HIGH PERFORMANCE RESEARCH COMPUTING

HPRC Primer

Using the Slurm Scheduler on the ACES Cluster

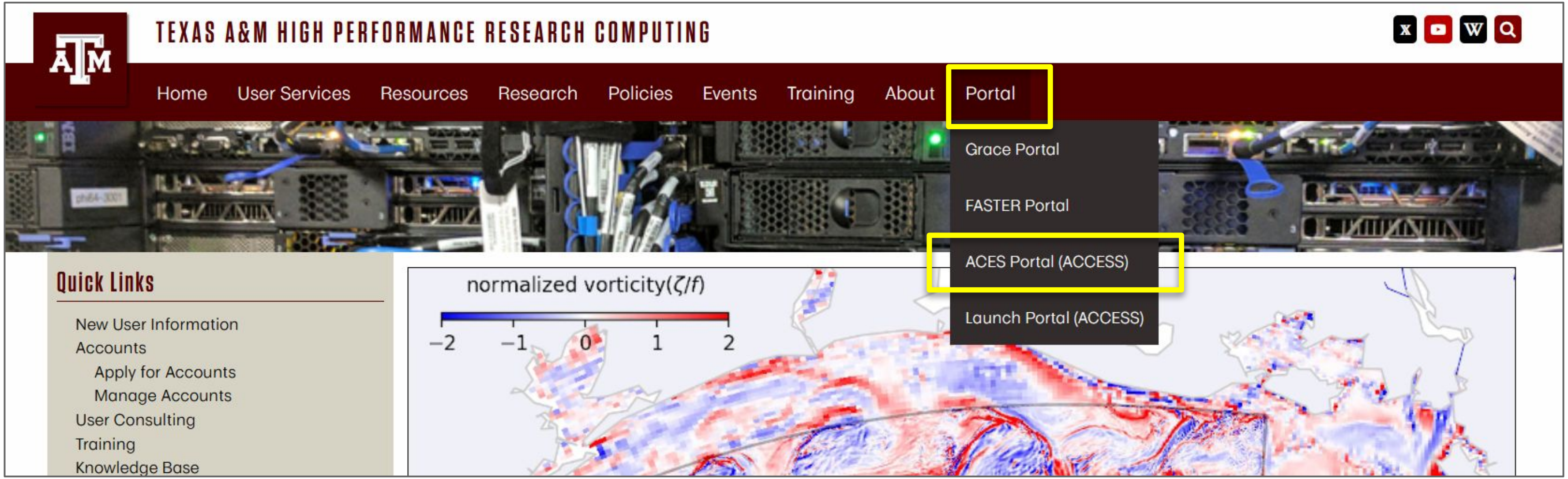
February 20, 2026

Victor Valdez

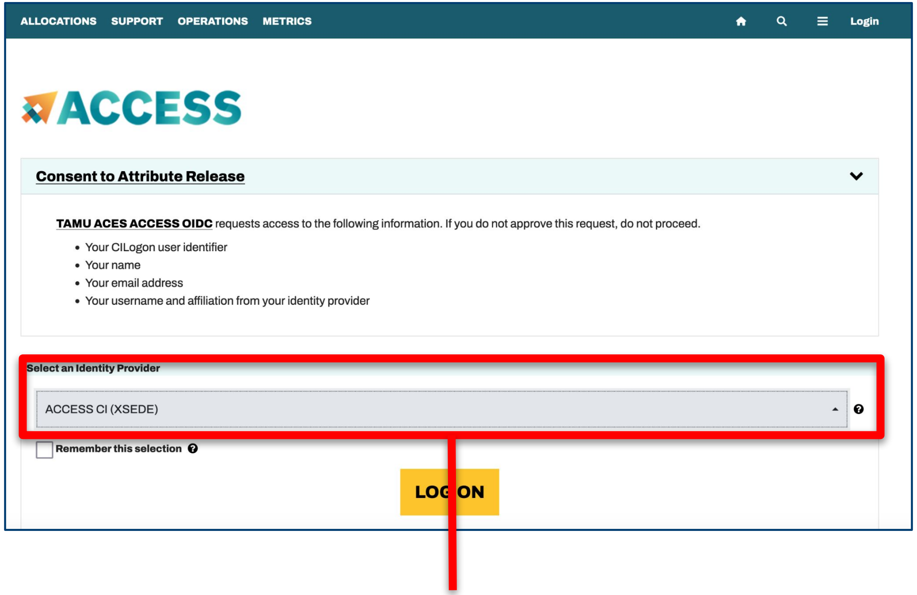
High Performance Research Computing
DIVISION OF RESEARCH

Accessing the ACES Portal

- HPRC webpage: hprc.tamu.edu
- Aces portal shortcut: portal-aces.hprc.tamu.edu
- Requires an ACCESS ID!



Accessing ACES via the Portal (ACCESS)



Consent to Attribute Release

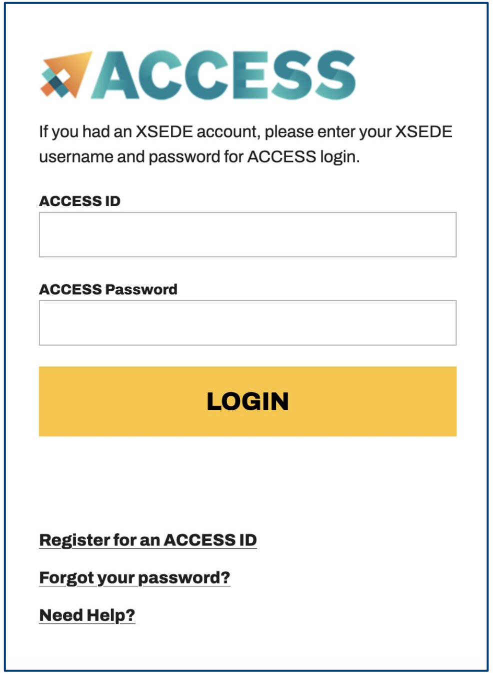
TAMU ACES ACCESS OIDC requests access to the following information. If you do not approve this request, do not proceed.

- Your CILogon user identifier
- Your name
- Your email address
- Your username and affiliation from your identity provider

Select an Identity Provider: ACCESS CI (XSEDE)

□ Remember this selection

Select the Identity Provider appropriate for your account.



If you had an XSEDE account, please enter your XSEDE username and password for ACCESS login.

ACCESS ID

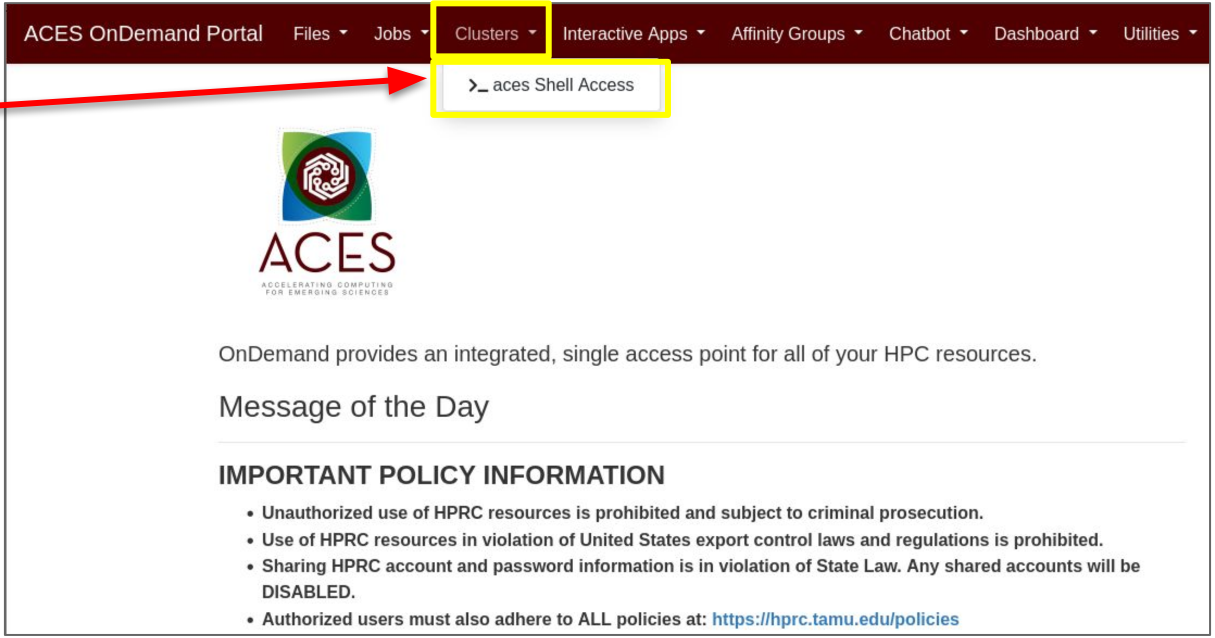
ACCESS Password

Log-in using your ACCESS or institutional credentials.

