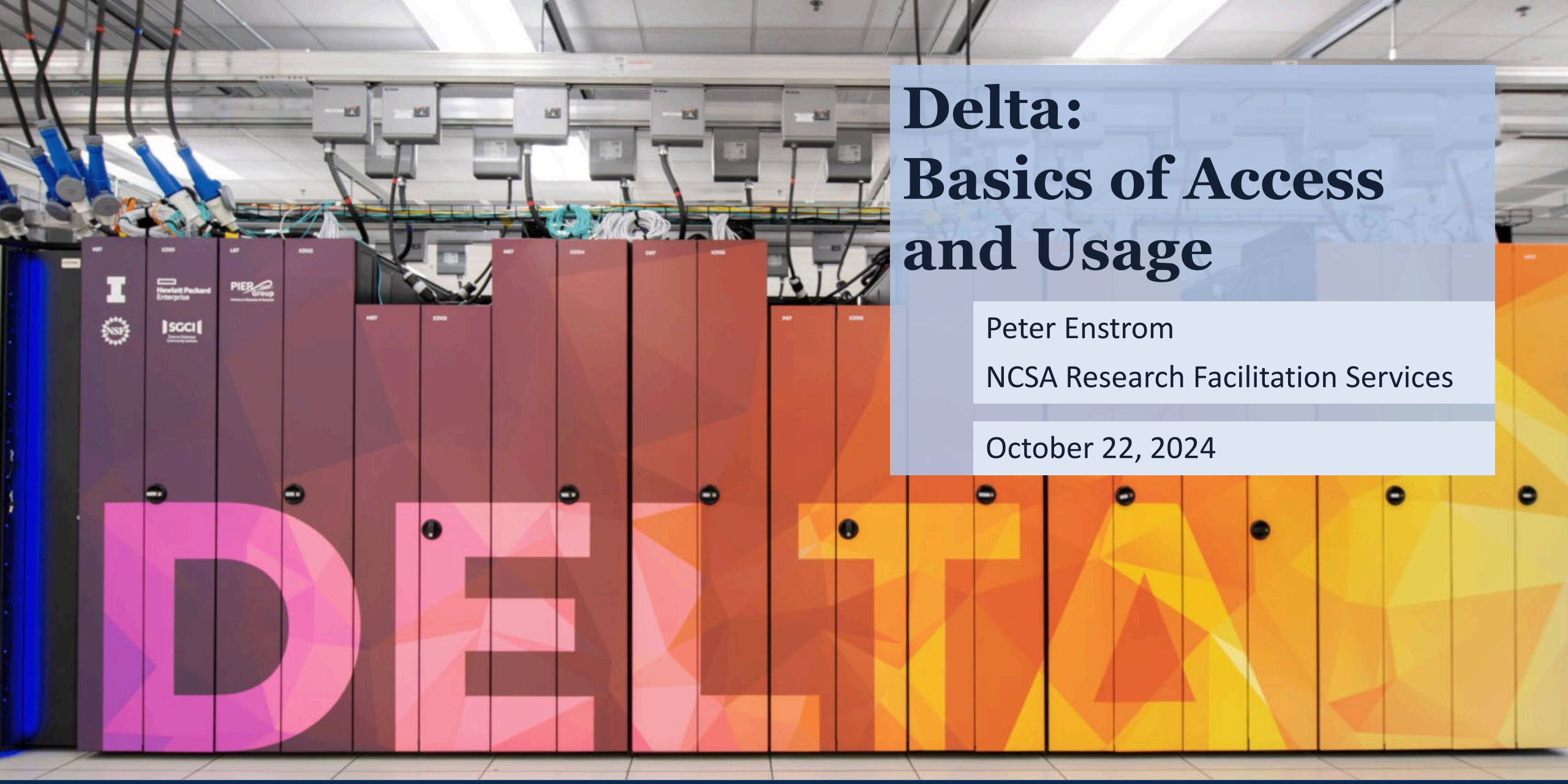


Delta: Basics of Access and Usage

Peter Enstrom

NCSA Research Facilitation Services

October 22, 2024

Outline

Block A

- What is Delta?
- What is Delta designed for?
- How to request an allocation

- How to access Delta
- How to navigate the file system
- How to run jobs
- How to transfer files to/from Delta

Block B

- Q&A session

NCSA DELTA

Code of Conduct

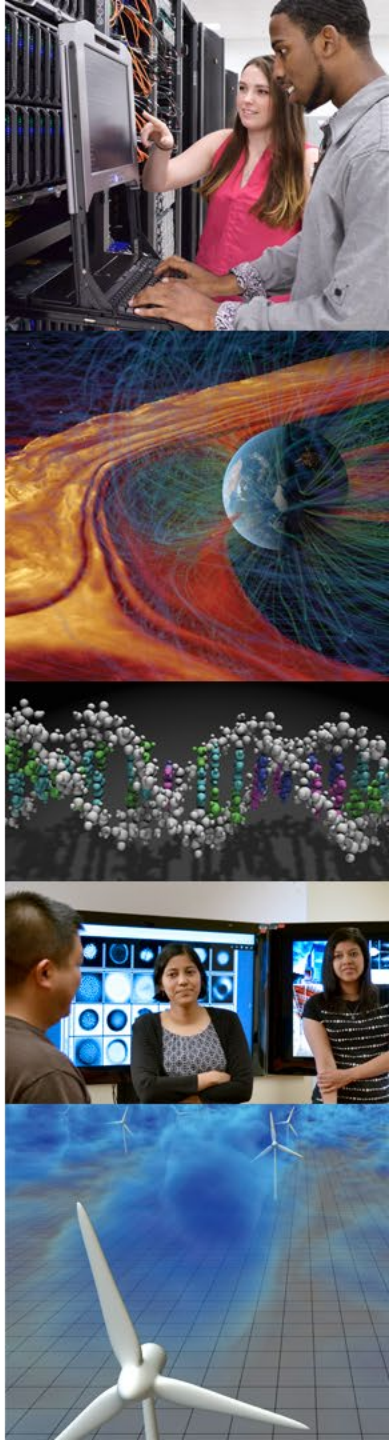
- ACCESS Code of Conduct - https://access-ci.org/code-of-conduct/

> As a program that aims to share ideas and freedom of thought and expression, it is essential that the interaction between participants, users of ACCESS services and ACCESS staff take place in an environment that recognizes the inherent worth of every person by being respectful of all. All ACCESS participants strive to be empathetic, respectful, welcoming, friendly, and patient. We strive to be collaborative and use language that reflects our values.
>
> The ACCESS program does not tolerate harassment in any form. Harassment is any form of behavior intended to exclude, intimidate or cause discomfort. Harassment includes, but is not limited to, the use of abusive or degrading language, intimidation, stalking, harassing photography or recording, inappropriate physical contact, and unwelcome sexual attention.

- UIUC Code of Conduct - https://www.ethics.uillinois.edu/compliance/university_code_of_conduct

What is Delta?

- Delta is a dedicated, ACCESS-allocated resource designed by HPE and NCSA.
- Delivers a highly capable GPU focused compute environment for GPU and CPU workloads.
- A mix of standard and reduced precision GPU resources.
- Provides high performance node-local SSD scratch file systems.
- Features a rich base of preinstalled applications, based on user demand.



Delta Hardware

GPU compute nodes

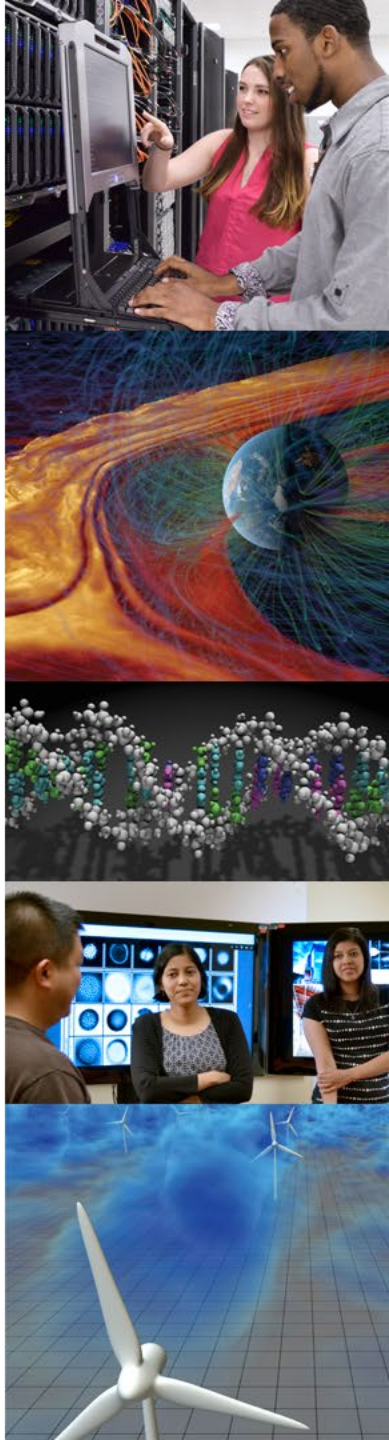
- 4-way NVIDIA A100 GPUs
  - 100 nodes
- 4-way NVIDIA A40 GPUs
  - 100 nodes
- 8-way NVIDIA A100 GPUs
  - 6 nodes
- 8-way AMD MI100 GPUs
  - 1 node

