

# HIGH PERFORMANCE RESEARCH COMPUTING

## - HPRC Primer - Introduction to Grace: An HPRC Resource

*September 5, 2025*



High Performance  
Research Computing  
DIVISION OF RESEARCH



# Table of Contents

## **By the end of this primer you will know**

- What Grace is as well as its resources
- How to log in and manage files/storage
- How to load software modules
- How to run jobs on the clusters using the SLURM scheduler

## **Prerequisites to Primer:**

- Have an HPRC account
- Basic knowledge of Linux

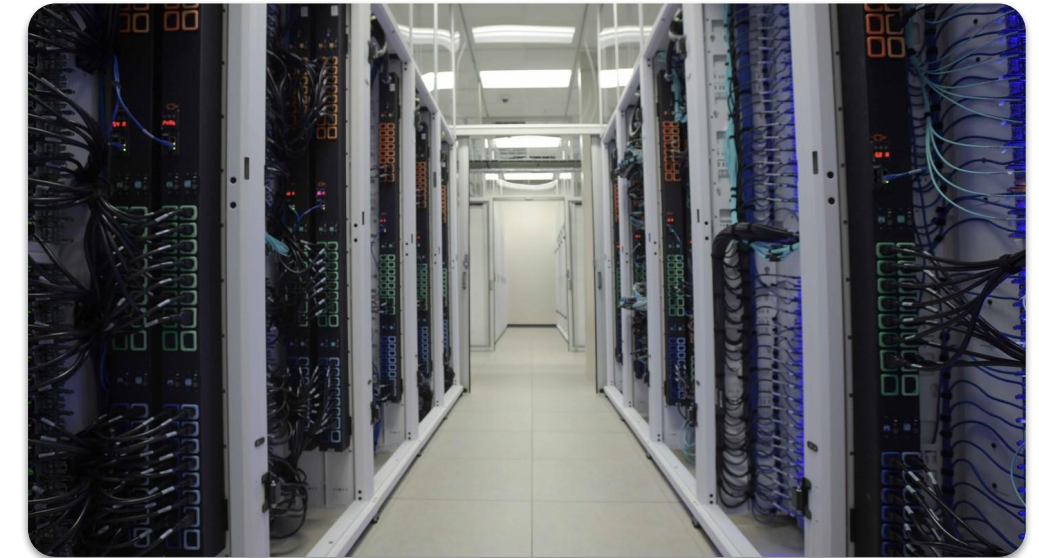
# Grace Hardware

Grace is a 940-node Intel cluster from Dell with an InfiniBand HDR-100 interconnect, A100 GPUs, RTX 6000 GPUs and T4 GPUs. The 940 nodes are based on the Intel Cascade Lake processor.

48 cores/node

3TB Large Memory-80 cores/node  
Login Nodes: 10 GbE TAMU network connection

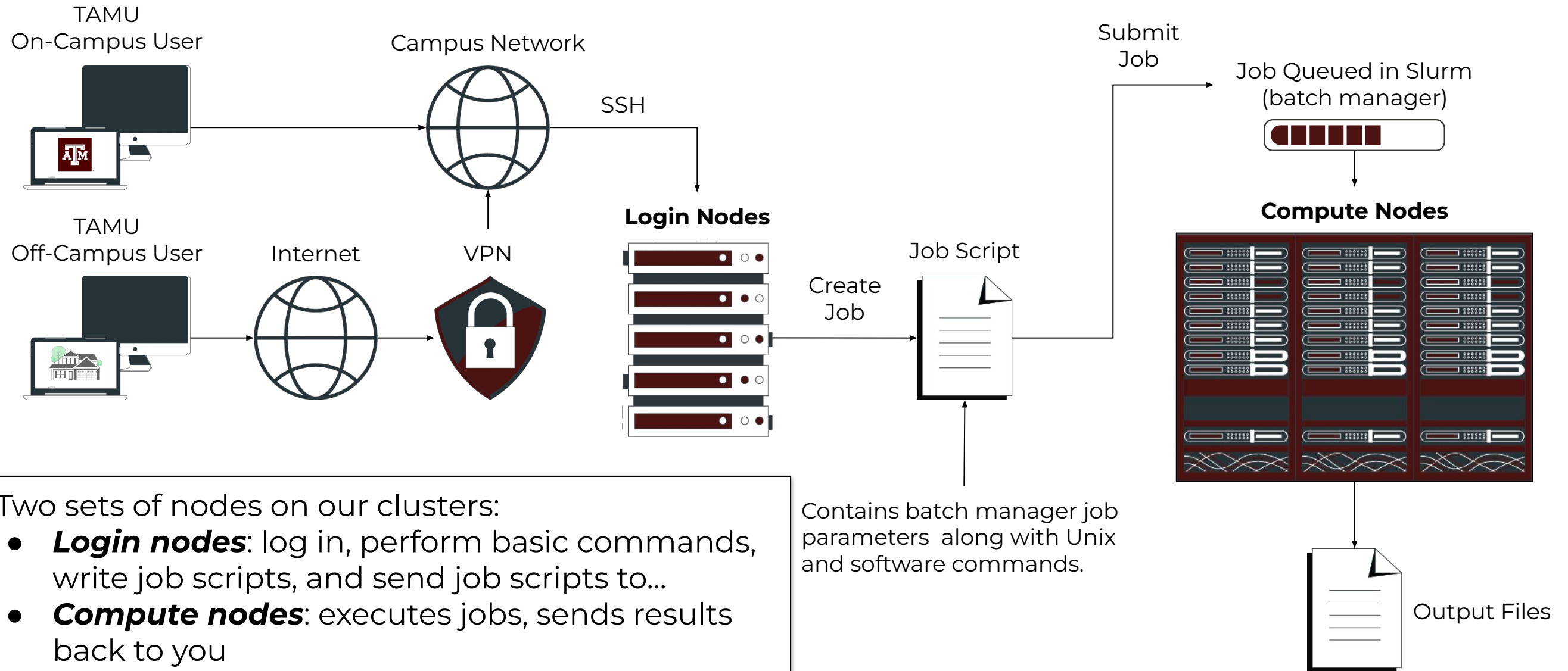
| Resource                               | Count |
|----------------------------------------|-------|
| Login Nodes                            | 5     |
| 384GB memory general compute nodes     | 800   |
| GPU - A100 nodes with 384GB memory     | 100   |
| GPU - RTX 6000 nodes with 384GB memory | 9     |
| GPU - T4 nodes with 384GB memory       | 8     |
| 3TB Large Memory                       | 8     |



