Ptolemy: High Performance Computing Cluster

Getting Started Guide

Anup Zope

Assistant Research Professor

Michael Navicky

Director

High Performance Computing Collaboratory
Mississippi State University

September 26, 2024

Overview

Outline

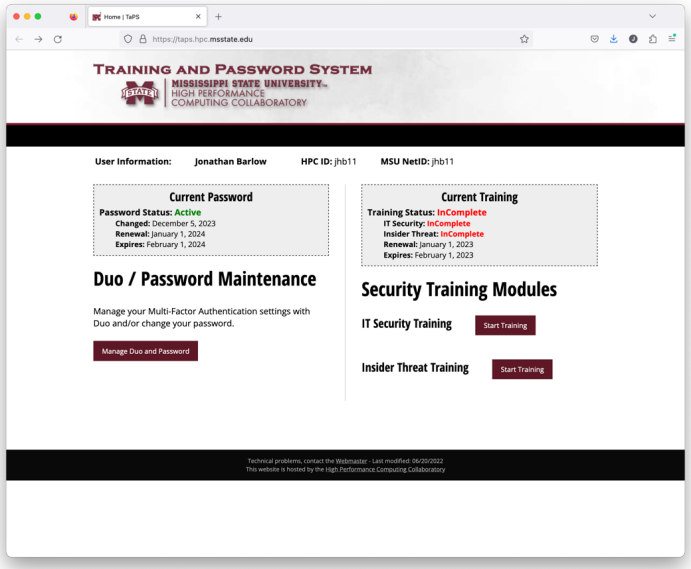
Introduction to Ptolemy

- Ptolemy is a supercomputer broadly available to MSU students and researchers.
- The cluster consists of
  - 1024 processor cores, 8 Terabytes (TB) of RAM
  - 64 NVIDIA A100 GPGPUs (80 GB RAM per GPU)
  - NVIDIA/Mellanox HDR200 InfiniBand Network (200Gb/s)
- Funded by
  - Offices of the President, Provost and Executive Vice President
  - Office of Research and Economic Development
  - The Division of Agriculture, Forestry and Veterinary Medicine
  - The Division of Finance and Administration

Outline

How to Get Access to Ptolemy?

- Email HelpDesk (help@hpc.msstate.edu) with a description of the purpose and compute requirements.
- When you receive an invitation email, click on the link in the email to fill out requested information, complete training.
- Wait to hear that your account has been created.
- Follow the instruction in the email and setup Ptolemy credentials with 2FA.



- Class instructors who plan to use the cluster for class projects need to setup class accounts by submitting helpdesk ticket.

Outline

How to Access Ptolemy?

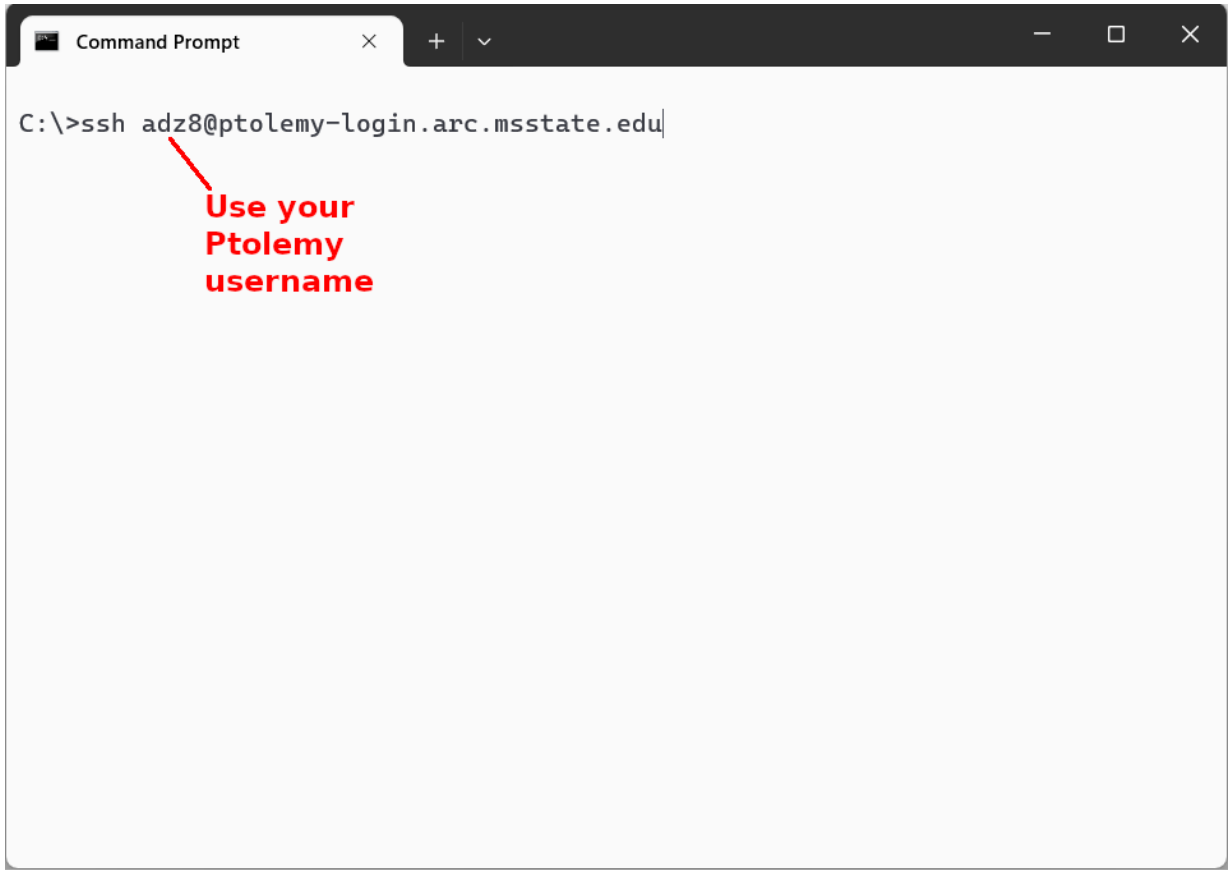
- Note: Most up to date documentation on Ptolemy can be found at `https://www.hpc.msstate.edu/computing/ptolemy`.
- What is required?
  - Active Ptolemy Account
  - Smartphone with activated DuoMobile 2FA
  - Laptop/desktop (a.k.a. client) with Windows/Linux/MacOS
  - Internet connection
- What are ways to connect to Ptolemy:
  - via Secure Shell a.k.a. SSH
  - via Open OnDemand

Connect to Ptolemy via Secure Shell (SSH)

- What is SSH?
  - A network protocol that allows users to securely access remote computers such as the Ptolemy cluster.
- This method provides a command line interface to Ptolemy. You need to be familiar with Linux bash commands to effectively use this method.
- You need an SSH client installed on your device.
  - On Windows, several options are available:
    - Windows Terminal (included in the OS)
    - Windows Subsystem for Linux (WSL)
    - Putty (`https://www.putty.org/`)
    - Cygwin (`https://www.cygwin.com/`)
    - MSYS2 (`https://www.msys2.org/`)
  - On Linux and MacOS: Terminal app included in the OS

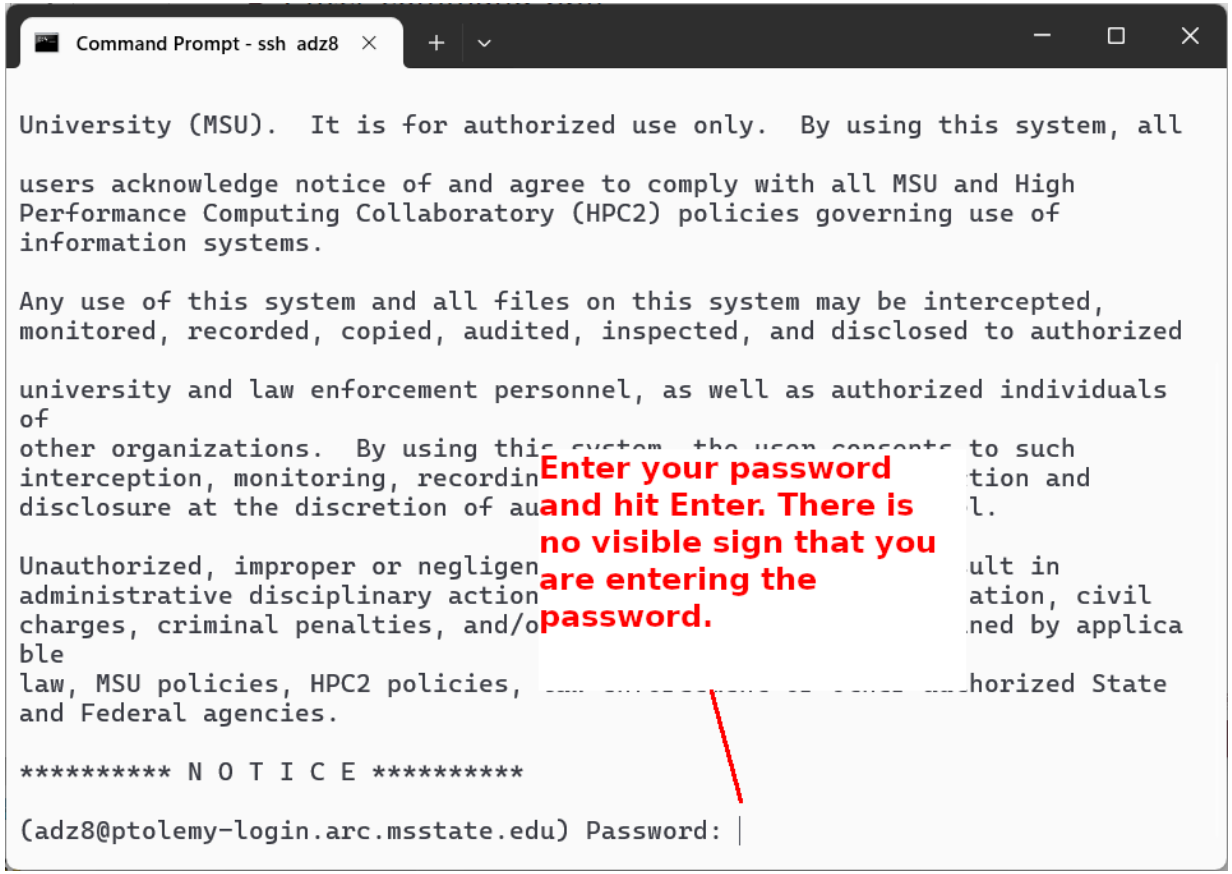
Connect to Ptolemy via Secure Shell (SSH): Terminal

- Enter command `ssh <ptolemy-user-name>@ptolemy-login.arc.msstate.edu` and hit `Enter`.



