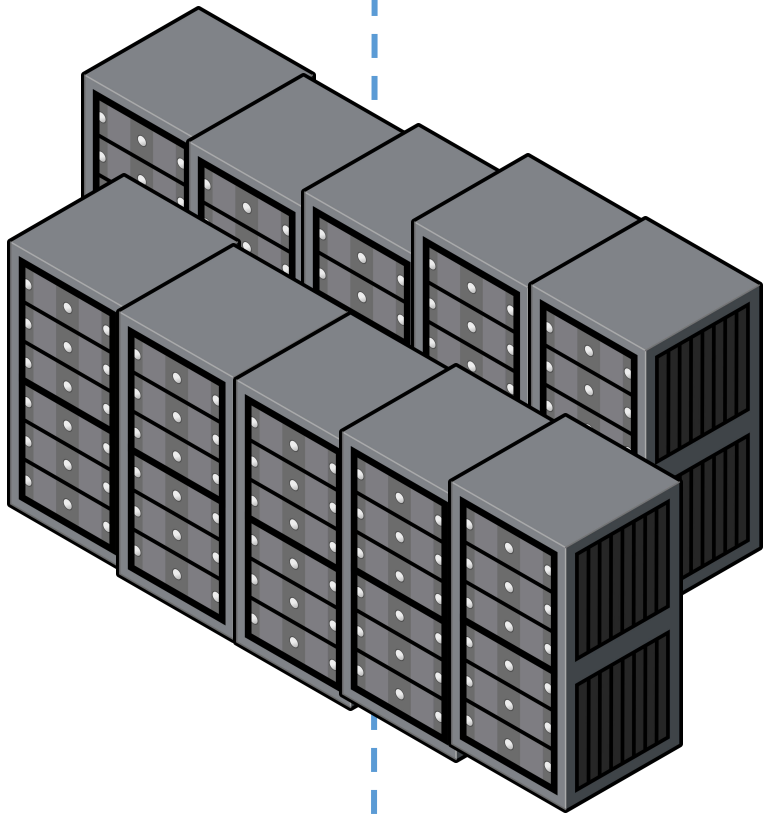
Specialized nodes (e.g. scheduler, data transfer, haproxy, proxmox VM, data portals, etc.)

Fast network interconnect

SSH connection



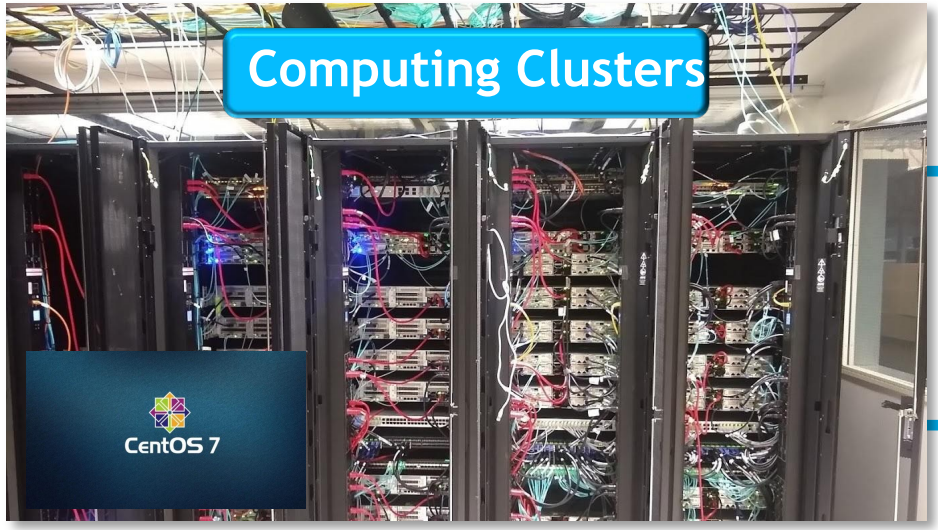
Load-balanced Login Nodes



Clusters of Compute Nodes

File systems with large arrays of disk drives

# Key UCI Computing Resources



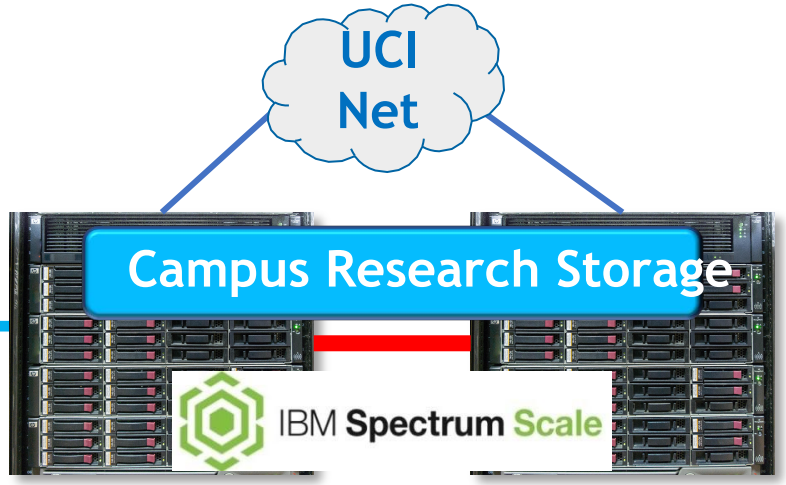

## HPC3

- ~6900 Cores/162 Hosts (expanding to ~8500/200)
- 52 Tesla V100 16Gb Nvidia GPUs
- EDR (100Gbps) Infiniband
- 10GbE Ethernet
- Minimum
  - 4GB memory/core
  - AVX2 instruction set (Epyc/Intel CPUs)


## Six Parallel File Systems

DFS2, DFS3a, DFS3b, …

- 3.9PB usable storage
- ~6GB/sec bandwidth/System
- Single Copy/No Snapshots




## CRSP – Campus Research Storage Pool

- 1 PB usable storage
- Available anywhere on UCI Network
- Dual Copy of All Data
- Snapshots
- Highly available

# Exercise #1

Connecting to UCI HPC3

# Logging onto HPC3

**Step 1** If connecting from outside of UCI network, connect first to UCI campus VPN, see instructions **UCI campus VPN**

**Step 2** Open your Terminal application and start ssh session for hpc3.rcic.uci.edu . Alternatively, you could use the jupyterhub interface at https://hpc3.rcic.uci.edu/biojhub3/ directly in your browser

**Step 3** Either in ssh session or jupyterhub interface, you will need to use your regular UCI credentials (UCINetID and password) to connect

**Step 4** For ssh connection, you will also be prompted for Multifactor Authentication (Duo).

# Terminal

To use ssh, you need to use one of Terminal applications and depending on a user laptop they can be:

| | |
|---|---|
| **Linux** | your favorite Terminal application (Ctrl-Alt-T for Ubuntu Linux) |
| **Mac** | Terminal or **iTerm2** |
| **Windows** | **PuTTY** or **MobaXterm** |
| **Windows 10/11** | **PuTTY**, Windows Terminal, **Linux Subsystem for Windows** or **MobaXterm** |

# SSH

Use your UCINetID and associated password to connect to an HPC3 login node (which are several load-balanced, systems) hpc3.rcic.uci.edu.

Your login name can be specified as ether user@hostname or given with the -l option, for example a user with UCINetID panteater can use:

```
ssh panteater@hpc3.rcic.uci.edu
```

or

```
ssh hpc3.rcic.uci.edu -l panteater
```

# Multifactor Authentication (Duo)

After the June 15, 2022 maintenance, HPC2/HPC3 will require multifactor authentication using **UCI's Duo infrastructure** for all password-based logins. When DUO is active, you will be prompted to enter a code (backup or generated by your DUO device) or request a push to your enrolled DUO-enabled device. A prompt looks similar to

```
ssh panteater@hpc3.rcic.uci.edu
Password:
Duo two-factor login for panteater


Enter a passcode or select one of the following options:


1. Duo Push to XXX-XXX-1212


Passcode or option (1-1): 1
Success. Logging you in...
  Last login:  ....
```

# SSH Keys (and Duo)

The design/implementation of the DUO-supplied **PAM module** makes it possible to use an SSH-key to login without entering a DUO code or receiving a DUO push. HPC3 supports the use of ssh-keys for remote login

We have written local guides for:

**Setting up and using ssh key-based login**

**Ssh with DUO**

In essence, the system from which you are initiating ssh (e.g. your laptop or workstation) should have a locally-generated and *password protected* ssh private key. The public key corresponding to that private key is placed on HPC2/HPC3 in your .ssh/authorized_keys file.