Shell Access via the HPRC Portal

Once in the Portal, select at the top: "Clusters" → "aces Shell Access"

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



OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

IMPORTANT POLICY INFORMATION

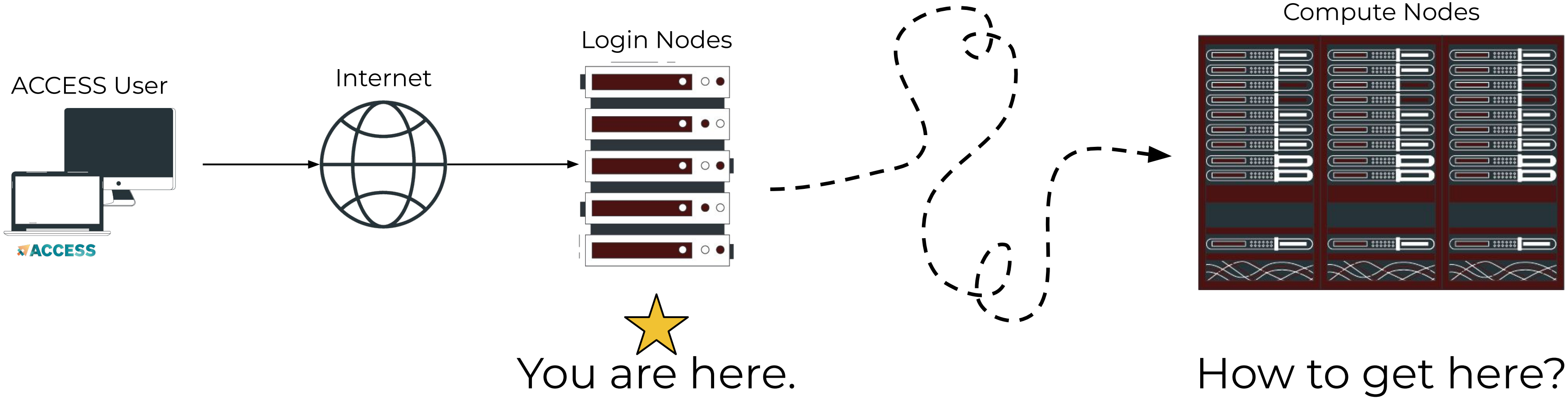
- **Unauthorized use of HPRC resources is prohibited and subject to criminal prosecution.**
- **Use of HPRC resources in violation of United States export control laws and regulations is prohibited.**
- **Sharing HPRC account and password information is in violation of State Law. Any shared accounts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

Hands-On Activity - 2 Minutes

- Connect to ACES now through the portal using portal-aces.hprc.tamu.edu
- Get a shell on the ACES cluster from the Clusters menu

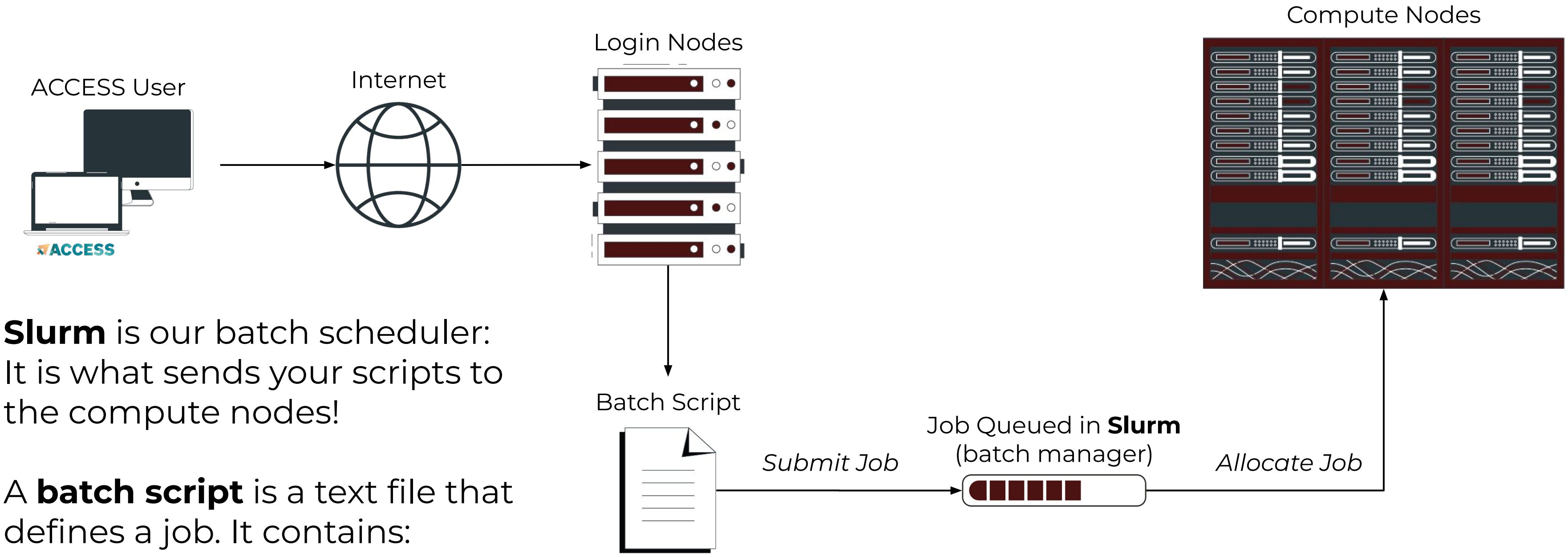
What is the hostname of the machine you connected to?

Batch Computing on HPRC Clusters



- Types of nodes:
    - Login node - a shared machine for light editing
    - Compute node - an allocated machine for heavy computation

Batch Computing on HPRC Clusters



Slurm is our batch scheduler: It is what sends your scripts to the compute nodes!

A **batch script** is a text file that defines a job. It contains:

1. software commands *and*
2. batch parameters

Basic Slurm

Sample Job Script Structure

```
#!/bin/bash

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample1
#SBATCH --time=01:30:00
#SBATCH --ntasks-per-node=48
#SBATCH --nodes=2
#SBATCH --mem=488G
#SBATCH --output=Example1Out.%j
#SBATCH --partition=gpu
#SBATCH --gres=gpu:h100:1

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

# load required module(s)
module purge
module load GCCcore/13.3.0
module load Python/3.12.3
module load CUDA/12.6.0

# Run your program
python my_gpu_script.py
```

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

These parameters describe your job to the job scheduler.
The lines starting with #SBATCH are NOT just comments!
Bash ignores them, but Slurm does not!
See the Knowledge Base for more info.

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

Submit a Job and Check Job Status

Submit job

```
sbatch example01.slurm
```

```
sbatch: Job submitted using account 1234567891011
sbatch: Current account balance = 9794.371667
Submitted batch job 982272
```

Key:

```
User inputs in green
```

```
Terminal output in blue
```

Note the job ID!

