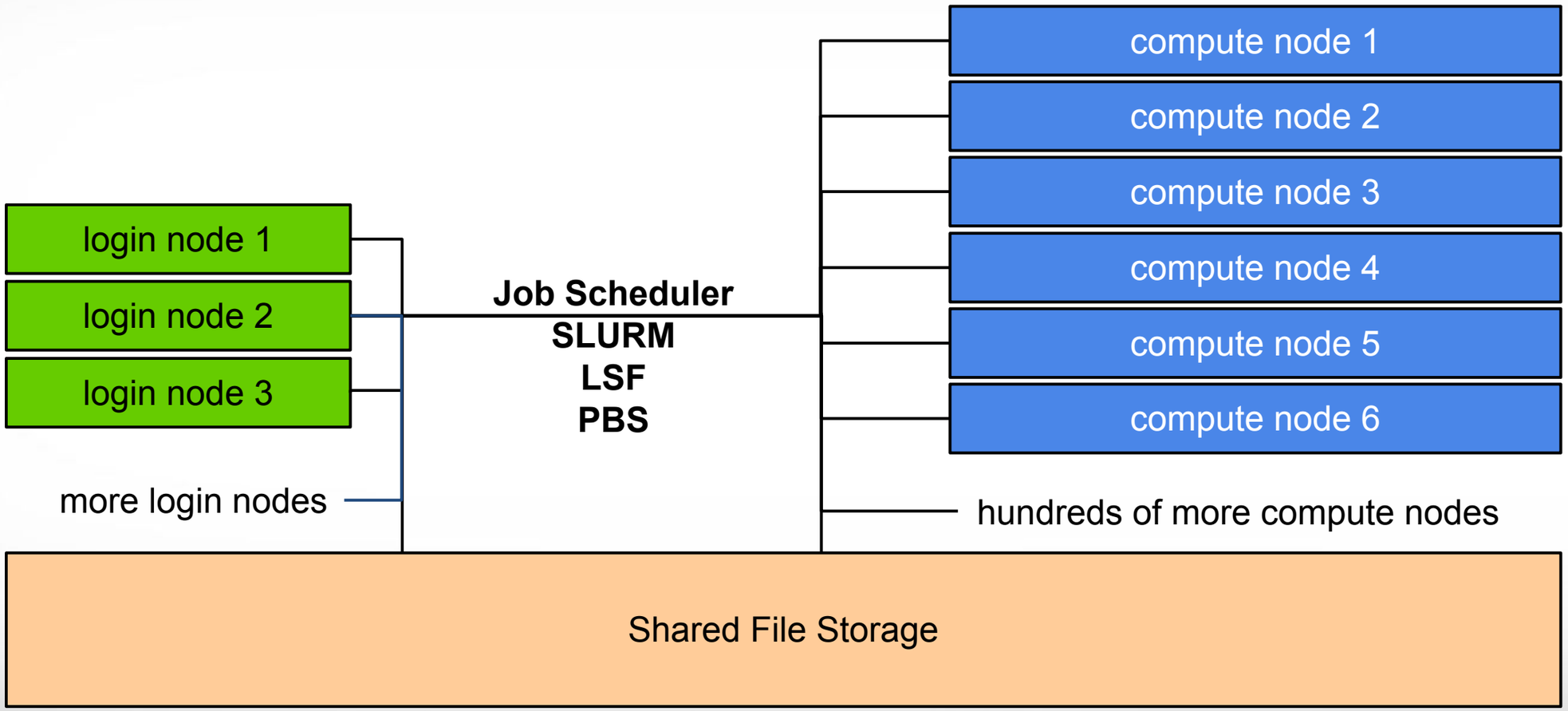
Introduction to HPRC at TAMU

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

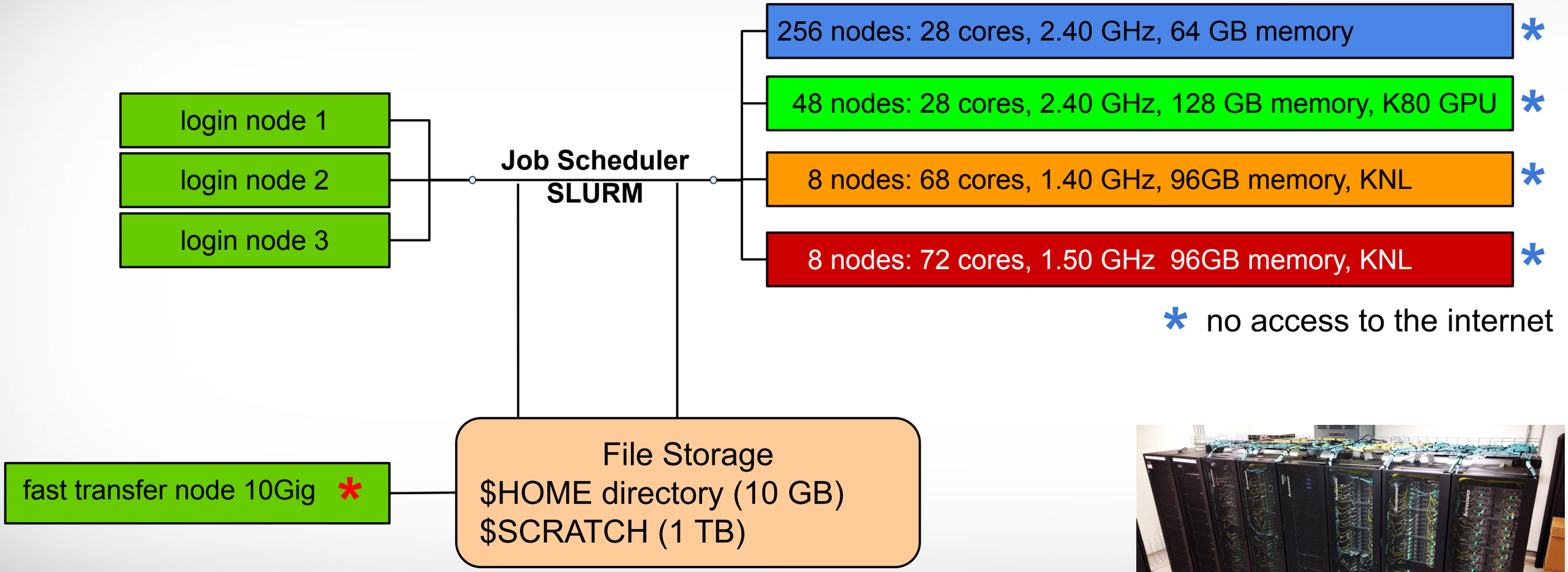
for OSOS Workshop, Sep 1, 2019

- HPRC Clusters Overview
- Accessing HPRC Clusters
- Service Units
- Software Modules
- Batch Job Scripts
- R (command line)
- HPRC Portal
  - RStudio (Ada)
  - Beauti, Beast (Terra)
  - FigTree, Tracer (Terra)

HPC Diagram


compute node 1
compute node 2
compute node 3
compute node 4
compute node 5
compute node 6
login node 1
login node 2
login node 3
Job Scheduler
SLURM
LSF
PBS
more login nodes
hundreds of more compute nodes
Shared File Storage