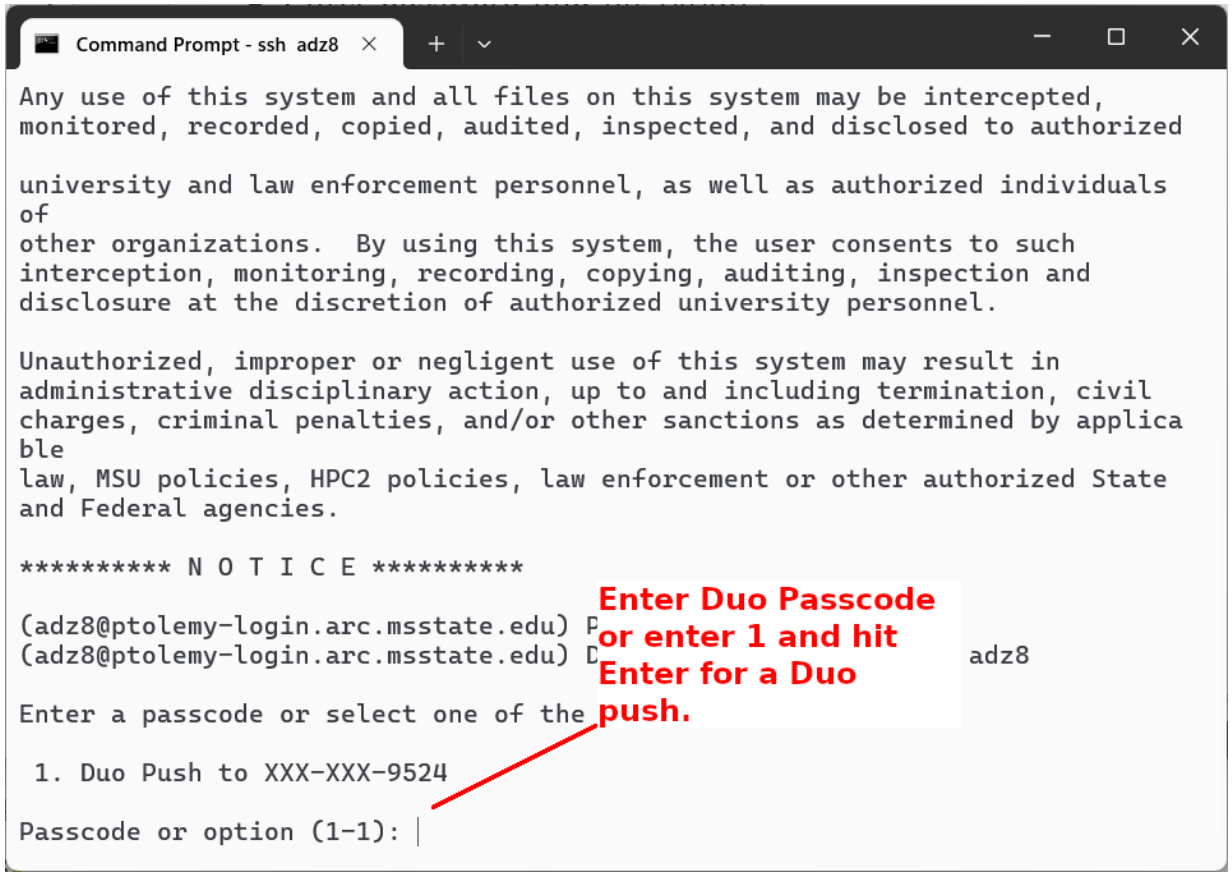
Connect to Ptolemy via Secure Shell (SSH): Terminal

- Enter password and hit `Enter`.



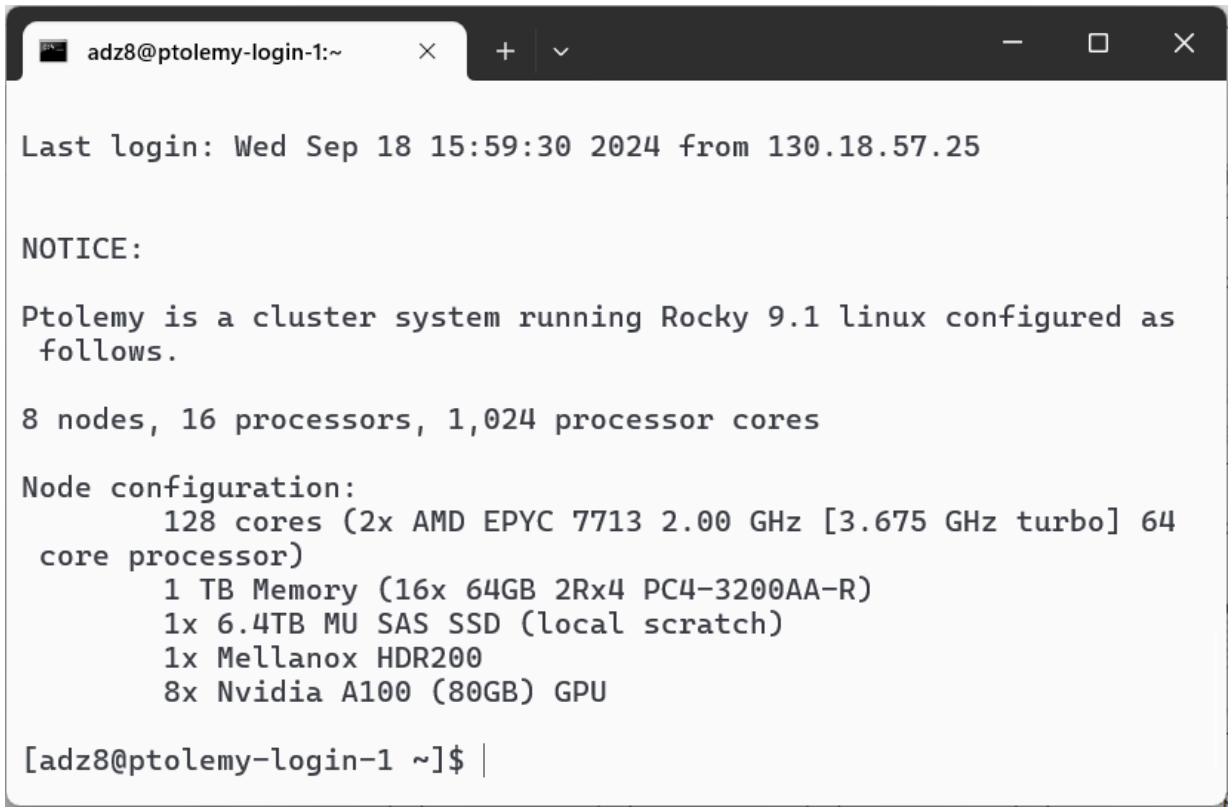
Connect to Ptolemy via Secure Shell (SSH): Terminal

- Enter Duo passcode and hit `Enter` or enter 1 for Duo push notification.



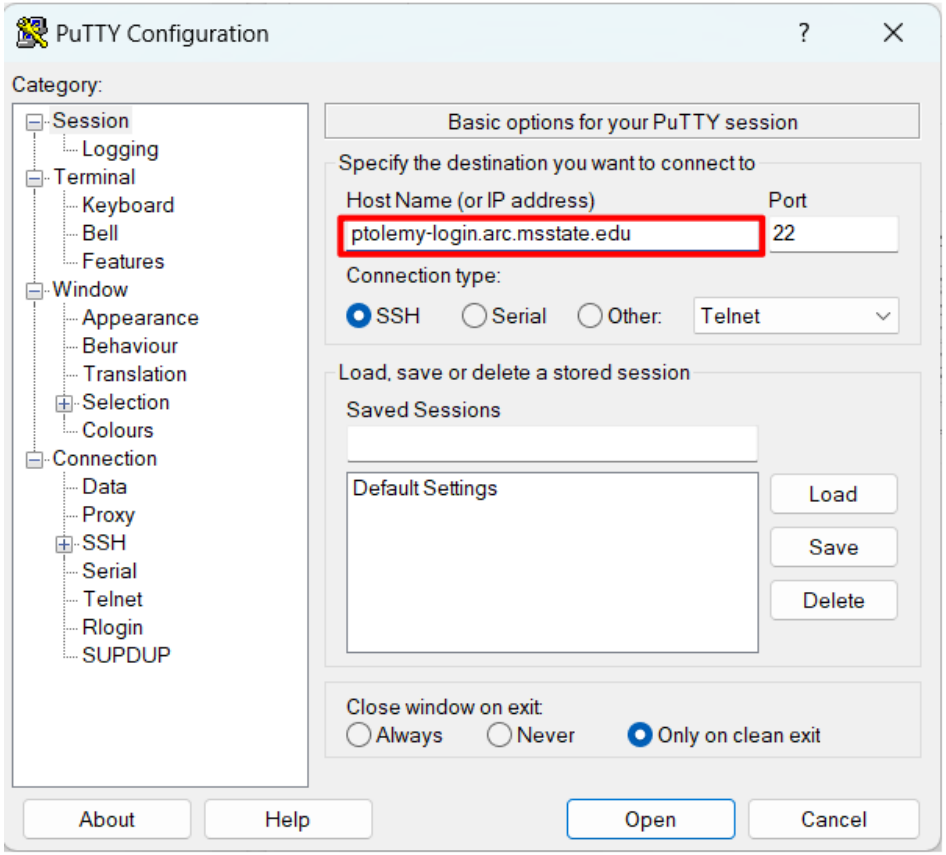
Connect to Ptolemy via Secure Shell (SSH): Terminal

- After successful login, you will see following screen.
- Use Linux bash shell commands to create/copy/move files, submit and monitor compute jobs etc.



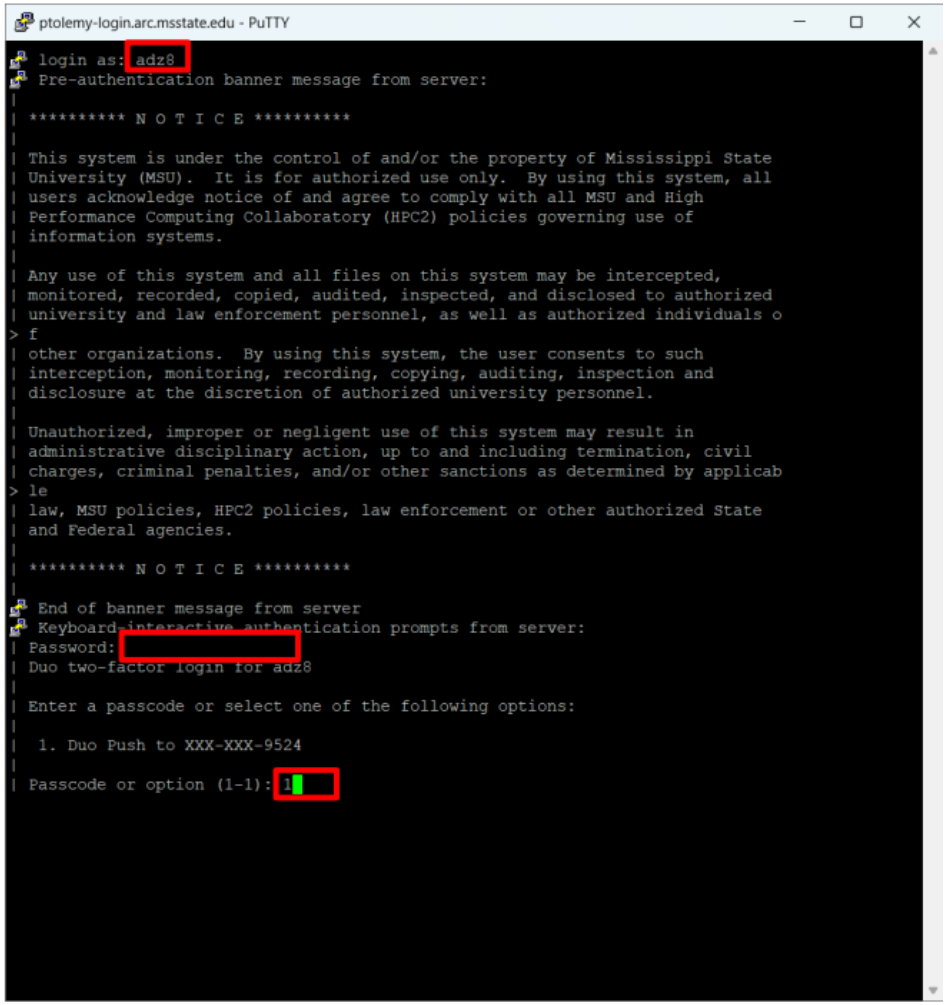
Connect to Ptolemy via Secure Shell (SSH): PuTTY

- Download and install PuTTY.
- Start PuTTY client.