Check status

```
squeue --me
```

```
JOBID   NAME     USER     PARTITION  NODES  CPUS  STATE    TIME        TIME_LEFT  START_TIME          REASON  NODELIST
982272  jobname  someuser  cpu       1      96    RUNNING  0:15        23:59:45   2025-03-06T11:49    None     ac058
6853257 jobname  someuser  xlong     2      96    RUNNING  3-07:36:56  16:23:04   2023-01-23T17:27:2  None   c[523-524]
```

Hands-On Activity- 10 mins

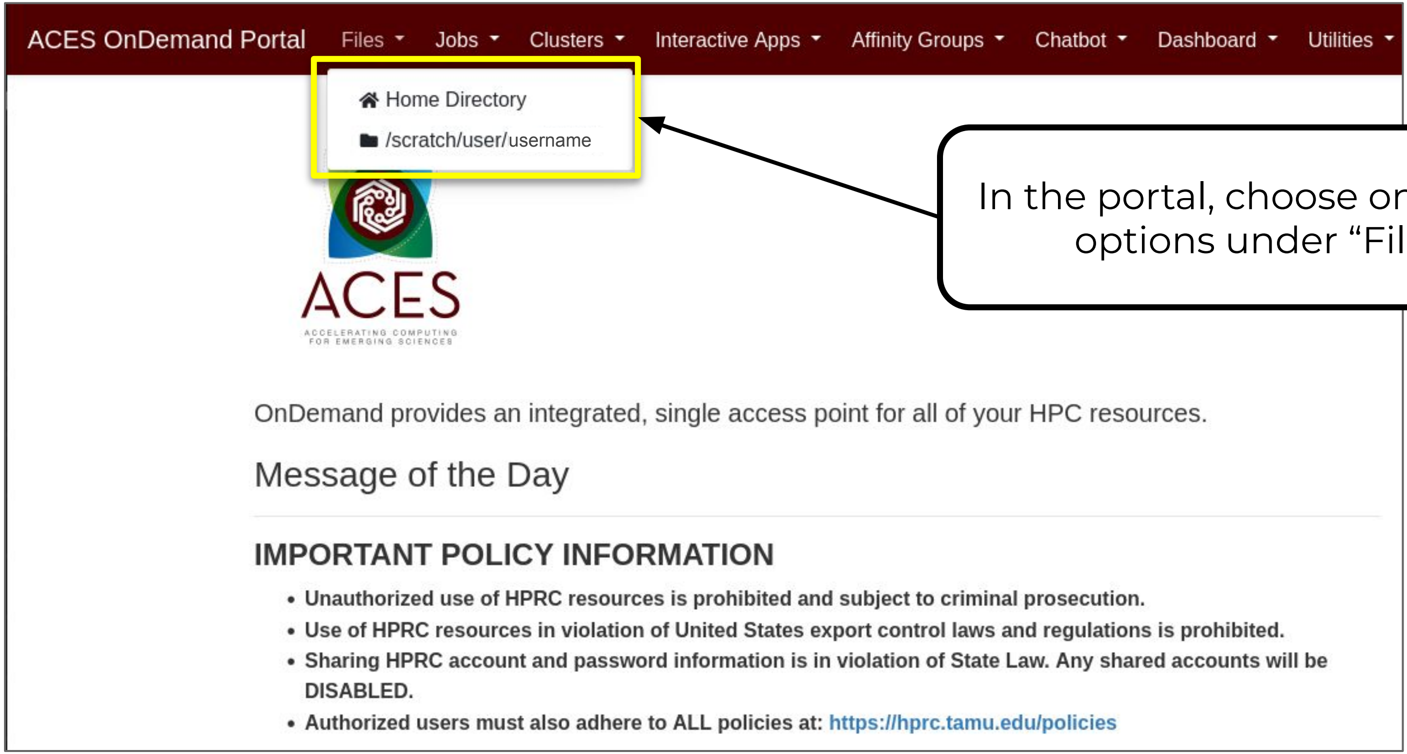
There are example files located at **`/scratch/training/slurm_scheduler`**

1. Copy **`hello_world.py`** and **`hello_world.slurm`** to your scratch directory
2. Submit the `.slurm` file using **`sbatch`**.
3. Check that the job is running in a Slurm queue with **`squeue`**.
4. Check the contents of the output file.
5. Edit the batch file and submit again. Check the new output for changes.

The following slides demonstrate how to move and edit the files without using the command line.

(Hands-on Activity Detail)

Open the File Navigator



OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

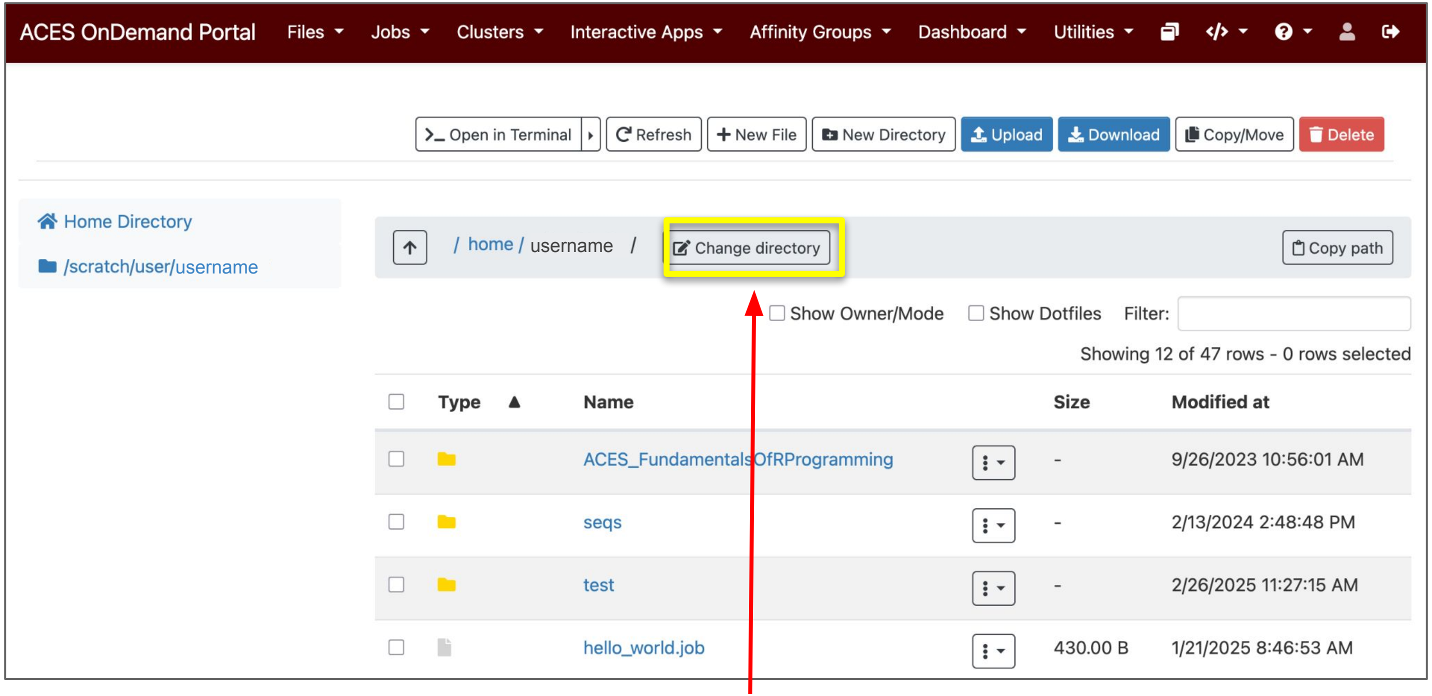
IMPORTANT POLICY INFORMATION

- **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 State Law. Any shared accounts will be DISABLED.**
- **Authorized users must also adhere to ALL policies at: https://hprc.tamu.edu/policies**

In the portal, choose one of the options under “Files”

(Hands-on Activity Detail)