After a successful login you will see a screen similar to the following:

```
Last login: Thu Jul 15 15:25:59 2021 from 10.240.58.4
+-----------------------------------------+
|                                         |
| | | ____   __ _(_)_ __        (_) | ___|  |
| | |/   \ /  ` | | '  \       | | |    \  |
| | | ( ) | ( | | | | | |      | | |   ) | |
| | |\   / \  , | | | | |      | | |    /  |
|          |   /                          |
+-----------------------------------------+
 Distro:  CentOS 7.8 Core
 Virtual: NO

 CPUs:    40
 RAM:     191.9GB
 BUILT:   2020-03-02 13:32

 ACCEPTABLE USE: https://rcic.uci.edu/documents/RCIC-Acceptable-Use-Policy.pdf
  login-i15 2001%
```

# Connecting to HPC3 Interactive Computing Environment

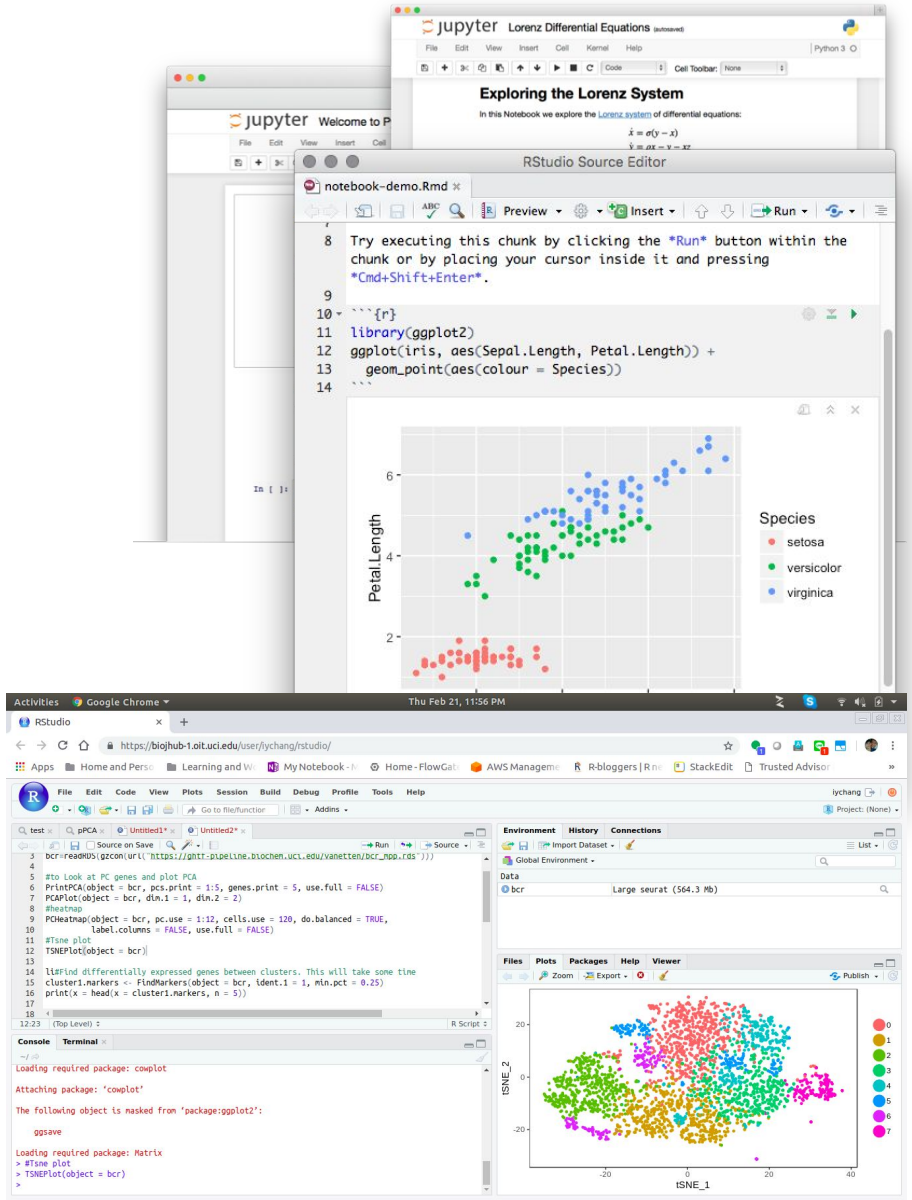
User Authentication


https://hpc3.rcic.uci.edu/biojhub3/

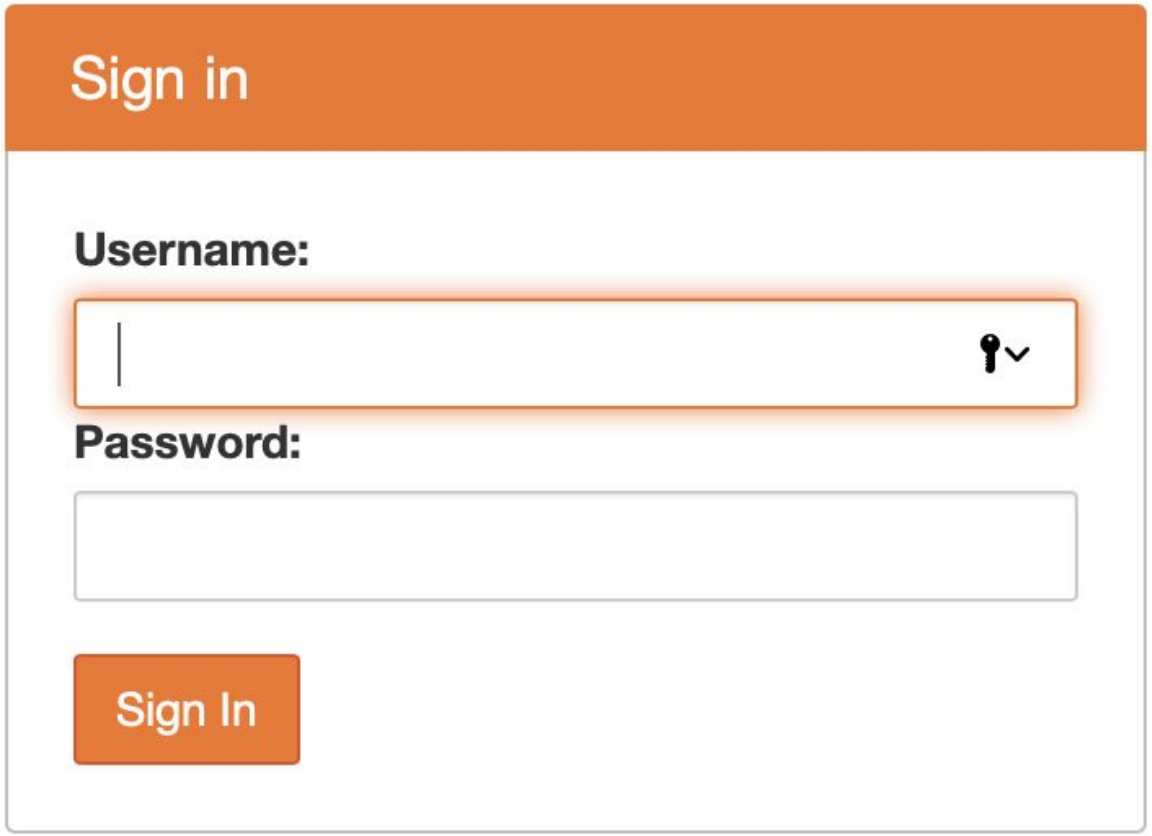
Container Selection

- Seurat2
- Seurat3
- ScanPy
- CellXGene

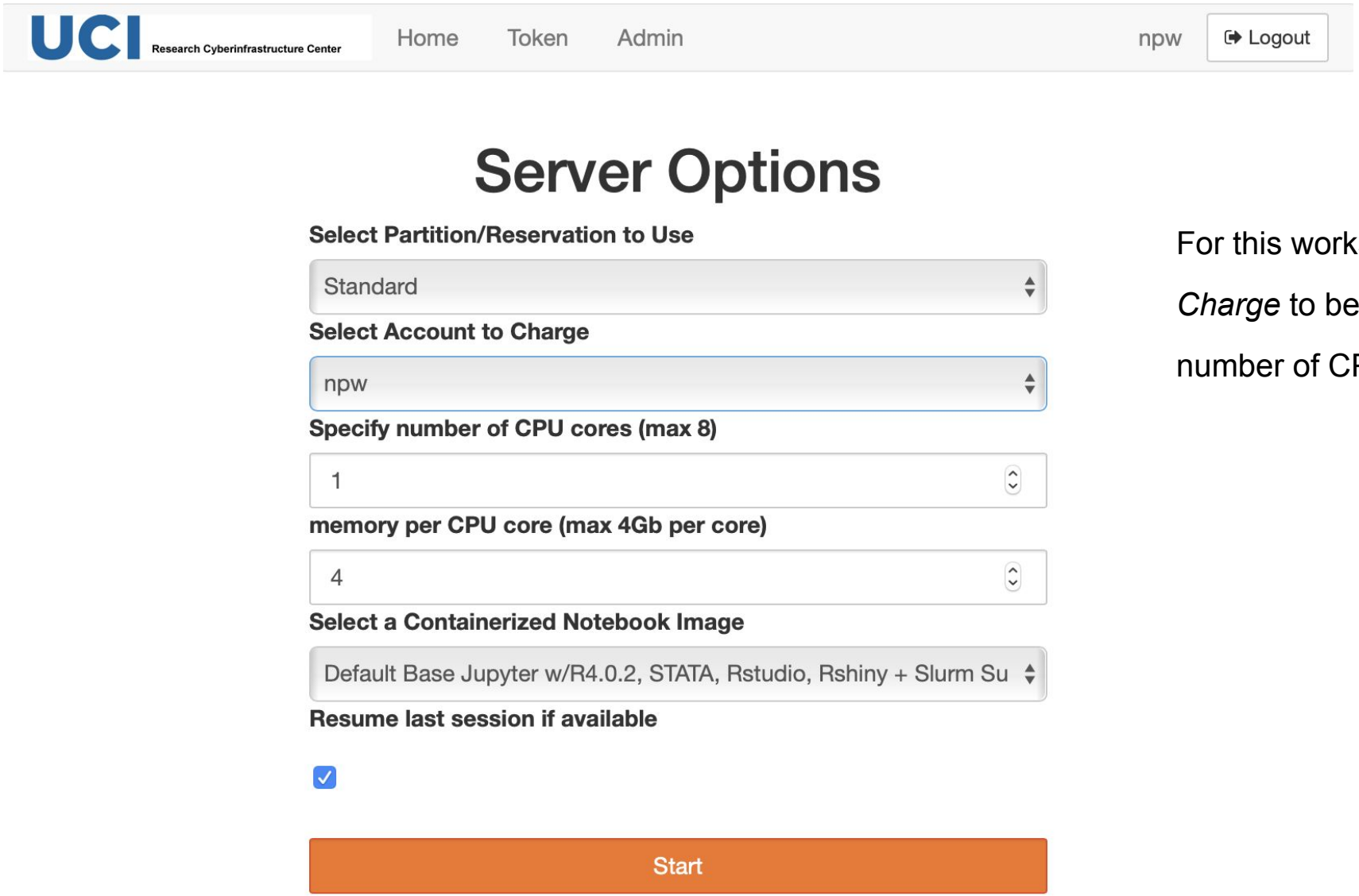
. . .