CPU compute nodes

- 132 CPU nodes
  - AMD EPYC 7763 “Milan”
  - 64 cores per node
  - 128 threads per node
  - 256 GB of RAM

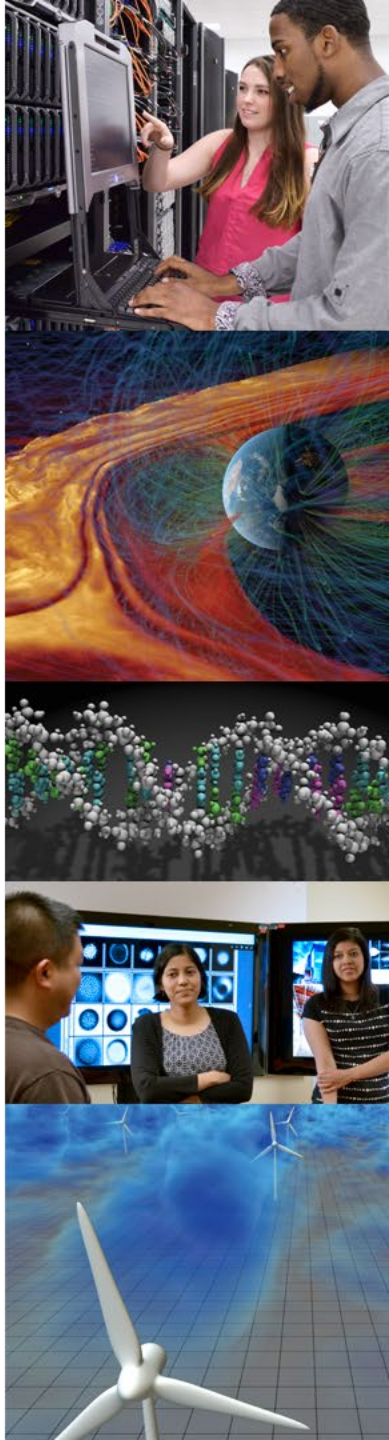
See the Delta user documentation for complete hardware specifications:

https://docs.ncsa.Illinois.edu/systems/delta



What is Delta Designed for?

- Delta is designed to help applications transition from CPU-only to GPU or hybrid CPU-GPU codes.
- The Delta CPUs are designed for general purpose computation across a range of domains that have algorithms that have not yet moved to the GPU.
- The Delta GPUs are designed to support accelerated computation across a range of domains including machine learning and visualization.



How to Request a Delta Allocation

- Most of Delta is allocated through ACCESS (https://allocations.access-ci.org/).
  - Explore, Discover, and Accelerate opportunities are open and processed continuously.
  - ACCESS awards are suitable for:
    - General research allocations
    - Educational allocations
    - Startup allocations
    - Campus Champion allocations

- A portion of Delta is available for allocation from the NSF NAIRR Pilot program for AI research. See the NAIRR Pilot Program Allocations page (https://nairrpilot.org/allocations) for more information.

Outline

Block A

- What is Delta?
- What is Delta designed for?
- How to request an allocation

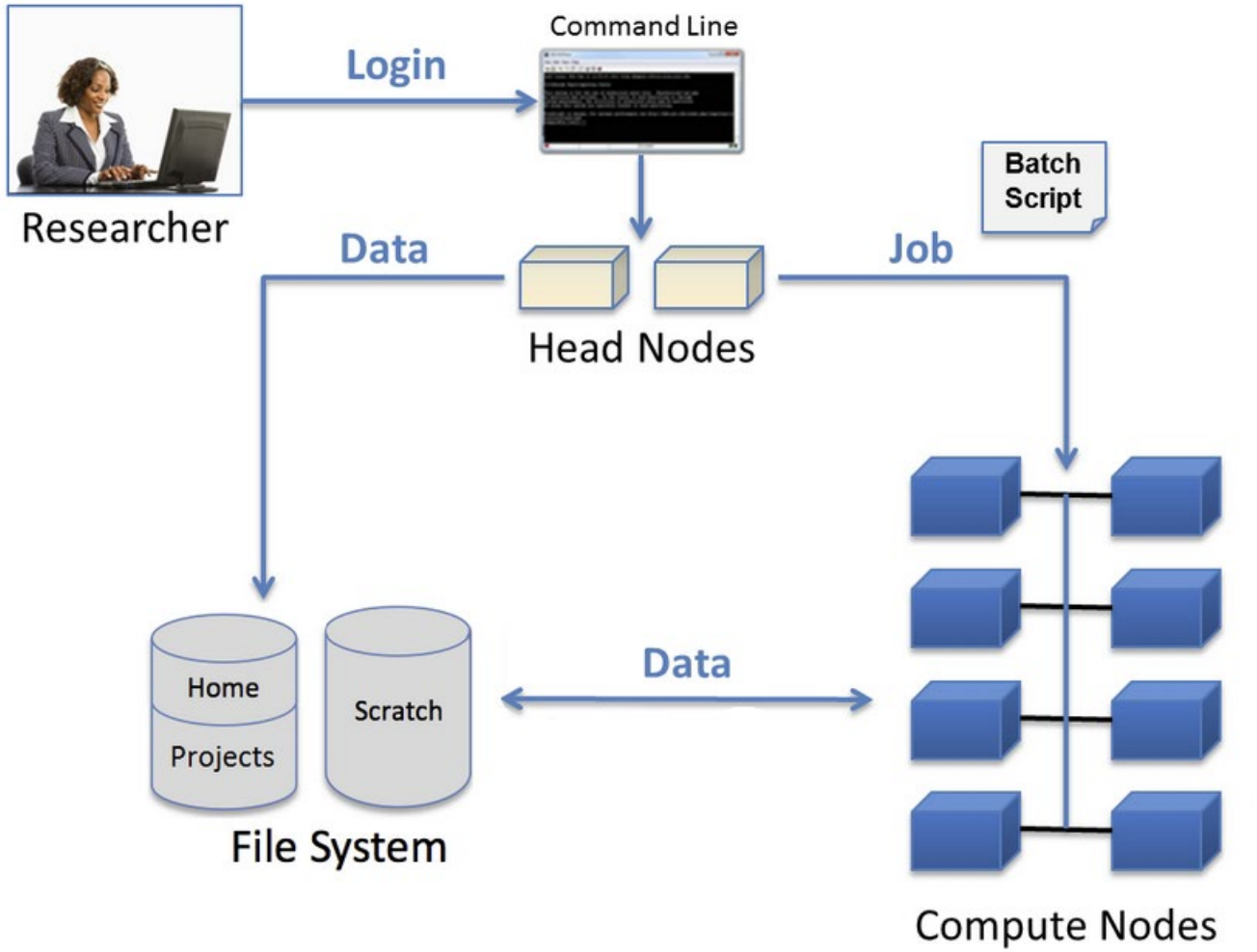
- How to access Delta
- How to navigate the file system
- How to run jobs
- How to transfer files to/from Delta

Block B

- Q&A session

NCSA | DELTA

Delta Access and Usage Overview

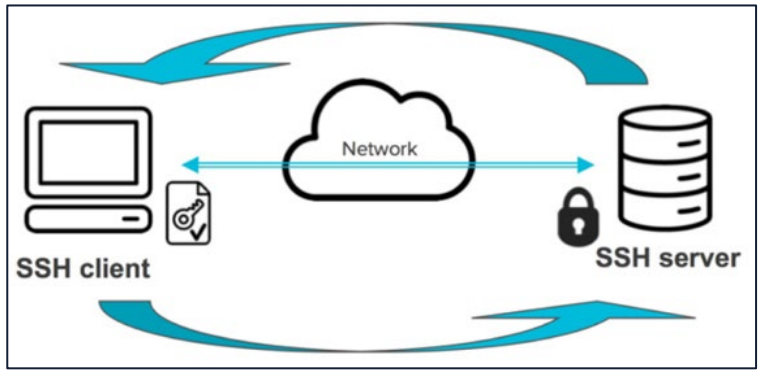


We will discuss each of these components with 6 hands-on exercises:

1. Accessing via SSH
2. The file system
3. Copying and moving file
4. Job scripts
5. Keeping track of your jobs
6. Transferring files

SSH – Secure Shell Protocol

- SSH Provides a secure channel over an unsecured network by using client-server architecture.