Navigating to Training Directory



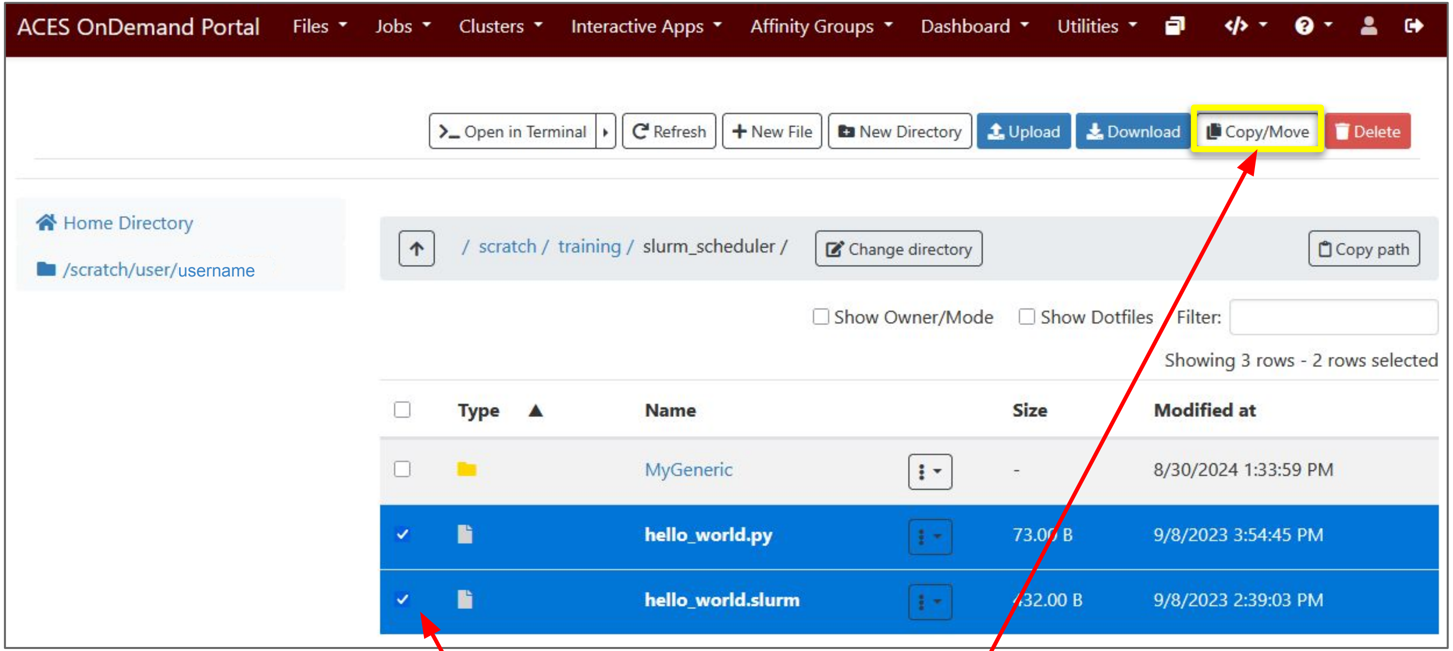
Showing 12 of 47 rows - 0 rows selected

□	Type ▲	Name		Size	Modified at
□	📁	ACES_FundamentalsOfRProgramming	⋮ ▾	-	9/26/2023 10:56:01 AM
□	📁	seqs	⋮ ▾	-	2/13/2024 2:48:48 PM
□	📁	test	⋮ ▾	-	2/26/2025 11:27:15 AM
□	📄	hello_world.job	⋮ ▾	430.00 B	1/21/2025 8:46:53 AM

Click on "change directory" and type **/scratch/training/slurm_scheduler**

(Hands-on Activity Detail)

Copy the Example Files



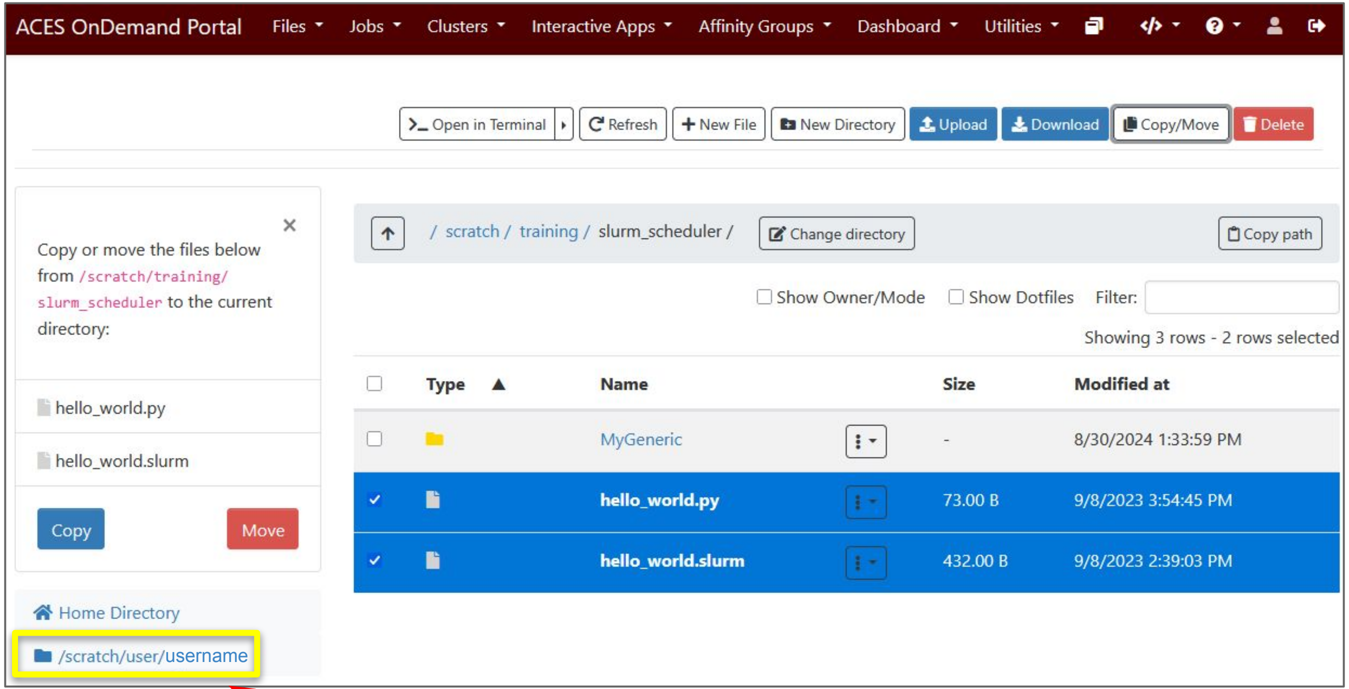
Showing 3 rows - 2 rows selected

☐	Type ▲	Name	Size	Modified at
☐		MyGeneric	-	8/30/2024 1:33:59 PM
☑		hello_world.py	73.00 B	9/8/2023 3:54:45 PM
☑		hello_world.slurm	432.00 B	9/8/2023 2:39:03 PM

Check box the two files and click Copy/Move

(Hands-on Activity Detail)

Move to Scratch



Copy or move the files below from /scratch/training/slurm_scheduler to the current directory:

- hello_world.py
- hello_world.slurm

/ scratch / training / slurm_scheduler /

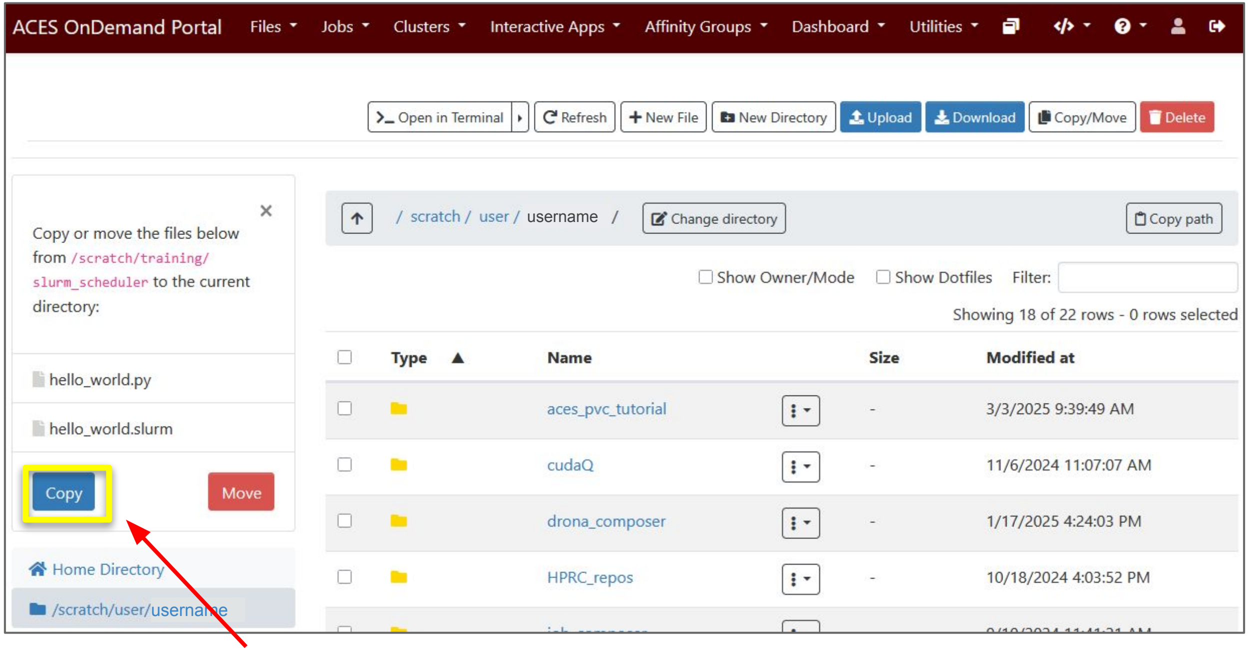
□ Show Owner/Mode □ Show Dotfiles Filter:

Showing 3 rows - 2 rows selected

□	Type ▲	Name	Size	Modified at
□		MyGeneric	-	8/30/2024 1:33:59 PM
☑		hello_world.py	73.00 B	9/8/2023 3:54:45 PM
☑		hello_world.slurm	432.00 B	9/8/2023 2:39:03 PM

Select the directory to copy to

Paste the Example Files



Copy or move the files below from `/scratch/training/slurm_scheduler` to the current directory:

hello_world.py

hello_world.slurm

/ scratch / user / username /

□ Show Owner/Mode □ Show Dotfiles Filter:

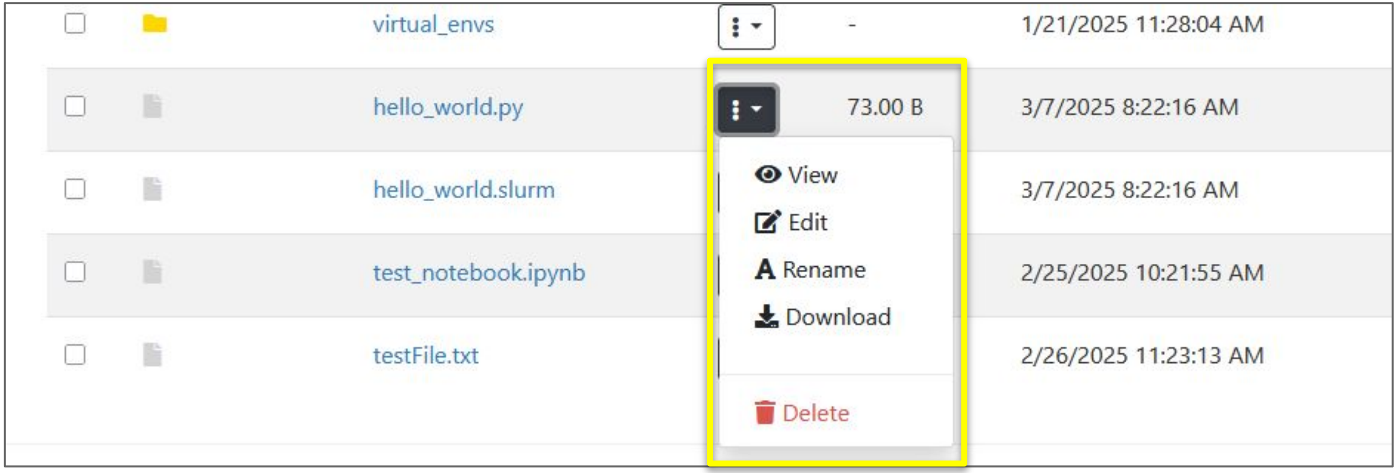
Showing 18 of 22 rows - 0 rows selected

□	Type ▲	Name	Size	Modified at
□		aces_pvc_tutorial	-	3/3/2025 9:39:49 AM
□		cudaQ	-	11/6/2024 11:07:07 AM
□		drona_composer	-	1/17/2025 4:24:03 PM
□		HPRC_repos	-	10/18/2024 4:03:52 PM

The files on the right will change to show the directory you chose. Hit "Copy" to actually copy the files to that directory.

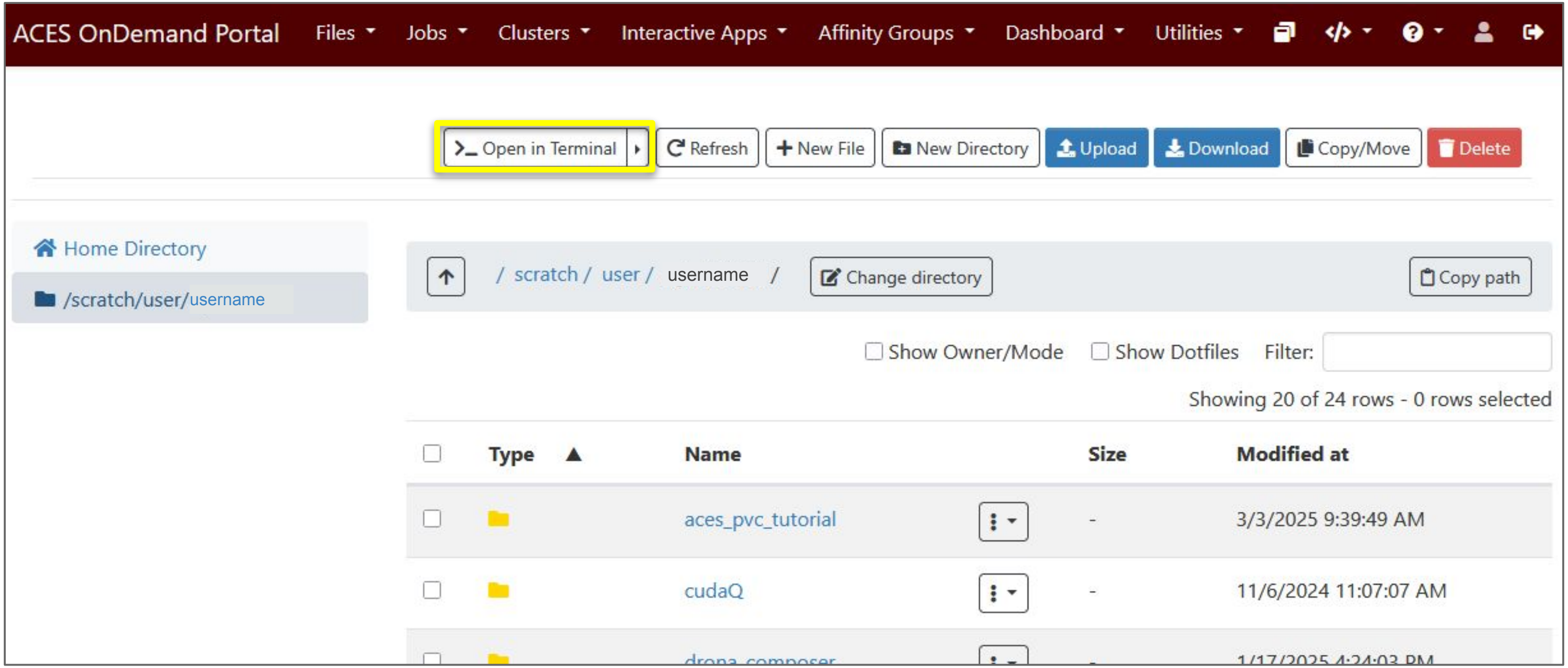
(Hands-on Activity Detail)

View and Edit the Example Files



		Name		Size	Date
□	📁	virtual_envs	⋮ ▾	-	1/21/2025 11:28:04 AM
□	📄	hello_world.py	⋮ ▾	73.00 B	3/7/2025 8:22:16 AM
□	📄	hello_world.slurm			3/7/2025 8:22:16 AM
□	📄	test_notebook.ipynb			2/25/2025 10:21:55 AM
□	📄	testFile.txt			2/26/2025 11:23:13 AM

Open a Terminal



/ scratch / user / username /

□ Show Owner/Mode □ Show Dotfiles Filter:

Showing 20 of 24 rows - 0 rows selected

□	Type ▲	Name		Size	Modified at
□	📁	aces_pvc_tutorial	⋮ ▾	-	3/3/2025 9:39:49 AM
□	📁	cudaQ	⋮ ▾	-	11/6/2024 11:07:07 AM
□	📁	drona_composer	⋮ ▾	-	1/17/2025 4:24:03 PM

Hands on Activity- Cont.