Terra HPC Diagram

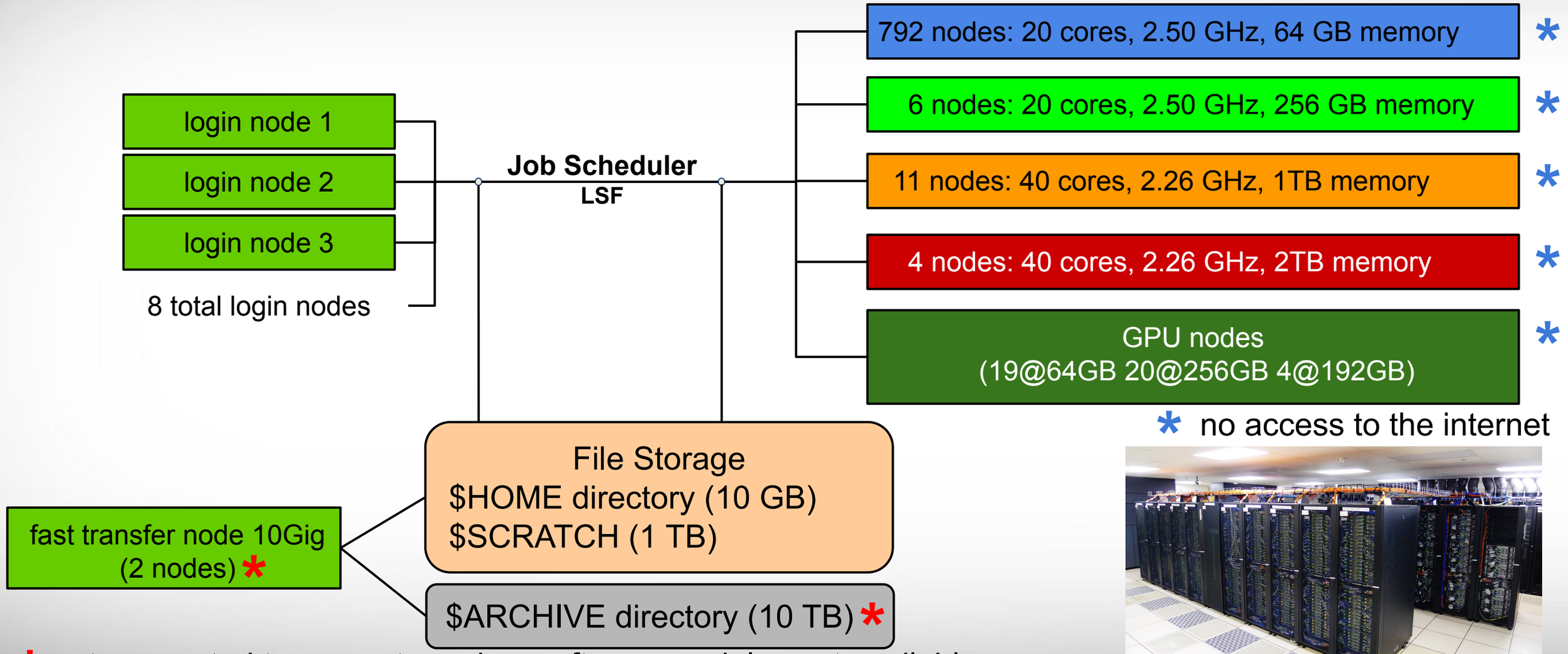


* no access to the internet

* not connected to compute nodes; software modules not available
* the amount of usable memory is node type minus 10GB for the OS

Production Date Spring 2017

Ada HPC Diagram



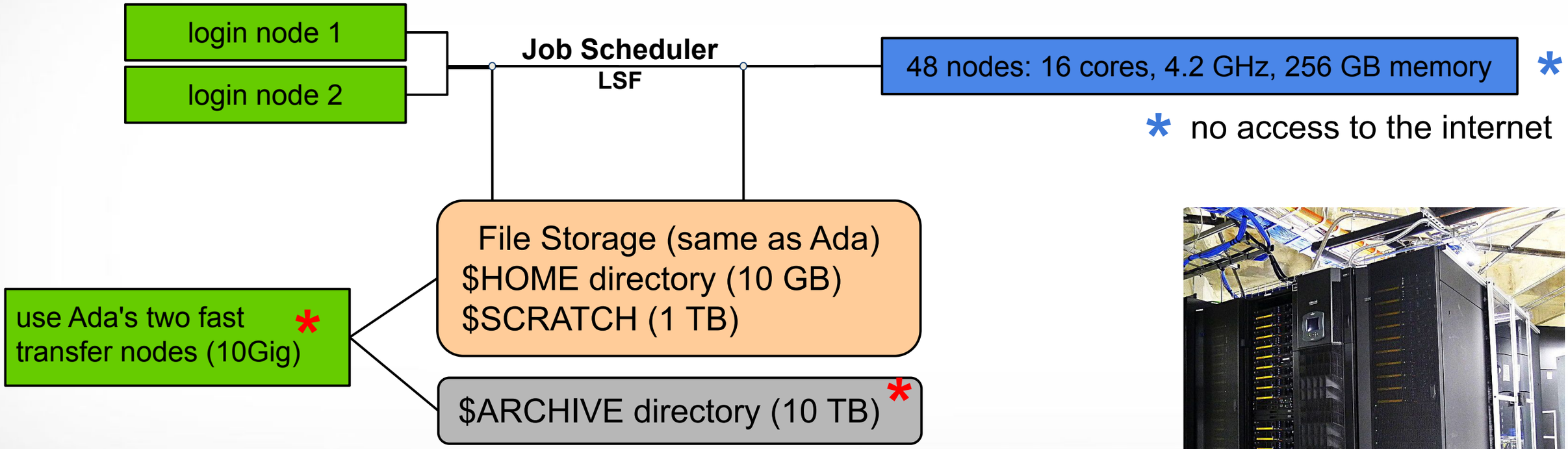
* no access to the internet

Production Date Sept 2014

* not connected to compute nodes; software modules not available

* the amount of usable memory is node type minus 10GB for the OS

Curie HPC Diagram

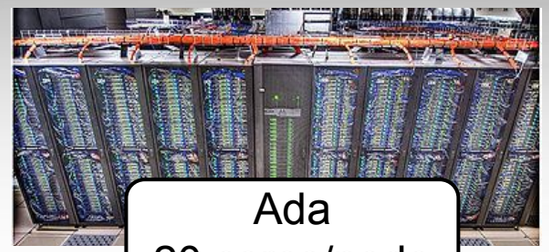


* not connected to compute nodes; software modules not available
* the amount of usable memory is node type minus 10GB for the OS