- SSH access is usually done via a *command line interface (CLI)*. Operating systems (Mac, Windows, and Linux) have a default *Terminal* installed that you can use as a CLI to SSH into Delta.

Exercise 1: Access Delta via Terminal

1. Open your Terminal

Mac OS	Windows OS	Linux OS
"Terminal" in Spotlight Search (Command+Space)	Start -> Windows System -> Command Prompt	Ctrl+Alt+T

2. Connect to Delta via your Terminal:

```
ssh username@login.delta.ncsa.illinois.edu
```

3. After entering the SSH command, you will be prompted to your **NCSA identity password**.

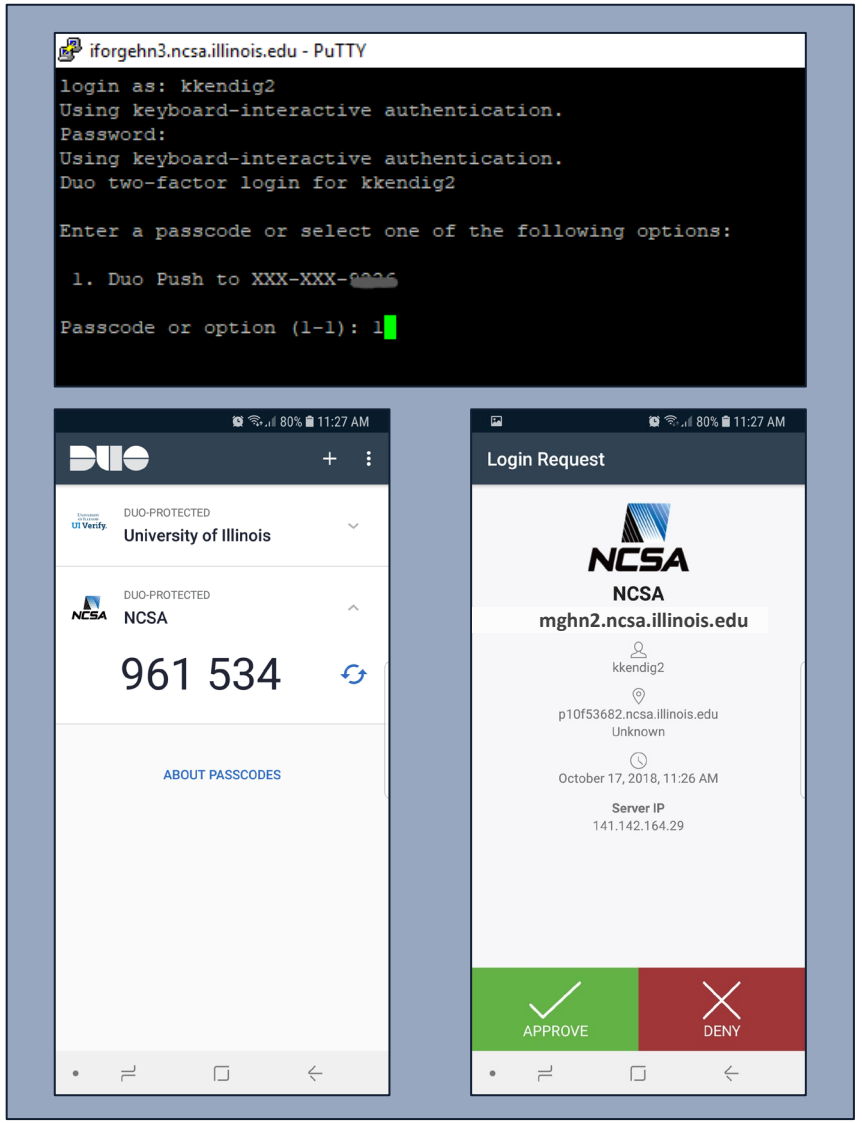
NCSA Duo Authentication

NCSA Duo MFA is required after logging in.
Authenticate with **ONE** of the following:

- **Passcode:** Enter the passcode generated in the Duo app on your phone into your terminal (without spaces).

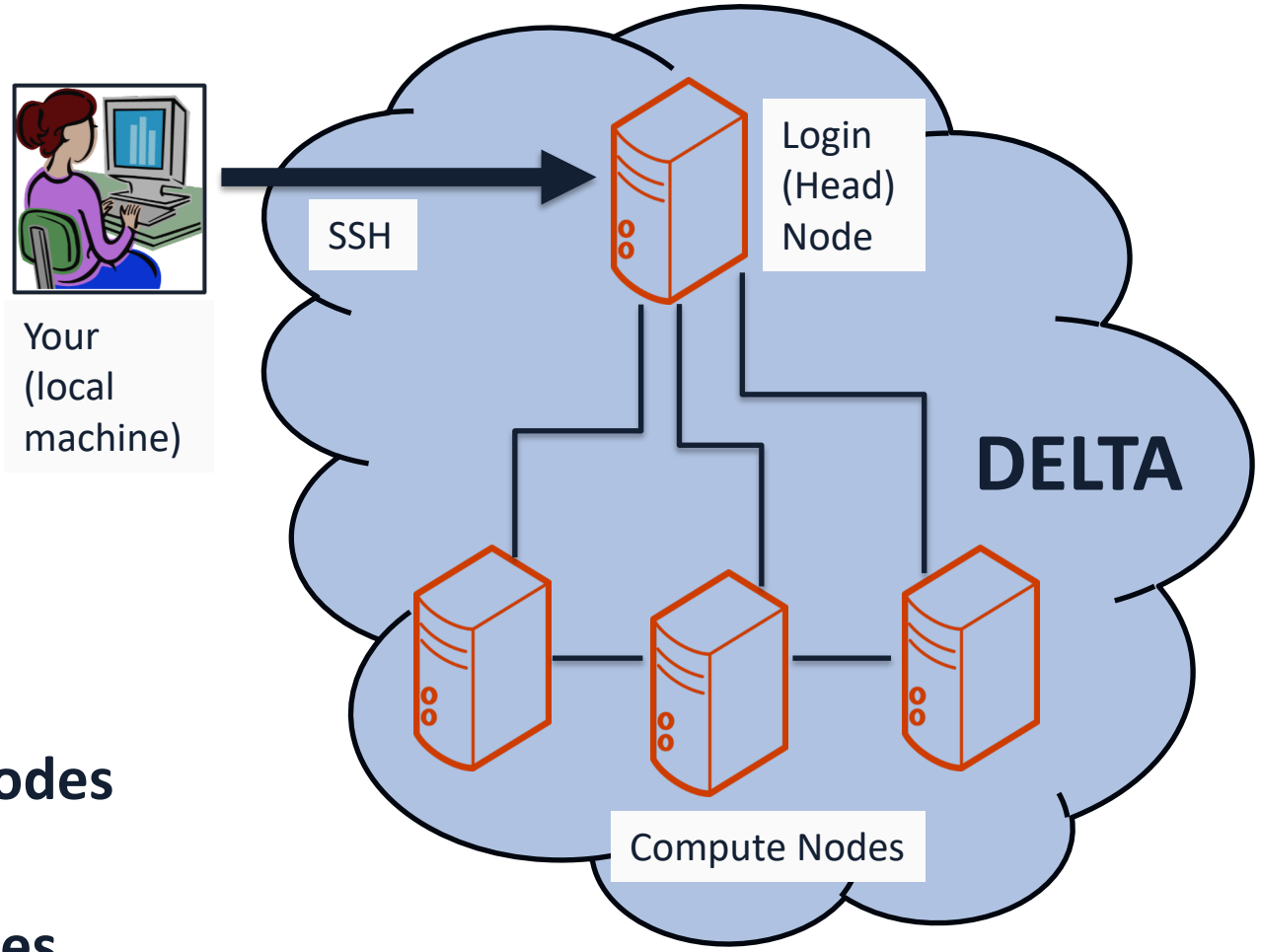
- **Push:** Type **1** in the terminal and hit **enter**. Within a few seconds, you will receive a push notification to your phone to validate your login attempt, which you can then approve with one click.

Your **NCSA Duo** account is separate from your **UIUC Duo** account.