If you haven’t already:

1. Investigate the two example files
   - Make a small edit to personalize. (e.g., “Hello MyName”)
2. Submit the batch file on the command line using:
   ```
   sbatch hello_world.slurm
   ```
3. Check the job status using:
   ```
   squeue --me
   ```
4. Once the job is completed, inspect the output file:
   ```
   cat hello_world_log.<job_id>
   ```

Job Script Details

Important Batch Job Parameters

Slurm	Comment
`#SBATCH --time=HH:MM:SS`	Specifies the time limit for the job. Must specify seconds SS on ACES
`#SBATCH --ntasks=x`	Total number of tasks (cores) for the job.
`#SBATCH --ntasks-per-node=xx`	Specifies the maximum number of tasks (cores) to allocate per node

`#SBATCH --mem=xxxxM`
or
`#SBATCH --mem=xG`
	Sets the maximum amount of memory (MB) per *node*.
	G for GB is supported on ACES

`#SBATCH --nodes=x`	Specifies the number of nodes to use

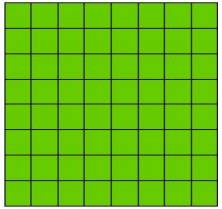
(These go in your job script file)

Mapping Jobs to Cores per Node on ACES

A.

🟩 allocated cores

□ unused cores

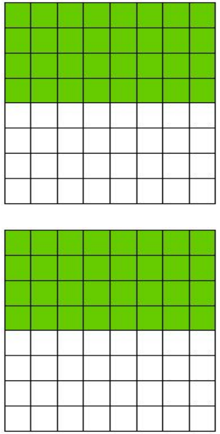


96 cores on
1 compute node

```
#SBATCH --ntasks=96
#SBATCH --ntasks-per-node=96
```

Preferred Mapping
(if applicable)

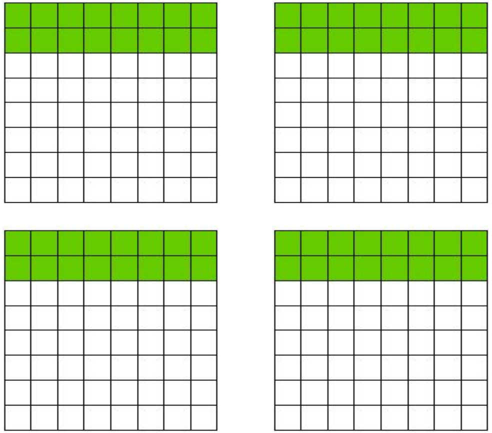
B.



96 cores on
2 compute nodes

```
#SBATCH --ntasks=96
#SBATCH --ntasks-per-node=48
```

C.



96 cores on
4 compute nodes

```
#SBATCH --ntasks=96
#SBATCH --ntasks-per-node=24
```

Pop Quiz

```
#SBATCH --job-name=JobExample2
#SBATCH --time=48:00:00
#SBATCH --ntasks=96
#SBATCH --ntasks-per-node=24
#SBATCH --nodes=4
#SBATCH --mem=48G
#SBATCH --output=stdout.%J
#SBATCH --error=stderr.%J
```

How many cores is this job requesting?

A. 1024
B. 96
C. 960
D. 4

Job Environment Variables

Each job has access to several self-referential variables:

- **`$SLURM_JOBID`** = job id
- **`$SLURM_SUBMIT_DIR`** = directory from where job was submitted
- **`$TMPDIR`** = high speed 1.5TB disk on the job's compute node

You can also use non-Slurm variables such as:

- **`$HOME`** = /home/username
- **`$SCRATCH`** = /scratch/user/username

https://hprc.tamu.edu/kb/Helpful-Pages/Batch-Translation/#environment-variables

Job Management Details

Job Submission and Tracking

Slurm	Description
`sbatch jobfile`	Submit jobfile to batch system
`squeue [--me] [-j job_id]`	List jobs
`scancel job_id`	Kill a job
`sacct -X -j job_id`	Show information for a job (can be when job is running or recently finished)
`sacct -X -S YYYY-MM-DD`	Show information for all of your jobs since YYYY-MM-DD

(These are command-line commands, not part of your job script)

Hands on Activity- 5 mins

See information about your recent jobs:

1. See what jobs you submitted today:
   `sacct -X -S 2026-02-20`
2. See details about your most recent job:
   `sacct -X -j <job_id>`
3. Try submitting new jobs and check on them with `sacct`

Consumable Computing Resources

The clusters are shared resources. Jobs must have some limits:

- Resources which we can specify in a job/slurm file:
  - Processor cores
  - Memory
  - Walltime
  - GPU and other accelerators
- Other resources:
  - SUs
  - Software license/token
    - Use **`license_status`** to query
    - License Checker: hprc.tamu.edu/kb/Software/useful-tools/License_Checker/

Find available license for "FEMZIP":

```
license_status -s matlab
```

```
License status for Matlab:
-------------------------------------------------------------------------
|License Name                   | # Issued| # In Use | # Available  |
-------------------------------------------------------------------------
|Matlab                         |    50   |     0    |      50      |
-------------------------------------------------------------------------
```

More information on this command:

```
license_status -h
```

How Does Slurm Assign Jobs?

- Job submissions are auto-assigned to batch *queues* (also called *partitions*) based on the resources requested
  - number of cores/nodes and walltime limit
  - specific resources requested
- Some jobs can be directly submitted to a queue:
  - If gpu nodes are needed, use the gpu partition/queue:
    `#SBATCH --partition=gpu`

https://hprc.tamu.edu/kb/User-Guides/ACES/Batch/#batch-queues

sinfo: Queue Status

- **sinfo** shows you the status of the job queues:

```
PARTITION         AVAIL  TIMELIMIT    JOB_SIZE    NODES(A/I/O/T)    CPUS(A/I/O/T)
cpu*              up     3-00:00:00   1-64        69/1/3/73         5960/760/288/7008
gpu               up     2-00:00:00   1-8         6/0/1/7           211/365/96/672
gpu_debug         up     2:00:00      1           1/1/1/3           16/176/96/288
pvc               up     2-00:00:00   1-30        15/13/2/30        892/1796/192/2880
bittware          up     2-00:00:00   1           0/2/0/2           0/192/0/192
nextsilicon       up     2-00:00:00   1           1/1/0/2           51/141/0/192
nec               up     2-00:00:00   1           0/1/0/1           0/48/0/48
staff             up     2-00:00:00   1-110       86/18/6/110       6696/3288/576/10560
```

Shows job queue status

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

pestat : Processor Status

- **pestat** allows you to check the status of the nodes on ACES
- -p allows you to show a specific partition
- Example:

```
pestat -p gpu -G
```

```
Print only nodes in partition gpu
GPU GRES (Generic Resource) is printed after each JobID
Hostname  Partition Node  Num CPU  CPUload  Memsize  Freemem  GRES/node             Joblist
                    State Use/Tot  (15min)   (MB)     (MB)                          JobID User GRES/job ...
ac041         gpu  alloc  96   96    1.95*   500000    491133   gpu:h100:8(S:0)   649110 u.rl117197 gpu:h100=6
ac045         gpu  alloc  96   96    1.92*   500000    489023   gpu:h100:8(S:0)   649110 u.rl117197 gpu:h100=6
ac049         gpu  mix    24   96   18.64*   500000    277410   gpu:h100:4(S:0)   750075 u.xw127610 gpu:h100=8 *
ac055         gpu  alloc  96   96    1.94*   500000    462834   gpu:h100:4(S:0)   649110 u.rl117197 gpu:h100=6
ac064         gpu  idle   0    96    0.00    500000    508735   gpu:a30:2(S:0)
ac065         gpu  idle   0    96    0.00    500000    510542   gpu:a30_1g.6gb:8(S:0)
ac096         gpu   mix   24   96    18.95*  500000    276939   gpu:h100:4(S:0)   750075 u.xw127610 gpu:h100=8 *
```

Job Memory Requests on ACES

- If you request more resources than is allowed, Slurm will reject the job
- To check the maximum requestable amount, use: **`maxconfig`**

```
 ACES partitions:   cpu  gpu  gpu_debug  pvc  bittware  memverge  nextsilicon  nec
 ACES GPUs in gpu partition:  a30:2  h100:2  h100:4  h100:8  pvc:2  pvc:4  pvc:6  pvc:8  ve:8

 Showing max parameters (cores, mem, time) for partition cpu
 CPU-billing * hours * nodes =   SUs
          96 *    72 *     1 = 6,912

#!/bin/bash
#SBATCH --job-name=my_job
#SBATCH --time=3-00:00:00
#SBATCH --nodes=1          # max 64 nodes for partition cpu
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=96
#SBATCH --mem=488G
#SBATCH --output=stdout.%x.%j
#SBATCH --error=stderr.%x.%j
```

Check your Service Unit (SU) Balance

- List the SU Balance of your Account(s).