Using your favorite browser go to: **https://hpc3.rcic.uci.edu/biojhub3/hub/login** You will see the following screen where you will Use your usual HPC3 credentials to sign in:

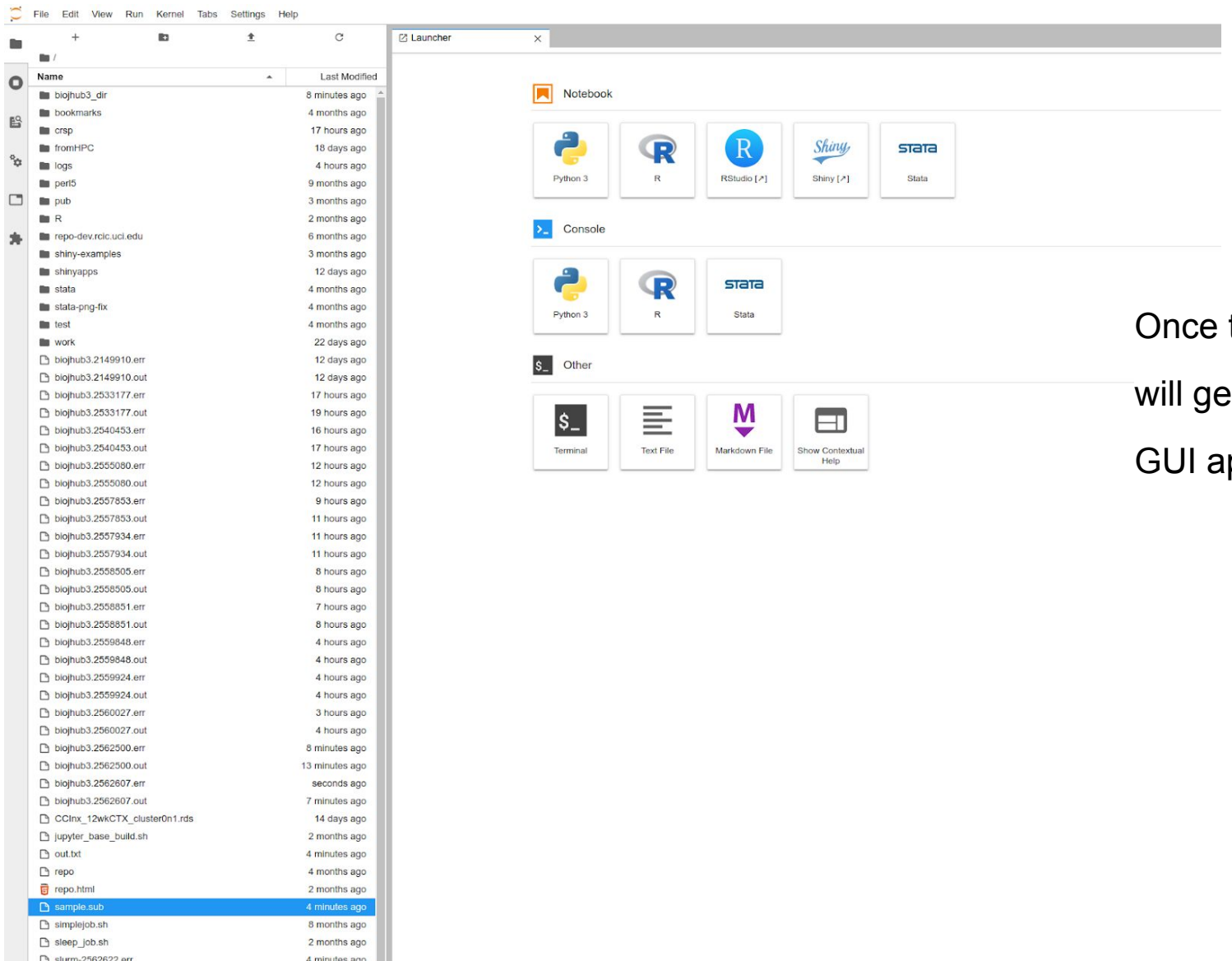


After authentication you will see a screen with server options as in the figure below:



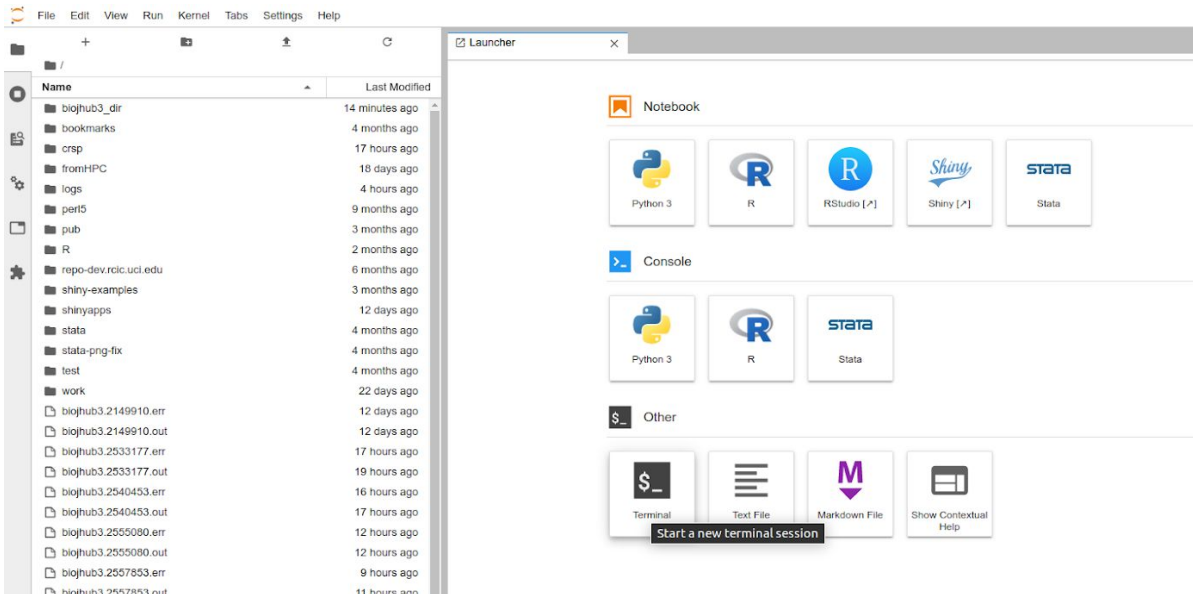

For this workshop, modify the *Select Account to Charge* to be one of your Slurm accounts, change number of CPUs to 2 and press `Start`.