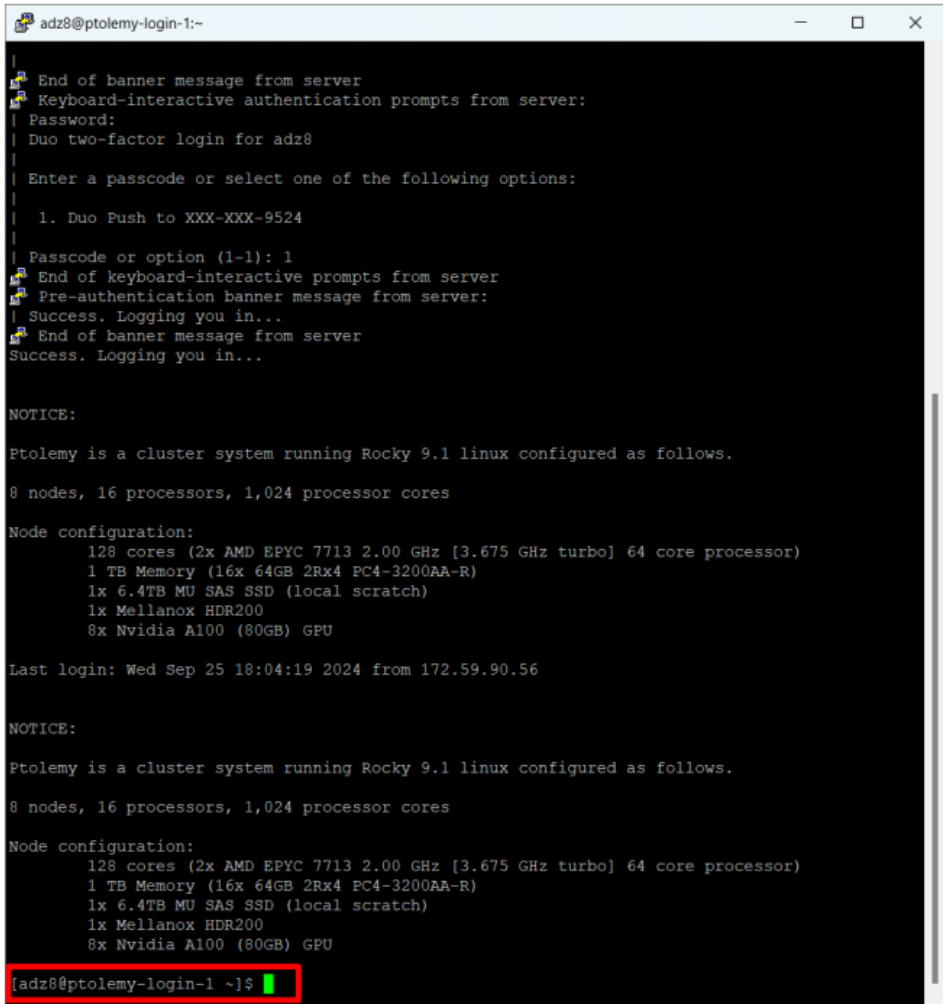
Connect to Ptolemy via Secure Shell (SSH): PuTTY

- Enter username, password, and Duo passcode or 1 for Duo push.



Connect to Ptolemy via Secure Shell (SSH): PuTTY

- After successful login:



Some Linux Bash Commands

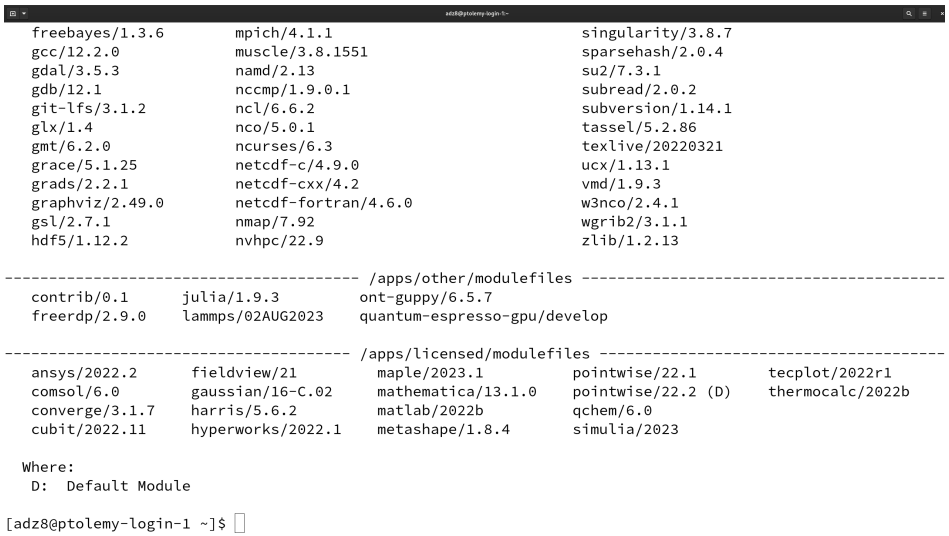
- `pwd`: Prints present working directory path.
- `ls <path-to-directory>`: List directory contents.
- `mkdir <directory-path>`: Create directory.
- `rmdir <directory-path>`: Deletes directory.
- `cd <directory-path>`: Change to the specified directory.
- `touch <file-path>`: Creates a file with specified path.
- `mv <src> <dst>`: Moves/renames files/folders.
- `cp <src> <dst>`: Copies files/folders.
- `head <file-path>`: Displays first few lines in the specified file.
- `tail <file-path>`: Displays last few lines in the specified file.
- `chmod`: Change permissions.
- `exit`: Logout and exit the terminal window.
- `top`: Command-line Task Manager.

Some Utility Commands and Tricks

- Utility Commands
  - `vi`, `emacs`, `nano`: Command line text file editors.
  - `awk`: Command line programmable manipulation of text-based data files, e.g. Excel-like calculations on columnar data.
  - `sed`: Stream editor. text-file transformations, e.g. find-replace.
  - `grep`: Search for files/folders with specified pattern in the name or in the content.
- Tricks
  - Use `<TAB>` for autocompletion.
  - Use `<Up Arrow>` and `<Down Arrow>` keys to browse through command history.
  - Use `man <command-name>` or `info <command-name>` or `<command-name> --help` to seek help on the specified command.

Software Stack on Ptolemy

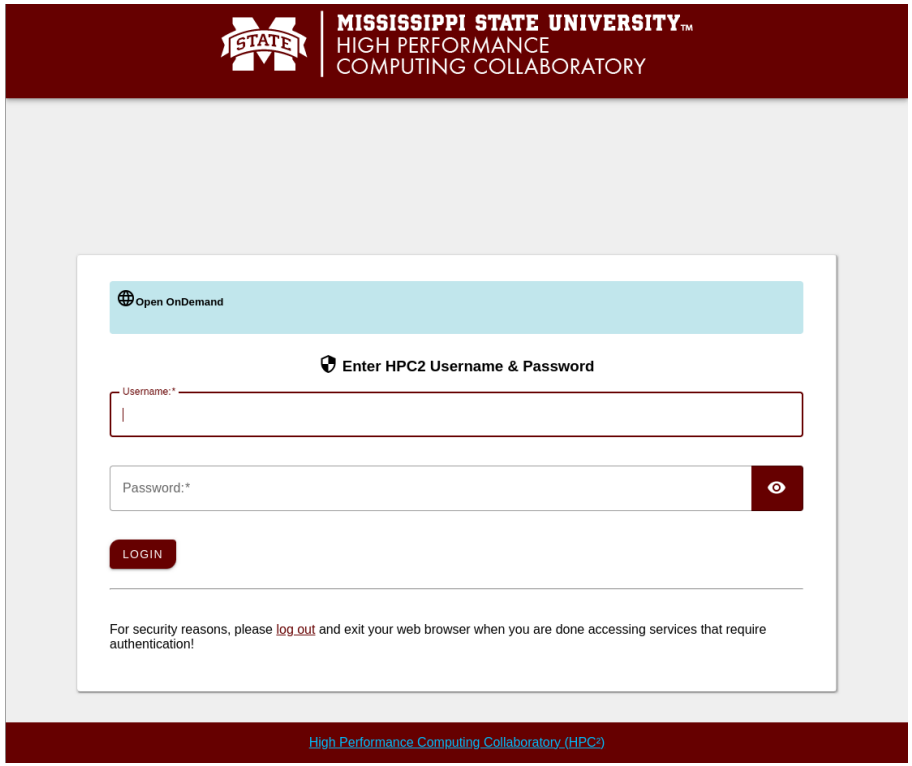
- The software stack on Ptolemy can be queried from command line using `module avail`.
- Use `module load <software-name>` to load the software for use.



- These software products can be used via their command line interfce (CLI). See the respective product documentation.
- Many compilers and libraries are available which you can use to compile your code or open source software.

Connect to Ptolemy via Open OnDemand

- What is Open OnDemand? A browser-based High Performance Computing portal.
- URL: `https://ptolemy-ood.arc.msstate.edu`
- Use ARC credentials to login.