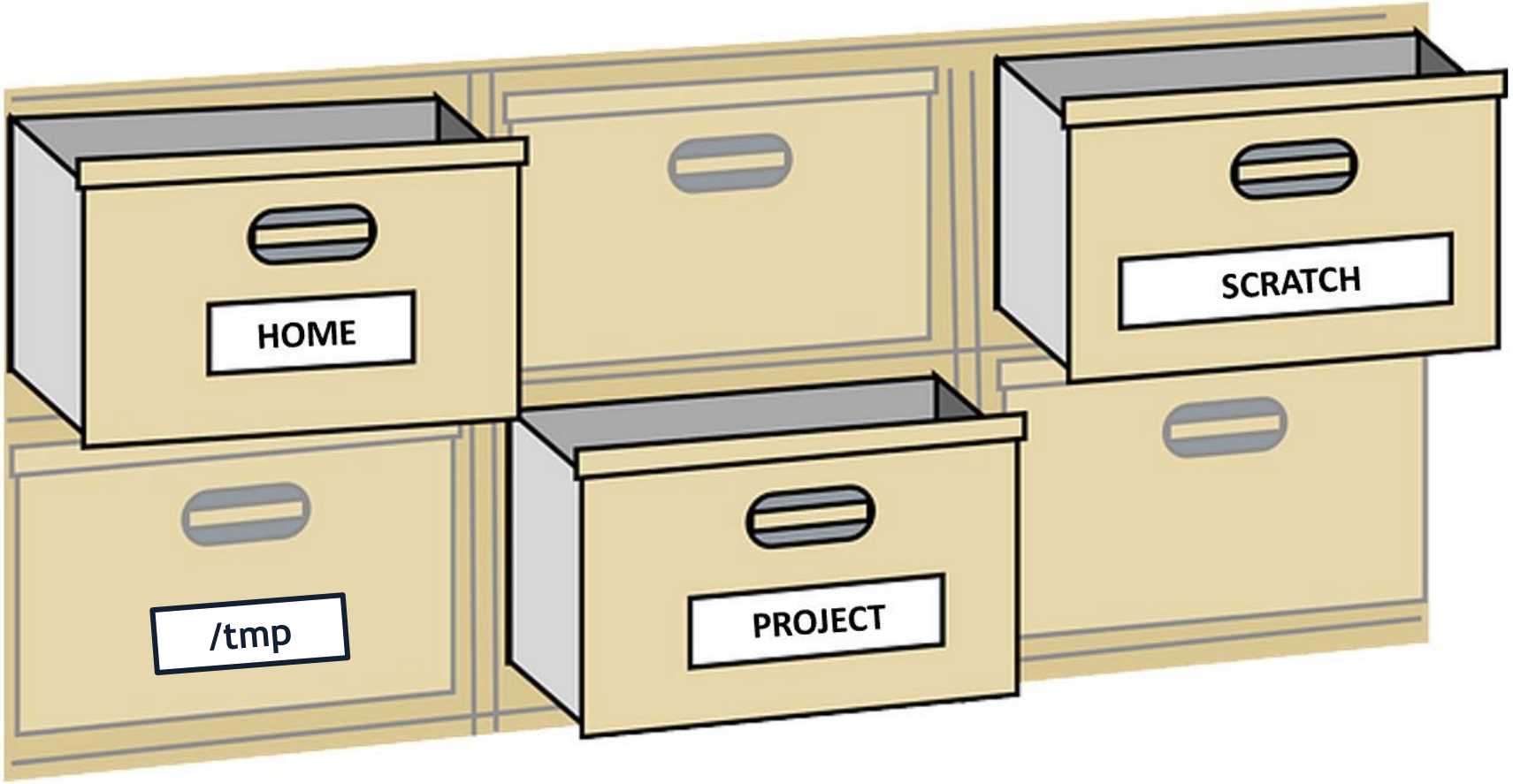


Where are you After Connecting?

- You start on one of the **login (head) nodes**, which is shared by many users.
- Your terminal window is now essentially a Bash Shell in the Linux cluster environment.
- Use **login nodes** for tasks such as:
  - file editing
  - code compilation
  - job submission and tracking
- ! DO NOT run applications on the **login nodes** (short, test runs are okay).
  - Run applications on the **compute nodes**.



The File System



Home Directory

- Directory name: **/u/<username>**
- Default space upon login
- 90GB quota
- Store files you want to keep long-term such as source code, scripts, data input files, and software
- Not for job input/output due to its smaller quota
- Not purged

Project Directory

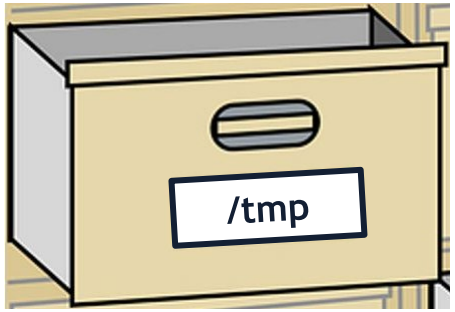
- Directory name: **/projects/<project_name>/**
- Minimum quota is 500GB; can be larger based on allocation request
- /projects is hosted on NCSA's center-wide (Taiga) file system
- Area for shared data for a project, common data sets, software, results, and so on
- Not purged

Scratch Directory

- Directory name: **/scratch/<project_name>/**
- Minimum quota is 100GB; can be larger based on allocation request
- A temporary space for job output
- Larger temporary files – like intermediary files in a pipeline that you don't need to keep
- Not purged

/tmp Directory

- Directory name: **/tmp**
- Unique to each node and job – not a shared file system
- No quotas in place, 0.74TB (CPU) or 1.50TB (GPU) shared or dedicated depending on node usage by job(s)
- Locally attached disk for fast small file input/output
- Purged after each job