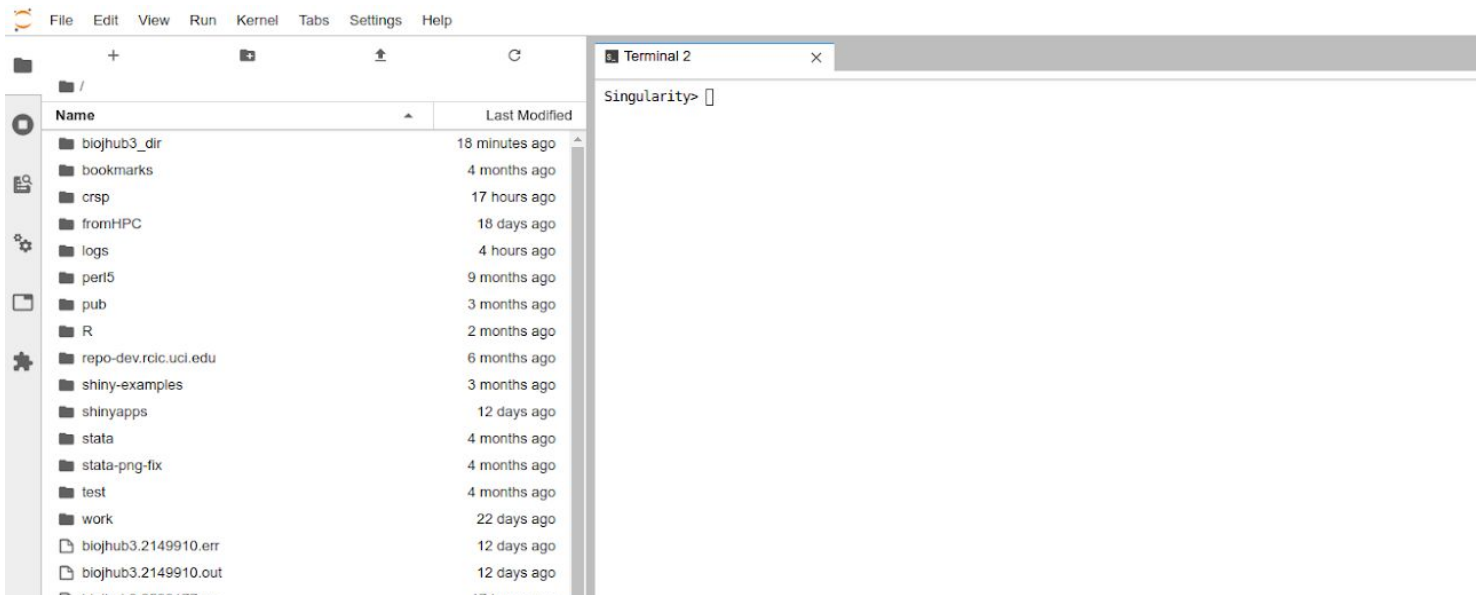
# Main Jupyter Interface



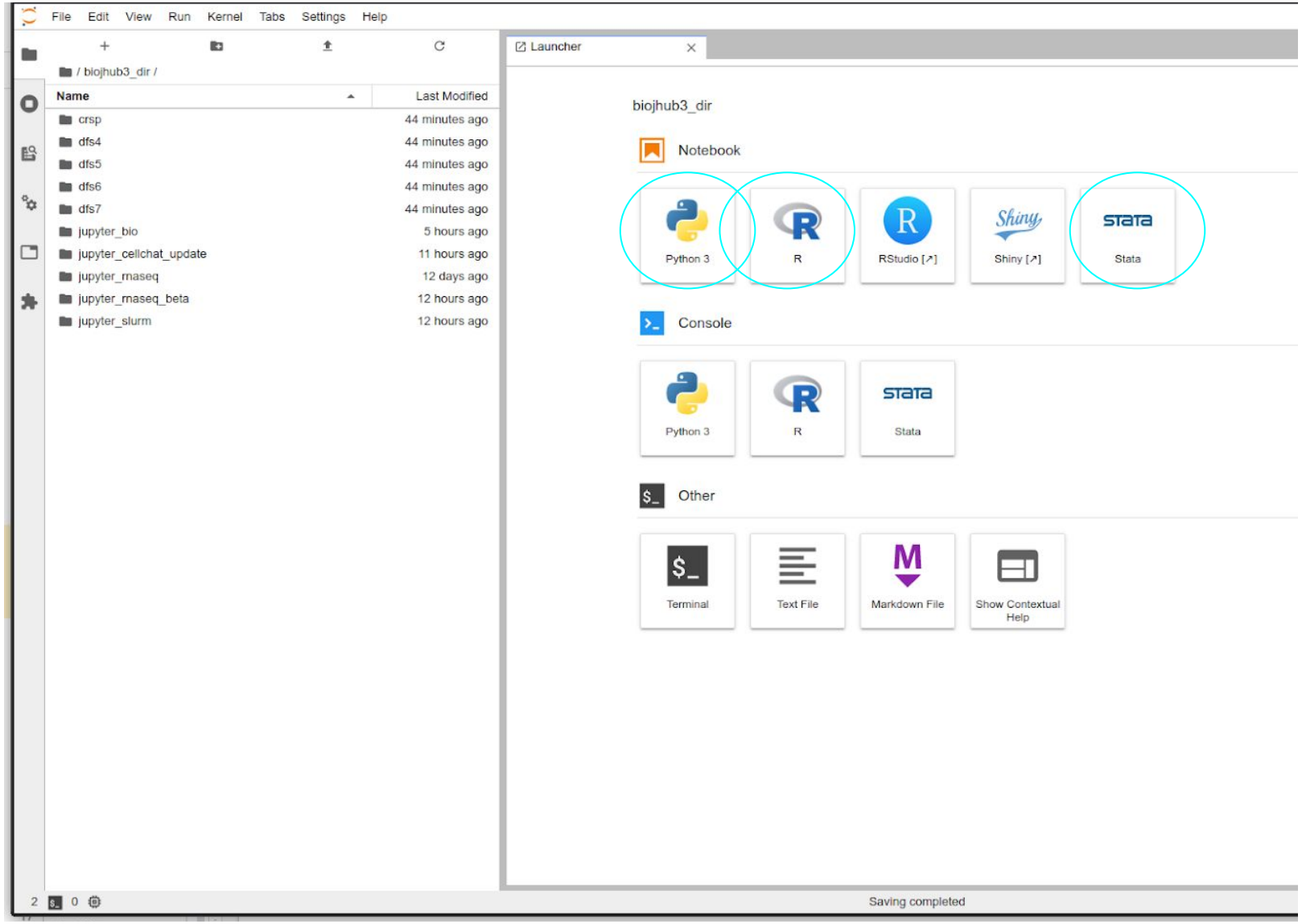

Once the notebook is done spawning, you will get a Launcher screen with a number GUI apps you can use.

# Starting a New App (Terminal)



# Jupyter Notebook




In this container, the user can open either a python, R, or Stata Jupyter computational notebook that will connect to the respective backend kernel and programming language environment.

Support for Matlab, Mathematica, and Julia are also available.

# Jupyter Notebook

File Edit View Run Kernel Tabs Settings Help

/ ··· / ucightf / workshop /

| Name | Last Modified |
|---|---|
| CCInx | 5 hours ago |
| CellChat | 5 hours ago |
| cellranger_out | 19 hours ago |
| SingleCellSignatureViewer | 13 hours ago |
| Seurat3_U54P30workshop_Jan21.ipynb | 15 hours ago |

Terminal 3 | Seurat3_U54P30workshop_

Markdown

## Seurat 3 Standard Processing of Single Sample

First load required libraries:

[28]:
```
library(Seurat)
library(dplyr)
```

Set up the pathway to Cellranger output and read the output to memory

[29]:
```
rootdir="/dfs6/pub/ucightf/workshop/cellranger_out"
pbmc.data <- Read10X(data.dir = rootdir)
```

Create a Seurat Object from the cellranger output

[30]:
```
#pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200)
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k")
pbmc
```

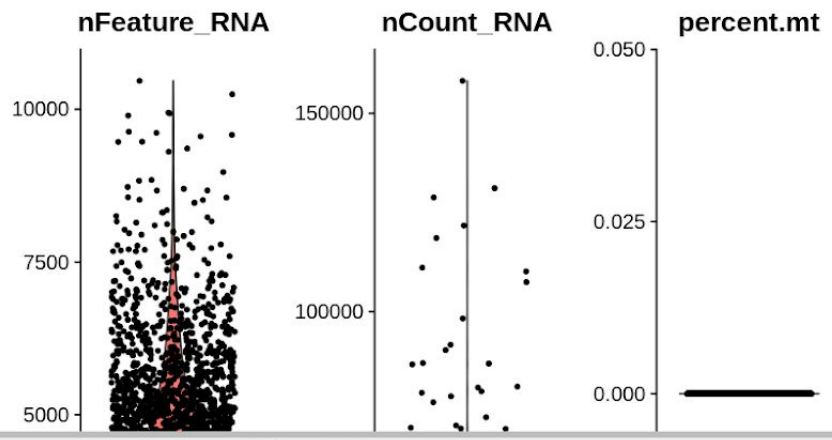
```
An object of class Seurat
29075 features across 8240 samples within 1 assay
Active assay: RNA (29075 features, 0 variable features)
```

## 1. Basic QC: compute percent of mitochondrial reads and add it to meta data. Violin plots for visual inspection. ¶

[31]:
```
# The [[ operator can add columns to object metadata. This is a great place to stash QC stats
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
```