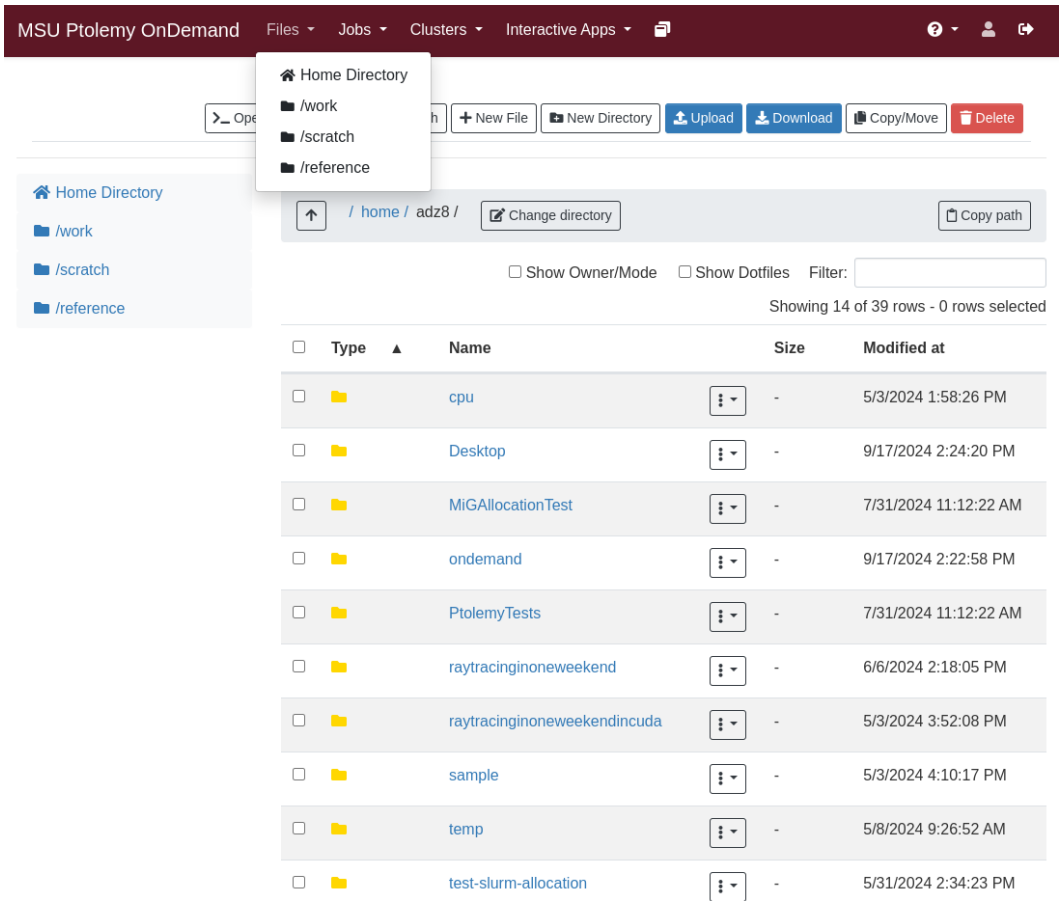


Connect to Ptolemy via Open OnDemand

- Main dashboard.

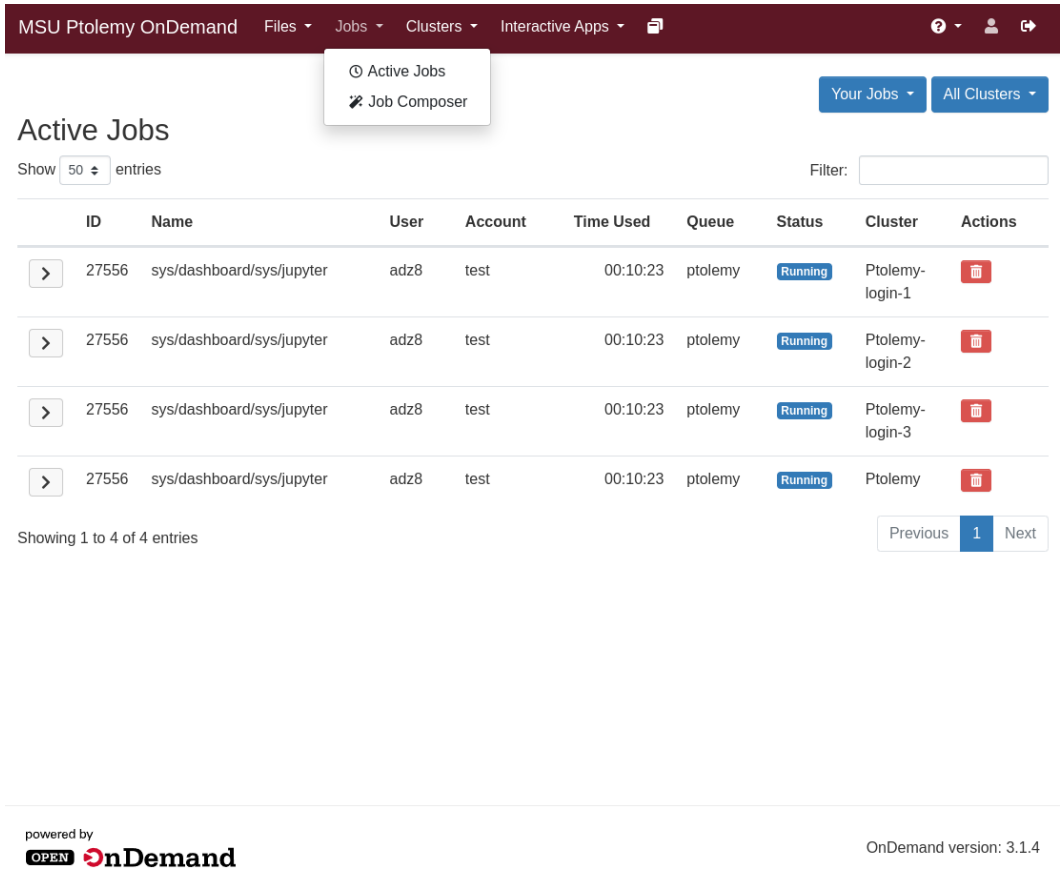
Connect to Ptolemy via Open OnDemand

- Can be used to browse folders, download/upload/copy/move data.
- Only for data **less than 1 GB in size**.



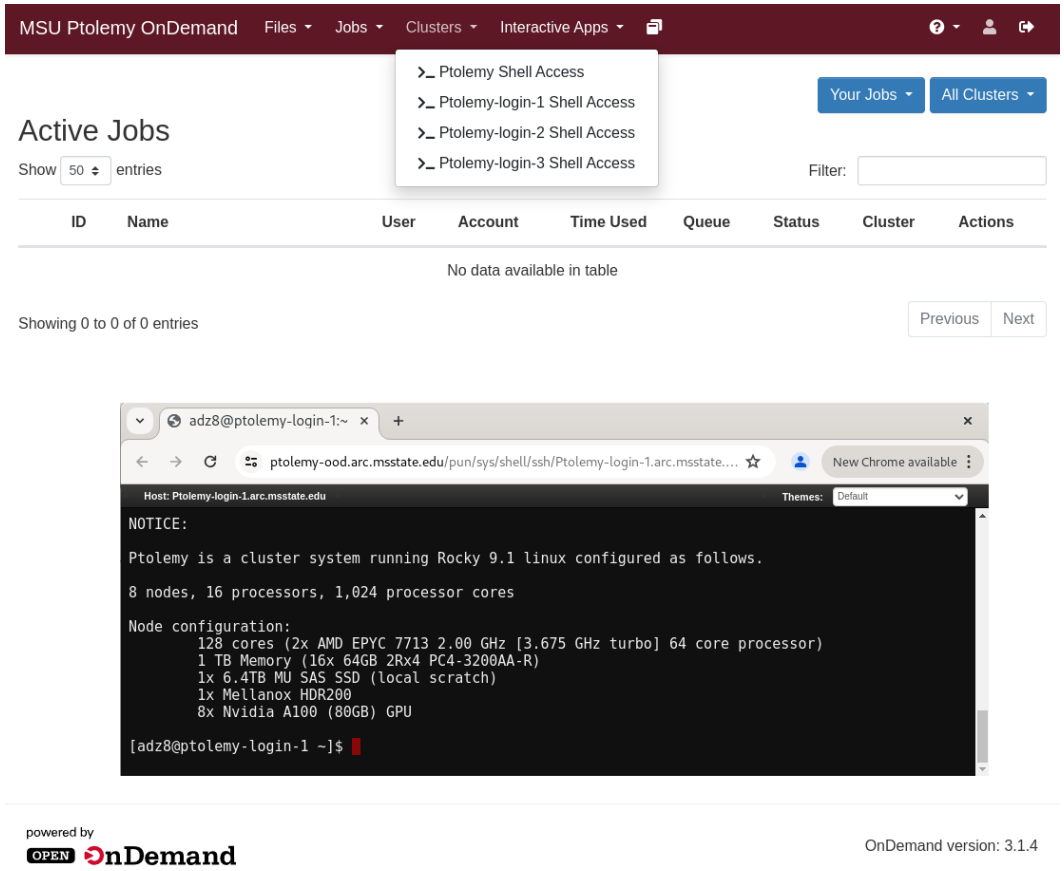
Connect to Ptolemy via Open OnDemand

- Can be used to monitor cluster jobs.



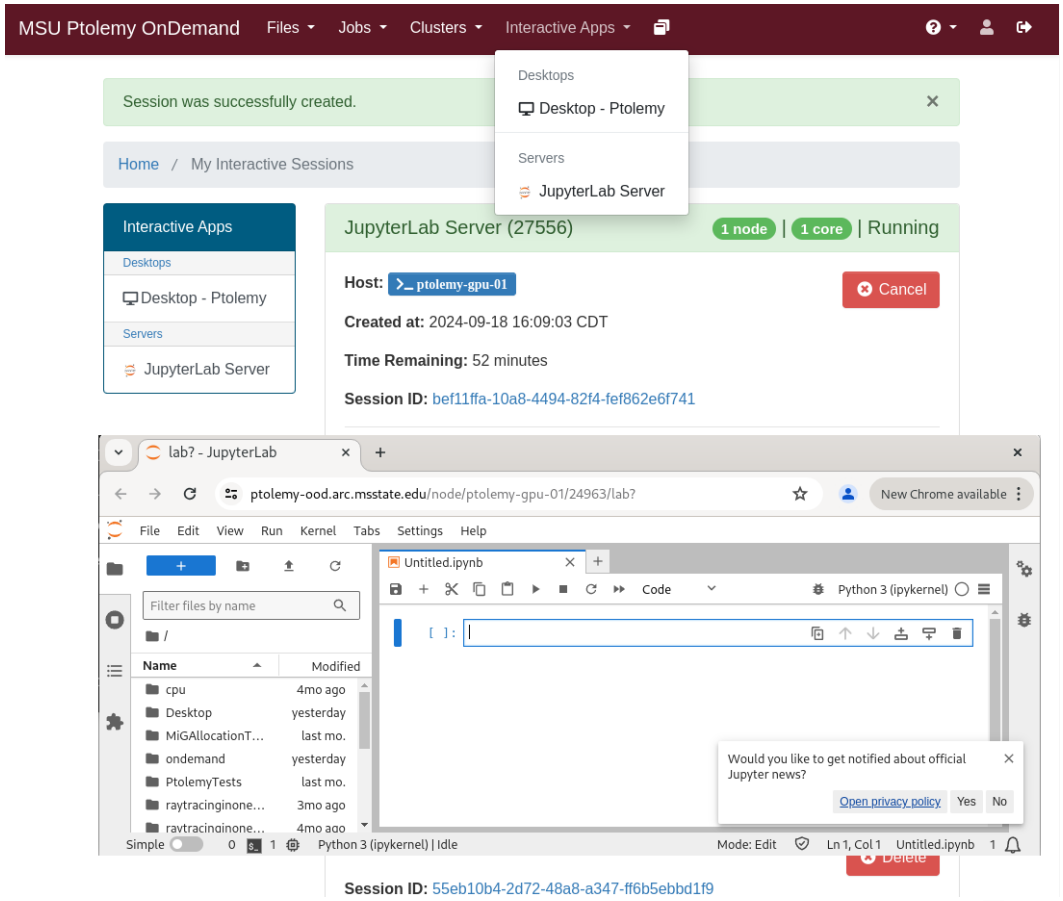
Connect to Ptolemy via Open OnDemand

- Can be used to obtain shell access.