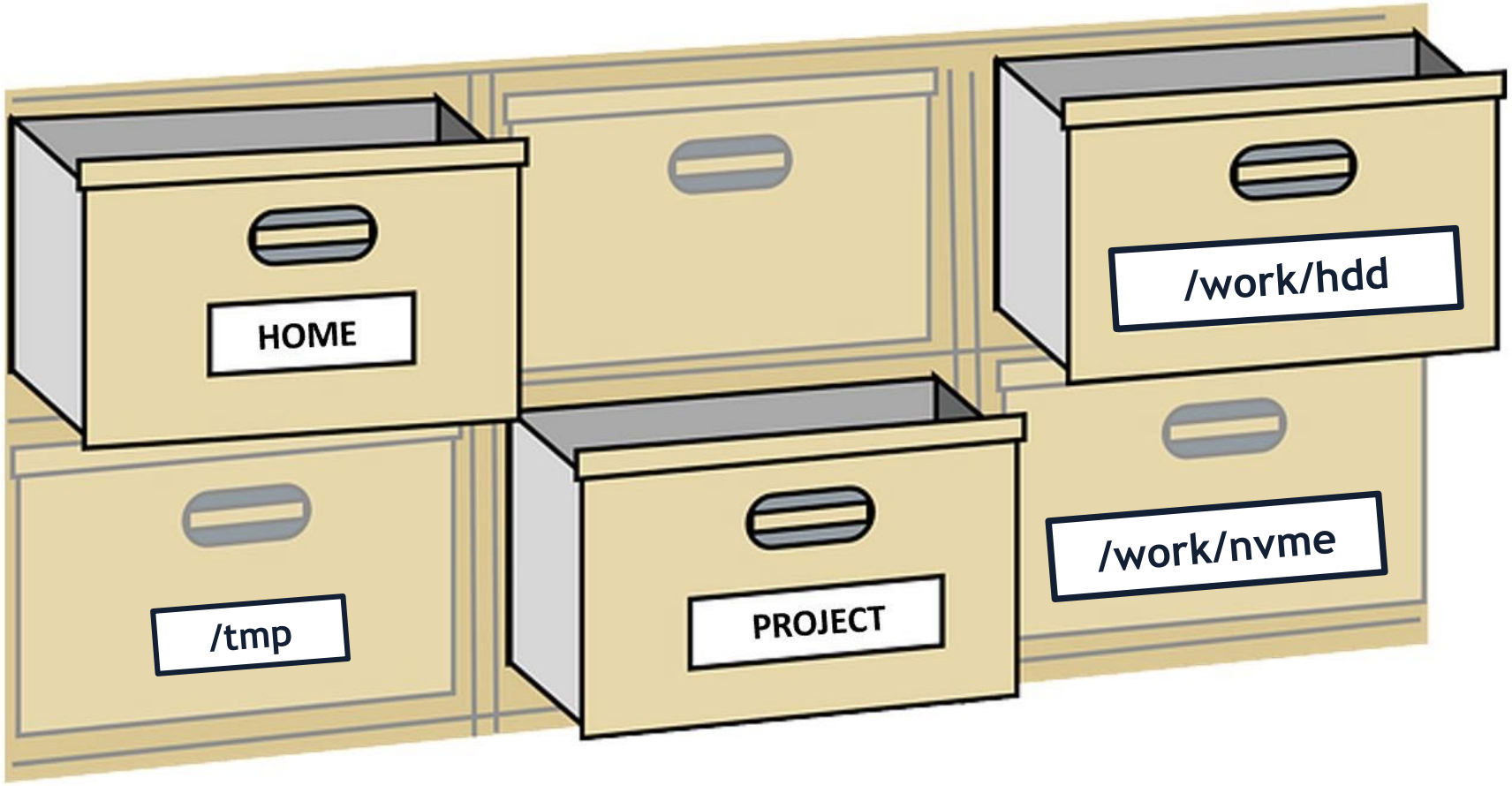


Quotas

- Use the **quota** command to view your system use and use by your project(s).
- This example output is for a person, with the username “<user>”, that is in two projects: “aaaa” and “bbbb”.

```
<user>@dt-login01 ~]$ quota
Quota usage for user <user>:
--------------------------------------------------------------------------------------------
| Directory Path | User | User | User  | User | User   | User |
|                | Block| Soft | Hard  | File | Soft   | Hard |
|                | Used | Quota| Limit | Used | Quota  | Limit|
--------------------------------------------------------------------------------
| /u/<user>      | 20k  | 50G  | 27.5G | 5    | 600000 | 660000 |
--------------------------------------------------------------------------------
Quota usage for groups user <user> is a member of:
--------------------------------------------------------------------------------
| Directory Path | Group | Group | Group | Group | Group  | Group |
|                | Block | Soft  | Hard  | File  | Soft   | Hard  |
|                | Used  | Quota | Limit | Used  | Quota  | Limit |
--------------------------------------------------------------------------------------------
| /projects/aaaa | 8k    | 500G  | 550G  | 2     | 300000 | 330000 |
| /projects/bbbb | 24k   | 500G  | 550G  | 6     | 300000 | 330000 |
| /scratch/aaaa  | 8k    | 552G  | 607.2G| 2     | 500000 | 550000 |
| /scratch/bbbb  | 24k   | 9.766T| 10.74T| 6     | 500000 | 550000 |
--------------------------------------------------------------------------------------------
```

Happening now, file system changes



Exercise 2: Navigating the file system

1. Use **pwd** to verify your current folder/area.
2. Use **ls** to see what files/folders you have there.
3. Use **cd** to change current directory to /usr/bin. Repeat steps 1 and 2.

```
cd <target_directory>
```

4. Use **quota** to verify your usage in each area.

Symbol	Target
.	Current directory
..	Parent directory
~	Home directory
-	Previously accessed directory

Exercise 3: Copying and moving files

1. Use **mkdir** to create a new folder in your home directory named **HelloWorld**.

```
cd; mkdir HelloWorld
```

2. Use **cp** to copy the **HelloWorld.py** file located at **/sw/training/introWS/** to your home folder.

```
cp /sw/training/introWS/HelloWorld.py $HOME
```

3. Use **mv** to move this file to the folder you created in step 1.

```
mv $HOME/HelloWorld.py HelloWorld/
```

4. Use **cp** to copy the file that you just moved. Copy it from **HelloWorld** to home.
5. Verify that copies are indeed in both folders using **ls**.
6. Remove the copy in your home folder using **rm**.

```
rm $HOME/HelloWorld.py
```

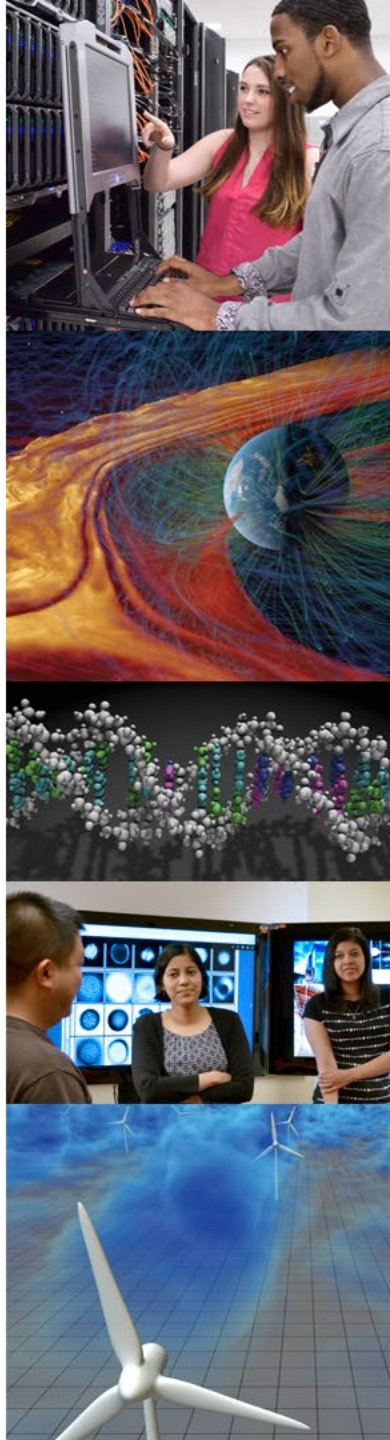
Tip: after each item, use **ls** to check if it worked!

Editing Files

- Two common Linux file editors are **vi** and **nano**.
  - **nano** is a text editor that is friendly for inexperienced Linux users.
  - **vi** (and and improved version, **vim**) is very powerful but harder to use.
- **nano**:
  - The general syntax to open a file in **nano** is: `nano <filename>`
  - If the file exists, it will open. Otherwise, a new file with that name is created.
  - Press control**(Ctrl)+X** to exit (you will be prompted to save).
  - Press control**(Ctrl)+O** to save without exiting.

Installed Software

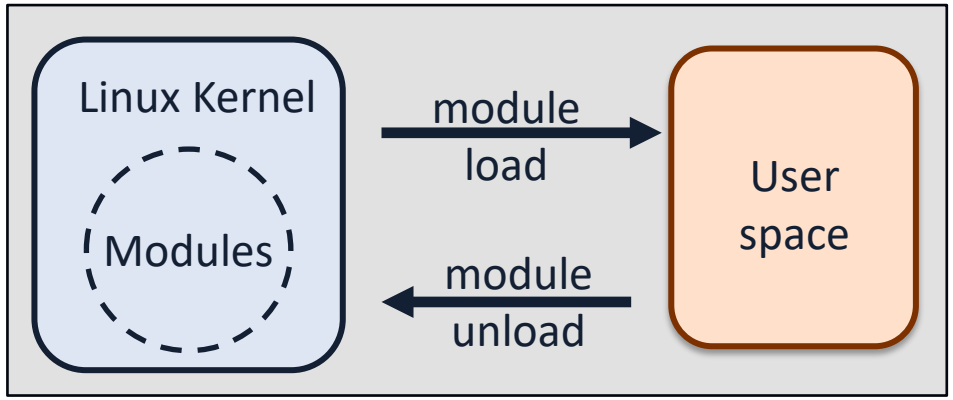
- There is a lot of software installed on Delta. Before you try to install an application, see if it's already there!
- To install software that isn't already available:
  - **Single user or single project use cases**: Use the Spack software package manager to install the software locally against the system Spack installation.
  - **General installation requests**: Submit a request through the NCSA Help Portal (https://help.ncsa.Illinois.edu). The Delta project office will review the request for broad use and installation effort.



Working with modules

Modules allow you to load and unload components of the operating system as you need them!

The same approach applies to loading a variety of software and libraries.



Command	Description
module **list**	Lists modules loaded in your session
module **avail**	Lists all available modules
module **help** *<module_file>*	Information about *<module_file>*
module **load** *<module_file>*	Loads *<module_file>* to your environment
module **unload** *<module_file>*	Removes *<module_file>* from your environment

Demonstration: Working with modules

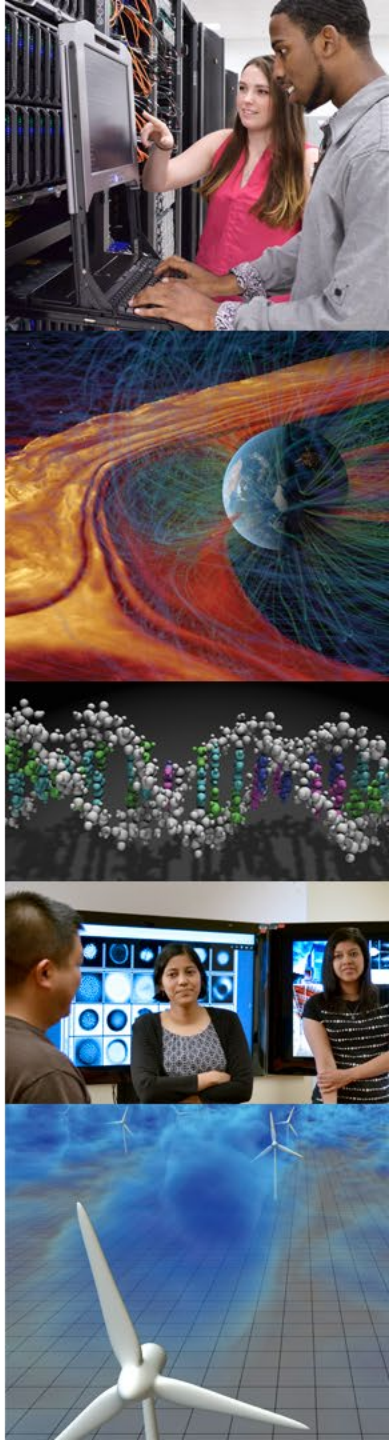
```
[enstrom@dt-login01 ~]$ module load anaconda3_cpu
[enstrom@dt-login01 ~]$ which python
/sw/external/python/anaconda3_cpu/bin/python

[enstrom@dt-login01 ~]$ python --version
Python 3.9.18

[enstrom@dt-login01 ~]$ python
Python 3.9.18 (main, Sep 11 2023, 13:41:44)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.

>>> import torch
>>> torch.cuda.is_available()
False
```