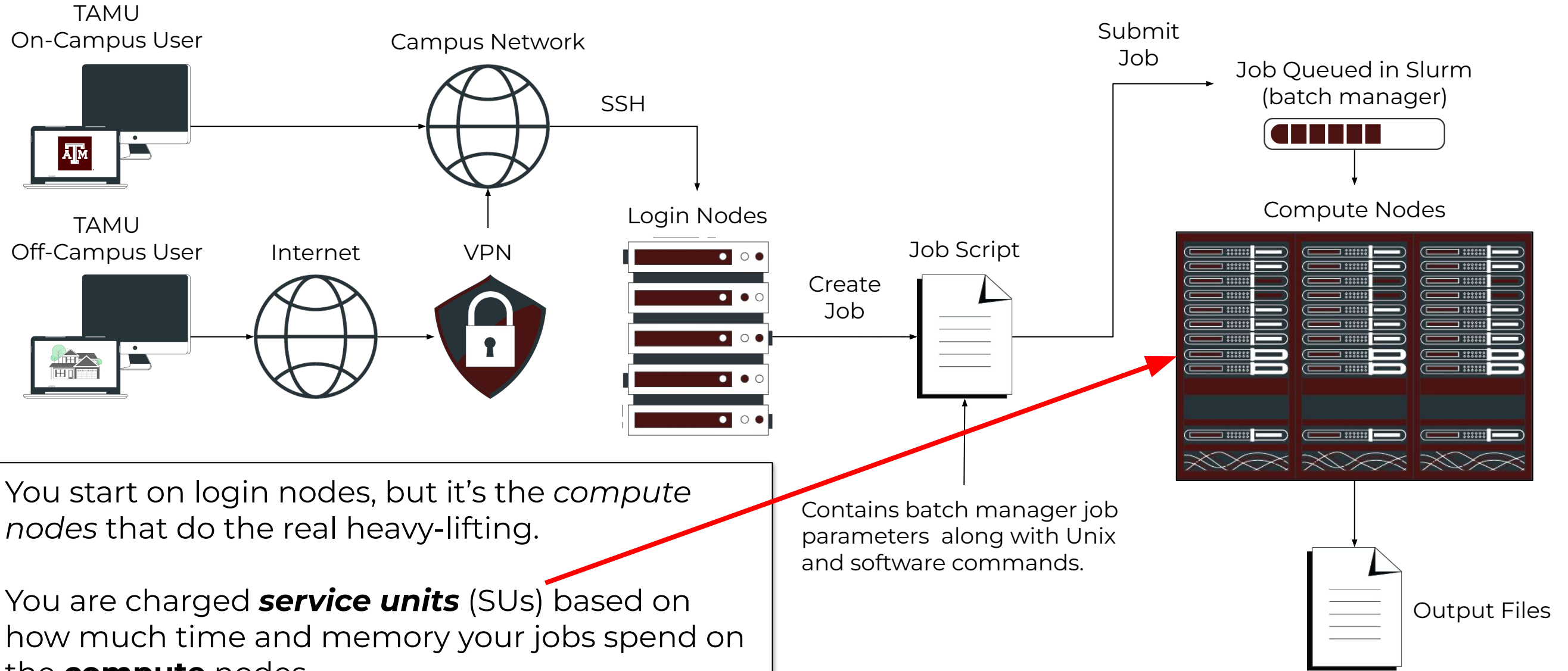
For more information:

<https://hprc.tamu.edu/kb/User-Guides/Grace/>

# Computing on HPRC Clusters



# Batch Jobs on HPRC Clusters



# File Quotas and Resource Allocations

- Two things to keep track of when computing

Quota

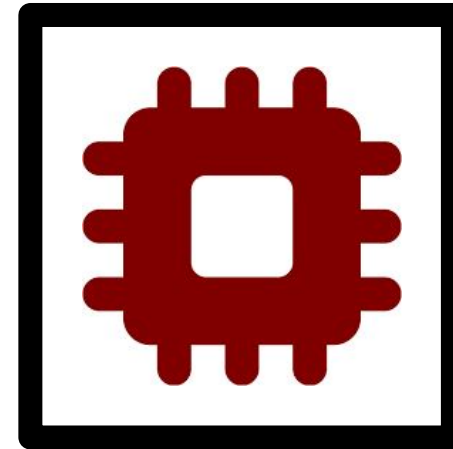


File Storage

*File Count*

*File Space*

Allocation

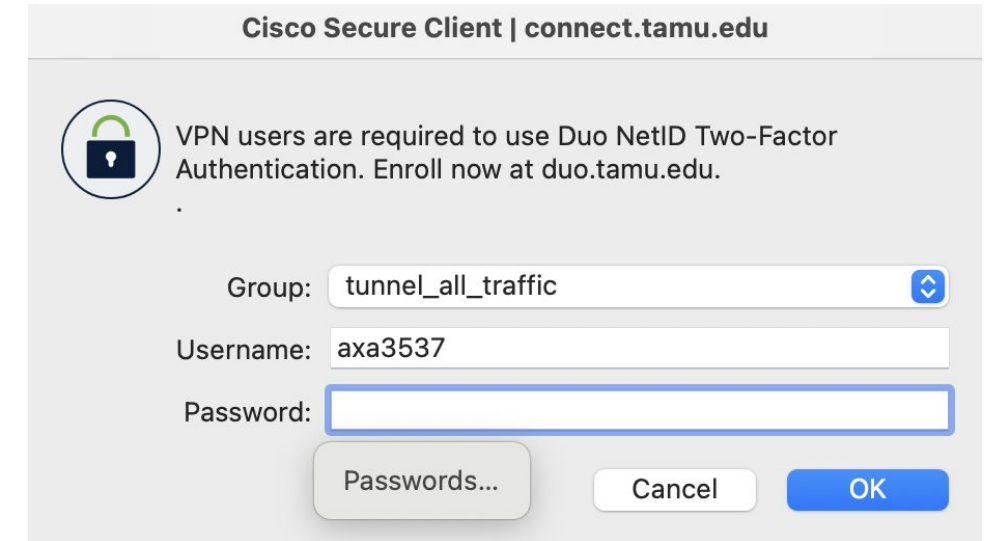


Computing Resources

*Service Units (SUs)*

# Accessing Grace: Setup

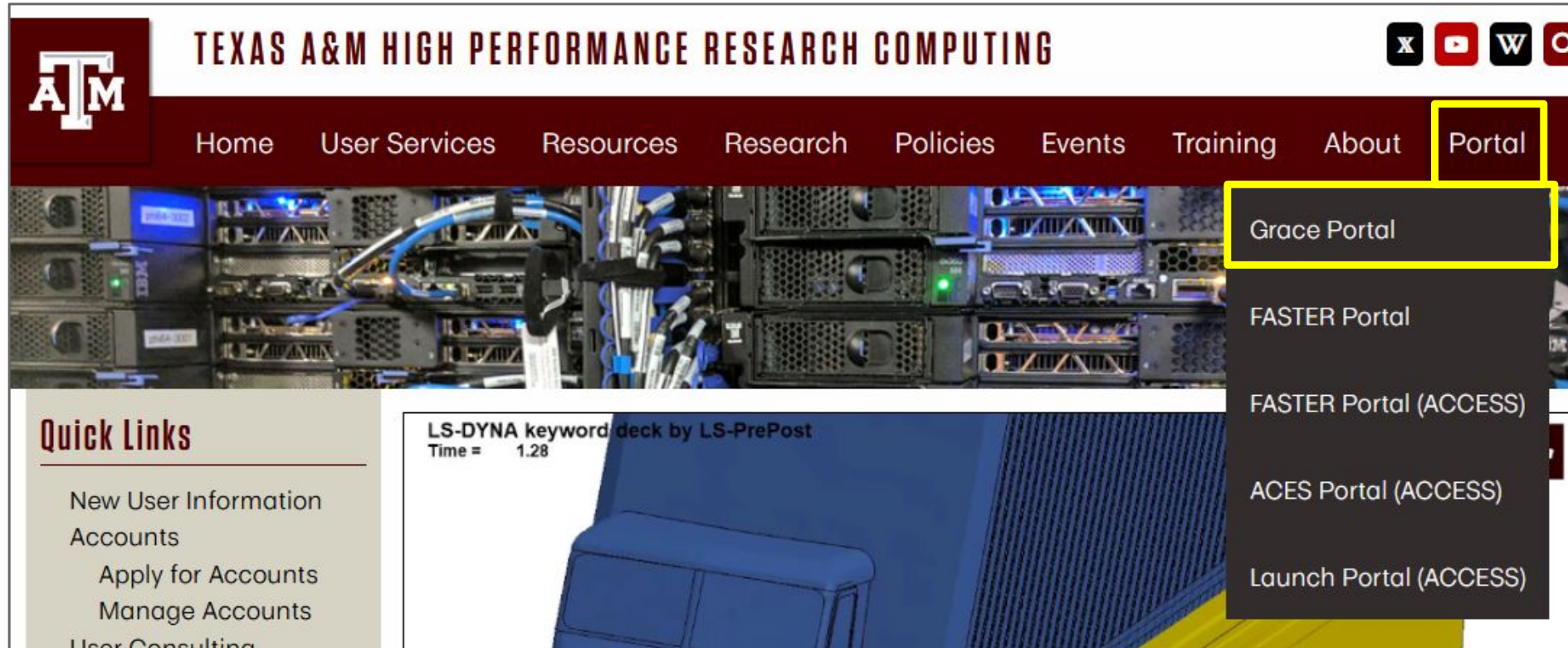
- If off-campus: Set up and start VPN (Virtual Private Network):  
[u.tamu.edu/VPnetwork](http://u.tamu.edu/VPnetwork)
- *Two-Factor Authentication* required



The image shows a screenshot of the Cisco Secure Client interface for connecting to connect.tamu.edu. It features a message about Duo NetID Two-Factor Authentication, a dropdown menu for the Group (set to tunnel\_all\_traffic), and input fields for Username (axa3537) and Password. There are buttons for Passwords..., Cancel, and OK.

- Today we'll access Grace via the online Portal, but you can also use ssh. `ssh <NetID>@grace.hprc.tamu.edu`
- See <https://hprc.tamu.edu/kb/User-Guides/Grace/Access/> for more details.

# Accessing Grace via the Portal



Access the HPRC portals through most web browsers:

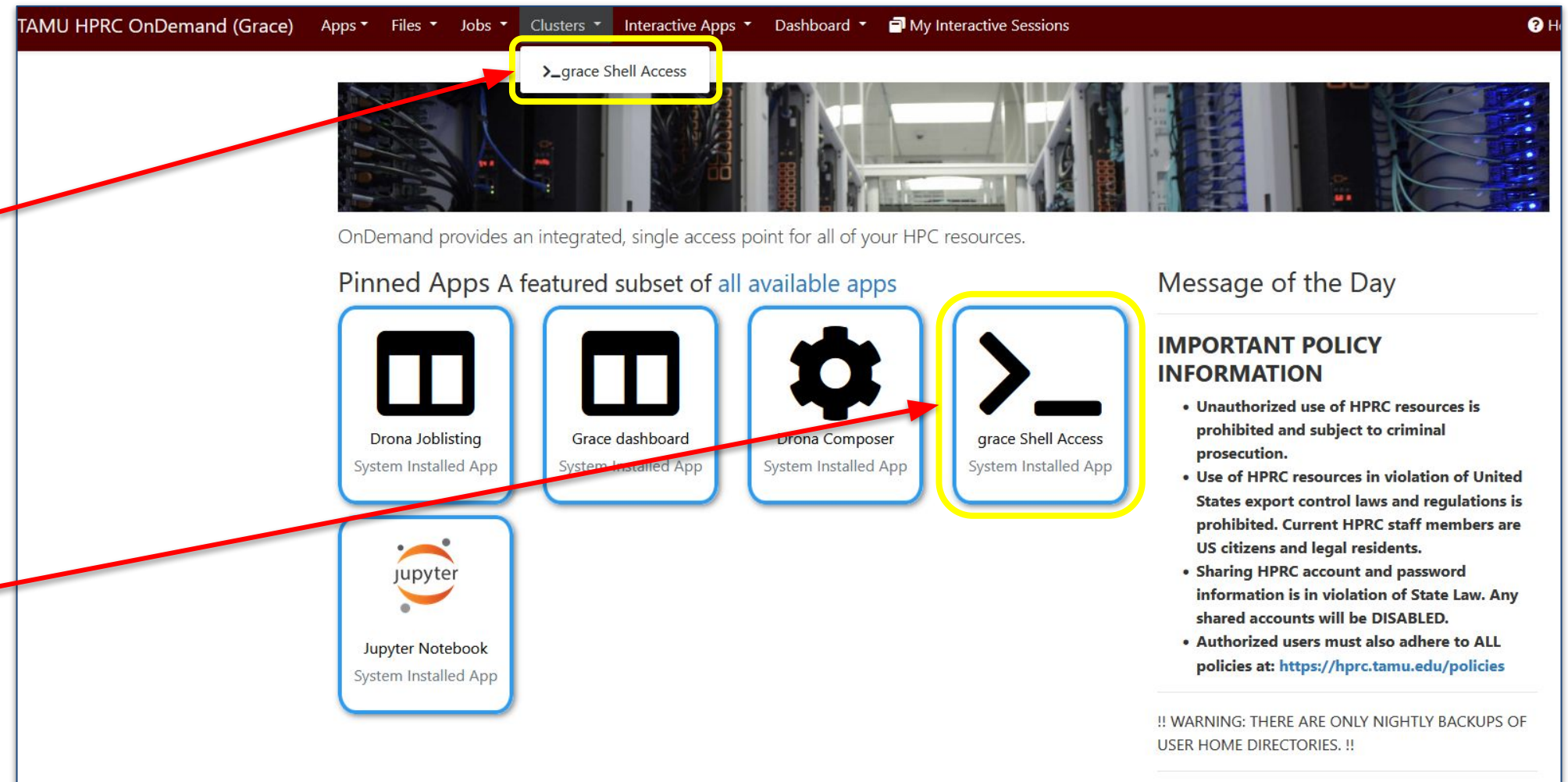
1. Go to [portal.hprc.tamu.edu](https://portal.hprc.tamu.edu) or use the Portal dropdown menu on the HPRC homepage: <https://hprc.tamu.edu/>
2. Choose **Grace Portal**