Production Date May 2015

HPRC Clusters

Terra
28 cores/node



Ada
20 cores/node

Curie
16 cores/node

Login Nodes	3	8	2
64 GB memory compute nodes	256	792	-
PHI 64 GB memory compute nodes	-	9	-
KNL (Knights Landing) (no SUs charged)	16 (68 cores/node)	-	-
K20 GPU 64 GB memory compute nodes	-	10	-
K80 GPU 128 GB memory compute nodes	48	-	-
K20 GPU 256 GB memory compute nodes	-	20	-
V100 192 GB memory compute nodes	-	4 (24 cores/node)	-
256 GB memory compute nodes	-	6	70
1 TB memory compute nodes	-	11 (40 cores/node)	-
2 TB memory compute nodes	-	4 (40 cores/node)	-

hprc.tamu.edu/resources

Historical HPRC Cluster Usage

hprc.tamu.edu

Cluster Status

Terra

Nodes	293/318 (92%)
Cores	7439/9496 (78%)
Jobs	89R-263Q

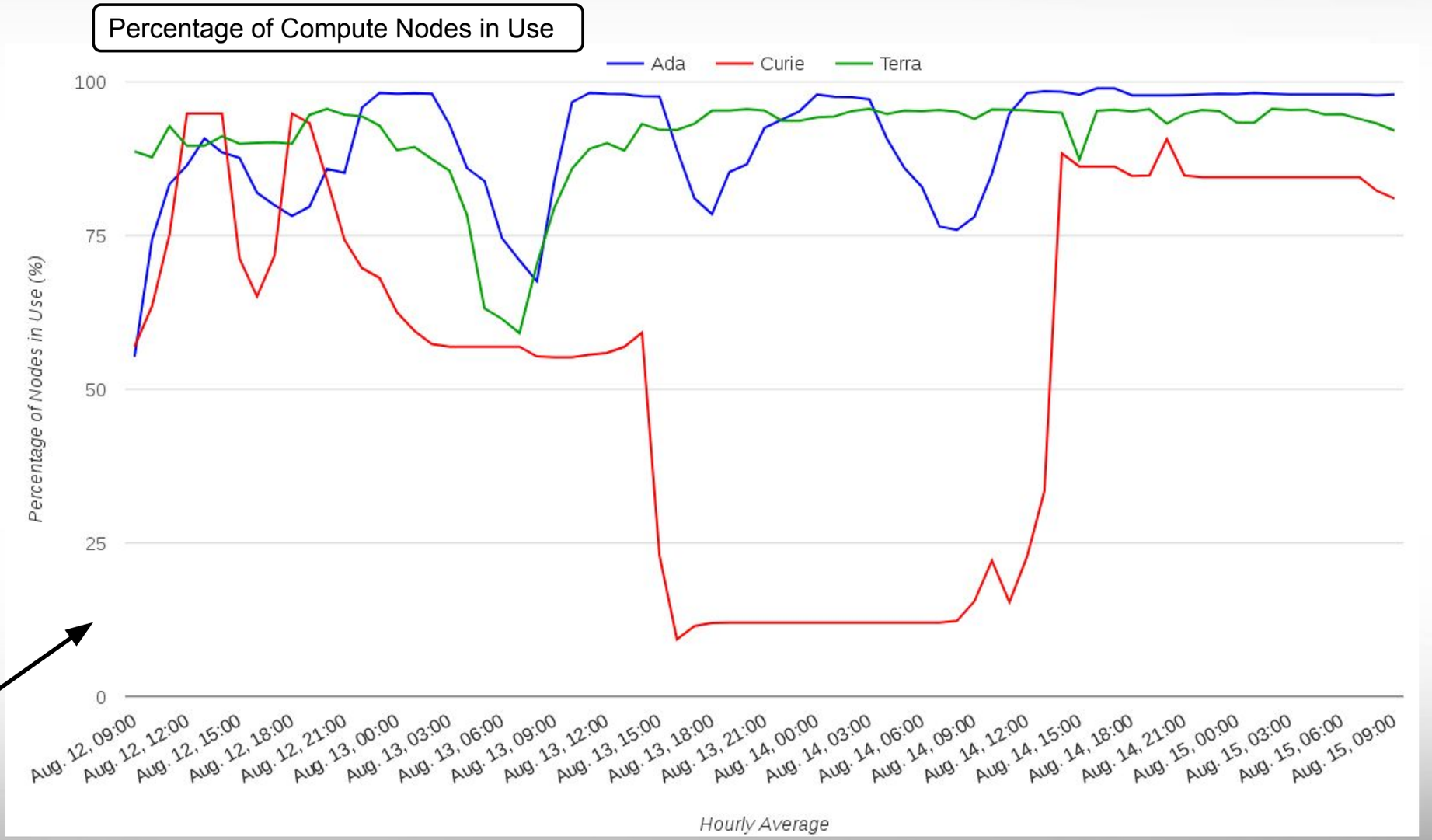
Ada

Nodes	797/814 (98%)
Cores	15539/16572 (94%)
Jobs	623R-343Q

Curie

Nodes	47/58 (81%)
Cores	752/927 (81%)
Jobs	47R-0Q

Historical Status



Which HPRC Cluster to Use?

- **Terra**
  - lots of common bioinformatics tools available
    - BEAST, BEAUti, (best accessed via portal), FastTree (portal), SAMtools, BWA, minimap2, R 3.6.0
  - plus some tools that require a newer operating system
    - Arrow (ArrowGrid_HPRC), Purge_Haplotigs, pbbioconda
  - Compute nodes have 64GB memory, GPU compute nodes have 96GB and 128GB memory
  - HPRC Portal apps: Tracer, FigTree, BEAUti, VNC (for BEAST), JBrowse
- **Ada**
  - has more bioinformatics tools installed but Terra is not far behind
    - BEAST, FastTree, SAMtools, BWA, minimap2, R 3.6.0
  - more software with complex installations and large databases is available
    - Maker, BLAST nt and nr databases
  - 1TB and 2TB memory compute nodes are available
  - HPRC Portal apps: RStudio (R 3.4.3 and 3.6.1), IGV, Galaxy
- **Curie**
  - *No SUs are charged* (filesystem and file quotas are the same as Ada)
  - some software cannot be built on Curie due to the IBM POWER7 architecture
  - some common tools available
    - BamTools, BCFtools, Biopython, CD-HIT, R 3.6.0
  - can run Canu assembler in grid mode (multi-node assembly)
  - all 48 compute nodes have 256GB memory and processor speed 4.2 GHz

HPRC Account and Service Units

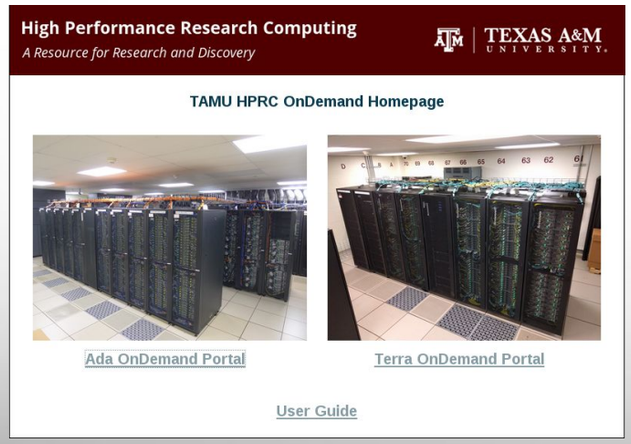
- Apply for an HPRC account: hprc.tamu.edu/apply
- HPRC clusters are free to use for TAMU faculty, staff and students
  - Usage measured by Service Units
  - 1 Service Unit (SU) = one core or 2GB memory usage per hour (2.5GB on Ada)
  - ***Basic*** = 5,000 Basic Service Units for Ada and Terra
    - No SUs charged on Curie
  - ***Startup*** = up to 200,000 SUs and two startup accounts per fiscal year
  - ***Research*** = millions of SUs, for large projects or PIs with many researchers
  - *Startup* and *Research* accounts are reserved for Faculty and Staff
  - Use **`myproject`** at the command line to query your balance

