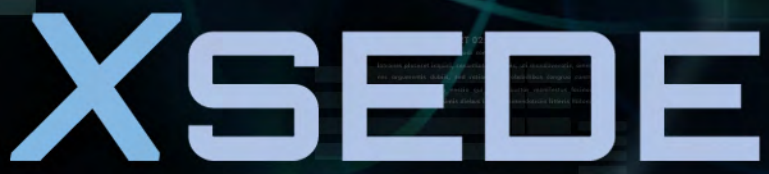
May 13, 2016

XSEDE New User Tutorial



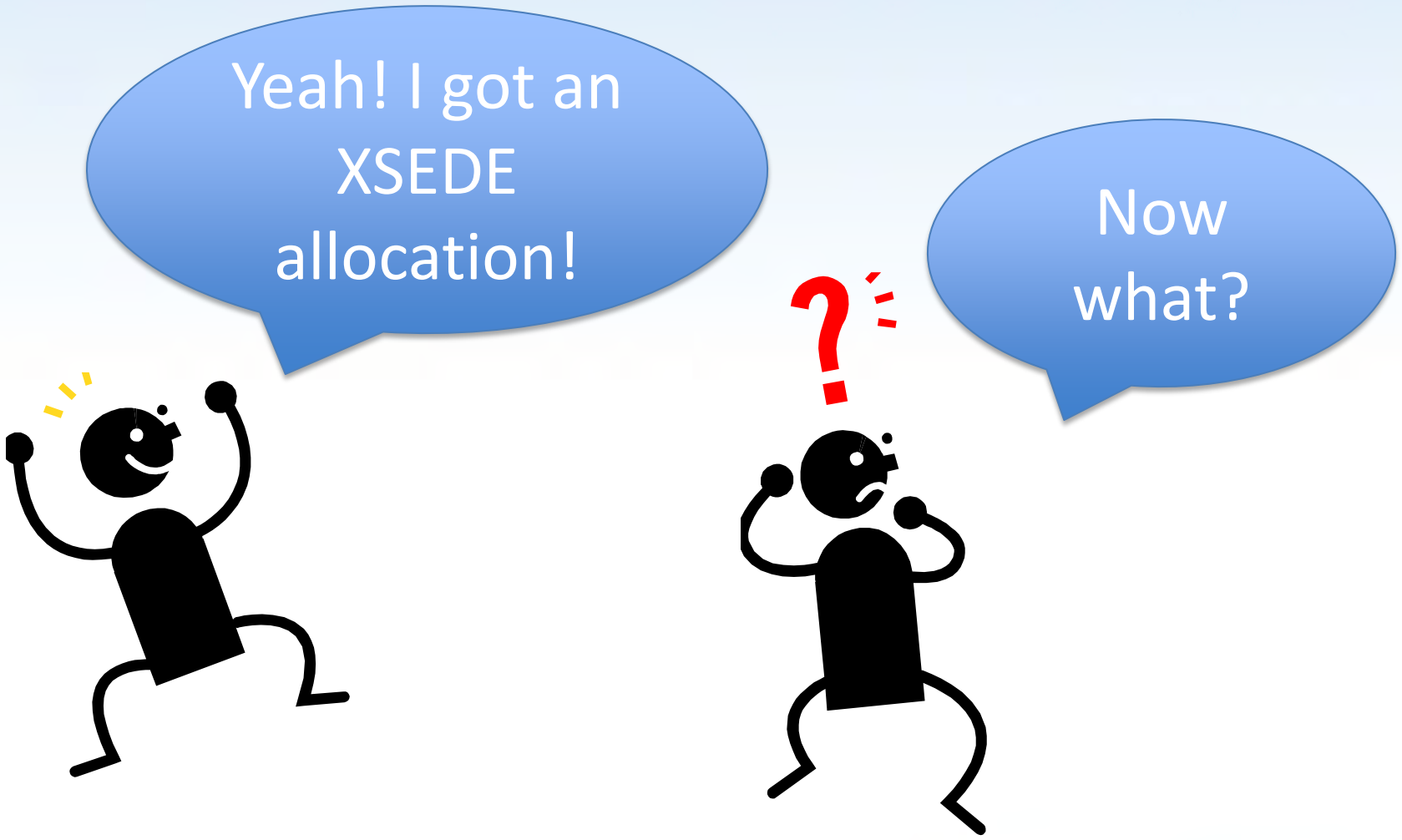
Extreme Science and Engineering Discovery Environment