Connect to Ptolemy via Open OnDemand

- Can be used to launch interactive applications.
- To request additional software, submit HelpDesk ticket at `help@hpc.msstate.edu`



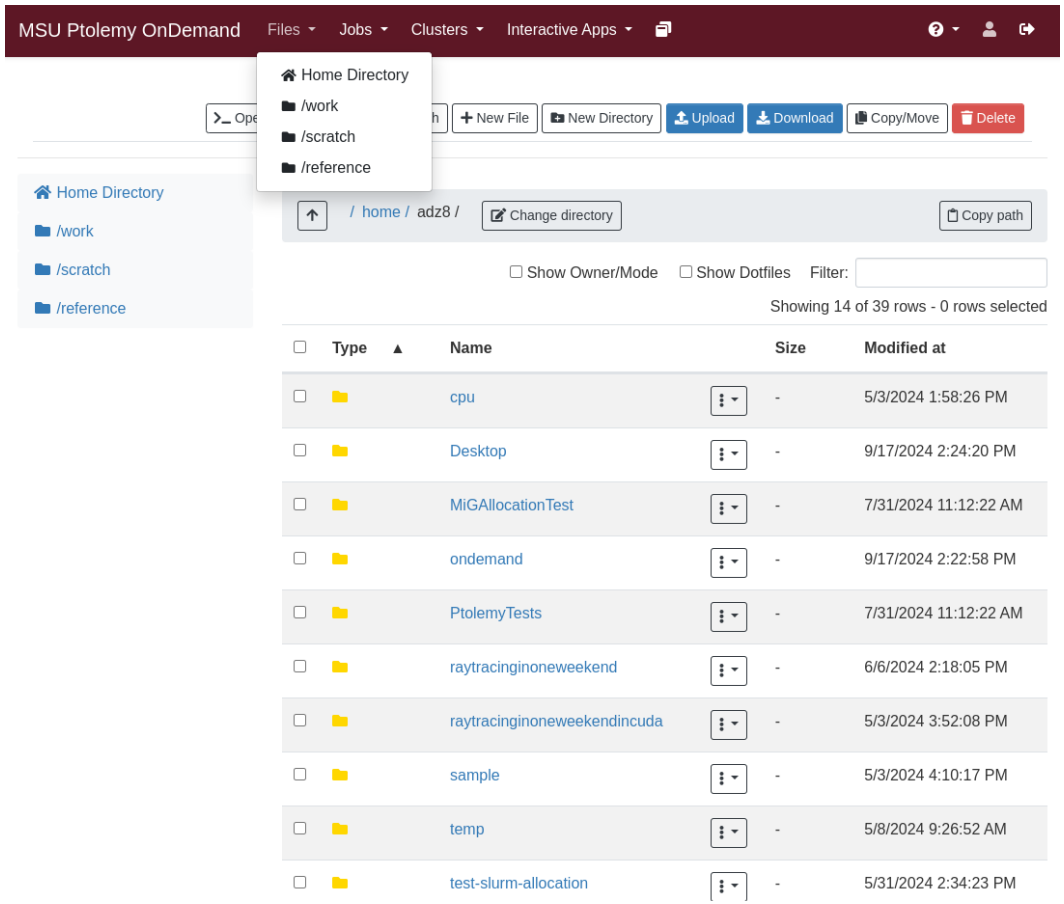
Outline

Storage Space

- Home Folder
  - Each user has a home folder `/home/<user-name>` (5GB quota).
- Scratch and Work Folder
  - Due to limited `$HOME` space, it is not suitable for parallel computations that read or write large data.
  - Instead, use
    - `/scratch/ptolemy/users/<user-name>` for temporary transient data.
    - `/work/ptolemy/users/<user-name>` for user specific data
    - `/work/ptolemy/projects/<project-name>` for project specific data
    - `/work/ptolemy/reference` for reference data sets
  - See **/work/README** and **/scratch/README** for detailed info.
  - `/scratch` is large, but older content is purged.
  - `/work` is not purged, but has limited disk quota. Submit a HelpDesk ticket for getting allocation here.
  - None of this data is backed up. So always remember to backup important files on your device.

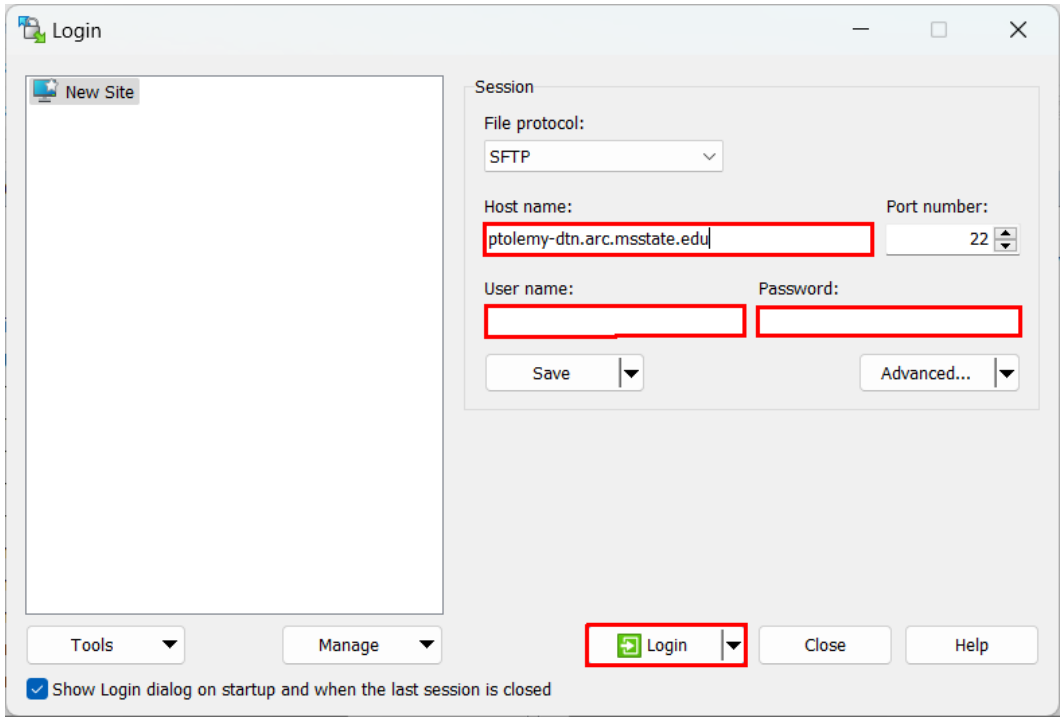
Transferring Data Between Your Device and Ptolemy

- Using Ptolemy Open OnDemand
- Use this method **only for data less than 1 GB**.



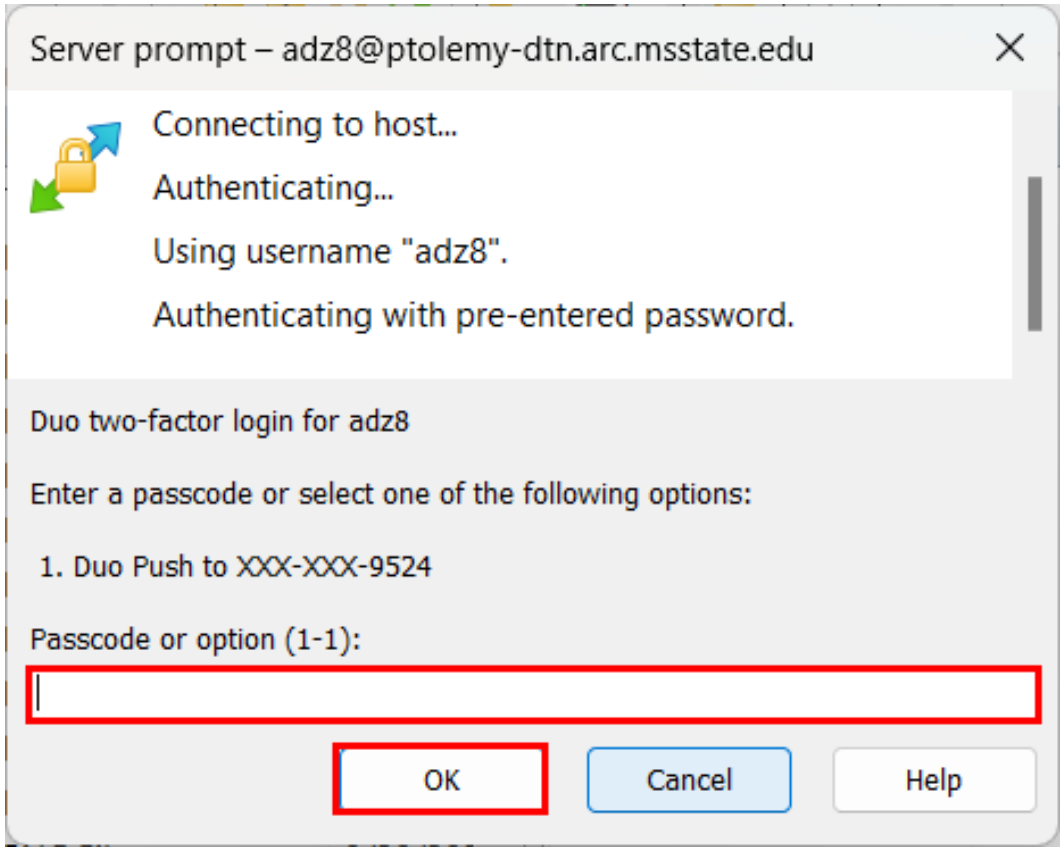
Transferring Data Between Your Device and Ptolemy

- Using SCP client such as WinSCP on Windows.