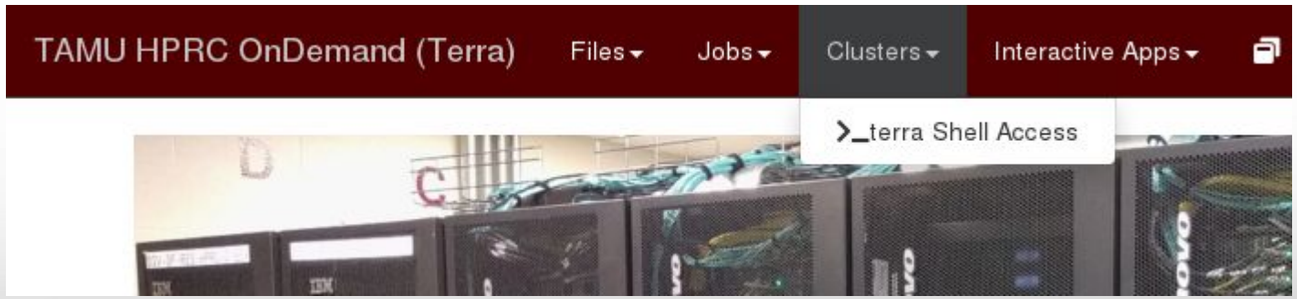
```
================================================================================
               List of YourNetID's Project Accounts
--------------------------------------------------------------------------------
|   Account    |   FY   | Default | Allocation |Used & Pending SUs|   Balance   |     PI     |
--------------------------------------------------------------------------------
|1228000223136|  2020  |       N|    10000.00|              0.00|    10000.00|Doe, John   |
--------------------------------------------------------------------------------
|1428000243716|  2020  |       Y|     5000.00|            -71.06|     4928.94|Doe, Jane   |
--------------------------------------------------------------------------------
```

- Users with Basic accounts will need to contact PI to have more SUs allocated from their PI's account if the user runs out of their SUs hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit

Accessing Terra

- SSH command is required for accessing HPRC Clusters command line:
  - On campus:
    - **`ssh NetID@terra.tamu.edu`**
  - Off campus:
    - Install and start VPN (Virtual Private Network): u.tamu.edu/VPnetwork
    - Then: **`ssh NetID@terra.tamu.edu`**
  - SSH Unix terminal is available on Linux and Mac
  - SSH programs for Windows; MobaXterm recommended
- Access GUI apps or the Unix command line through portal.hprc.tamu.edu

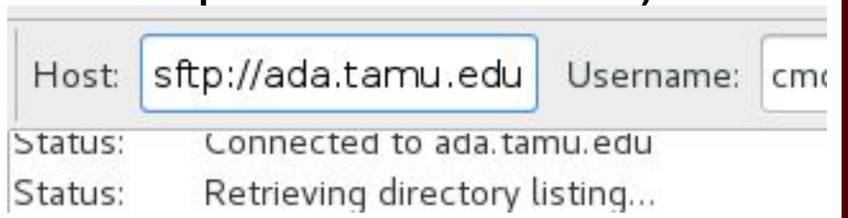




hprc.tamu.edu/wiki/HPRC:Access

File Transfers

- Simple File Transfers:
  - scp: command line (Linux, Mac)
  - rsync: command line (Linux, Mac); ***can resume transfer***
  - MobaXterm: GUI (Windows)
  - WinSCP: GUI (Windows)
  - FileZilla: GUI (Linux, Mac, Windows) (use sftp protocol sftp://ada.tamu.edu)
  - Portal: portal.hprc.tamu.edu
- Bulk data transfers:
  - Use fast transfer nodes
    - data transfer processes will not timeout at 60 minutes
    - **`terra-ftn.hprc.tamu.edu`**
    - **`ada-ftn1.tamu.edu or ada-ftn2.tamu.edu`**
    - Globus Connect (hprc.tamu.edu/wiki/SW:GlobusConnect)
    - GridFTP



hprc.tamu.edu/wiki/HPRC:FileTransfers

File Systems and User Directories

Directory	Environment Variable	Space Limit	File Limit	Intended Use
/home/$USER	$HOME	10 GB	10,000	Small to modest amounts of processing.
/scratch/user/$USER	$SCRATCH	1 TB	50,000	Temporary storage of large files for on-going computations. Not intended to be a long-term storage area.
/tiered/user/$USER	$ARCHIVE	10 TB	50,000	Intended to hold valuable data files that are not frequently used **(on Ada and Curie only)**

- **`$HOME` `$SCRATCH`** and **`$ARCHIVE`** directories are not shared between Ada and Terra clusters.
- **`$HOME` `$SCRATCH`** and **`$ARCHIVE`** directories are shared between Ada and Curie clusters.
- View usage and quota limits using the command: **`showquota`**
- Quota and file limit increases are only available for **`$SCRATCH`** and **`$ARCHIVE`** directories
- Request a group directory for sharing files.
- Do not share your **`$HOME` `$SCRATCH`** or **`$ARCHIVE`** directories.

hprc.tamu.edu/wiki/Ada:Filesystems_and_Files
hprc.tamu.edu/wiki/Terra:Filesystems_and_Files

Software

- See the Software wiki page for instructions and examples
  - hprc.tamu.edu/wiki/Bioinformatics
  - hprc.tamu.edu/wiki/SW
  - hprc.tamu.edu/software/terra
- Contact us for software installation help/request
  - User can install software in their **`$SCRATCH`** dir
  - Do not run the "*sudo*" command when installing software

Computing Environment

- Software modules are used to load software into the working environment
- Each version of a software, application, library, etc. is available as a module.
  - Toolchain is the type and version of compiler used to build software

software_name / sw_version toolchain [Python-version]

```
TopHat/2.1.1-intel-2017A-Python-2.7.12
```