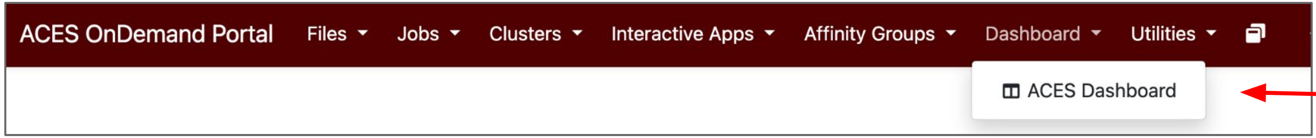
```
myproject -m
```

```
 =================================================================================================
                List of u.lp46691's Project Accounts
------------------------------------------------------------------------------------------------
|  Account   |  FY  | Default | Allocation |Used & Pending SUs|   Balance   |          PI        |
------------------------------------------------------------------------------------------------
|154669186753|  2025|        Y|   951450.00|          -2443.94|   949006.06|Perez, Lisa          |
------------------------------------------------------------------------------------------------
|155466915524|  2025|        N|   602891.00|             -3.51|   602887.49|Perez, Lisa          |
------------------------------------------------------------------------------------------------
===========================
   Account         Project
---------------------------
 154669186753   TRA220029
 155466915524   TRA240008
---------------------------
```

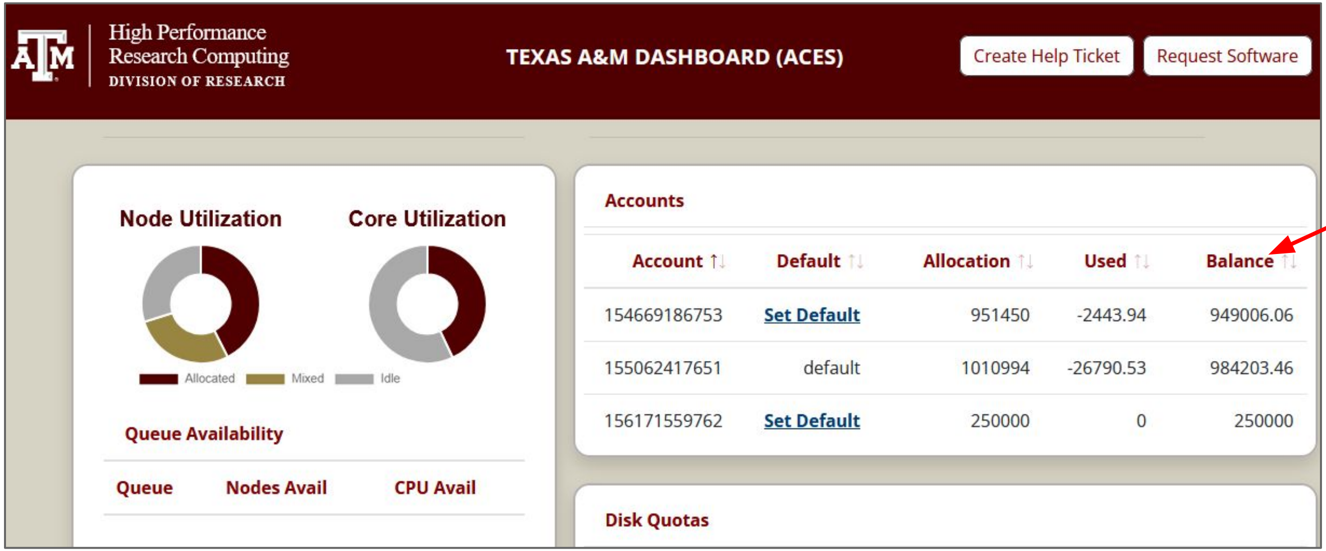
- Run "myproject -d *Account#*" to change default project account.
- Run "myproject -h" to see more options.

https://hprc.tamu.edu/kb/User-Guides/AMS/#service-unit
https://hprc.tamu.edu/kb/User-Guides/AMS/#ams-user-interfaces

Check your Service Unit (SU) Balance



Select Dashboard -> ACES Dashboard in the portal



Accounts

Account ↑↓	Default ↑↓	Allocation ↑↓	Used ↑↓	Balance ↑↓
154669186753	Set Default	951450	-2443.94	949006.06
155062417651	default	1010994	-26790.53	984203.46
156171559762	Set Default	250000	0	250000

Queue Availability

Queue	Nodes Avail	CPU Avail

Disk Quotas

Here you can view your accounts SU Balance

ACCESS Charging Scheme

Resource(s)	SUs (per hour)	ACCESS Credits (per hour)
Intel SPR / Icelake	1	0.1
Intel PVC GPUs	30	3.75
Bittware Agilex FPGA	30	3.75
Intel Optane Memory	30	3.75
Graphcore IPU Classic	45	5.625
NextSilicon Co-processor	50	6.25
Graphcore IPU Bow	60	7.5
NVIDIA A30	64	8
NEC Vector Engine	75	9.375
NVIDIA H100	128	16

8 SUs on ACES = 1 ACCESS credits

Continued Learning

HPRC YouTube

HPRC Homepage

ACES Quick Start Guide

ACES Portal (ACCESS users)

help@hprc.tamu.edu

HIGH PERFORMANCE
RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

https://hprc.tamu.edu

HPRC Helpdesk:

help@hprc.tamu.edu
Phone: 979-845-0219

Take our short course survey!

https://u.tamu.edu/hprc_shortcourse_survey

HPRC Survey

https://u.tamu.edu/hprc_shortcourse_survey

Help us help you. Please include details in your request for support, such as, Cluster (ACES, FASTER, Grace, Launch), NetID (UserID), Job information (JobID(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.

High Performance
Research Computing
DIVISION OF RESEARCH

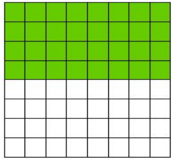
Thank you.

Any questions?

u.tamu.edu/hprc_shortcourse_survey

Slurm Job File (multi core, single node)

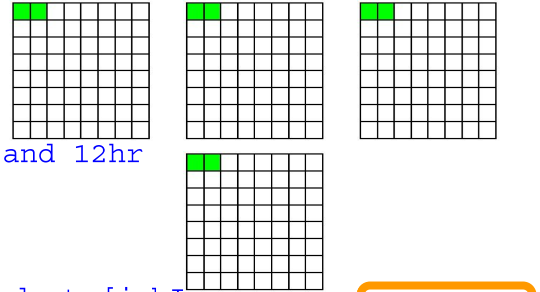
```
#!/bin/bash

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample2      # Set the job name to "JobExample2"
#SBATCH --time=6:30:00              # Set the wall clock limit to 6hr and 30min
#SBATCH --nodes=1                        # Request 1 node
#SBATCH --ntasks-per-node=32        # Request 32 tasks(cores) per node
#SBATCH --mem=28G                   # Request 28GB 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=email_address   # Send all emails to email_address

# purge module
module purge
# load required module(s)
module load GCC/12.1.0

# run your program
./my_multicore_program
```



SUs = 208

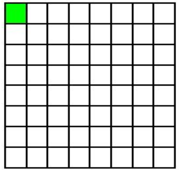
Slurm Job File (multi core, multi node)

```
#!/bin/bash

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample3         # Set the job name to "JobExample3"
#SBATCH --time=1-12:00:00              # Set the wall clock limit to 1 Day and 12hr
#SBATCH --ntasks=8                     # Request 8 tasks (cores)
#SBATCH --ntasks-per-node=2            # Request 2 tasks(cores) per node
#SBATCH --mem=2.5G                     # Request 2.5 GB  per node
#SBATCH --output=stdout.%j                  # Send stdout and stderr to "stdout.[jobID]"

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456               # Set billing account to 123456 (find your account with
"myproject")
#SBATCH --mail-type=ALL                # Send email on all job events
#SBATCH --mail-user=email_address      # Send all emails to email_address

# purge and then load the modules required for your software
module purge
module load GCC/11.3.0 OpenMPI/4.1.4

# run program with MPI
mpirun my_multicore_multinode_program
```