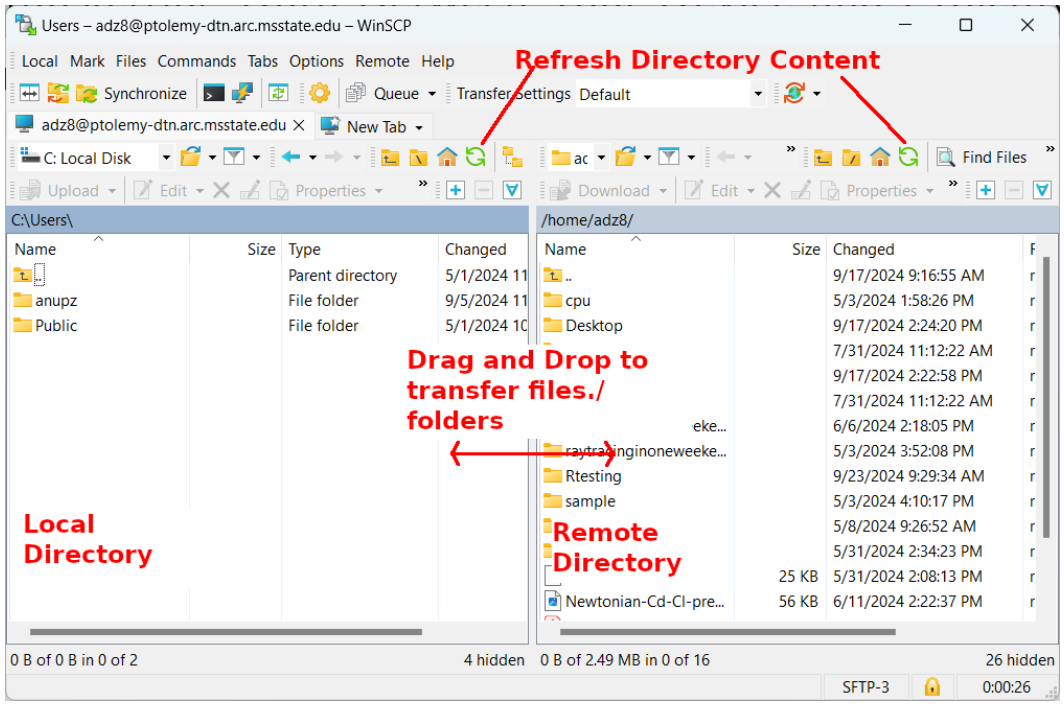
Transferring Data Between Your Device and Ptolemy

- Using SCP client such as WinSCP on Windows.



Transferring Data Between Your Device and Ptolemy

- Using SCP client such as WinSCP on Windows.



Transferring Data Between Your Device and Ptolemy

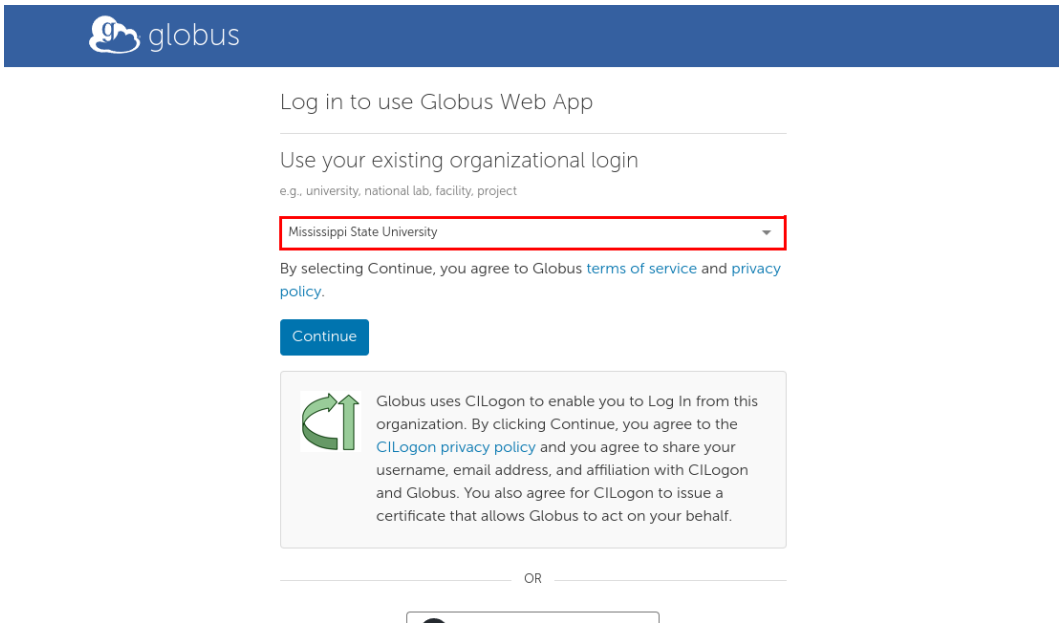
Using Globus Online

- URL: `https://www.globus.org/`

Transferring Data Between Your Device and Ptolemy

Using Globus Online

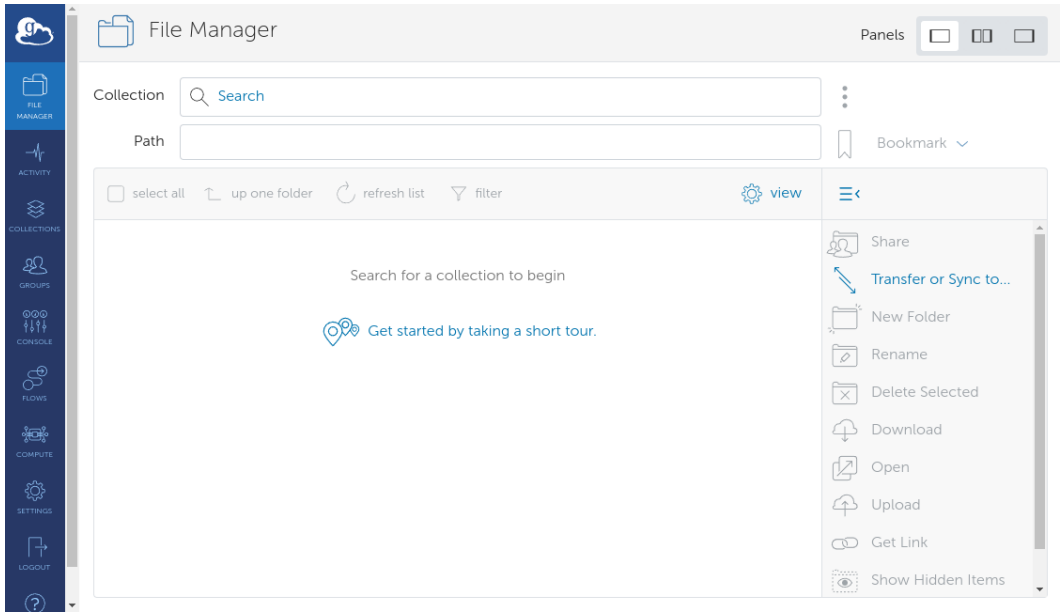
- Login using MSU NetID and password (Central Auth Service credentials), and 2FA.



Transferring Data Between Your Device and Ptolemy

Using Globus Online

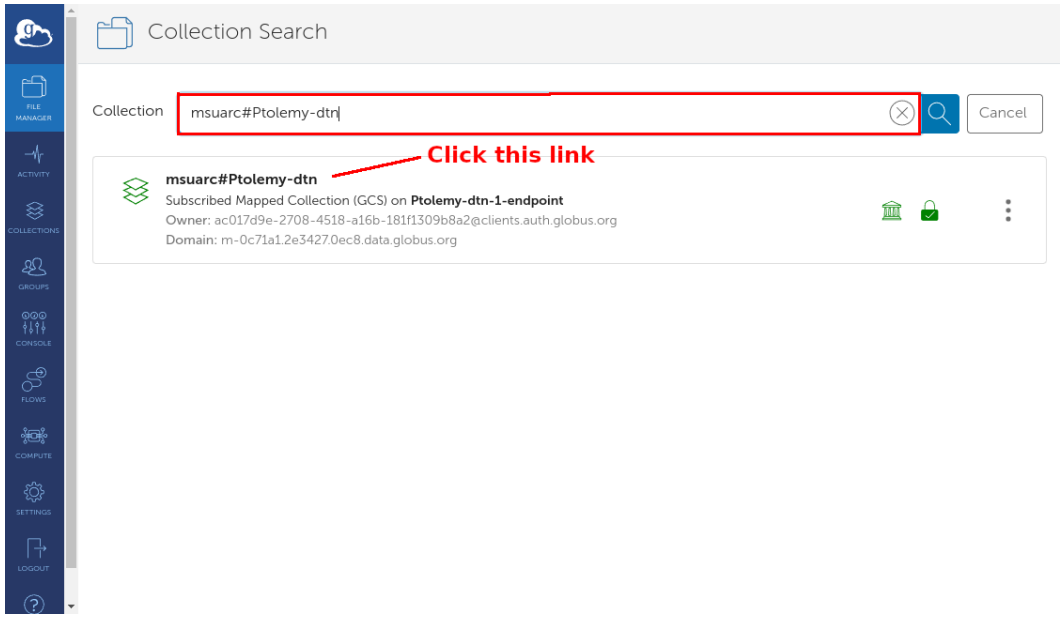
- File Manager Dashboard:



Transferring Data Between Your Device and Ptolemy

Using Globus Online

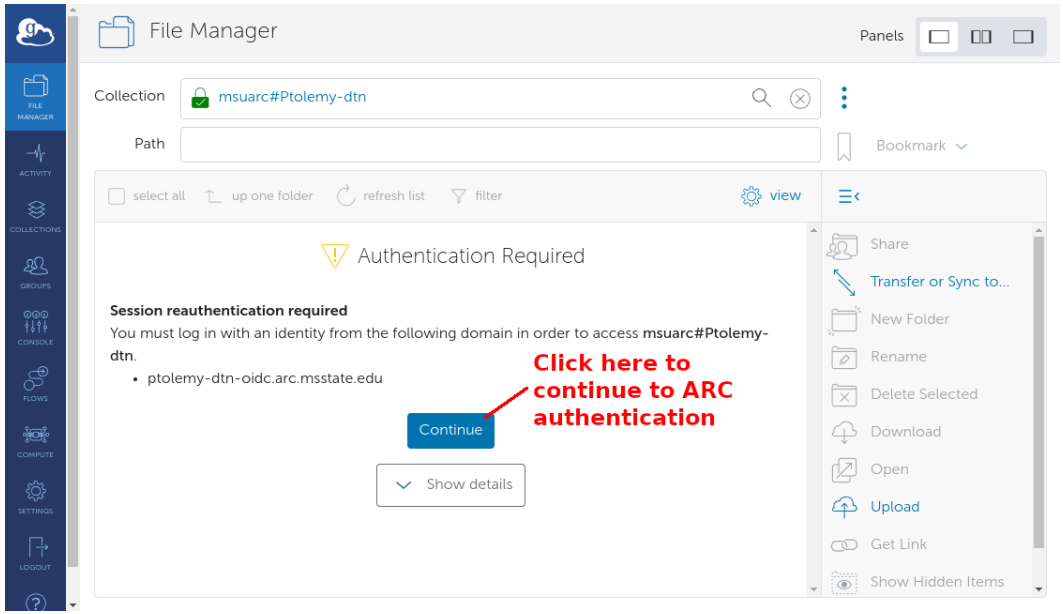
- Search for collection (a.k.a. end-point): `msuarc#Ptolemy-dtn`
- Click the link to the collection



Transferring Data Between Your Device and Ptolemy

Using Globus Online

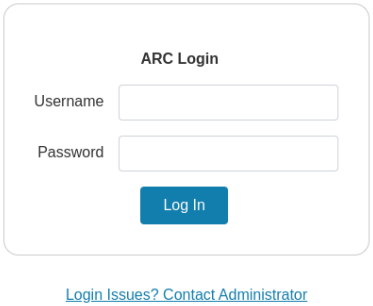
- It asks for authentication to access the end-point.