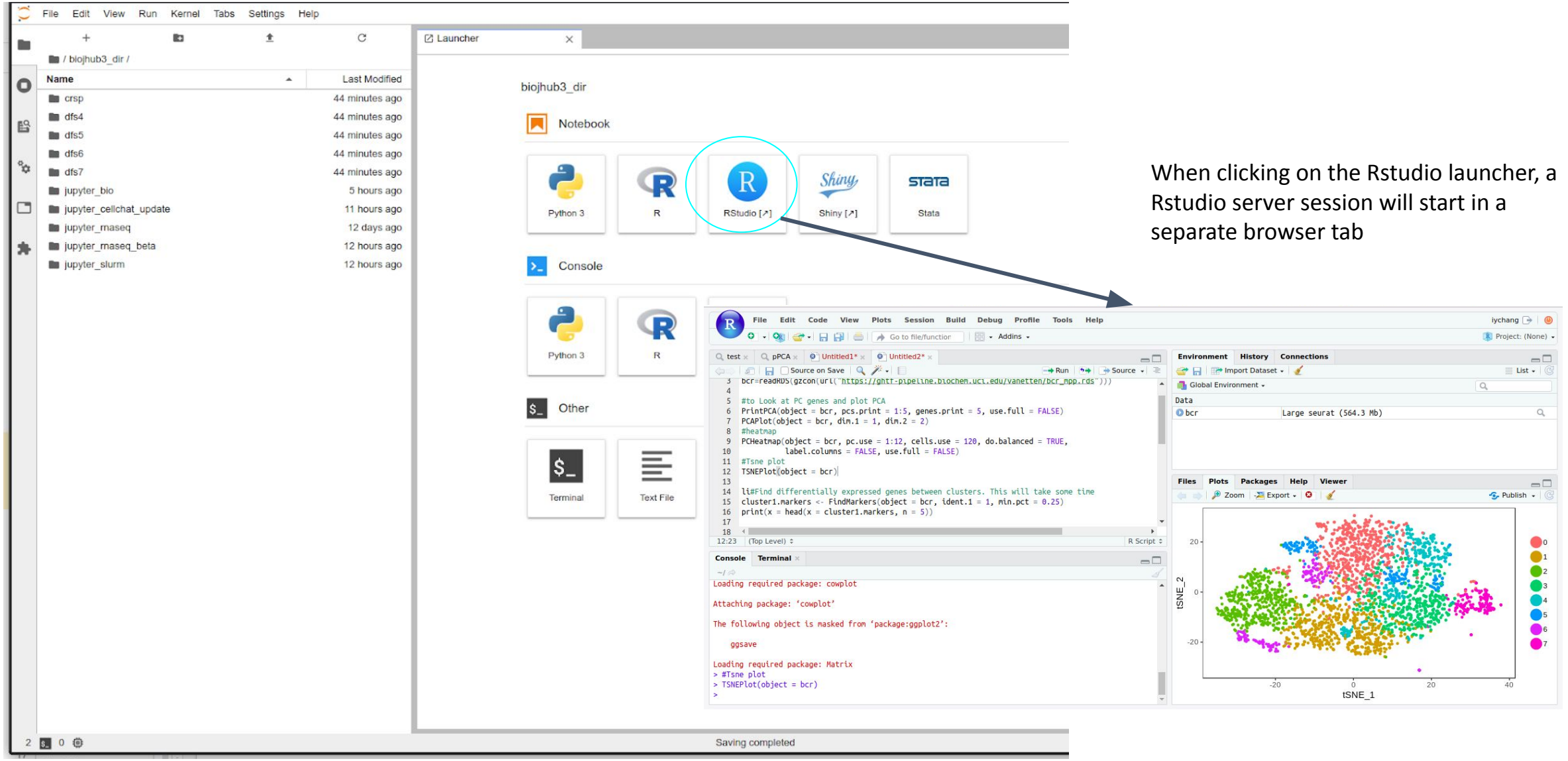
[32]:
```
# Visualize QC metrics as a violin plot
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```

```
Warning message in SingleExIPlot(type = type, data = data[, x, drop = FALSE], idents = idents, :
"All cells have the same value of percent.mt."
```




3 1 R | Idle | Saving completed | Mode: Command | Ln 1, Col 1 | Seurat3_U54P30workshop_Jan21.ipynb

# RStudio



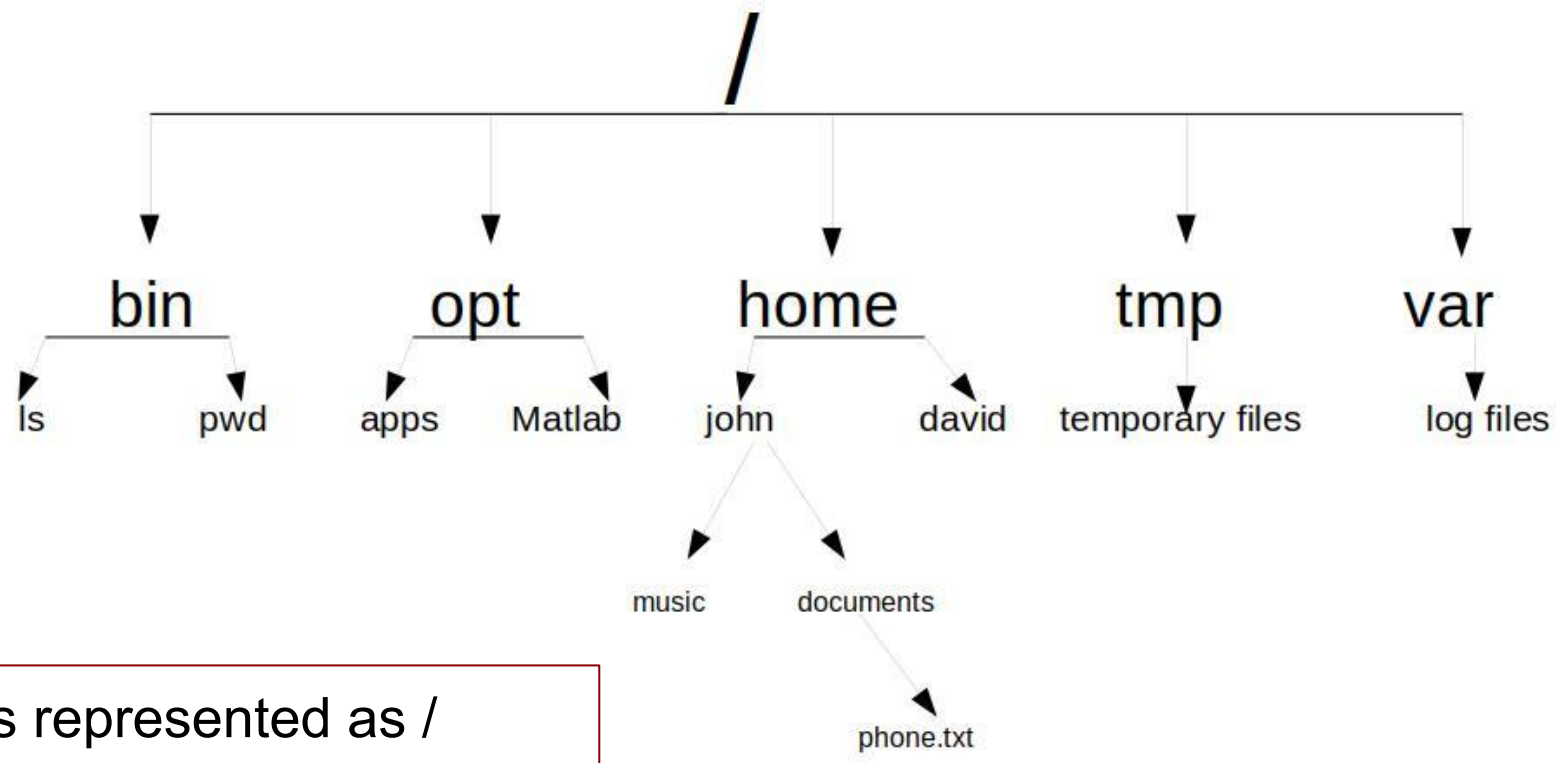

When clicking on the Rstudio launcher, a Rstudio server session will start in a separate browser tab

# Summary for Exercise #1

- SSH securely connects your computer to HPC3's login nodes in the command line interface.
- biojhub3 (jupyterhub) allows you to connect directly to a HPC3 compute node via the web browser interface.
- Jupyterhub provides a convenient and portable access to HPC3, but does not currently support X11 GUI programs and file transfer is limited to web upload protocol.

# Basic Linux commands

# Linux file system – directory tree




The root directory is represented as /
The home directory is represented as ~
The current directory is represented as .
The parent directory is represented as ..
The previous directory is represented as -

# Absolute and relative paths

- An *absolute path* begins with the root directory and follows the directory tree branch by branch until the path to the desired directory or file is completed. (e.g. `/home/john/documents/phone.txt`)

- A *relative path* starts from the current working directory (e.g. `./documents/phone.txt`)

# File Permissions

**Every file in Linux has the following access modes**:

| | |
|---|---|
| **read**, denoted as **r** | the capability to read or view the contents of the file. |
| **write**, denoted as **w** | the capability to modify and remove the content of the file. |
| **execute**, denoted as **x** | the capability to run a file as a program. |
| **sticky bit**, denoted as **s** | additional capability to set permissions for Set User ID (SUID) and Set Group ID (SGID) bits. |

**Every file in Unix has the following attributes or permissions** :

| | |
|---|---|
| **owner** | determine what actions the owner of the file can perform on the file. |
| **group** | determine what actions a user, who is a member of the group that a file belongs to, can perform on the file. |
| **other** (world) | determine what action all other users can perform on the file. |