Jay Alameda
National Center for Supercomputing Applications

XSEDE Training Survey

- Please complete a short on-line survey about this module at http://bit.ly/hamptonxsede. We value your feedback, and will use your feedback to help improve our training offerings.
- Slides from this workshop are available at

  http://hpcuniversity.org/trainingMaterials/219


Yeah! I got an XSEDE allocation!
Now what?

Learning Outcomes

After completing this tutorial, you will be able to:

- Use the XSEDE User Portal
- Access your XSEDE resources
- Manage files
- Run jobs
- Get help

XSEDE User Portal (XUP)

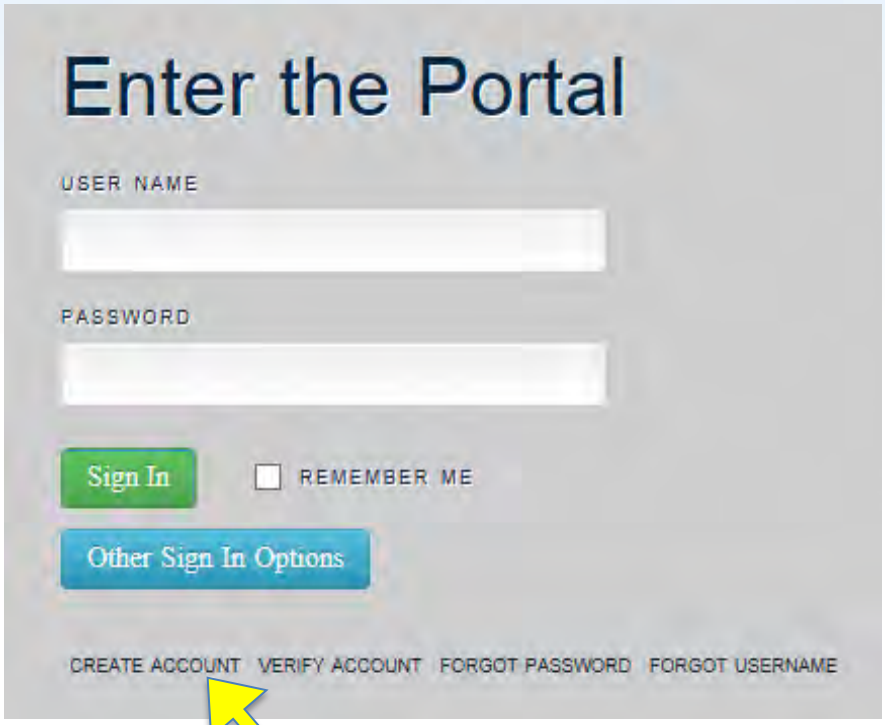
- URL: portal.xsede.org
- Single point-of-entry to information about XSEDE services and utilities for using them
- Anyone can create an XUP user account and access non-project features
- Only XSEDE allocation project members can access project features

Using the XUP

- Create and login to your XUP Account
- Use XSEDE resources responsibly
- Get added to your XSEDE project
- Navigate your personal My XSEDE webpage
- Navigate the information in the XUP