Transferring Data Between Your Device and Ptolemy

Using Globus Online

- Use ARC credentials.

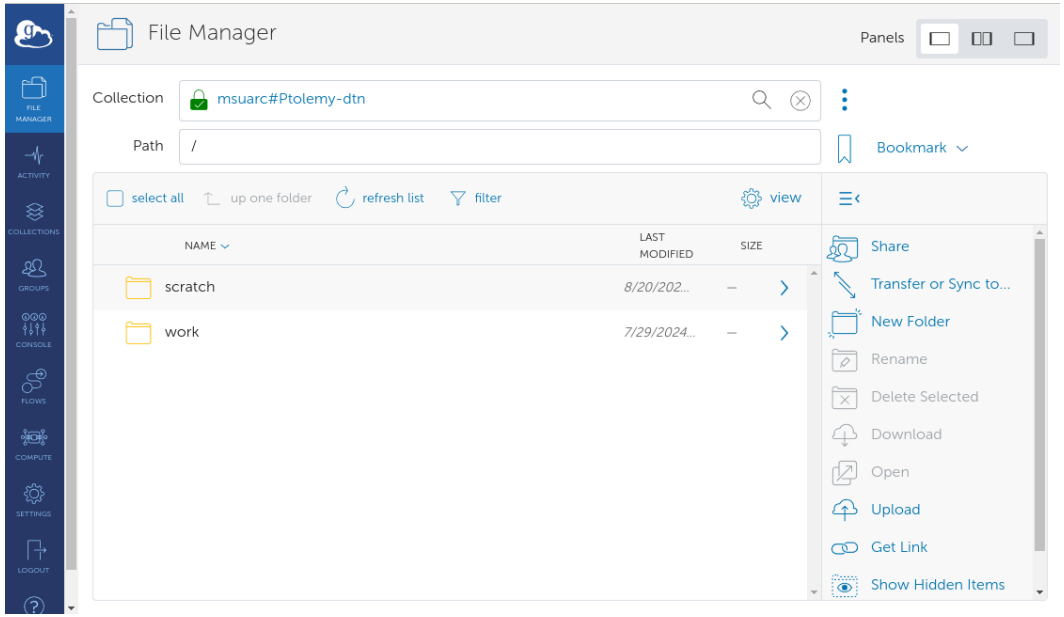


Use Ptolemy username and password.

Transferring Data Between Your Device and Ptolemy

Using Globus Online

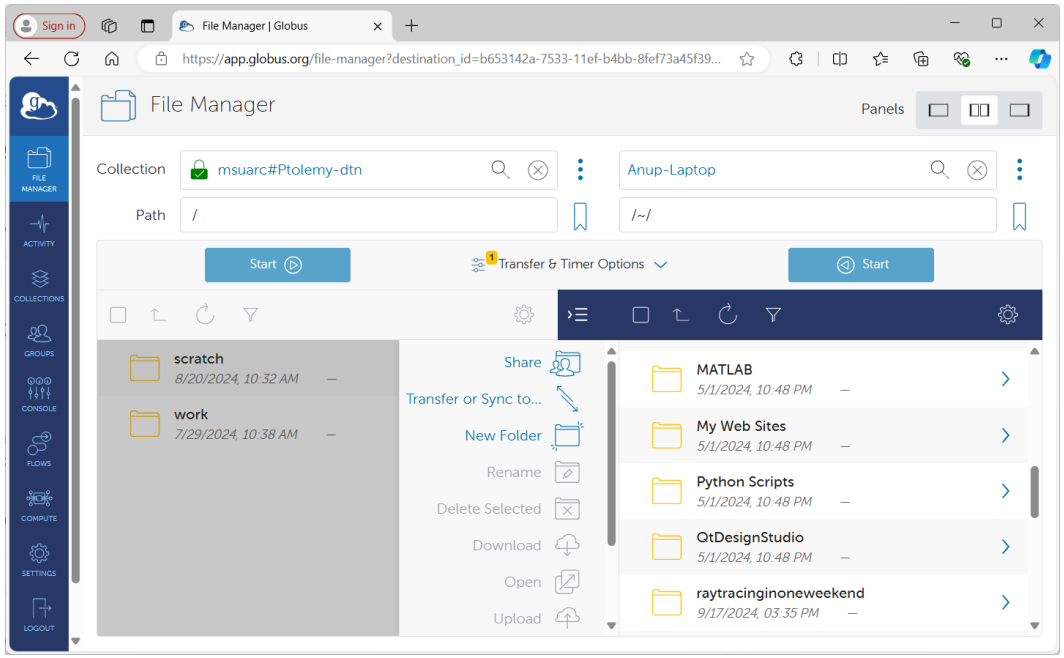
- Once connected to the end-point, folders can be browsed, data can be uploaded/downloaded.



Transferring Data Between Your Device and Ptolemy

Using Globus Online

- You can setup data-sync between Ptolemy and your device.
- Needs installation and configuration of Globus Personal Connect on your device:`https://www.globus.org/globus-connect-personal`.



Transferring Data Between Your Device and Ptolemy

Using SSH bash command line interface.

- Use `scp` or `rsync`
- Linux/MacOS: Use Terminal
- Windows: Use WSL/Cygwin/MSYS2 Terminal
- Syntax:
  - `scp -r <source> <destination>`
  - `rsync -avE <source> <destination>`
- Transfer from your workstation to Ptolemy
  - `<source>` is your workstation folder
  - `<destination>` is `<uname>@ptolemy-dtn.arc.msstate.edu:<scratch-folder>`
- Transfer from Ptolemy to your workstation
  - `<source>` is `<uname>@ptolemy-dtn.arc.msstate.edu:<scratch-folder>`
  - `<destination>` is your workstation folder

  Notice that the role of `<source>` and `<destination>` has changed.

Outline

Types of Nodes on Ptolemy

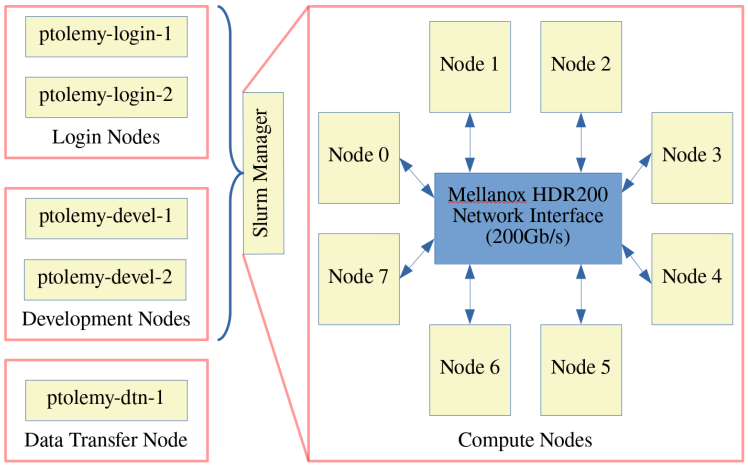


Figure: Types of nodes on Ptolemy

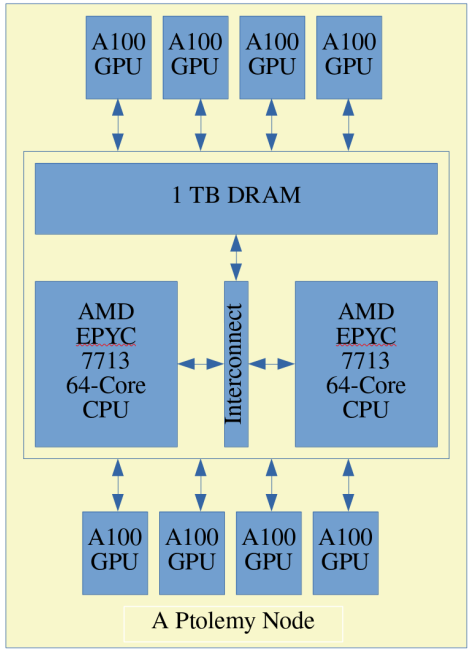


Figure: Schematic of a Compute Node

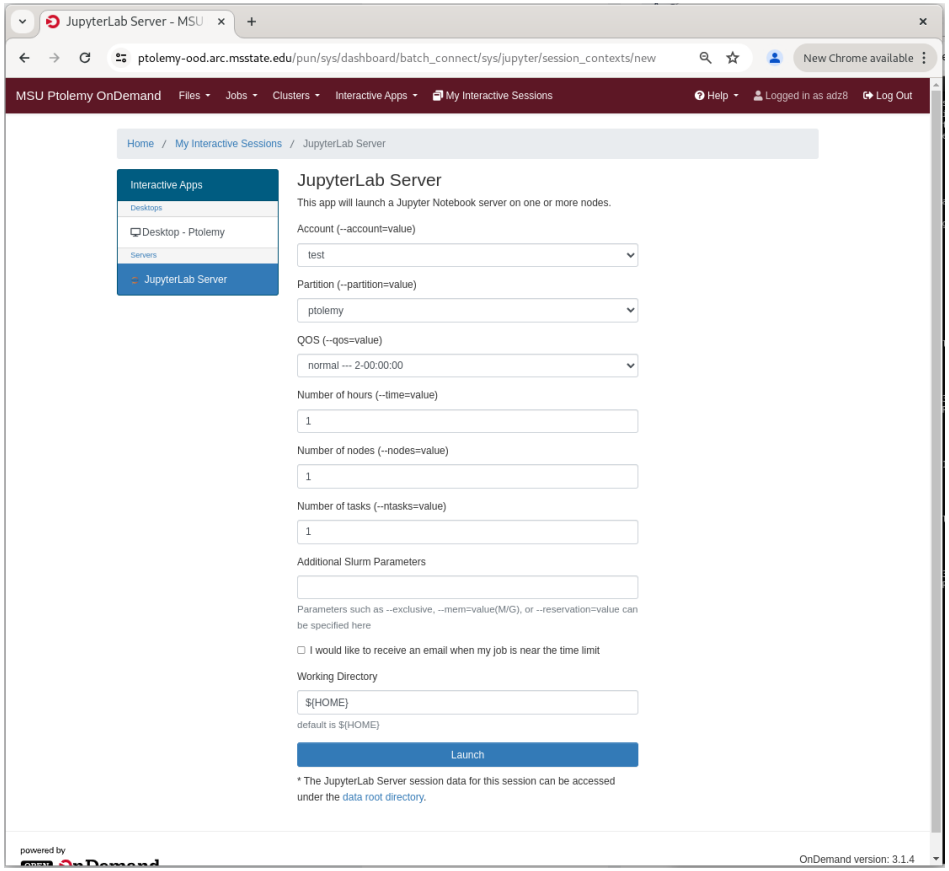
- **NEVER** run jobs on the login nodes - this is **PARAMOUNT**. Users that abuse this rule will have access to the cluster removed.

So, How to Run a Software on Ptolemy?