software_name / sw_version precompiled Linux binary

```
SPAdes/3.13.0-Linux
```

- load software into working environment with the module command

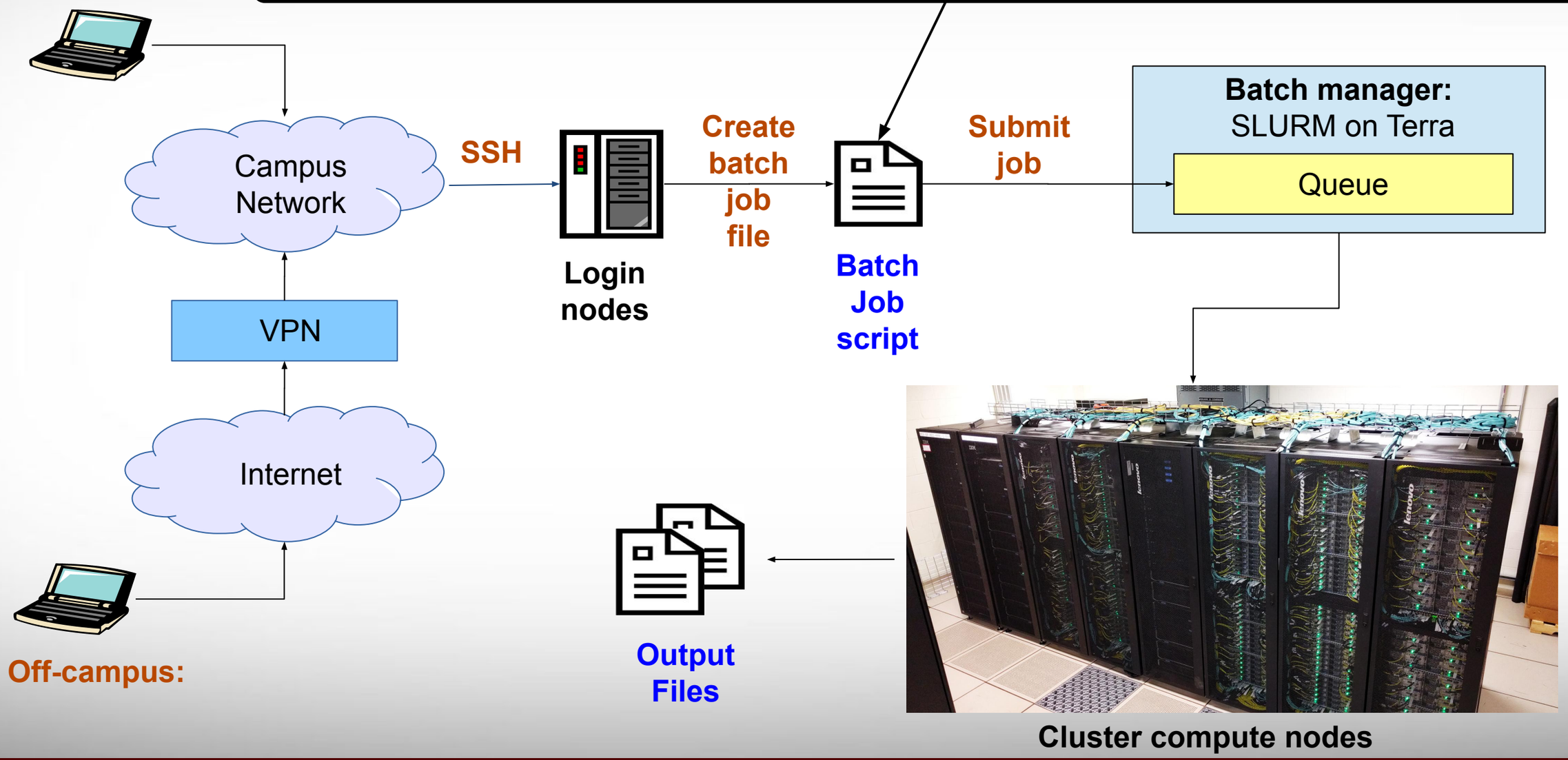
```
module load TopHat/2.1.1-intel-2017A-Python-2.7.12
```

hprc.tamu.edu/wiki/Terra:Computing_Environment#Modules

Batch Job Scripts

Batch Computing on HPRC Clusters

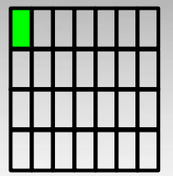
A batch job script is just a text file that contains Unix and software commands and Batch manager job parameters



Terra Job Script (Serial Example)

```
# ENVIRONMENT SETTINGS; NO NEED TO CHANGE
#!/bin/bash
#SBATCH --export=NONE
#SBATCH --get-user-env=L

# NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=my_job
#SBATCH --time=1-00:00:00
#SBATCH --ntasks=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=2G
#SBATCH --output=stdout.%j
#SBATCH --error=stderr.%j

# OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456
#SBATCH --mail-type=ALL
#SBATCH --mail-user=user_email

# load required module(s)first
module load Beast/2.6.0-GCC-8.2.0-2.31.1-Java-1.8.0

# run your program
beast
```

Terra Job Script (Serial Example)



using 1 of 28 CPUs on one Terra compute node

```
# ENVIRONMENT SETTINGS; NO NEED TO CHANGE
#!/bin/bash
#SBATCH --export=NONE
#SBATCH --get-user-env=L

# NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=my_job
#SBATCH --time=1-00:00:00
#SBATCH --ntasks=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=2G
#SBATCH --output=stdout.%j
#SBATCH --error=stderr.%j

# OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456
#SBATCH --mail-type=ALL
#SBATCH --mail-user=user_email

# load required module(s)first
module load Beast/2.6.0-GCC-8.2.0-2.31.1-Java-1.8.0

# run your program
beast
```

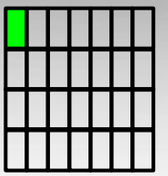
These parameters describe your job to the job scheduler

This is single line comment and not run as part of the script

Load the required module(s) first specifying version

This is a command that is executed by the job

Terra Job Script (Serial Example)

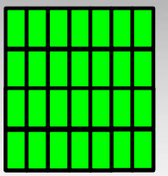


```
# ENVIRONMENT SETTINGS; NO NEED TO CHANGE
#!/bin/bash
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

# NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=my_job            # Set the job name as my_job
#SBATCH --time=1-00:00:00            # Set the wall clock limit to 1 day
#SBATCH --ntasks=1                   # Request 1 task (command)
#SBATCH --ntasks-per-node=1          # Request 1 task (command) per node
#SBATCH --cpus-per-task=1            # Request 1 cpu per task (command)
#SBATCH --mem=2G                     # Request 2GB per node
#SBATCH --output=stdout.%j           # Send stdout to "stdout.[jobID]"
#SBATCH --error=stderr.%j            # Send stderr to "stderr.[jobID]"

# OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456             # Set billing account to 123456
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=user_email       # Send all emails to email_address

# load required module(s)first
module load Beast/2.6.0-GCC-8.2.0-2.31.1-Java-1.8.0

# run your program
beast
```

using 1 of 28 CPUs on one Terra compute node

SUs = 24

Terra Job Script (Parallel Example)



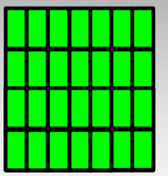
```
# ENVIRONMENT SETTINGS; NO NEED TO CHANGE
#!/bin/bash
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

# NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=my_job            # Set the job name as my_job
#SBATCH --time=1-00:00:00            # Set the wall clock limit to 1 day
#SBATCH --ntasks=1                   # Request 1 task (command)
#SBATCH --ntasks-per-node=1          # Request 1 task (command) per node
#SBATCH --cpus-per-task=28           # Request 28 cpus per task (command)
#SBATCH --mem=54G                    # Request 54GB per node
#SBATCH --output=stdout.%j           # Send stdout to "stdout.[jobID]"
#SBATCH --error=stderr.%j            # Send stderr to "stderr.[jobID]"

# OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456             # Set billing account to 123456
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=user_email       # Send all emails to email_address

# load required module(s)first
module load Beast/2.6.0-GCC-8.2.0-2.31.1-Java-1.8.0

# run your program
beast -threads 28
```

using 28 of 28 CPUs on one Terra compute node

SUs = 672

match software threads with #SBATCH cpus

Terra RScript Job Script



using 28 of 28 CPUs on one Terra compute node

SUs = 672

```
# ENVIRONMENT SETTINGS; NO NEED TO CHANGE
#!/bin/bash
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

# NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=my_job            # Set the job name as my_job
#SBATCH --time=1-00:00:00            # Set the wall clock limit to 1 day
#SBATCH --ntasks=1                   # Request 1 task (command)
#SBATCH --ntasks-per-node=1          # Request 1 task (command) per node
#SBATCH --cpus-per-task=28           # Request 28 cpus per task (command)
#SBATCH --mem=54G                    # Request 54GB per node
#SBATCH --output=stdout.%j           # Send stdout to "stdout.[jobID]"
#SBATCH --error=stderr.%j            # Send stderr to "stderr.[jobID]"

# OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456             # Set billing account to 123456
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=user_email       # Send all emails to email_address

# load R_tamu module which has additional R packages installed
module load R_tamu/3.6.0-iomkl-2018b-recommended-mt

# run the Rscript command with your file of R commands
Rscript mycommands.R
```