Let's Talk About Jobs

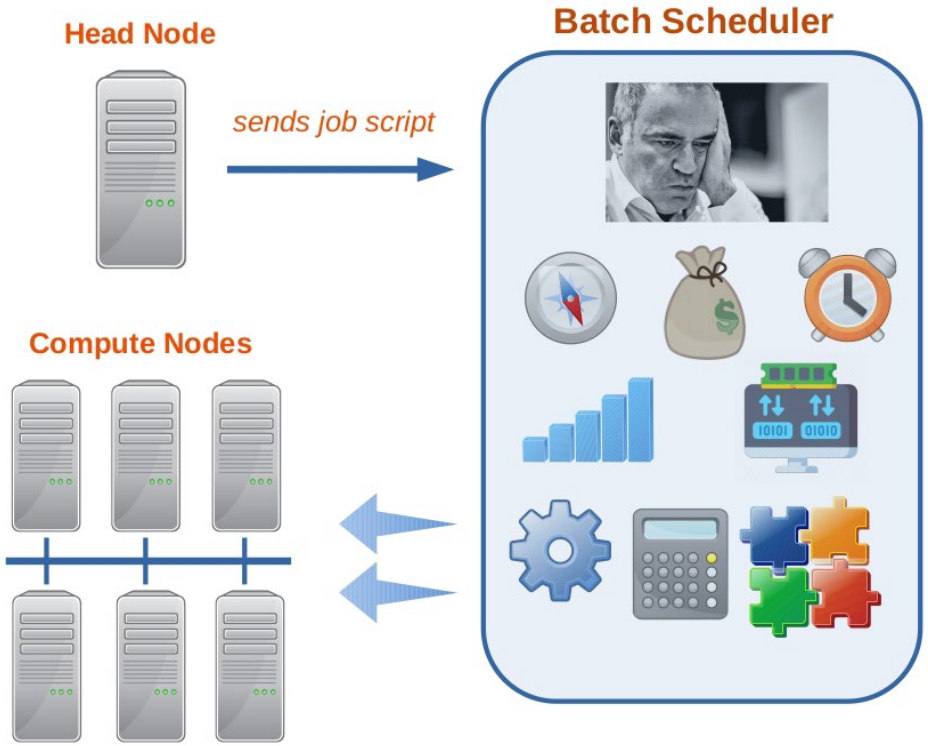
Now that we know how to:

- Log in to Delta
- Navigate the file system
- Load modules

It's time to use the cluster's power!



Batch schedulers (Slurm)



Batch processing runs jobs that can run without end-user interaction or can be scheduled to run as resources permit.

The **batch scheduler** is a computer application that weighs several factors to determine where, how, and when a certain request is going to run on the compute nodes.

Delta uses Slurm for batch scheduling.

Batch job scripts

For Slurm to do its magic, we need to provide it with some information...

Job Script

```
#!/bin/bash

#SBATCH --time=XX:YY:ZZ
#SBATCH --nodes=N
#SBATCH --ntasks-per-node=M
#SBATCH --partition=Where
#SBATCH --account=account_name
#SBATCH --output=OutFile
#SBATCH --error=ErrorFile
#SBATCH --mail-user=Email
#SBATCH --mail-type=When

./HelloWorld.exe
```

We will send bash commands

Required time

Number of nodes

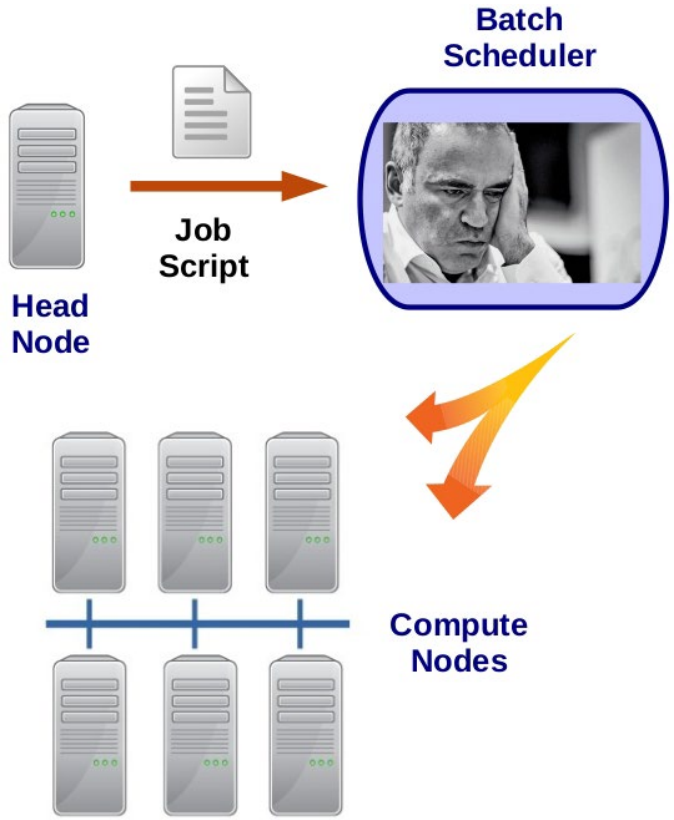
Where to run it

Account to charge

Who to email

When to email

What to run



Exercise 4: Job script for HelloWorld

1. Copy the file **HelloWorld.sbatch** to your HelloWorld folder.

```
cp /sw/training/introWS/HelloWorld.sbatch $HOME/HelloWorld/
```

2. Use `cat` to verify the information that is being passed to the scheduler.

```
cd ~/HelloWorld; cat HelloWorld.sbatch
```

3. Submit your job scrip to the batch scheduler using **sbatch**.

```
sbatch HelloWorld.sbatch
```

4. Take note of your JobID.

Exercise 5: Keeping track of your job

Use **squeue** commands to monitor the status of your job

1. List the JobIDs tied to your username.

```
squeue -u <username>
```

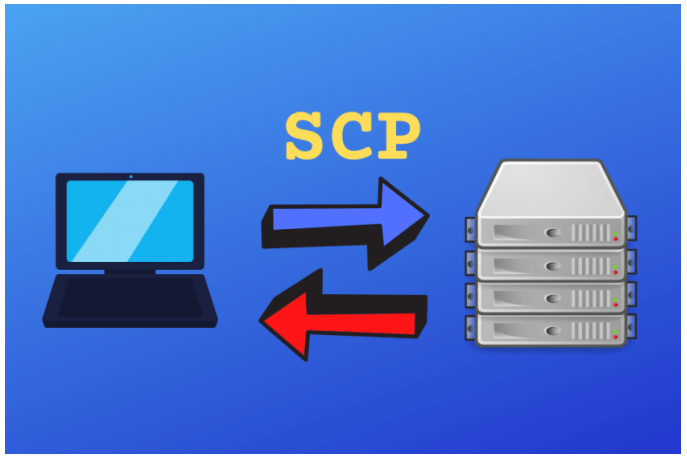
2. Look up the status of your job based on the JobID

```
squeue -j <JobID>
```

Command	Action
squeue -a	List status of all jobs in the batch system
squeue –u <username>	List status of all your jobs
squeue –j <JobID>	Lists information about a job
scancel <JobID>	Kills a job

File transfers using SCP

- Secure Copy Protocol (SCP)
  - Works basically like SSH
  - Safe channel to copy/transfer data to/from your local machine
  - Command line interface



- Pulling files from Delta to your machine:

```
scp <username>@dt-login.delta.ncsa.illinois.edu:<path_to_file> <destination>
```

- Pushing files to Delta from your machine:

```
scp <path_to_file> <Username>@dt-login.delta.ncsa.illinois.edu:<destination>
```