Create and login to your XUP account

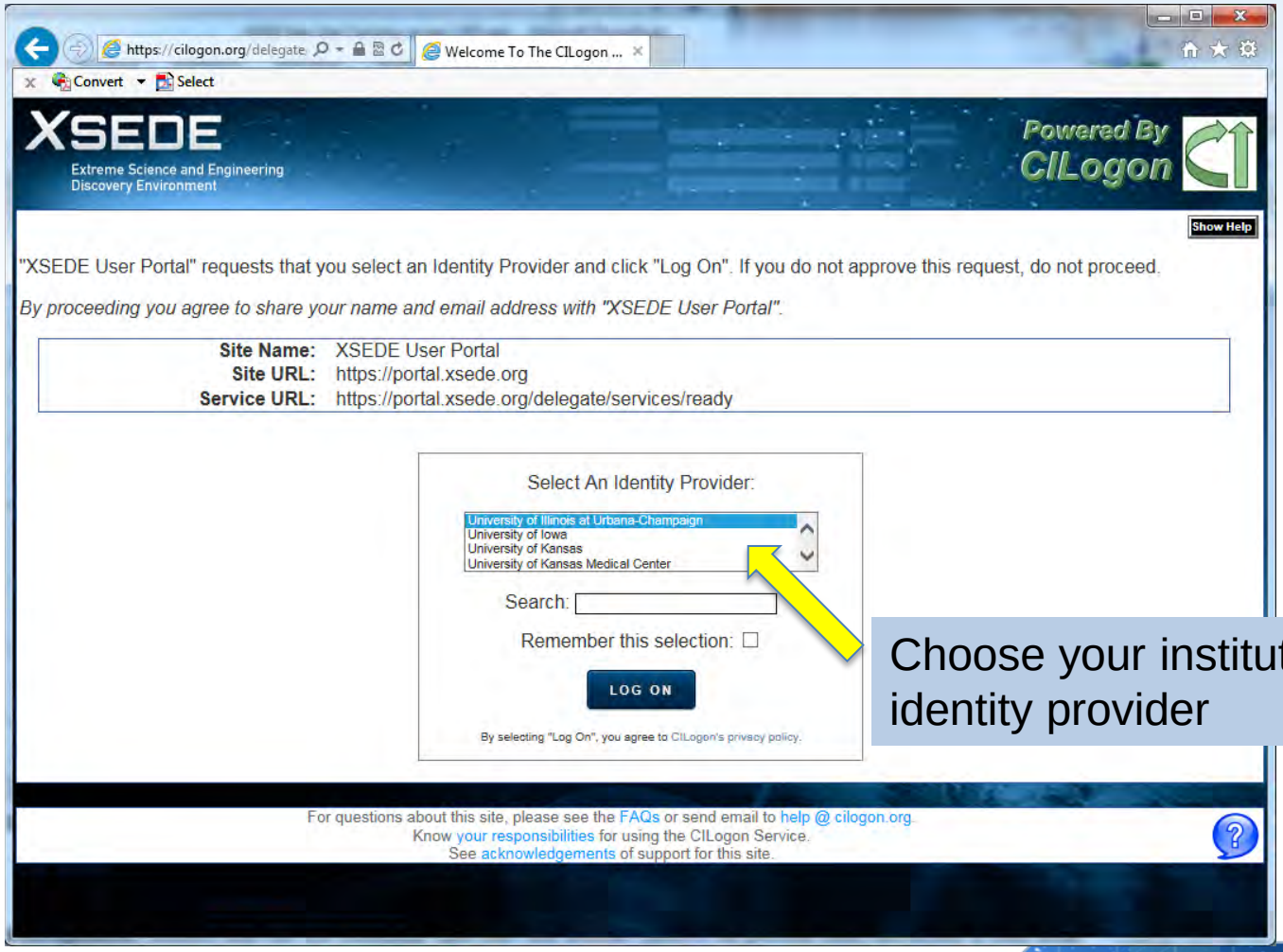
portal.xsede.org



1. From the XUP homepage, click CREATE ACCOUNT
2. Complete the User Account Form
3. Verify your account request
4. Select your username and password
5. Login to the XUP