```
[user@login-x:~]$ ls -l
total 55524423
drwxrwsr-x  7 panteater bio                7 Aug  5  2019 biofiles
-rw-r--r--  1 panteater panteater  4294967296 May 31  2019 performance.tst
```

# Exercise #2

Navigating file systems

# Live demo Linux navigation commands

| | |
|---|---|
| pwd | print working directory |
| cd | change directory |
| ls | list (files) |
| clear | clear terminal text (ctrl - l) |
| history | history of past commands |
| man | show the manual for a linux command |
| chown | change file ownership |
| chmod | change file permission |
| <Tab> | autocomplete |
| <Up>/<Down> | go back/forward one command |
| <Ctrl-r> | reverse history search |

# Exercise #3

Working with files (creating, viewing, copying, and editing)

# Live demo Linux commands

| | |
|---|---|
| touch | touch up the access bit of the file |
| mkdir | make directory |
| rm | remove (file/directory) |
| cp | copy file |
| mv | move file |
| ln | link files (default hardlink and *-s* softlink) |
| file | check file info |
| cat / tac | concatenate / reverse concatenate |
| head / tail | print the beginning/ending 10 lines of the file |
| more | show the contents of the file one page at a time |
| less | scroll up and down the contents of the file using navigation keys |
| vi / nano | command line text editing programs |

Useful code to generate test file: `for (( i=1; i<=100; i++)); do echo $i >> testline.txt;done`

# HPC3 specific commands

# Exercise #4

Managing your storage

**The HPC3 filesystem storage is generally in 3 areas. Please see the links below for detailed information about each filesystem.**

**HOME** **The HOME area** has a 50GB quota for each user. In addition, there is a space for snapshots. Total for home and snapshots is 100GB. Each user HOME is in /data/homezvolX/<account>

**DFS** **The BeeGFS Parallel storage File System (DFS)** access remains the same. All users have /pub/<account> area. Depending on a lab affiliation, users may have space in /dfs2, /dfs3a, /dfs3b, /dfs4, /dfs5 and /dfs6.

**CRSP** **The Campus Research Storage Pool (CRSP)** is available in /share/crsp. Depending on a lab affiliation, users may have space in /share/crsp/lab/<labname./<account>

# Check $HOME quota

**To see your current quota usage do:**

```
[user@login-x:~]$ df -h ~
Filesystem                          Size  Used Avail Use% Mounted on
 10.240.58.6:/homezvol0/panteater   50G  3.5G   47G   7% /data/homezvol0/panteater
```

The **~** stands for your $HOME. The output above shows that user panteater used 3.5Gb of its 50Gb allocation.

# If you want to see the usage by files and directories in $HOME

```
[user@login-x:~]$ cd
[user@login-x:~]$ ls
bin                examples     local        perl5
biojhub3_dir       info         mat.yaml     R
classify-image.py  keras-nn.py  modulefiles  sbank-out
```

change to your $HOME directory

list contents of $HOME

```
[user@login-x:~]$ du -s -h *
7.0M    bin
166M    biojhub3_dir
8.5K    classify-image.py
647K    examples
91K     info
4.5K    keras-nn.py
126M    local
4.5K    mat.yaml
60K     modulefiles
512     perl5
1.2G    R
 25K    sbank-out
```

find disk usage for each file and directory in $HOME. The output shows disk usage in kilobytes (K), megabytes (M) or gigabytes (G). For directories, all contents inside are included. For example, a directory **R** uses 1.2Gb of disk space.

# To see the quotas for user panteater on DFS pool /dfs6

```
[user@login-x:~]$ dfsquotas panteater dfs6
==== [Group Quotas on dfs6]

Quota information for storage pool Default (ID: 1):

      user/group     ||            size            ||    chunk files
     name      |  id  ||    used    |    hard    ||  used   |  hard
---------------|------||------------|------------||---------|---------
 panteater_lab|012345||   26.25 TiB|   50.00 TiB||  1310459| 18500000
   alpha users|158537||      0 Byte|      1 Byte||        0|        1
      panteater|000865||  755.59 GiB| 1024.00 GiB||   258856|unlimited
```

# Exercise #5

Data transfer from your computer to HPC3

# Data Transfer to HPC3

Often users need to brings data from other servers and laptops. To transfer data one needs to use `scp` (secure copy) or `rsync` (file copying tool). Alternatively, one can use graphical tools (Filezilla, MountainDuck, or WinSCP) to transfer files between a local laptop and the cluster. Follow each program instructions how to do this.

In all of the transfer application you will need to use hpc3.rcic.uci.edu to indicate a remote server (where you want to transfer your files) and use your UCNetID credentials for your user name and password.

# SCP (secure file transfer protocol)

Scp allows one to connect to a remote server and transmit desired files via the connection.

For example, a user has an access to a group allocation /dfsX/panteater_lab/panteater and want to transfer data there.

On your laptop or other server:

```
scp -r mydata panteater@hpc3.rcic.uci.edu:/dfsX/panteater_lab/panteater
```

where -r allows recursive copying of subdirectories

# RSYNC

Rsync is a program that allows to greatly speed up file transfers.

For example, for a recursive copy use:

```
rsync -rv mydata panteater@hpc3.rcic.uci.edu:/dfsX/panteater_lab/panteater
```

# Exercise #6

Selecting software modules

# Loading software modules

- **Environment module** is a user interface to the Modules package which provides for the dynamic modification of the user's environment via modulefiles.
- Each **modulefile** contains all the info needed to configure the shell to use a specific application.
- Command **module load** interprets the modulefiles and
  - Sets aliases
  - Sets environment variables
  - Loads depended modules
- Command **module avail** lists all installed software and their versions

General info for Linux https://modules.readthedocs.io/en/latest/
Read User guide for HPC3 https://rcic.uci.edu/hpc3/software-tutorial.html

# Environment modules update your environment

## Case 1: usage of multiple versions of software

```
login-i16 which R
/usr/bin/which: no R in (/usr/local/bin:/usr/bin:/usr/sbin:/data/homezvol0/npw/bin)
login-i16 module avail R
---------- /opt/rcic/Modules/modulefiles/LANGUAGES -------
R/3.6.2 R/4.0.2
login-i16 module load R/4.0.2
login-i16 which R
/opt/apps/R/4.0.2/bin/R
login-i16 module list
Currently Loaded Modulefiles:

1) OpenBLAS/0.3.6      2) java/1.8.0  3) icu/65.1
login-i16 module unload R/4.0.2
login-i16 module list
No Modulefiles Currently Loaded.
login-i16 module load R/3.6.2
login-i16 which R
/opt/apps/R/3.6.2/bin/R
```

## Case 2: load/unload different software modules

```
login-i16 module load gcc/8.4.0
login-i16 module list
Currently Loaded Modulefiles:
  1) gcc/8.4.0
login-i16 module load hdf5/1.10.5/gcc.8.4.0
login-i16 module list
Currently Loaded Modulefiles:
  1) gcc/8.4.0 2) java/1.8.0  3) hdf5/1.10.5/gcc.8.4.0
login-i16 module unload hdf5/1.10.5/gcc.8.4.0
login-i16 module list
Currently Loaded Modulefiles:
  1) gcc/8.4.0
```

Always unload module in reverse order: FILO!

# Environment module commands summary

search

| | |
|---|---|
| $ module avail | shows all installed software environment |
| $ module avail R | modules show R modules |
| $ module keyword salmon | check all modules for a keyword |
| salmon/1.1.0 : Name salmon | |
| salmon/1.1.0 : | |

info

| | |
|---|---|
| $ module display R | shows environment modification + description |
| $ module help R | show module specific help (description) |

use

| | |
|---|---|
| $ module load R | loads R at whatever latest version not ideal |
| $ module load R/4.0.2 | loads R at specified version preferred method |
| $ module list | lists currently loaded modules |
| $ module unload R/4.0.2 | unloads specified module (in reverse order if many) |
| $ module purge | removes all loaded modules |

# Exercise #7

Sample job submission

# HPC3 SLURM

**Slurm is an open-source workload manager for Linux clusters and provides:**

1. access to resources (computer nodes) to users so they can run their applications.
2. framework to start, execute, and monitor work on a set of allocated nodes.
3. management of a queue for pending work.

Helpful UCI HPC3 specific slurm guide:

https://rcic.uci.edu/hpc3/slurm.html

# Simple code of conduct for running applications on HPC3

1. **All jobs, batch or interactive must be submitted to the scheduler**
2. **Do not run computational jobs on login nodes** this adversely affects many users. Login nodes are meant for light editing or compilation and for submitting jobs. Any job that runs for more than an hour or is using significant memory and CPU within an hour should be submitted to Slurm either as interactive or batch job.
3. **Ssh access to the compute nodes is turned off** to prevent users from starting jobs bypassing Slurm. See **attaching to running job** below.
4. **Do not run Slurm jobs in your $HOME**.
5. **Make sure you stay within your disk quota**. File system limits are generally the first ones that will negatively affect your job. See **storage guides**

# Cluster Partitions

HPC3 has different kinds of hardware, memory footprints, and nodes with GPUs. All nodes (servers) all are separated into groups according to their resources. Slurm uses the term partition to signify a queue of resources. We have a few separate partitions, most users will need to use *standard* and *free* partitions:

- **standard partition** is for jobs that should not be interrupted. Usage is charged against the user's Slurm bank account. Each user gets FREE one time allocation of 1000 core hours to run jobs here. **Users are NOT CHARGED ANY $**. If all allocation is used, users can run jobs only if they are associated with labs that have core hours in their lab banks. Usually, lab bank is a PI lab account.
- **free partition** is for jobs that can be preempted (killed) by standard jobs. Users can run jobs in this partition even if they have only 1 core-hour left. There are no charges for this partition.

# HPC3 Policies for CPU and memory scheduling

| Partition | Default memory/core | Max memory/core | Default / Max runtime | Cost | Jobs preemption |
|---|---|---|---|---|---|
| **CPU Partitions** | | | | | |
| **standard** | 3GB | 6GB | 2day / 14day | 1 / core-hr | No |
| **free** | 3GB | 18GB | 1day / 3day | 0 | Yes |
| **debug** | 3GB | 18GB | 15min / 30min | 1/core-hr | No |
| **highmem** | 6GB | 10GB | 2day / 14day | 1/core-hr | No |
| **hugemem** | 18GB | 18GB | 2day / 14day | 1/core-hr | No |
| **GPU Partitions** | | | | | |
| **gpu** | 3GB | 9GB | 2day / 14day | 1/core-hr, 32/GPU-hr | No |
| **free-gpu** | 3GB | 9GB | 1day / 3day | 0 | Yes |
| **gpu-debug** | 3GB | 9GB | 15min / 30min | 1/core-hr, 32/GPU-hr | No |

# Checking your allocations

`sbank` is short for "Slurm Bank". `Sbank` is used to display HPC3 user account information. In order to run jobs on HPC3, a user must have available CPU hours. To check how many CPU hours are available in your personal account, run the command with your account name:

```
[user@login-x:~]$ sbank balance statement -a panteater
User           Usage |        Account     Usage | Account Limit Available (CPU hrs)
---------- --------- + -------------- --------- + ------------- ---------
 panteater*        58 |      PANTEATER        58 |         1,000       942
```

To check how many CPU hours are available in all accounts that you have access to and how much you used:

```
[user@login-x:~]$ sbank balance statement -u panteater
User           Usage |        Account     Usage | Account Limit Available (CPU hrs)
User           Usage |        Account     Usage | Account Limit Available (CPU hrs)
---------- --------- + -------------- --------- + ------------- ---------
panteater*        58 |      PANTEATER        58 |         1,000       942
 panteater*    6,898 |         PI_LAB     6,898 |       100,000    93,102
```

# Slurm interactive jobs

To request an interactive job, use the `srun` command. Suppose you are enabled to charge to the panteater_lab account then, to start an interactive session you can use one of 3 methods :

```
[user@login-x:~]$ srun --pty /bin/bash -i                    (1)
[user@login-x:~]$ srun --pty -p free /bin/bash -i            (2)
[user@login-x:~]$ srun -A panteater_lab --pty /bin/bash -i   (3)
```

After you are done use `logout` command to logout:

```
[user@hpc3-l18-04:~]$ logout
```

1) you will be put on an available node in standard partition using your default Slurm bank account

2) you will be put on an available node in free partition using your default Slurm bank account

3) you will be put on an available node in standard partition using panteater_lab account

See 2.4. Interactive Job of https://rcic.uci.edu/hpc3/slurm.html for more options

# Slurm Batch Job

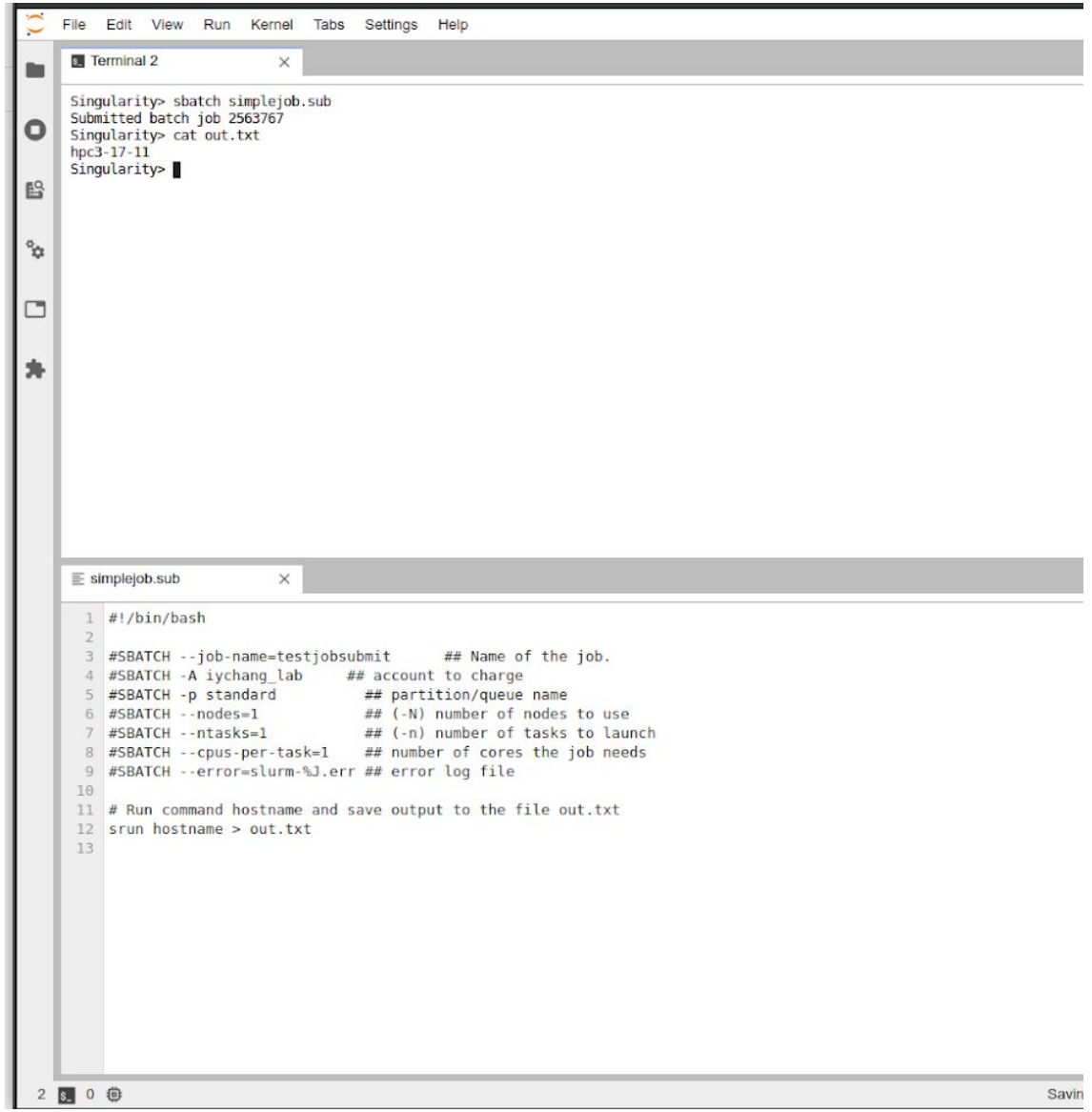
**simplejob.sub**

```
#!/bin/bash

#SBATCH --job-name=test       ## Name of the job.
#SBATCH -A panteater_lab      ## account to charge
#SBATCH -p standard           ## partition/queue name
#SBATCH --nodes=1             ## (-N) number of nodes to use
#SBATCH --ntasks=1            ## (-n) number of tasks to launch
#SBATCH --cpus-per-task=1     ## number of cores the job needs
#SBATCH --error=slurm-%J.err ## error log file

# Run command hostname and save output to the file out.txt
  srun hostname  > out.txt
```

To submit the job do:

```
[user@login-x:~]$ sbatch simplejob.sub
  Submitted batch job 362
```

Please look through https://rcic.uci.edu/hpc3/examples.html for different job examples



# Job status

To check the status of your job in the queue:

```
[user@login-x:~]$ squeue -u panteater
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               362  standard     test panteater R       0:03      1 hpc3-17-11
```

To get detailed info about the job:

```
[user@login-x:~]$ scontrol show job 362
```

The output will contain a list of key=value pairs that provide job information.

To cancel a specific job:

```
[user@login-x:~]$ scancel <jobid>
```

# Job history

We have a cluster-specific tool to print a ledger of jobs based on specified arguments. Default is to print jobs of the current user for the last 30 days:

```
[user@login-x:~]$ /pub/hpc3/zotledger -u panteater
      DATE       USER    ACCOUNT PARTITION    JOBID  JOBNAME ARRAYLEN CPUS WALLHOURS     SUs
2021-07-21  panteater  panteater  standard  1740043     srun        -    1      0.00    0.00
2021-07-21  panteater  panteater  standard  1740054     bash        -    1      0.00    0.00
2021-08-03  panteater     lab021  standard  1406123     srun        -    1      0.05    0.05
2021-08-03  panteater     lab021  standard  1406130     srun        -    4      0.01    0.02
2021-08-03  panteater     lab021  standard  1406131     srun        -    4      0.01    0.02
    TOTALS          -          -         -        -        -    -      0.07    0.09
```

To find all available arguments use:

```
[user@login-x:~]$ /pub/hpc3/zotledger -h
```

# Job info

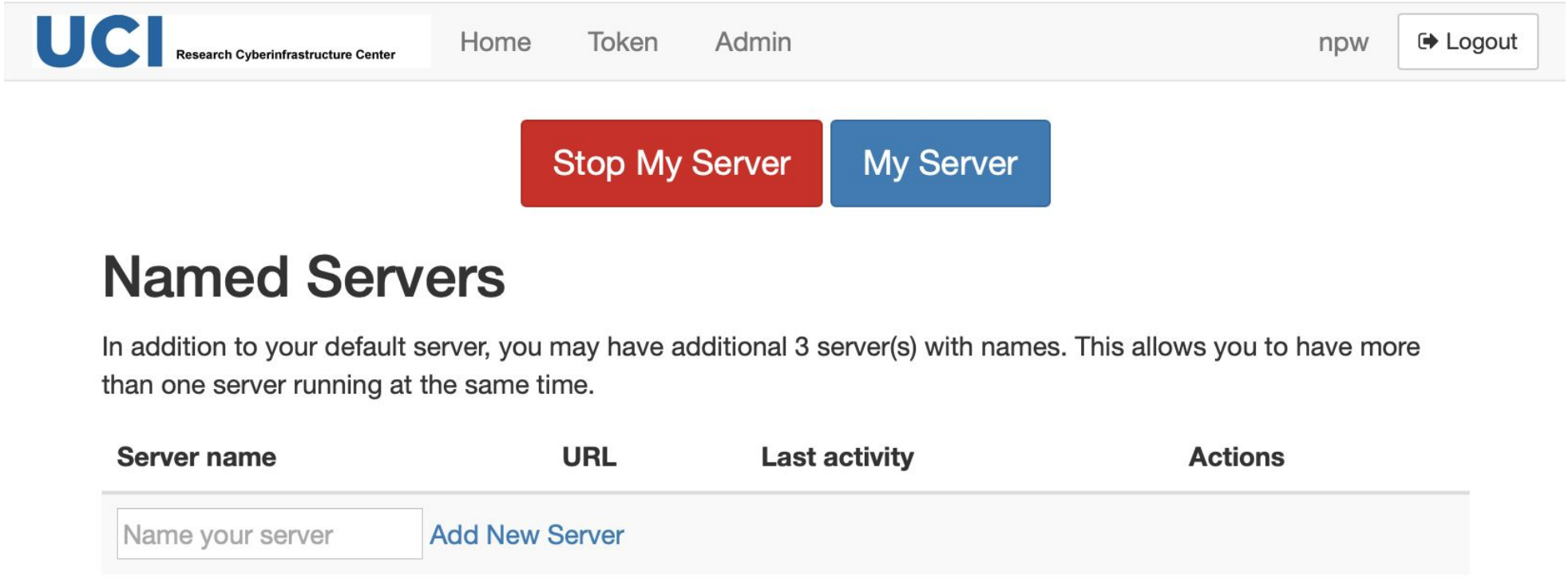
`sacct` can be used to see accounting data for all jobs and job steps. An example below shows how to use job ID for the command:

```
[user@login-x:~]$ sacct -j 43223
       JobID  JobName  Partition      Account  AllocCPUS      State ExitCode
------------ -------- ---------- ------------ ---------- ---------- --------
   36811_374    array   standard panteater_l+          1  COMPLETED      0:0
```

The above command uses a default output format. A more useful example will set a specific format for sacct that provides extra information:

```
[user@login-x:~]$ export
SACCT_FORMAT="JobID%20,JobName,Partition,Elapsed,State,MaxRSS,AllocTRES%32"
[user@login-x:~]$ sacct -j 600
     JobID JobName  Partition  Elapsed     State  MaxRSS AllocTRES
---------- -------  --------  -------- --------- ------- --------------------------------
       600    all1  free-gpu  03:14:42 COMPLETED         billing=2,cpu=2,gres/gpu=1,mem=+
 600.batch   batch            03:14:42 COMPLETED 553856K           cpu=2,mem=6000M,node=1
 600.extern  extern            03:14:42 COMPLETED       0 billing=2,cpu=2,gres/gpu=1,mem=+
```

Be sure to stop your Juputerhub notebook server after you are done. From the **File** menu choose **Hub Control Panel** and you will be forwarded to a screen similar where you can press on `Stop My Server` to shut down the server:



# Acknowledgement

GHTF: Suzanne Sandmeyer, Melanie Oakes, Jenny Wu, Christina Lin

RCIC: Phil Papadopoulos, Imam Toufique, Francisco Lopez, Nick Santucci,
Joulien Tatar, Nadya Williams

# Extra Slides

# HPC3 Reference Guides

(https://rcic.uci.edu/hpc3/hpc3-reference.html#connect)

| | |
|---|---|
| **Getting an account** | send email to **hpc-support@uci.edu** |
| **Logging in** | **Connecting to HPC3** |
| **Submitting your first job** | **SLURM tutorial** |
| **Available software** | **Environment modules tutorial** |
| **Purchasing Hardware/Core Hours** | **Beyond baseline allocation** |
| **All about accounting** | **Free and Accounted jobs** |
| **Storage** | **Home Area, Parallel File Systems, and CRSP** |
| **Getting Help** | **Ask for help or software install** |

# Talking to RCIC and to Each Other

- **How do I ask for help/talk to RCIC?**
  - Send email to hpc-support@uci.edu

    This automatically creates a help ticket
  - Read that fine website: https://rcic.uci.edu


- **What about talking to RCIC and the other users at UCI?**
  - Join the Google group https://groups.google.com/a/uci.edu/g/rcic-users
  - Chat with us on Slack: https://rcicos.slack.com/

# RCIC Recommended Online Tutorials

**The Missing Semester of Your CS Education** Many topics as separate lectures, including Shell Tools and Scripting, Editors (Vim), Command-line Environment, Data Wrangling, Git, security and more.

**The Software Carpentry** teaches basic skills via workshops and lessons, here are direct links:

- **The UNIX Shell** The Unix shell fundamentals
- **Introduction to Python** Learn the basics of Python programming language.
- **Introduction to R** Learn the basics of R programming language.

# Basics of being a good citizen on a cluster

1. Cluster is a shared resource, it is NOT your personal machine
2. What you do affects all the other users, so think before you hit that *Enter* key
   - Do not run interactive jobs on login nodes
   - Do not transfer data on login nodes
3. Secured from mischief and disasters.
   - We restrict users' ability (permissions) to install and run unwanted software applications
   - It is your responsibility to act secure
   - Be careful when bringing applications from unknown sources. DO NOT ask for sudo access
4. For your jobs: use resources you need, don't ask for more

   Study this Slurm guide https://rcic.uci.edu/hpc3/slurm.html
5. Be mindful how you submit tickets

   https://rcic.uci.edu/hpc3/getting-help.html#_how_to_ask_for_help

# High-level View of what things cost

## No Cost Allocations

| Role | HPC3 Core Hours | GPU Hours | Home Area Storage | DFS Storage | CRSP Storage |
|---|---|---|---|---|---|
| Faculty | 200K hours/year[1] | By Request ~2K hours/year[1] | 50GB | 1TB in Pub | 1 TB |
| Student | 1000 hours | --- | 50GB | 1 TB in Pub | --- |

## Cloud-like Costs

| | HPC3 Core Hours | GPU Hours | Home Area Storage | DFS Storage | CRSP Storage |
|---|---|---|---|---|---|
| Faculty | $.01/core hour | $0.32/GPU hour | Not expandable | $100/TB/5 years | $60/TB/year |
| AWS Equivalent | C5n.large $.063 | P3.2xlarge $1.95 | --- | --- | S3[2] Standard $242/TB/year |

[1] Exact amounts dependent on # requests/available hardware

[2] Comparison difficult - S3 has higher durability, CRSP has no networking fee.

# HPC3 Policies for CPU and memory scheduling

| Partition | Default memory/core | Max memory/core | Default / Max runtime | Cost | Jobs preemption |
|---|---|---|---|---|---|
| **CPU Partitions** | | | | | |
| **standard** | 3GB | 6GB | 2day / 14day | 1 / core-hr | No |
| **free** | 3GB | 18GB | 1day / 3day | 0 | Yes |
| **debug** | 3GB | 18GB | 15min / 30min | 1/core-hr | No |
| **highmem** | 6GB | 10GB | 2day / 14day | 1/core-hr | No |
| **hugemem** | 18GB | 18GB | 2day / 14day | 1/core-hr | No |
| **GPU Partitions** | | | | | |
| **gpu** | 3GB | 9GB | 2day / 14day | 1/core-hr, 32/GPU-hr | No |
| **free-gpu** | 3GB | 9GB | 1day / 3day | 0 | Yes |
| **gpu-debug** | 3GB | 9GB | 15min / 30min | 1/core-hr, 32/GPU-hr | No |

# Symbolic link

Often times, it is simpler to have a direct link to a long file path in your $HOME directory. It is also necessary to define such links to paths outside of $HOME in the Jupyterlab file navigation.

For this workshop, let's create a symbolic directory link "workshop" in your $HOME directory to the workshop example directory located at /dfs6/pub/ucightf/workshop using the *ln* command

The syntax for *ln* is as follows: *ln -s* **TARGET linkname**

so in our case, open your terminal window and navigate to $HOME by using the command *cd*

and then type in the command: *ln -s /dfs6/pub/ucightf/workshop workshop*

if you then change into *workshop* via the command: *cd $HOME/workshop*

you should be directed to the contents of */dfs6/pub/ucightf/workshop*