Exercise 6: Transfer files with SCP

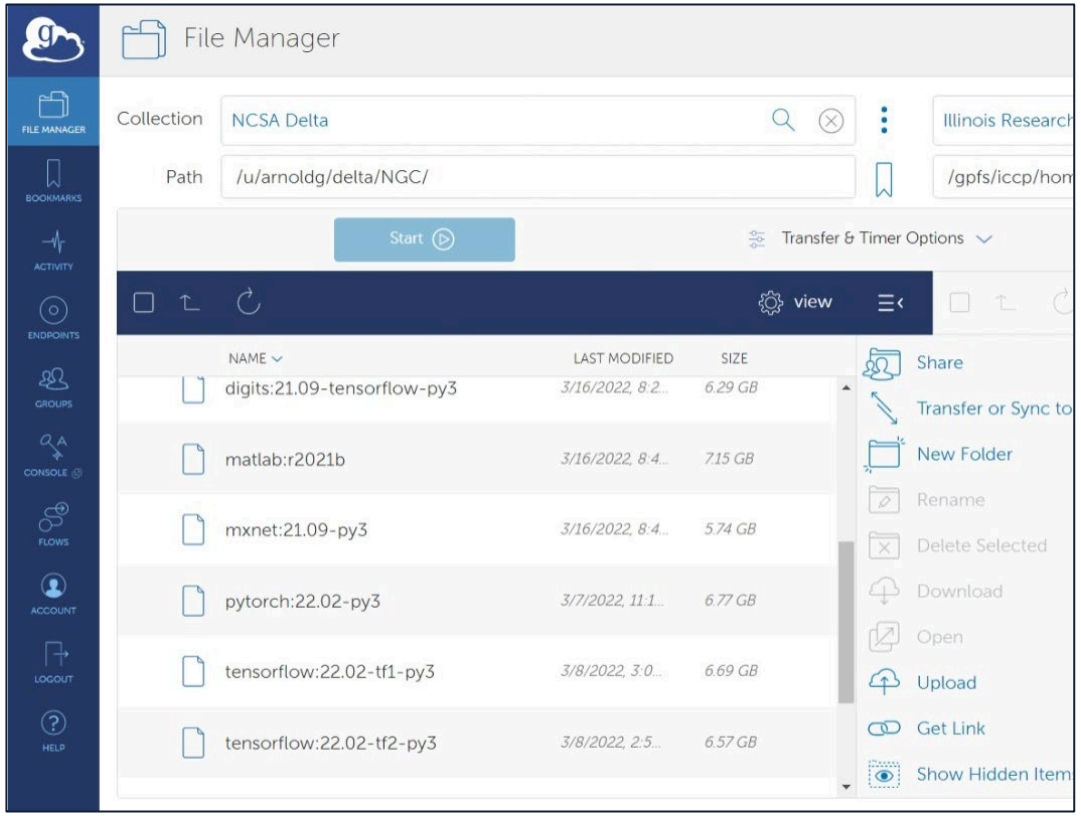
1. Log out of your cluster session using **logout**.

2. Use **scp** to copy the **HelloWorld.out** file to your local machine

```
scp <username>@delta.ncsa.illinois.edu:~/HelloWorld/HelloWorld.out .
```

3. Look for the file on your machine and try to open it using a text editor.

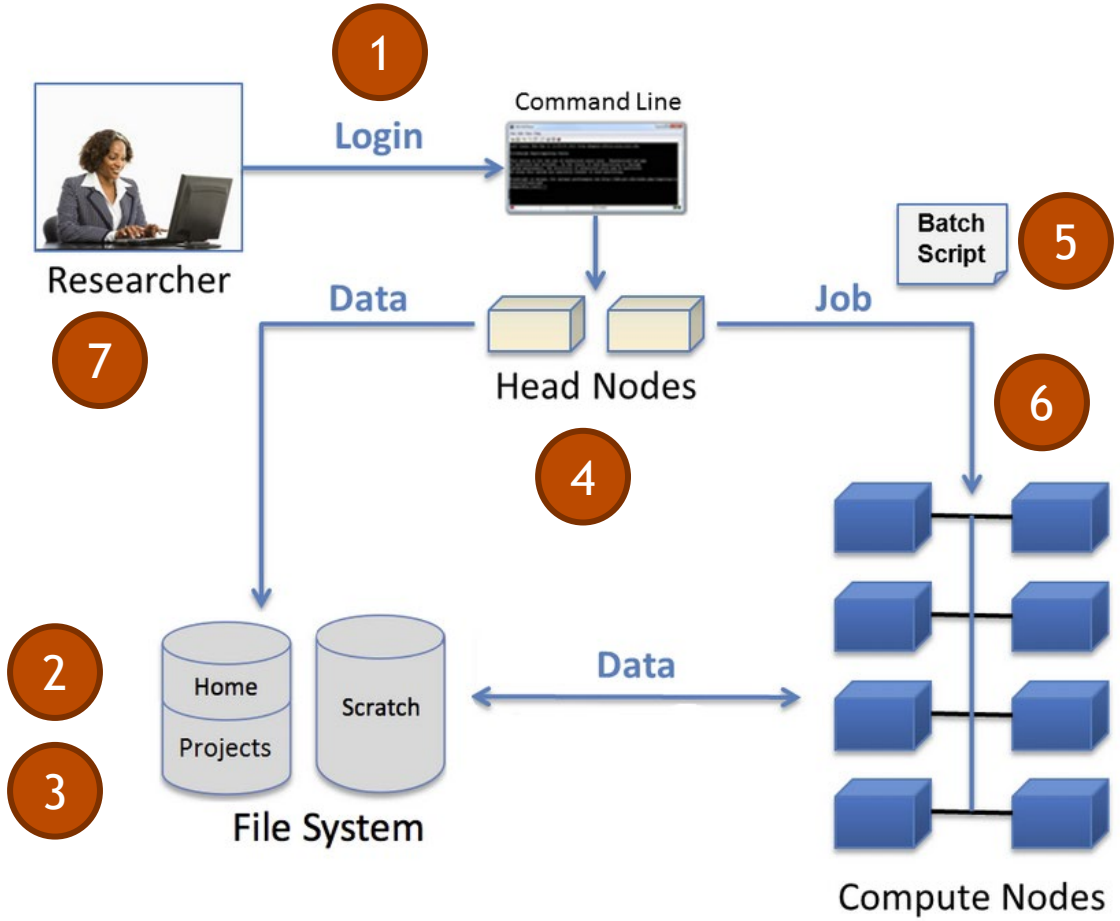
Large File Transfers with Globus



- Globus is recommended for large file transfers.
- Create a Globus account at www.globus.org
- Delta Globus endpoints:
  - “NCSA Delta”
  - “ACCESS Delta”

Cooling down

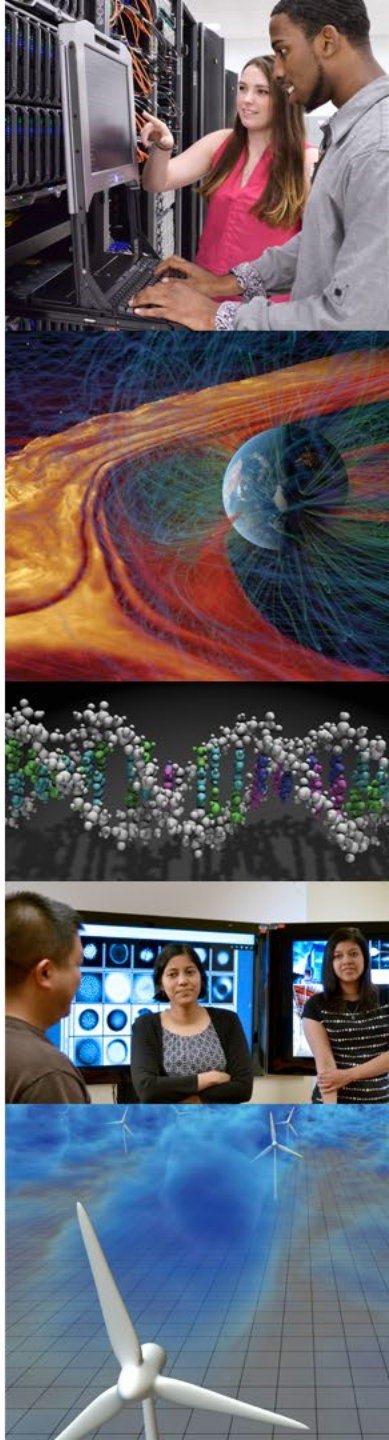
We now have a pretty good idea about navigating Delta...