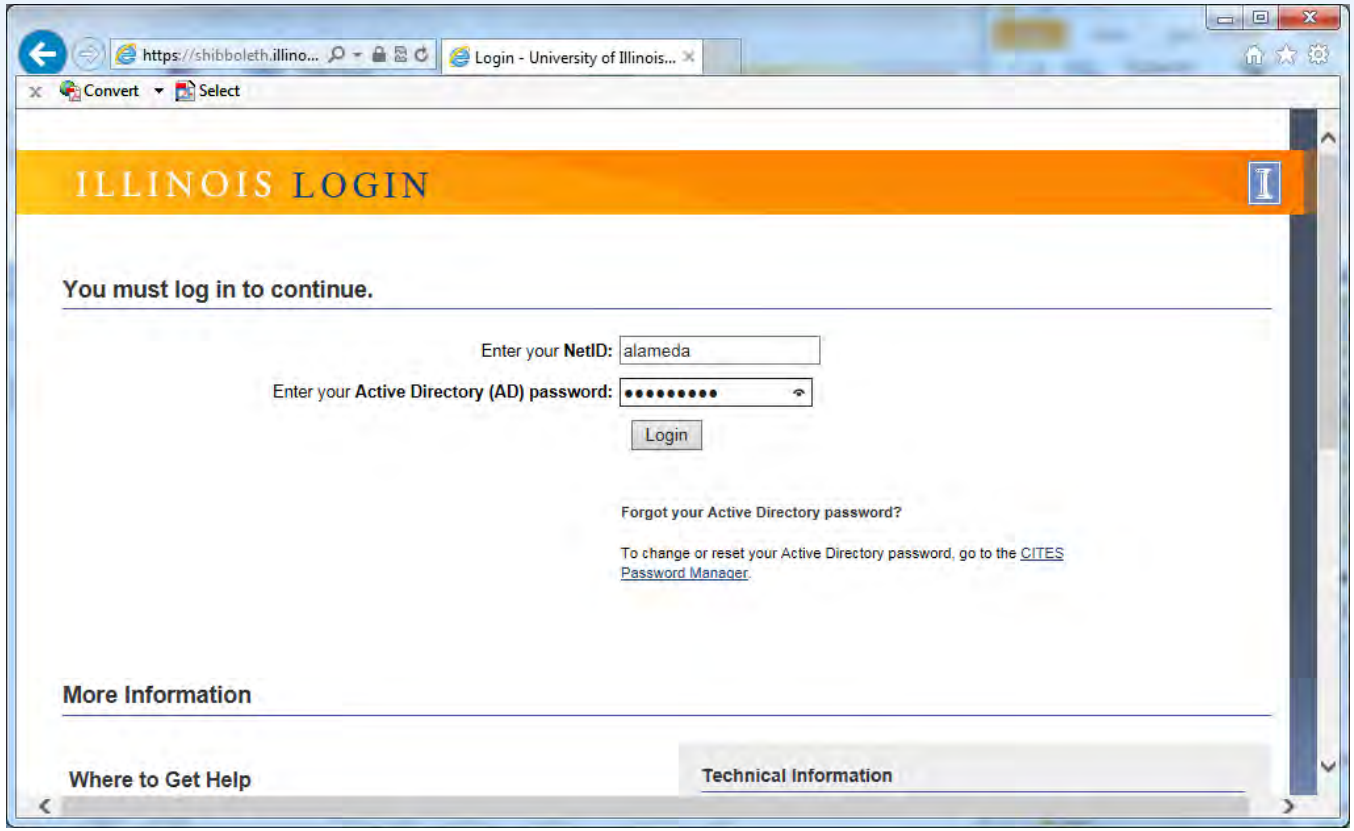
Click the CREATE ACCOUNT link to access the XUP User Account Form