<https://hprc.tamu.edu/kb/User-Guides/Grace/Access/>

# Accessing Grace via the Portal

Two ways to enter the Portal:

- (1) Select at the top:  
“Clusters” →  
“Grace Shell\*  
Access”
- (2) Select in Main  
Menu:  
“Grace Shell  
Access”



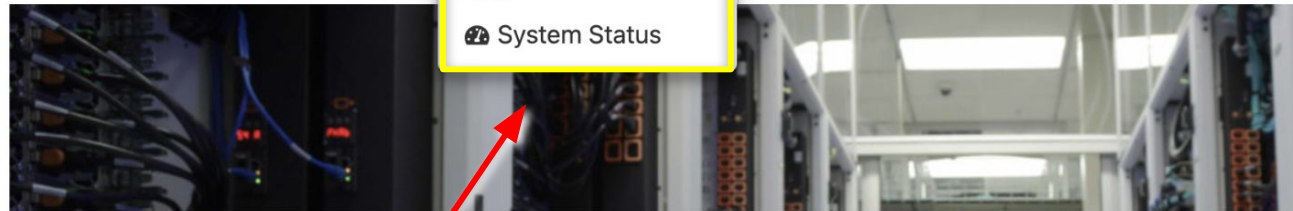
\**shell* is also called *terminal* or *command line*

<https://hprc.tamu.edu/kb/User-Guides/Grace/Access/>

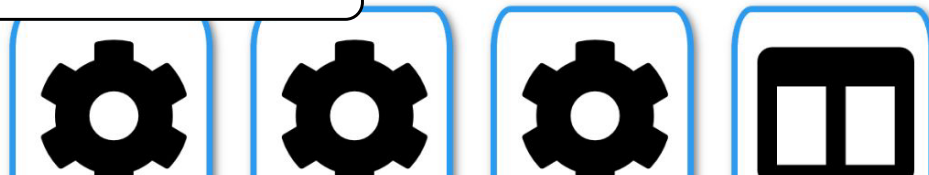
# Shell Access via the Portal



>\_ grace Shell Access  
🔌 System Status



Get a shell terminal  
right in your browser



Featured subset of **all available** Message of the

## IMPORTANT POLICY INFORMATION

- Unauthorized use of HPC resources is prohibited

```
Host: login.gracelocal

*****
This computer system and the data herein are available only for authorized
purposes by authorized users. Use for any other purpose is prohibited and may
result in disciplinary actions or criminal prosecution against the user. Usage
may be subject to security testing and monitoring. There is no expectation of
privacy on this system except as otherwise provided by applicable privacy laws.
Refer to University SAP 29.01.03.M0.02 Acceptable Use for more information.
*****

Last login: Thu Aug 21 13:32:56 2025 from 10.73.4.63
=====
Texas A&M University High Performance Research Computing
=====
Consulting:      help@hprc.tamu.edu (preferred) or (979) 845-0219
Website:         https://hprc.tamu.edu
Knowledgebase:   https://hprc.tamu.edu/kb/
YouTube Channel: https://www.youtube.com/texasamhprc
=====

*****
=== IMPORTANT POLICY INFORMATION ===
* - Unauthorized use of HPRC resources is prohibited and subject to
*   criminal prosecution.
* - Use of HPRC resources in violation of United States export control
*   laws and regulations is prohibited.
* - Sharing HPRC account and password information is in violation of
*   Texas State Law. Any shared accounts will be DISABLED.
* - Authorized users must also adhere to ALL policies at:
*   https://hprc.tamu.edu/policies/
*****

!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!

Please restrict usage to 8 CORES across ALL login nodes.
Users found in violation of this policy will be SUSPENDED.

To see these messages again, run the motd command.

Your current disk quotas are:
Disk                               Disk Usage      Limit      File Usage      Limit
/home/axa3537                      681M            10.0G      6394            10000
/scratch/user/axa3537              2.0G            1.0T       52122          250000
/scratch/group/hprc               4.8T            10.0T     494939         1000000
* Quota increase for /scratch/group/hprc will expire on Dec 31, 2026
Type 'showquota' to view these quotas again.
[axa3537@grace1 ~]$
```

# Hands-On Activity - 2 Minutes

Try to access a *shell*\* on Grace now, either through [portal.hprc.tamu.edu](http://portal.hprc.tamu.edu) or [hprc.tamu.edu](http://hprc.tamu.edu)

\*(also called *terminal* or *command line*)

What message do you see when you login?

Remember Grace has 5 login nodes. Which one does your command prompt say you were assigned?

# 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. Backed up nightly.                                                               |
| /scratch/user/\$USER | \$SCRATCH*           | 1 TB        | 250,000    | Temporary storage of large files for on-going computations. Not intended to be a long-term storage area. Not backed up. |

**\*Do NOT share your home or scratch directories.**

Request a group directory for sharing files.

**\$SCRATCH** is shared between the FASTER and Grace clusters.

View file usage and quota limits in the shell using the command:

**showquota**

```
[axa3537@grace5 ~]$ showquota
Your current disk quotas are:
Disk                Disk Usage    Limit    File Usage    Limit
/home/axa3537        681M         10.0G    6394          10000
/scratch/user/axa3537 2.0G         1.0T     52122         250000
/scratch/group/hprc  4.8T         10.0T    494939        1000000
```

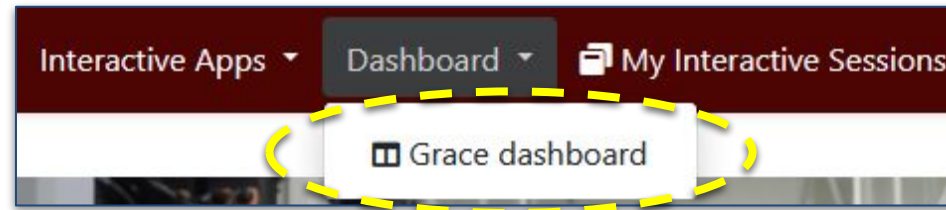
[https://hprc.tamu.edu/kb/User-Guides/Grace/Filesystems\\_and\\_Files/](https://hprc.tamu.edu/kb/User-Guides/Grace/Filesystems_and_Files/)