1. Login using SSH client
2. Understand and use the file system
3. Copy and move files around
4. Load and unload modules
5. Write and submit a job script
6. Keep track of your jobs
7. Transfer files to your local machine

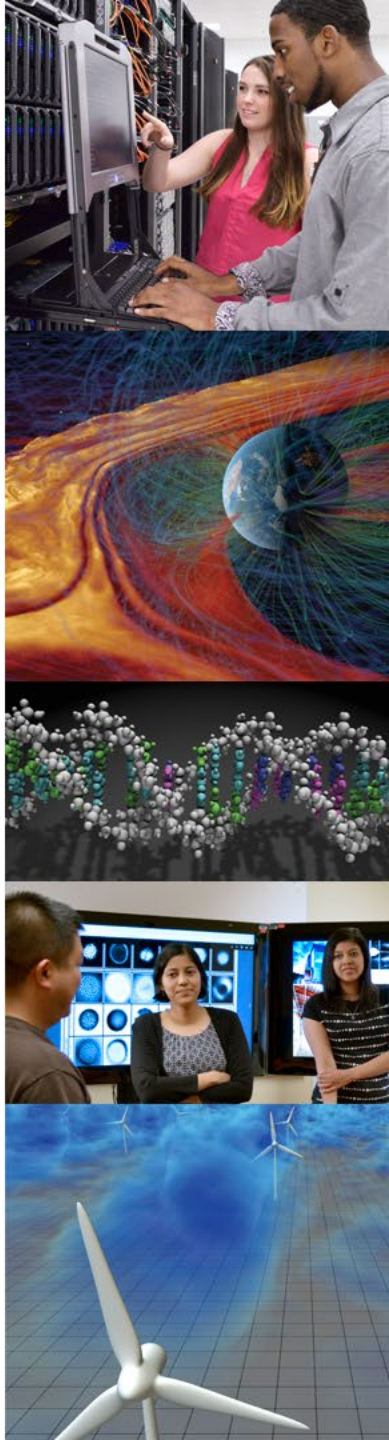
Resources

- Delta site:
  - https://delta.ncsa.Illinois.edu
- Delta user documentation:
  - https://docs.ncsa.Illinois.edu/systems/delta
- ACCESS allocation request:
  - https://allocations.access-ci.org
- Training on HPC topics (includes self-paced courses):
  - https://hpc-training.org



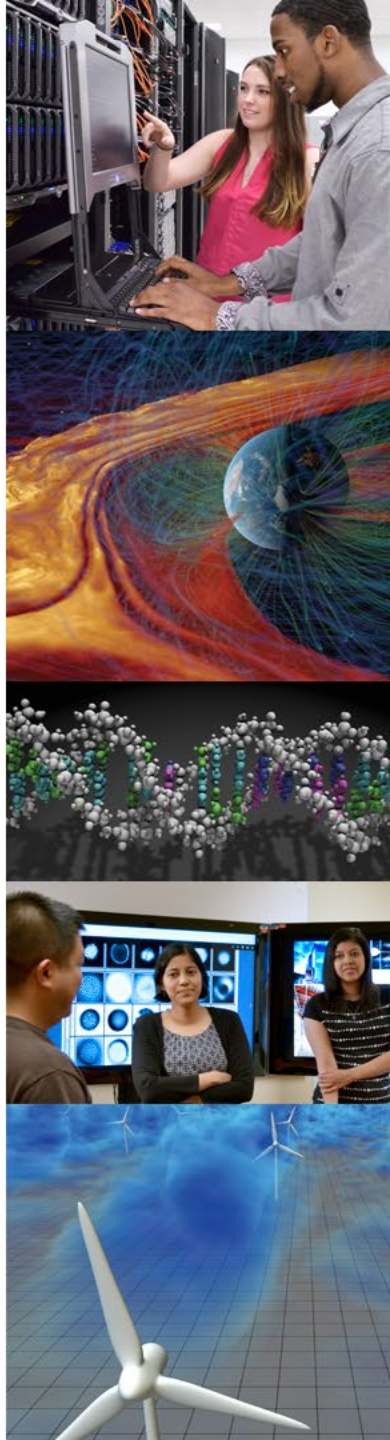
Resources

- Slurm documentation:
  - https://slurm.schedmd.com/documentation.html
- Environment modules documentation:
  - https://modules.readthedocs.io/en/latest
- Best practices for running jobs, in general:
  - https://docs.nersc.gov/jobs/best-practices
- Customizing Your Computer Environment Training:
  - https://www.hpc-training.org/moodle/course/view.php?id=77



Getting Help

- Go to the **NCSA Help Portal** (https://help.ncsa.Illinois.edu)
- When you submit a ticket, include:
  - What you were trying to do
  - How you tried to do it
  - Why you think it isn't working
  - Copy/paste the commands you ran and what output you saw on screen
- You will get emails as your ticket is worked on, please respond to questions that are asked.



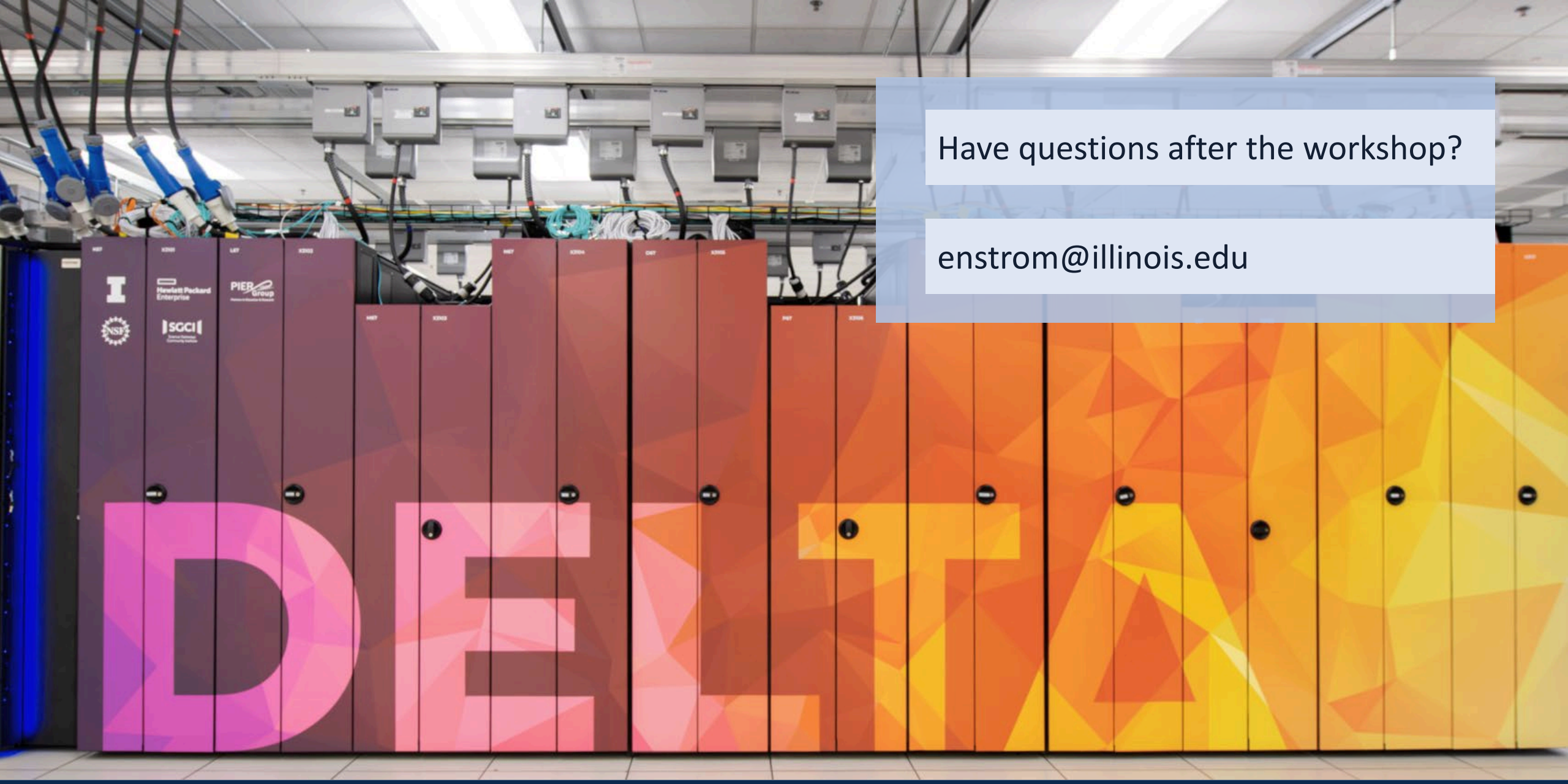

Hewlett Packard Enterprise
PIEB Group
SGCI
NSF
ChilledDoor
DELTA

Questions?

Please Provide Feedback!

- Feedback is vital for our continuous improvement as instructors/lecturers, and for our organization.

- Please take a couple of minutes to answer the anonymous survey. Any comments are extremely helpful and all of them will be carefully taken into consideration.



Have questions after the workshop?

enstrom@illinois.edu