Submit Your Terra Job Script & Check Job Status

Submit job

```
sbatch my_job_script.sh
```

```
Submitted batch job 161997
(from job_submit) your job is charged as below
                  Project Account: 122792016265
                  Account Balance: 1687.066160
                  Requested SUs:   672
```

Check status

```
squeue -u $USER
```

```
JOBID   NAME      USER      PARTITION  NODES  CPUS  STATE    TIME   TIME_LEFT  START_TIME          REASON      NODELIST
161997  my_ job   someuser  medium     1      28    PENDING  0:00   24:00      -                   Resources
154038  somejob   someuser  medium     4      112   RUNNING  2:49   17:11      2017-01-30T20:40:4  None        tnxt-[0401-0404]
```

Monitor Note Utilization on Terra: *pestat*

```
pestat [-u username]
```

lists the node utilization across all nodes for a running job.
to see more options use: `pestat -h`

Example 1:

```
pestat -u $USER
```

```
Hostname       Partition     Node  Num_CPU  CPUload  Memsize  Freemem  Joblist
                             State Use/Tot             (MB)     (MB)   JobId User ...
tnxt-0703          xlong     alloc  28   28   16.23*    57344    55506  565849 someuser
tnxt-0704          xlong     alloc  28   28   19.60*    57344    53408  565849 someuser
tnxt-0705          xlong     alloc  28   28   19.56*    57344    53408  565849 someuser
```

Low CPU load utilization highlighted in **Red**
(Freemem should also be noted)

Example 2:

```
pestat -u $USER
```

```
Hostname       Partition     Node  Num_CPU  CPUload  Memsize  Freemem  Joblist
                             State Use/Tot             (MB)     (MB)   JobId User ...
tnxt-0703          xlong     alloc  28   28   27.54    57344    55506  565849 someuser
tnxt-0704          xlong     alloc  28   28   27.50    57344    53408  565849 someuser
tnxt-0705          xlong     alloc  28   28   26.47*   57344    53408  565849 someuser
```

Good CPU load utilization highlighted in **Purple**
Ideal CPU load utilization displayed in White

Debug Terra job failures

- Debug job failures using the stdout and stderr files
- stdout and stderr files were created with the **`#SBATCH`** parameters
  - **`#SBATCH --output stdout.%j`**
  - **`#SBATCH --error stderr.%j`**

- Check for error messages generated by SLURM or the software

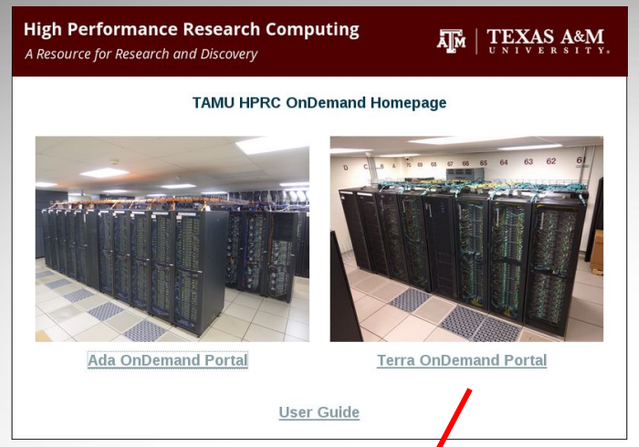
```
slurmstepd: error: Exceeded job memory limit at some point.
```

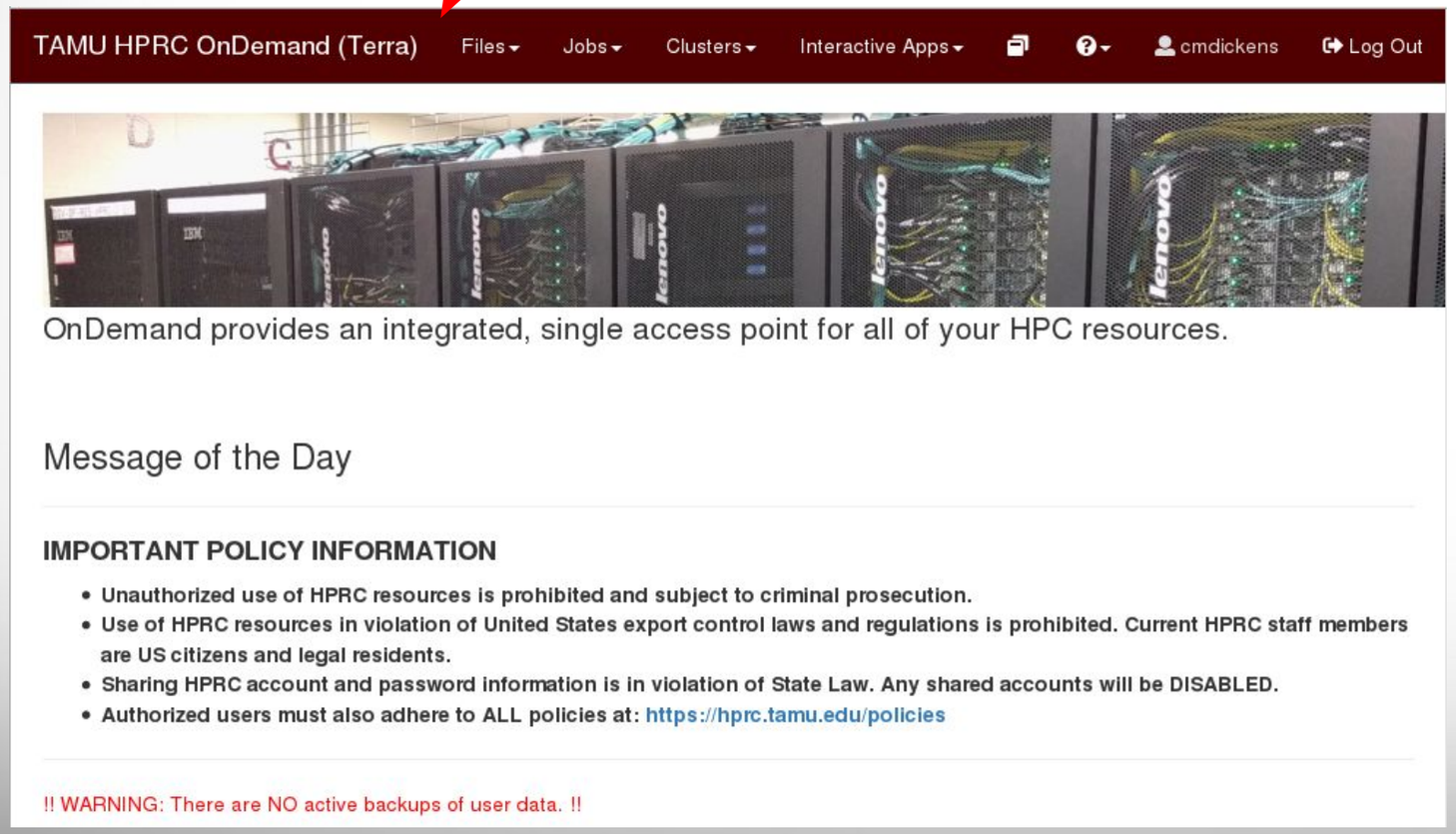
  - Job reached the requested memory
  - Make the necessary adjustments to the **`#SBATCH`** memory parameters in the job script and resubmit your job script