# Portal: Grace Dashboard

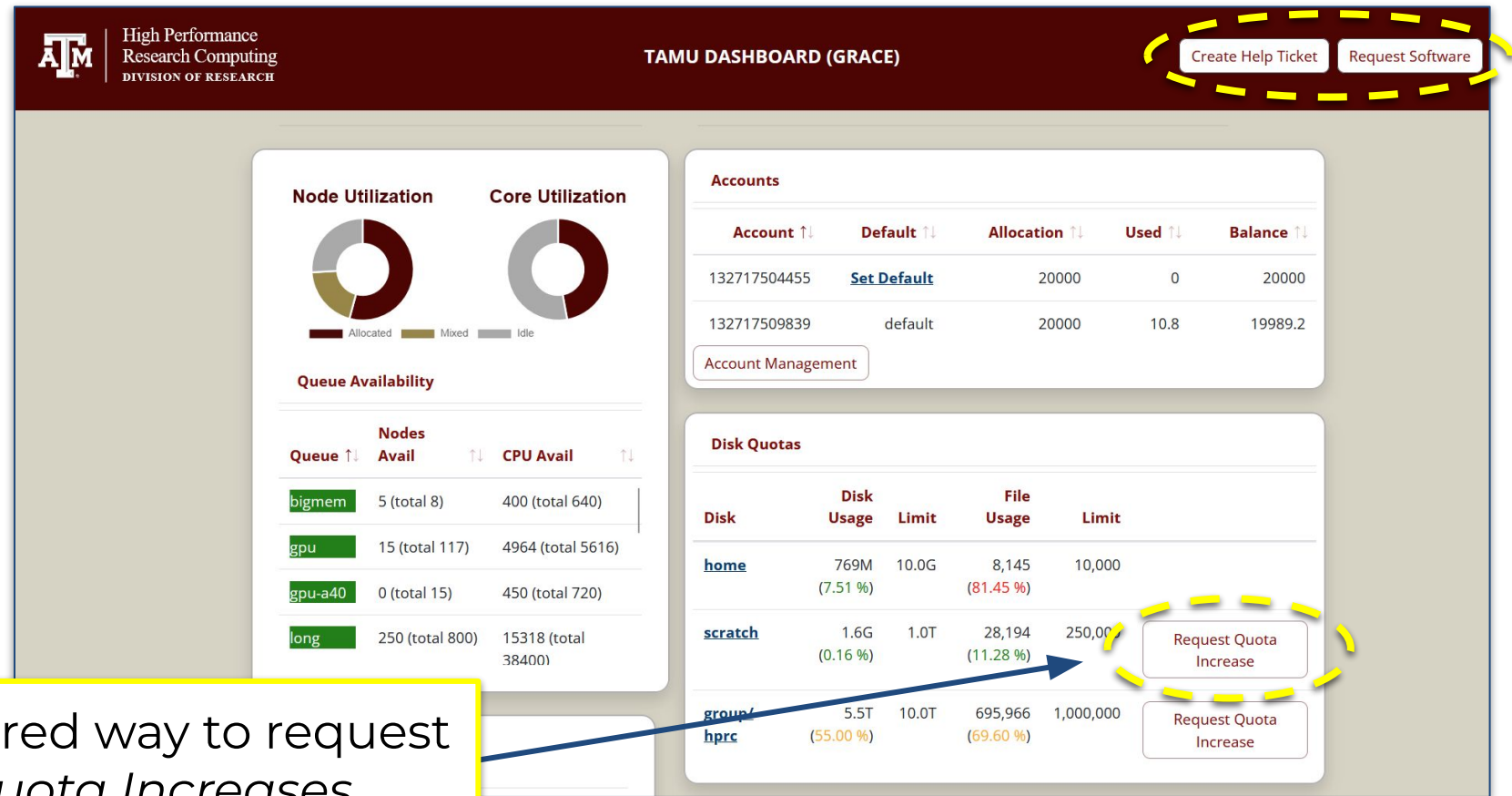
- Easily view Cluster Utilization, Storage Quotas, & Allocation Balances

- Ask for help and request software

- View current groups and job listings



Request help or software



Preferred way to request *Quota Increases*

# Hands-On Activity - 2 Minutes

1. Please try to access dashboard now through the portal.
2. Check your quotas both on the command line and on the dashboard.

`showquota`

# Checking Your Service Unit (SUs) Balance

- 1 SU = 1 core/hr (GPUs are more expensive per-hour!)  
Visit <https://hprc.tamu.edu/kb/User-Guides/AMS/#sus-for-jobs-requesting-gpus> for more information
- SUs are charged to default account when none is specified.

The screenshot displays the TAMU Dashboard (GRACE) interface. The top navigation bar includes the TAMU logo, the text 'High Performance Research Computing DIVISION OF RESEARCH', and the title 'TAMU DASHBOARD (GRACE)'. A yellow callout box with the text 'Change your Default Account' points to the 'Set Default' link in the 'Accounts' table. The 'Accounts' table lists two accounts: '132717504455' and '132717509839'. The '132717509839' account is marked as the 'default' and has a balance of 19989.2. The 'Queue Availability' section shows a table with columns for Queue, Nodes Avail, and CPU Avail. The 'Disk Quotas' section shows a table with columns for Disk, Disk Usage, Limit, File Usage, and Limit. The 'home' disk has a usage of 769M (7.51%) and a limit of 10.0G.

**Node Utilization**

**Core Utilization**

**Queue Availability**

| Queue   | Nodes Avail    | CPU Avail         |
|---------|----------------|-------------------|
| bigmem  | 5 (total 8)    | 400 (total 640)   |
| gpu     | 15 (total 117) | 4964 (total 5616) |
| gpu-a40 | 0 (total 15)   | 450 (total 720)   |

**Accounts**

| Account      | Default                     | Allocation | Used | Balance |
|--------------|-----------------------------|------------|------|---------|
| 132717504455 | <a href="#">Set Default</a> | 20000      | 0    | 20000   |
| 132717509839 | default                     | 20000      | 10.8 | 19989.2 |

**Disk Quotas**

| Disk | Disk Usage    | Limit | File Usage      | Limit  |
|------|---------------|-------|-----------------|--------|
| home | 769M (7.51 %) | 10.0G | 8,145 (81.45 %) | 10,000 |

# Checking Your SUs in the Shell

- List the SU Balance of your Account(s) with: **myproject**

```
=====
List of YourNetID's Project Accounts
-----
| Account | FY | Default | Allocation | Used & Pending SUs | Balance | PI |
-----
| 1228000223136 | 2025 | N | 10000.00 | 0.00 | 10000.00 | Doe, John |
-----
| 1428000243716 | 2025 | Y | 5000.00 | -71.06 | 4928.94 | Doe, Jane |
-----
| 1258000247058 | 2025 | N | 5000.00 | -0.91 | 4999.09 | Doe, Jane |
-----
```

- Run **myproject -d <Account#>** to change default project account  
(replace <Account#> with your number!)