Other Sign In Options

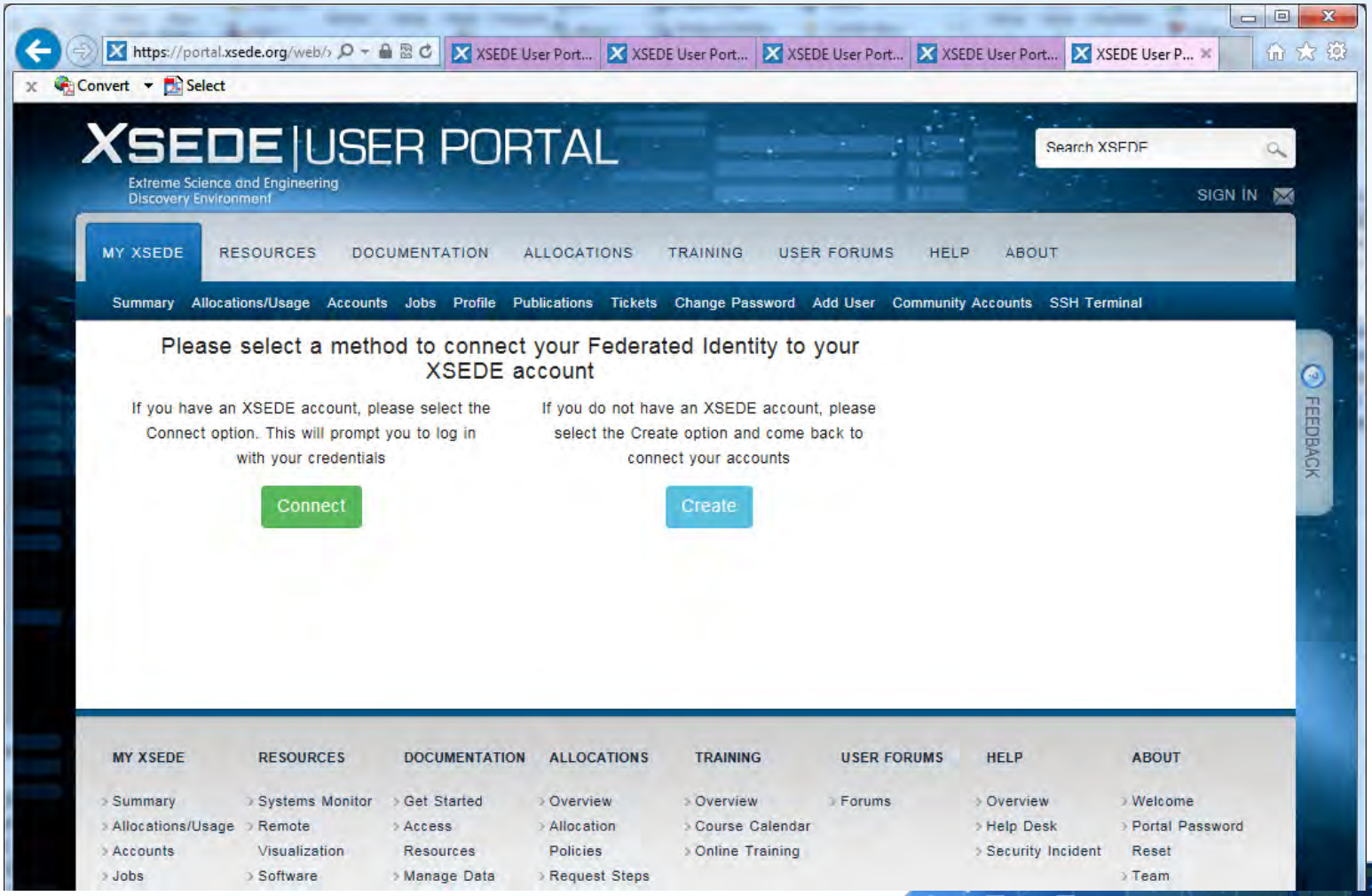


Choose your institutions's identity provider