SUs = 288

Slurm Job File (serial GPU)



```
#!/bin/bash

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample4      # Set the job name to "JobExample4"
#SBATCH --time=01:00:00                   # Set the wall clock limit to 1hr
#SBATCH --ntasks=1                        # Request 1 task (core)
#SBATCH --mem=2G                          # Request 2GB per node
#SBATCH --output=stdout.%j                # Send stdout and stderr to “stdout.[jobID]"
#SBATCH --gres=gpu:h100:1                 # Request 1 GPU
#SBATCH --partition=gpu                   # Request the GPU partition/queue

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

# purge and then load required module(s)
module purge
module load CUDA/11.7.0

# run your program
./my_gpu_program
```

SUs = 129

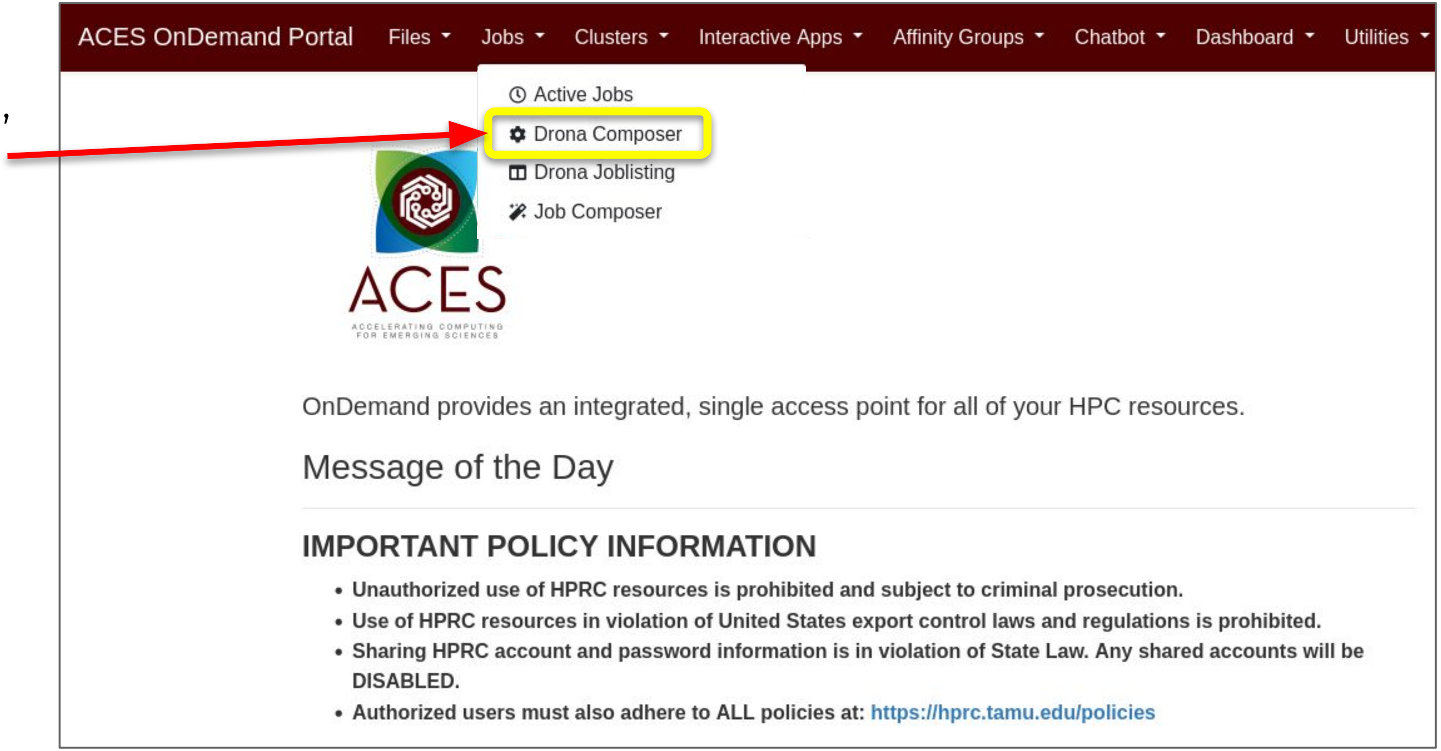
Other Type of Jobs

- MPI and OpenMP
- Visualization:
  - https://portal.hprc.tamu.edu/ (visualization jobs can be run on both FASTER and Grace)
- Large number of concurrent single core jobs
  - Check out ***tamulauncher***
    - https://hprc.tamu.edu/kb/Software/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, you can resubmit the same job to complete the unfinished commands in the input file

The Drona Composer

- A framework to assist you with generating and creating jobs

Once in the Portal, select at the top: “Jobs” → “Drona Composer”



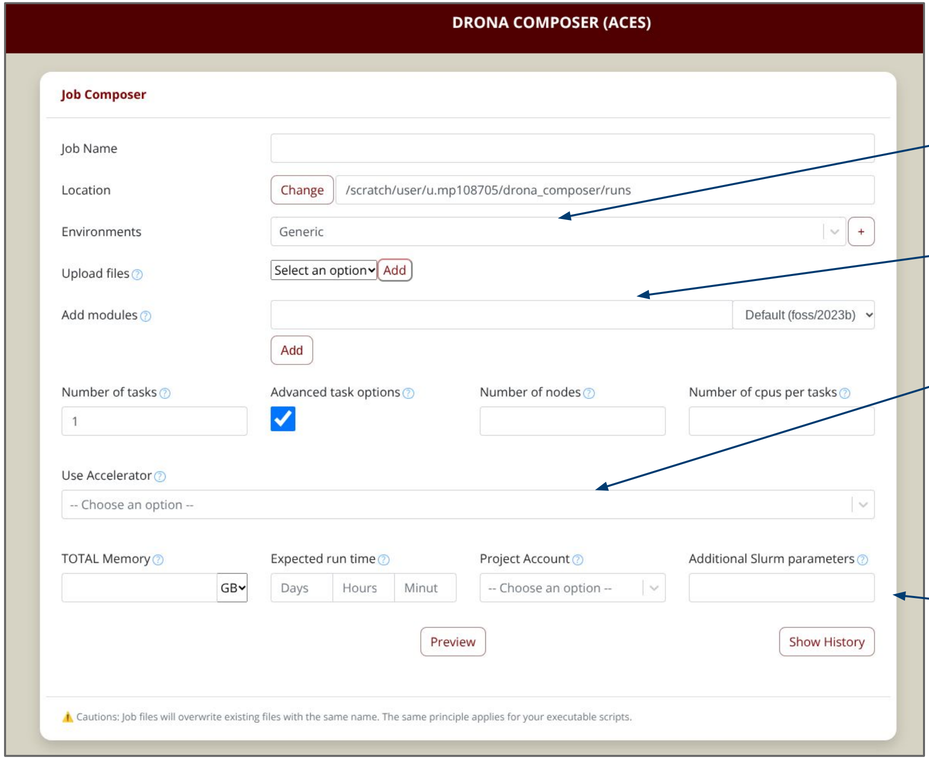
OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

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 State Law. Any shared accounts will be DISABLED.**
- **Authorized users must also adhere to ALL policies at: https://hprc.tamu.edu/policies**

DRONA COMPOSER - JOB SUBMISSION MADE EASY



DRONA COMPOSER (ACES)

Job Composer

Job Name

Location: Change /scratch/user/u.mp108705/drona_composer/runs

Environments: Generic +

Upload files: Select an option Add

Add modules: Default (foss/2023b) Add

Number of tasks: 1

Advanced task options: ☑

Number of nodes

Number of cpus per tasks

Use Accelerator: -- Choose an option --

TOTAL Memory: GB

Expected run time: Days Hours Minut

Project Account: -- Choose an option --

Additional Slurm parameters

Preview

Show History

⚠ Cautions: Job files will overwrite existing files with the same name. The same principle applies for your executable scripts.

Select environment.

Optionally add modules.

Select choice of accelerator.

Set optional job parameters (or let Drona pick defaults).