<https://hprc.tamu.edu/kb/User-Guides/AMS/#UI>

<https://hprc.tamu.edu/kb/User-Guides/AMS/#service-unit>

# Checking Your SUs in the Shell

- Run **myproject -j** to see SU usage for all jobs

```
[axa3537@grace3 ~]$ myproject -j 132663473580
```

| ----- List of Jobs ----- |          |               |                     |                     |                     |          |               |         |                    |  |  |
|--------------------------|----------|---------------|---------------------|---------------------|---------------------|----------|---------------|---------|--------------------|--|--|
| ProjectAccount           | JobID    | JobArrayIndex | SubmitTime          | StartTime           | EndTime             | Walltime | TotalSlots    | UsedSUs |                    |  |  |
| 132663473580             | 15564194 | 0             | 2025-06-17 10:39:31 | 2025-06-17 10:39:45 | 2025-06-17 10:43:03 | 198      | 1             | 0.06    |                    |  |  |
| 132663473580             | 16029806 | 0             | 2025-08-05 10:55:15 | 2025-08-05 10:55:31 | 2025-08-05 10:56:53 | 82       | 1             | 0.02    |                    |  |  |
| 132663473580             | 16030076 | 0             | 2025-08-05 11:05:38 | 2025-08-05 11:05:54 | 2025-08-05 11:14:14 | 500      | 1             | 0.14    |                    |  |  |
| 132663473580             | 16030256 | 0             | 2025-08-05 11:14:56 | 2025-08-05 11:15:04 | 2025-08-05 11:23:14 | 490      | 1             | 9.94    |                    |  |  |
| 132663473580             | 16030728 | 0             | 2025-08-05 11:33:57 | 2025-08-05 11:34:08 | 2025-08-05 11:45:01 | 653      | 1             | 13.24   |                    |  |  |
| 132663473580             | 16047456 | 0             | 2025-08-06 15:58:38 | 2025-08-06 15:58:50 | 2025-08-06 16:19:44 | 1254     | 1             | 0.35    |                    |  |  |
| 132663473580             | 16048154 | 0             | 2025-08-06 16:49:31 | 2025-08-06 16:49:45 | 2025-08-06 17:49:48 | 3603     | 1             | 73.06   |                    |  |  |
| 132663473580             | 16164088 | 0             | 2025-08-15 14:51:00 | 2025-08-15 14:51:08 | 2025-08-15 14:51:54 | 46       | 1             | 0.01    |                    |  |  |
|                          |          |               |                     |                     |                     |          | Total Jobs: 8 |         | Total Usage: 96.82 |  |  |

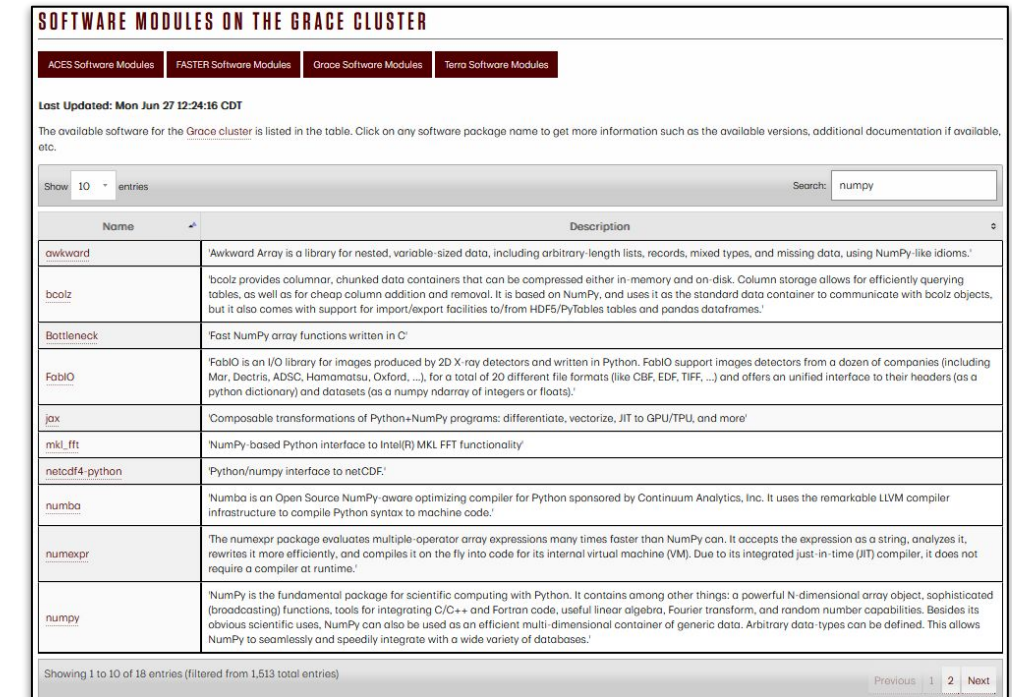
- Run **myproject -h** to see more options

# Hands-On Activity - 2 Minutes

1. Use **myproject** to check the SU balance of your accounts.
2. Use the dashboard to check the same information.

# Software

- Search for software modules on <https://hprc.tamu.edu/software/grace/>
- See the Software Knowledge Base page <https://hprc.tamu.edu/kb/Software/> for instructions and examples
- License-restricted software
  - Contact [help@hprc.tamu.edu](mailto:help@hprc.tamu.edu)
- Contact HPRC (can use the dashboard) for **software installation help/request**
  - User can install software in their home/scratch directory
  - **Do NOT** run the **sudo** command when installing software



SOFTWARE MODULES ON THE GRACE CLUSTER

ACES Software Modules FASTER Software Modules Grace Software Modules Terra Software Modules

Last Updated: Mon Jun 27 12:24:16 CDT

The available software for the Grace cluster is listed in the table. Click on any software package name to get more information such as the available versions, additional documentation if available, etc.

Show 10 entries Search: numpy

| Name                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">awkward</a>        | 'Awkward Array is a library for nested, variable-sized data, including arbitrary-length lists, records, mixed types, and missing data, using NumPy-like idioms.'                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">booz</a>           | 'booz provides columnar, chunked data containers that can be compressed either in-memory and on-disk. Column storage allows for efficiently querying tables, as well as for cheap column addition and removal. It is based on NumPy, and uses it as the standard data container to communicate with booz objects, but it also comes with support for import/export facilities to/from HDF5/PyTables tables and pandas dataframes.'                                                                                                                                     |
| <a href="#">Bottleneck</a>     | 'Fast NumPy array functions written in C'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">FabIO</a>          | 'FabIO is an I/O library for images produced by 2D X-ray detectors and written in Python. FabIO support images detectors from a dozen of companies (including Mar, Dectris, ADSC, Hamamatsu, Oxford, ...), for a total of 20 different file formats (like CBF, EDF, TIFF, ...) and offers an unified interface to their headers (as a python dictionary) and datasets (as a numpy ndarray of integers or floats).'                                                                                                                                                     |
| <a href="#">jax</a>            | 'Composable transformations of Python+NumPy programs: differentiate, vectorize, JIT to GPU/TPU, and more'                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">mkl_fft</a>        | 'NumPy-based Python interface to Intel(R) MKL FFT functionality'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">netcdf4-python</a> | 'Python/numPy interface to netCDF'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <a href="#">numba</a>          | 'Numba is an Open Source NumPy-aware optimizing compiler for Python sponsored by Continuum Analytics, Inc. It uses the remarkable LLVM compiler infrastructure to compile Python syntax to machine code.'                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">numexpr</a>        | 'The numexpr package evaluates multiple-operator array expressions many times faster than NumPy can. It accepts the expression as a string, analyzes it, rewrites it more efficiently, and compiles it on the fly into code for its internal virtual machine (VM). Due to its integrated just-in-time (JIT) compiler, it does not require a compiler at runtime.'                                                                                                                                                                                                      |
| <a href="#">numpy</a>          | 'NumPy is the fundamental package for scientific computing with Python. It contains among other things: a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.' |