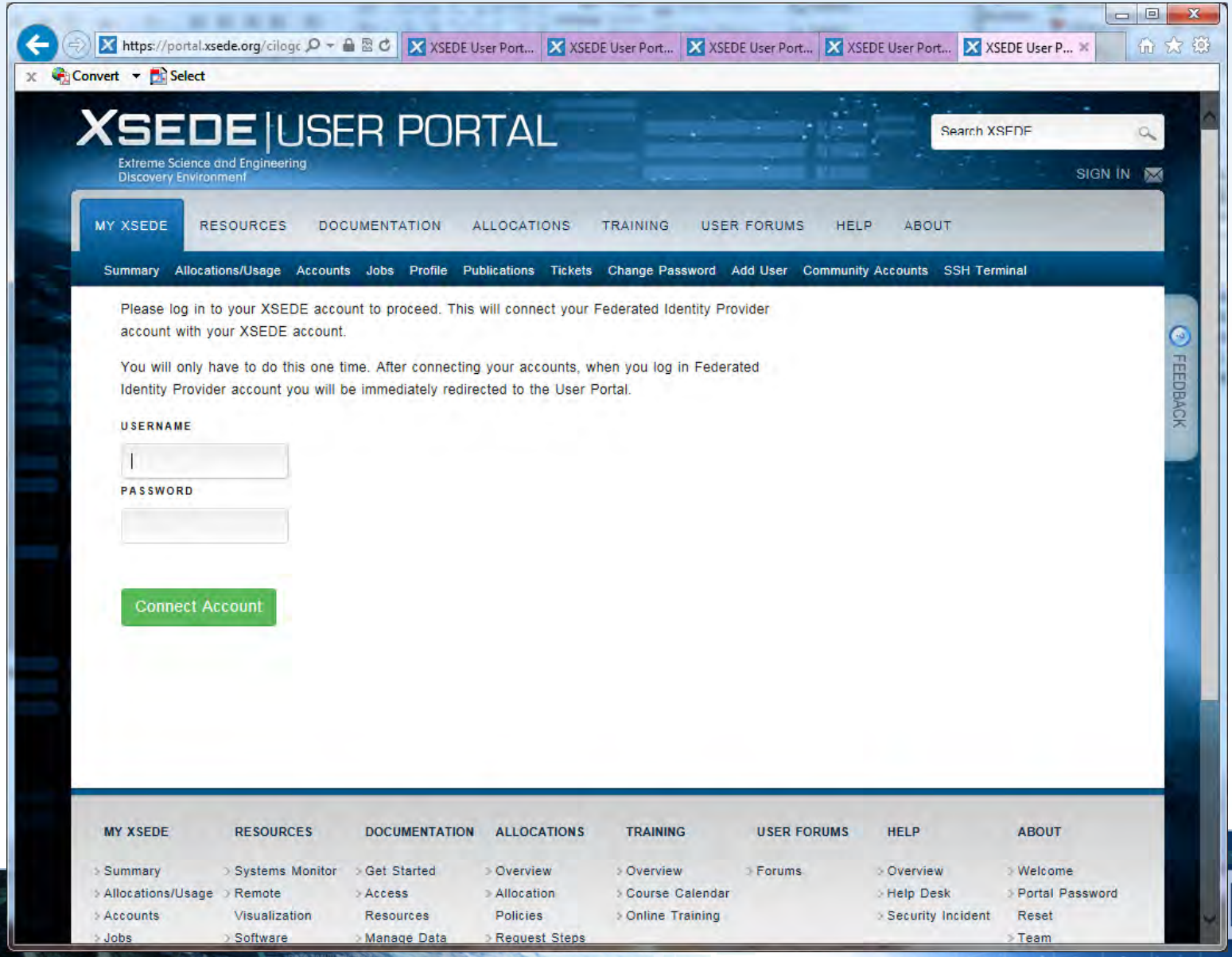
Example: Logging in with Illinois credentials