Other Type of Jobs

- Large number of concurrent single core jobs
  - **tamulauncher**
    - hprc.tamu.edu/wiki/SW:tamulauncher
    - Useful for running many single core commands concurrently across multiple nodes within a job
    - Can be used with serial or multi-threaded programs
    - Distributes a set of commands from an input file to run on the cores assigned to a job
    - Can only be used in batch jobs
    - If a tamulauncher job gets killed due to reaching the max walltime, you can resubmit the same job to complete the unfinished commands in the commands file
- Large number of concurrent multi-core jobs
  - Job Array
    - one multi-core command per node
- Multi-core Multi-node software
  - software supporting MPI for running one multi-core command across multiple nodes

portal.hprc.tamu.edu





The HPRC portal allows users to do the following

- Browse files on the HPRC filesystems
- Access the Ada, Terra or Curie Unix command line
  - no SUs charged for using command line
- Compose job scripts on Terra or Ada
- Launch jobs on Terra or Ada
  - SUs charged for launching jobs
- Launch interactive GUI apps (SUs charged)
  - Terra
    - Tracer, FigTree, BEAUti, BEAST (VNC)
    - JBrowse
  - Ada
    - RStudio
    - IGV
    - MATLAB
    - VNC
    - Galaxy
    - Jupyter Notebook
- Monitor and stop running jobs and interactive sessions

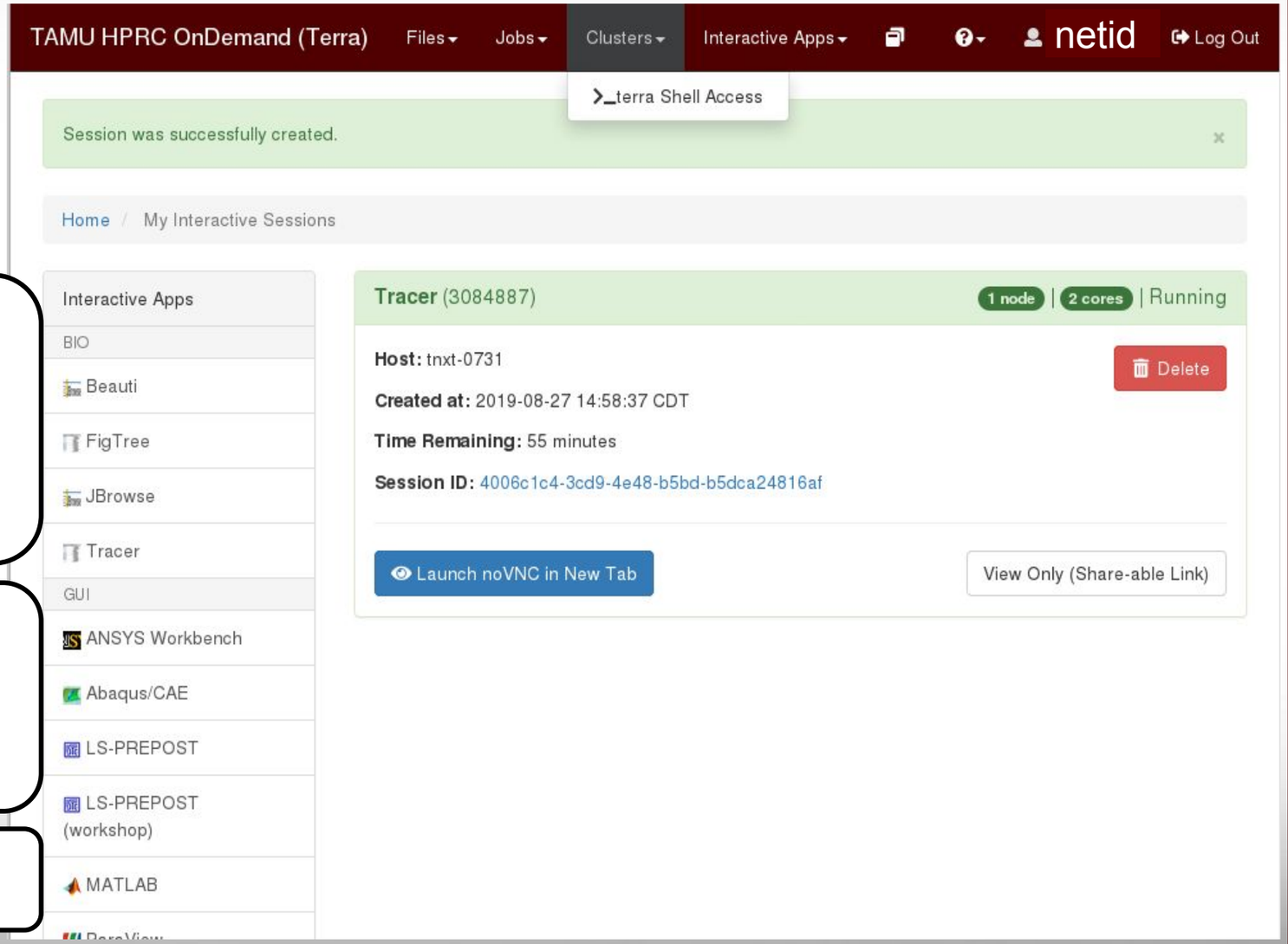
portal.hprc.tamu.edu

- Select the Interactive App you want to launch
- Configure the job parameters
  - number of cores
  - amount and memory
  - total walltime
- Click Launch
- Wait for the Launch noVNC button to appear

- Launching an app starts a job on a compute node
  - -- SUs will be charged for Interactive App jobs
  - -- compute nodes have no internet access
- You can close the browser and the job will remain running as you left it
- Delete the job when you are done working

- Start an ssh session by selecting
  - Clusters >_terra Shell Access
- Shell Access is run on the login node and does not charge SUs.