Showing 1 to 10 of 18 entries (filtered from 1,513 total entries) Previous 1 2 Next

# Software: Application Modules

- Installed applications are made available with the module system
- Grace uses a *software hierarchy* inside the module system
- In this hierarchy, the user loads a compiler which then makes available Software built with the currently-loaded compiler

```
module avail
```

← shows which software is available

```
module load GCC/13.2.0 OpenMPI/4.1.6
```

← load GCC compiler version 13.2.0 and OpenMPI version 4.1.6

```
module avail
```

← show which software is available to use with the loaded modules

```
module load PyTorch/2.7.0
```

← load PyTorch version 2.7.0

```
module list
```

← see what software you've loaded already

Module System Youtube Video → [www.youtube.com/watch?v=drxpbrOCPEw](https://www.youtube.com/watch?v=drxpbrOCPEw)

# Software: Modules and Toolchains

- Toolchains are what we call groups of compilers & libraries
- There's a variety of toolchains available on the clusters: **toolchains**
  - intel/2024a
  - foss/2024a *(more than just these versions)*
  - GCCcore/13.3.0
- Other module commands:

```
module spider  
module purge
```

← search for modules and their dependencies  
← removes all loaded modules

good practice!

helpful: press q to quit module spider

Module System Youtube Video → [www.youtube.com/watch?v=drxpbrOCPFw](https://www.youtube.com/watch?v=drxpbrOCPFw)

# Hands-On Activity - 5 Minutes

Remember:

```
module load
```

```
module purge
```

1. Please search for and load the following module:

```
OpenMPI/5.0.3
```

(Hint) Type this to show which compiler needs to be loaded:

```
module spider OpenMPI/5.0.3
```

(Hint) And check that it's been loaded with:

```
module list
```

2. Next remove (unload) all your current modules.

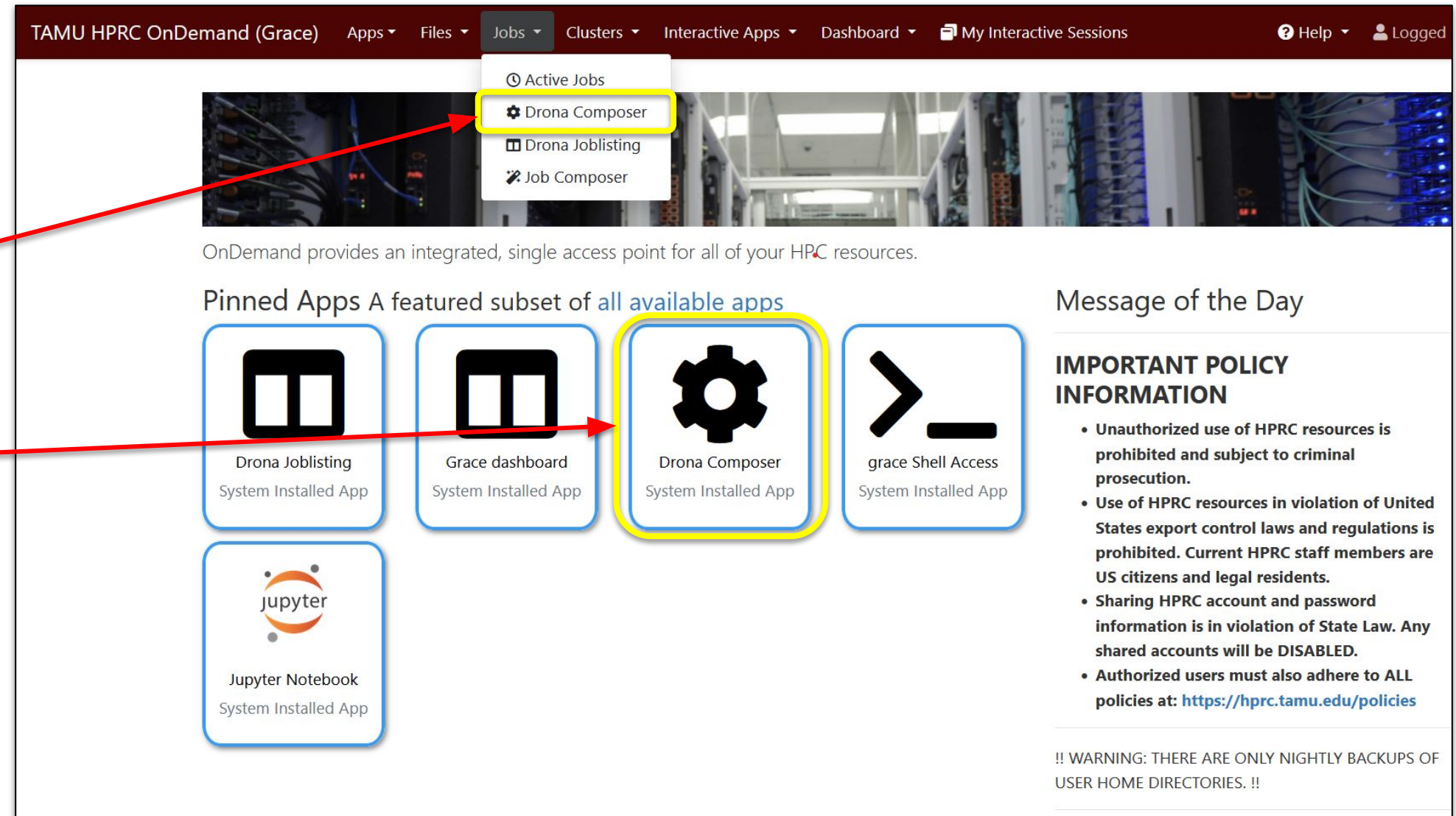
Module System Youtube Video → [www.youtube.com/watch?v=drxpbrOCPFw](http://www.youtube.com/watch?v=drxpbrOCPFw)