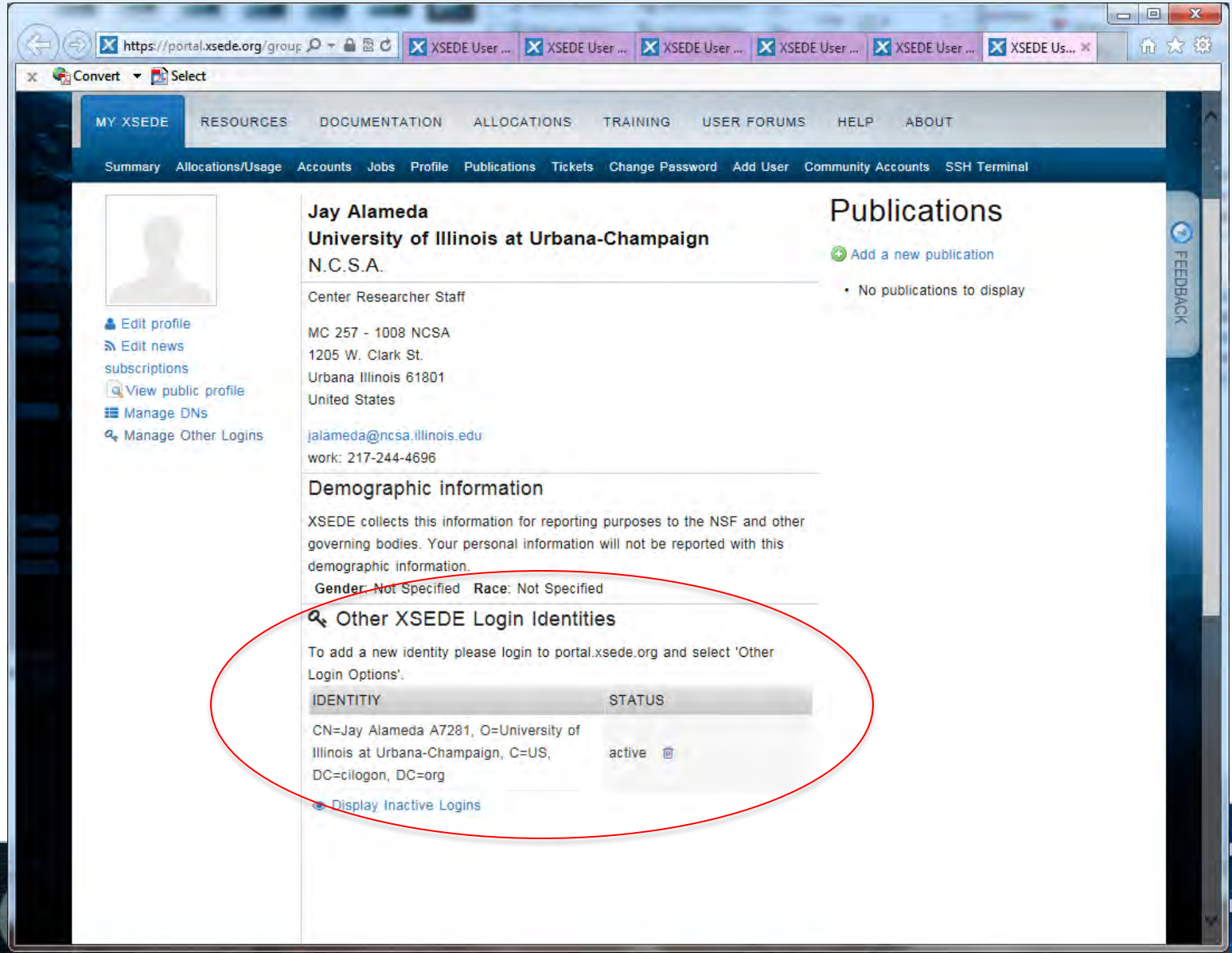
Link your campus identity and portal identity



Login to the portal to link identities



And inspect your new linked identity, via user profile



XSEDE Acceptable Use Policy

- Must accept the User Responsibilities Form after creating your XUP account and again at the beginning of each allocation you receive.
- Choose a strong password and protect it.
- Close SSH terminals and log out of the User Portal when you are finished with your session.
- Report Suspicious Activity : email help@xsede.org or call 1-866-907-2383 immediately, regardless of the time of day.