- Click Files tab to upload files from local to Terra

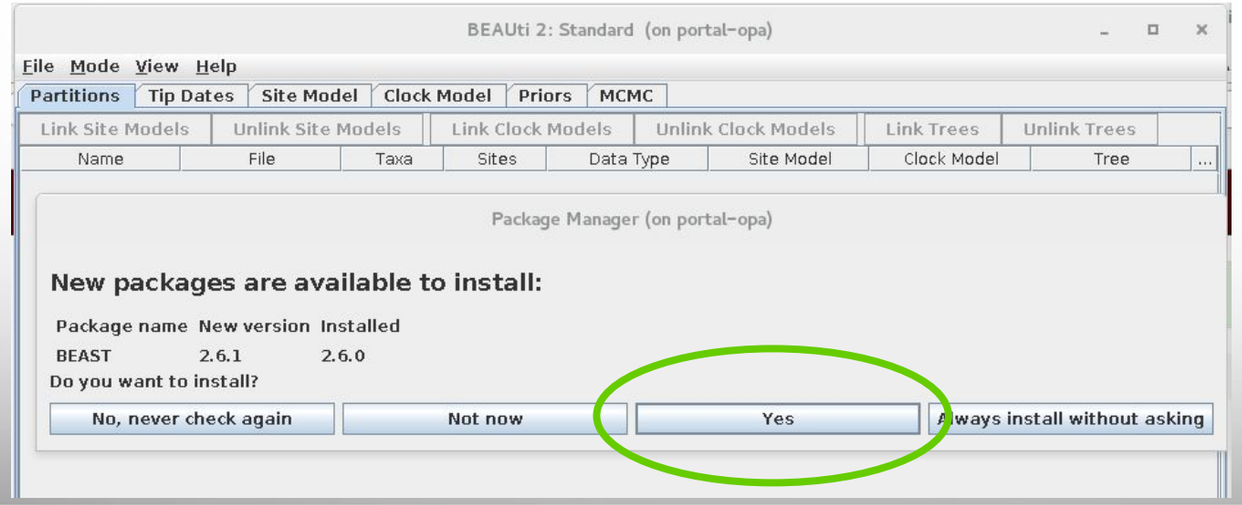


Upgrade BEAST to v2.6.1 with BEAUti v2.6.0

In order to upgrade BEAST to 2.6.1, you must start with version 2.6.0 of BEAUti

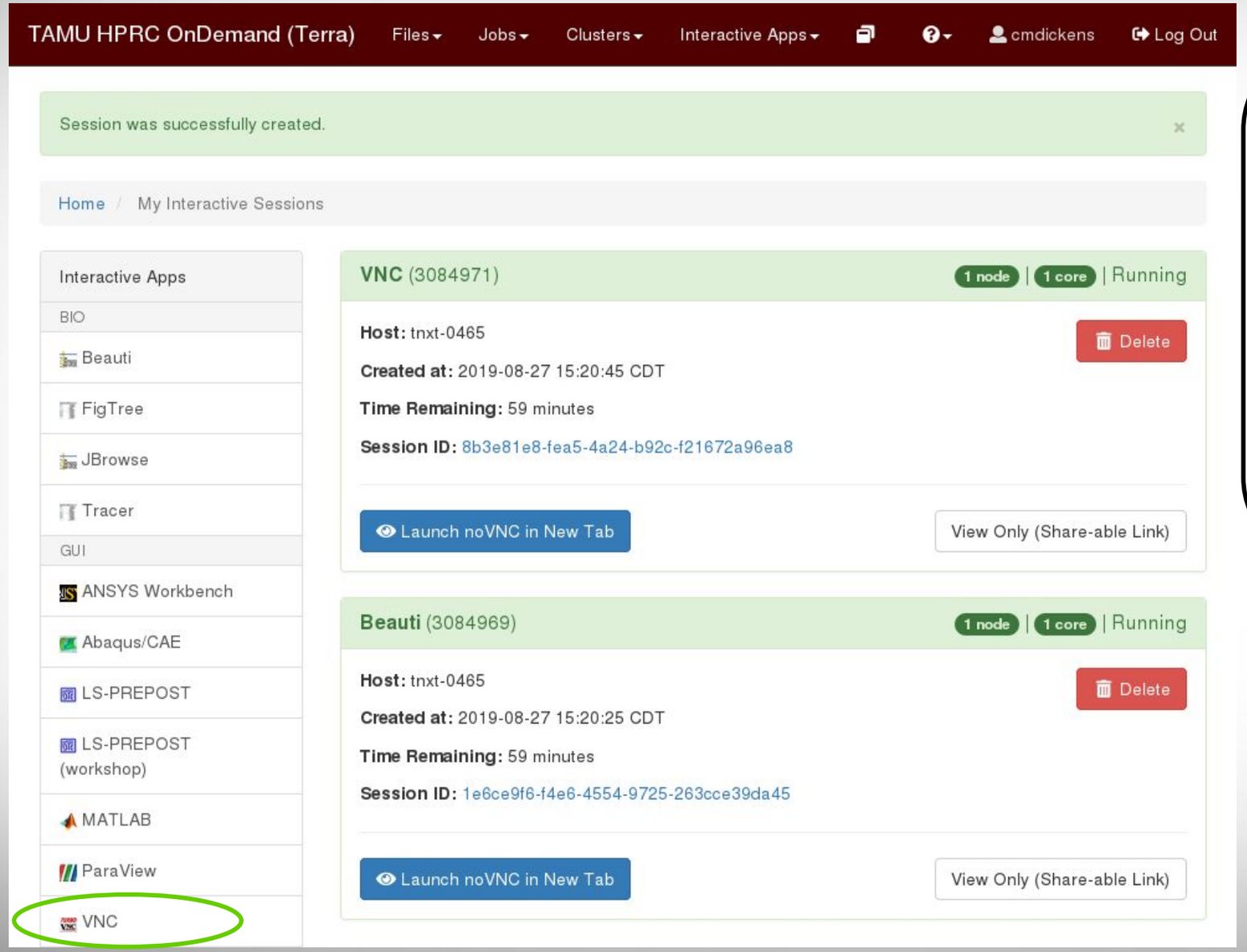
- ssh to Terra using an ssh terminal client (not the portal.hprc.tamu.edu shell or BEAUti Interactive App)
  - use ssh terminal in Linux or MacOS and use MobaXTerm in Windows
  - Mac users try **`ssh -Y`** instead of **`ssh -X`** if the BEAUti app does not show up after running beauti
- on the login node, load the Beast module and start beauti
- select **Yes** then exit beauti after BEAST has completed installation (takes less than a minute to download)
  - **`Installed BEAST in /home/netid/.beast/2.6/BEAST`**
- run BEAST v2.6.1 as a VNC job by starting beast with the Beast/2.6.0 module loaded (next slide)

```
[netid@localhost ~]$ ssh -X netid@terra.tamu.edu
[netid@terra3.tamu.edu ~]$ module load Beast/2.6.0-GCC-8.2.0-2.31.1-Java-1.8.0
[netid@terra3.tamu.edu ~]$ beauti
```



type Beast/2.6.0 then hit the tab key

Run BEAST as a VNC Interactive Job

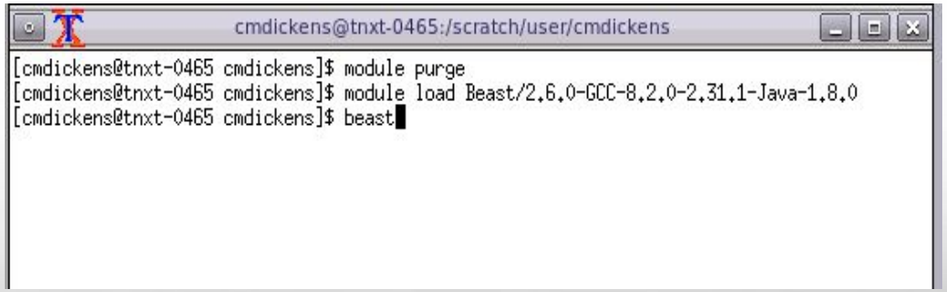


- Select the VNC Interactive App
- Configure the job parameters
  - number of cores
  - amount and memory
  - max walltime
- Click Launch
- Wait for the blue Launch noVNC button to appear then click it
- Type the following commands in the terminal

  module purge

  module load Beast/2.6.0 (then hit tab)

  beast



Need Help?

- First check the FAQ hprc.tamu.edu/wiki/HPRC:CommonProblems
  - Ada User Guide hprc.tamu.edu/wiki/Ada
    - Exercises hprc.tamu.edu/wiki/Ada:Exercises
  - Terra User Guide hprc.tamu.edu/wiki/Terra
    - Exercises hprc.tamu.edu/wiki/Terra:Exercises
- Email your questions to **help@hprc.tamu.edu**. (Managed by a ticketing system)
- Help us, help you -- we need more info
  - Which Cluster
  - UserID/NetID (*UIN is not needed!*)
  - Job id(s) if any
  - Location of your jobfile, input/output files
  - Application used if any
  - Module(s) loaded if any
  - Error messages
  - Steps you have taken, so we can reproduce the problem
- Or visit us @ 114A Henderson Hall
  - Making an appointment is recommended.

Thank you.

Any question?

hprc.tamu.edu/training