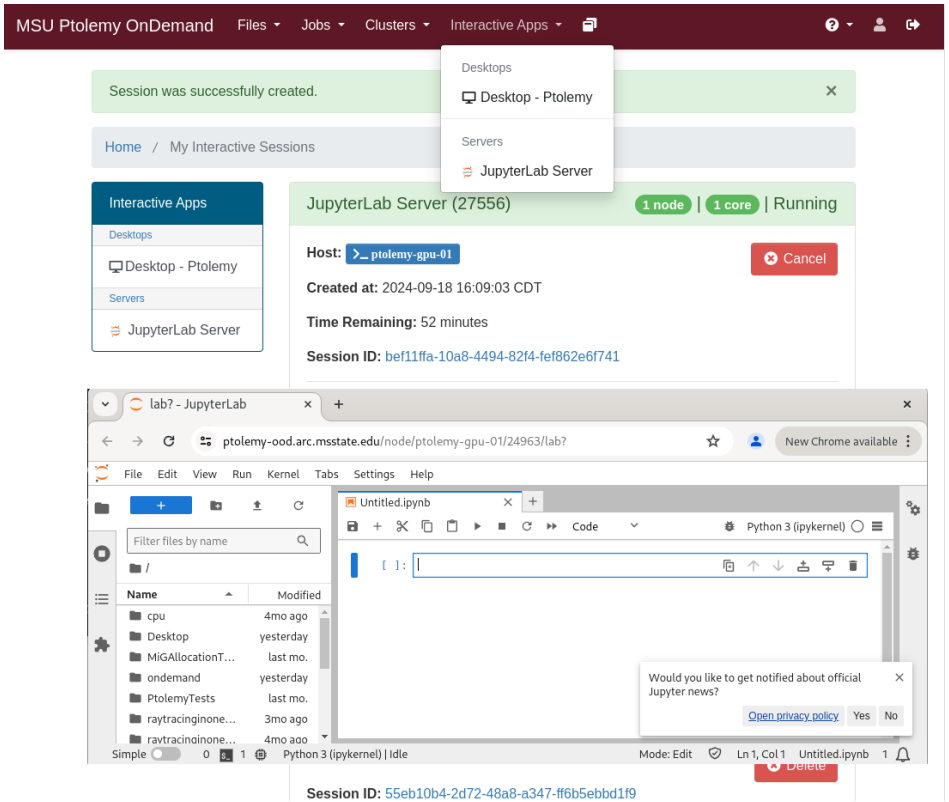
- Use Slurm to submit a compute job to the cluster.
- Slurm manages and allocates compute resources required for a compute job.
- There are two primary methods to run a software on Ptolemy are:
  - via Open OnDemand
  - via command line interface using Slurm
- Special cases:
  - Users can run SHORT DEVELOPMENT jobs on the devel nodes. But they must not monopolize the node's resources, and runs should be kept under 30 mins as a general rule.
  - Jobs are allowed on data transfer nodes via Slurm 'service' partition. Still requires Slurm, but allows automated data transfer.

Using Open OnDemand

Choose Slurm parameters:



Launch:



Using Command Line Slurm Submission

Two steps:

- Write a bash script for job submision

```
#!/bin/bash

#SBATCH --job-name=Test              # Name of the parallel job
#SBATCH --nodes=1                    # Number of nodes
#SBATCH --ntasks-per-node=1          # Number of cores per node
#SBATCH --partition=gpu-a100         # Name of the partition
#SBATCH --mem=10G                    # Amount of CPU memory requird per node
#SBATCH --account=test               # Account number
#SBATCH --gres=gpu:a100_1g.10gb:1    # Requests one Nvidia A100 GPU with 10 GB of memory.
#SBATCH --time=1:00:00               # Maximum wall time
#SBATCH --output=stdout-%j.out       # Output file (stdout), %j is job allocation number
#SBATCH --error=stderr-%j.out        # Error file (stderr), %j is job allocation number
#SBATCH --mail-type=ALL              # All types all emails (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<email-address> # Where to send the email

# Get a clean BASH environment
module purge

# Setup required software and BASH environment e.g.:
module load contrib cuda cuda-libs

# Command to start parallel program run, parallel job using
# nodes*ntasks-per-node CPU cores. e.g.:
srun ./convolutionFFT2D
```

- Use `sbatch` command to submit the job to Slurm

Slurm: Additional Commands

- To obtain the information on Slurm account (–account option) and QOS (–qos option):

```
sacctmgr show association where user=<user-name> \
format=cluster,account\%30,user,qos,maxjobs
```

- Obtain information of partitions and nodes managed by Slurm.

```
sinfo
```

- To submit the parallel job, use the `sbatch` command

```
sbatch <sbatch-script-file-name.sh>
```

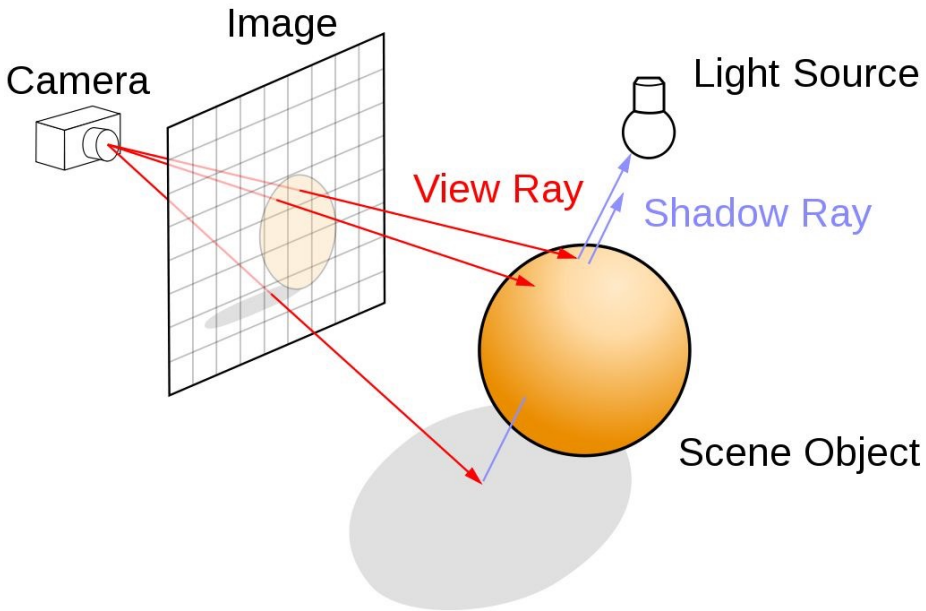
- To query the status of submitted job, use `squeue` command

```
squeue -u <user-name>
```

- To cancel a submitted job

```
scancel <job-id>
```

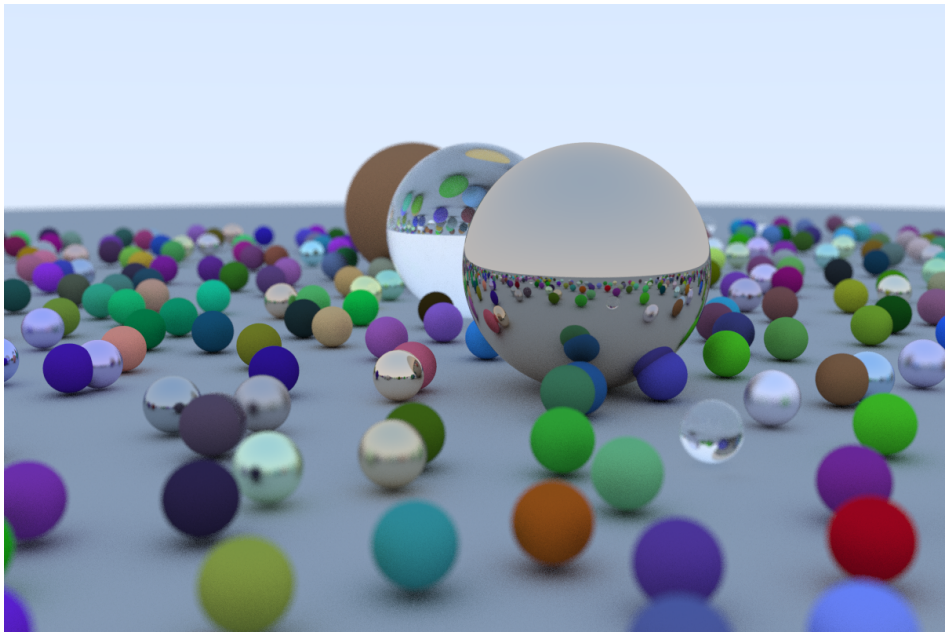
Ray Tracing Example - CPU vs GPGPU



1. Ray tracing is a technique to produce 2D images of 3D object model using the light transport mechanisms: absorption, reflection, refraction etc.
2. Calculation of the color of each pixel is an independent computation $\Longrightarrow$ embarassingly parallel computation.

Ray Tracing Example - CPU vs GPGPU

1 2



```
.
└── raytracinginoneweekend
    ├── cpu
    │   ├── camera.h
    │   ├── hitable.h
    │   ├── hitable_list.h
    │   ├── main.cc
    │   ├── Makefile
    │   ├── material.h
    │   ├── ray.h
    │   ├── README.md
    │   ├── sphere.h
    │   ├── submit.sh
    │   └── vec3.h
    └── cuda
        ├── camera.h
        ├── hitable.h
        ├── hitable_list.h
        ├── main.cu
        ├── Makefile
        ├── material.h
        ├── ray.h
        ├── README.md
        ├── sphere.h
        ├── stderr.txt
        ├── stdout.txt
        ├── submit.sh
        └── vec3.h
```

[1] Code taken from https://github.com/rogerallen/raytracinginoneweekendincuda.git

[2] Copy the code from /scratch/ptolemy/users/adz8/raytracinginoneweekend

Ray Tracing Example - CPU vs GPGPU

- Copy source code:

```
DIR=/scratch/ptolemy/users/<user-name>
cp -R /scratch/ptolemy/users/adz8/raytracinginoneweekend ${DIR}
```

- Load software required to compile the code:

```
module load gcc cuda
```

- Compile CPU code and submit to cluster:

```
cd ${DIR}/raytracinginoneweekend/cpu
make
sbatch submit.sh
```

- Compile GPU code and submit to cluster:

```
cd ${DIR}/raytracinginoneweekend/cuda
make
sbatch submit.sh
```

Question: How much faster is the CUDA version?

Using OCI Containers

- OCI containers are standalone, executable package of a software that includes all the library dependencies needed to run the software.
- Where to find OCI containers?
  - Use following commands in SSH terminal to print the directory location where containers are stored:

```
module load apptainer
echo $CONTAINERS
```

  - If not available in the `$CONTAINERS` directory, submit a HelpDesk ticket (`help@hpc.msstate.edu`) to request the required container.
  - You can build your own container. Submit a HelpDesk ticket to get assistance.

Using OCI Containers

- Write batch submission file:

```
#!/bin/bash

# The usual SBATCH options

# Get a clean environment
module purge

# Load apptainer
module load apptainer

# Launch the container
srun apptainer exec ${CONTAINERS}/<container-file-name.sif> <exec-command>
```

- Use sbatch to submit

Questions?