# The Drona Composer

- A simple app to assist you with composing jobs

Two ways to get to Drona Composer:

- (1) Select Drona Composer under “Jobs” tab
- (2) Select Drona Composer in Main Menu



# The Drona Composer

Create your own jobs through the app:

- (1) Pick a job name
- (2) Choose location and environments
- (3) Add modules
- (4) Choose Resources
- (5) Choose Time

The screenshot shows the 'Job Composer' interface of the Drona Composer (GRACE) application. The interface is divided into several sections with red arrows pointing to specific fields:

- Job Name:** A text input field.
- Location:** A text input field with a 'Change' button next to it. The current value is '/scratch/user/jwinchell/drona\_composer/runs'.
- Environments:** A dropdown menu showing 'Generic' with a '+' button to the right.
- Upload files:** A dropdown menu showing 'Select an option' with an 'Add' button.
- Add modules:** A text input field with a 'Default (foss/2023b)' dropdown and an 'Add' button.
- Number of tasks:** A text input field with a value of '1' and a 'Number of tasks' label.
- Advanced task options:** A checkbox labeled 'Advanced task options'.
- Use Accelerator:** A dropdown menu showing '-- Choose an option --'.
- TOTAL Memory:** A text input field with a 'GB' unit selector.
- Expected run time:** A section with three dropdown menus for 'Days', 'Hour', and 'Minu'.
- Project Account:** A dropdown menu showing '-- Choose an option --'.
- Additional Slurm parameters:** A text input field.

At the bottom right, there are two buttons: 'Preview' and 'Hide History'.

# Sample Job Script Structure

```
#!/bin/bash
##NECESSARY JOB SPECIFICATIONS
#SBATCH --export=NONE
#SBATCH --get-user-env=L
#SBATCH --job-name=JobExample1
#SBATCH --time=01:30:00
#SBATCH --ntasks=1
#SBATCH --mem=2G
#SBATCH --output=stdout.%j

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

# load required module(s)
module purge
module load GCCcore/14.2.0 Python/3.13.1

# Run your program
python my_program.py
```

These *parameters* describe your job to the Slurm job scheduler.

**The lines starting with #SBATCH are NOT comments!**

See the [Knowledge Base](#) for more info

← ntasks = cores

← Account number to be charged

Whatever commands or scripts you want to run. Here, we set up the modules we need for our environment and run a python program.

(We will practice with job files in a few slides!)

# Submit a Job and Check Job Status

Submit job

```
sbatch example01.job
```

```
Submitted batch job 6853258
(from job_submit) your job is charged as below
      Project Account: 122792016265
      Account Balance: 1687.066160
      Requested SUs:   3
```

Check status

matching JOBID

```
squeue -u netID
```

or

```
squeue --me
```

| JOBID   | NAME    | USER     | PARTITION | NODES | CPUS | STATE   | TIME       | TIME_LEFT | START_TIME         | REASON | NODELIST   |
|---------|---------|----------|-----------|-------|------|---------|------------|-----------|--------------------|--------|------------|
| 6853258 | jobname | NetID    | xlong     | 2     | 96   | RUNNING | 3-07:36:50 | 16:23:10  | 2025-01-23T17:27:3 | None   | c[180,202] |
| 6853257 | jobname | someuser | xlong     | 2     | 96   | RUNNING | 3-07:36:56 | 16:23:04  | 2025-01-23T17:27:2 | None   | c[523-524] |

# Hands-On Activity

1. Navigate to `/scratch/training/Intro-to-Grace`
2. Copy `hello_world.slurm` and `hello_world.py` to your home directory
3. Return to your home directory and submit the job file using `sbatch`.
4. Check that the job is running in a Slurm queue with `squeue`.
5. When your job completes, check the contents of the output file.

# Batch Queues

- Job submissions are auto-assigned to batch queues based on the resources requested (e.g. number of cores/nodes and walltime limit)
- Use **sinfo** to check their status:

```
[NetID@grace2 ~]$ sinfo
PARTITION      AVAIL  TIMELIMIT  JOB_SIZE  NODES (A/I/O/T)  CPUS (A/I/O/T)
short*         up      2:00:00    1-32      687/97/16/800    30786/6758/856/38400
medium         up      1-00:00:00  1-128     687/97/16/800    30786/6758/856/38400
long           up      7-00:00:00  1-64      687/97/16/800    30786/6758/856/38400
xlong          up      21-00:00:00 1-32      687/97/16/800    30786/6758/856/38400
vnc            up      12:00:00    1-32      104/12/1/117     895/4633/88/5616
gpu            up      4-00:00:00  1-32      104/12/1/117     895/4633/88/5616
bigmem         up      2-00:00:00  1-4       0/7/1/8          0/560/80/640
staff          up      infinite    1-infinite 791/109/17/917   31681/11391/944/4401
special        up      7-00:00:00  1-infinite 791/109/17/917   31681/11391/944/4401
gpu-a40        up      10-00:00:00 1-15      15/0/0/15        45/675/0/720
```

**For the NODES and CPUS columns:**

**A = Active** (in use by running jobs)

**I = Idle** (available for jobs)

**O = Offline** (unavailable for jobs)

**T = Total**

<https://hprc.tamu.edu/kb/User-Guides/Grace/Batch/#batch-queues>

# Need Help?

First check the [FAQ](#)

- [Grace User Guide](#)
- Email your questions to [help@hprc.tamu.edu](mailto:help@hprc.tamu.edu)

Help us help you -- when you contact us, tell us:

- Which Cluster you're using
- Your username
- 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

# Job Submission and Tracking

| Slurm queue command                      | Description                                                                     |
|------------------------------------------|---------------------------------------------------------------------------------|
| <b>sbatch</b> jobfile1                   | Submit jobfile1 to batch system                                                 |
| <b>squeue</b> [-u user_name] [-j job_id] | List jobs                                                                       |
| <b>scancel</b> job_id                    | Kill a job                                                                      |
| <b>sacct -X -j</b> job_id                | Show information for a job<br>(can be when job is running or recently finished) |
| <b>sacct -X -S</b> YYYY-HH-MM            | Show information for all of your jobs since YYYY-HH-MM                          |
| <b>lnu</b> job_id                        | Show resource usage for a job                                                   |
| <b>pestat -u</b> \$USER                  | Show resource usage for a running job                                           |
| <b>seff</b> job_id                       | Check CPU/memory efficiency for a job                                           |

<https://hprc.tamu.edu/kb/Helpful-Pages/Batch-Translation/>

# Continued Learning

[Intro to HPRC Video Tutorial Series](#)

[HPRC's Knowledge Base](#)



High Performance  
Research Computing  
DIVISION OF RESEARCH

*Give us feedback on the class with this  
survey:*

[https://u.tamu.edu/hprc\\_shortcourse\\_survey](https://u.tamu.edu/hprc_shortcourse_survey)

**Thank you**  
*Questions?*