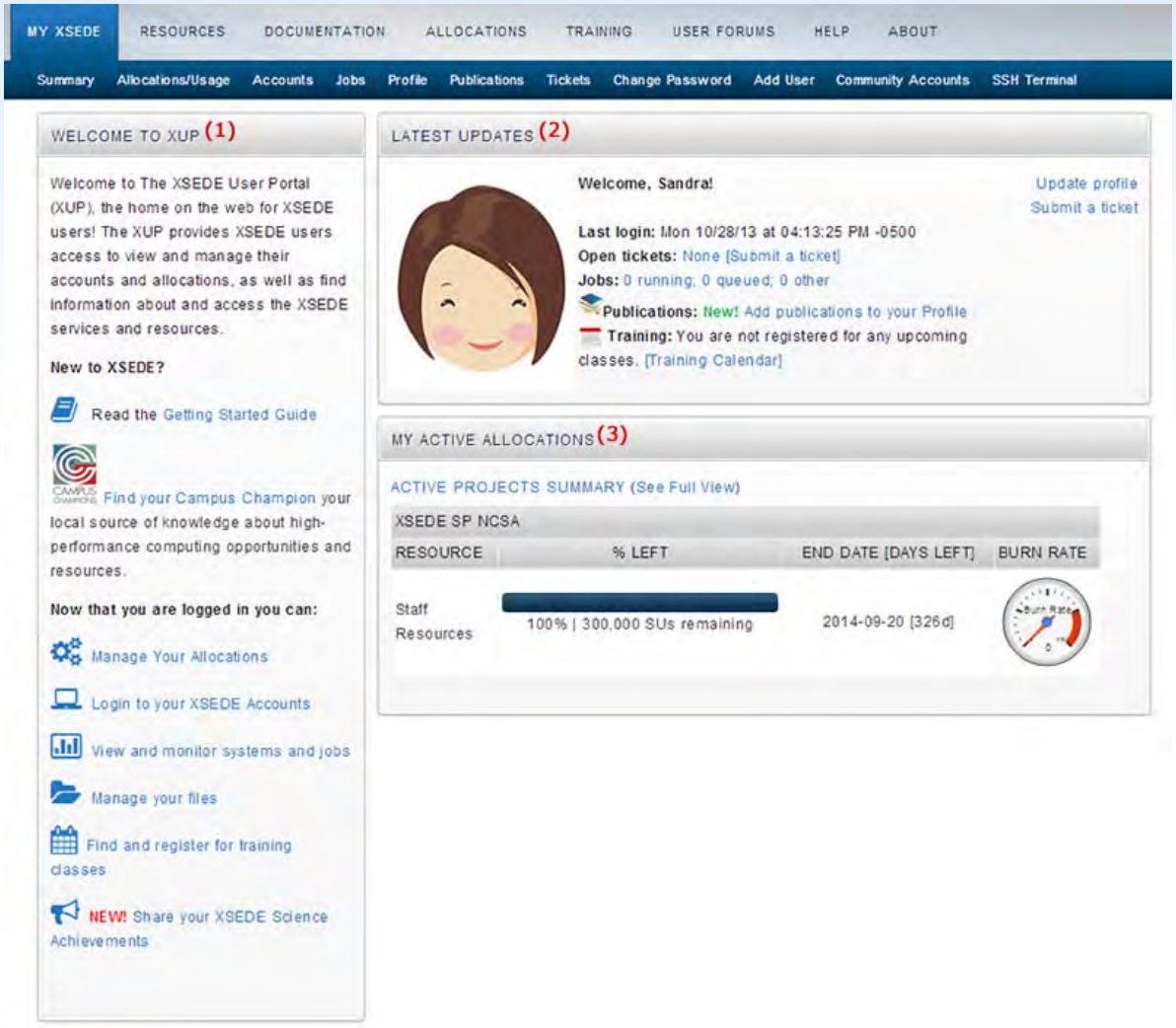
XSEDE Cybersecurity Tutorial

https://portal.xsede.org/web/xup/online-training

Get Added to Your XSEDE project

- PIs automatically have full access to their project's account.
- The PI is responsible for managing users on their account.
- Ask the PI, or their allocation manager, to add your XUP username to the project.

Your My XSEDE webpage



(1)

WELCOME TO XUP

- Quick access to commonly used features.

(2)

LATEST UPDATES

- Latest information specific to your user account.

(3)

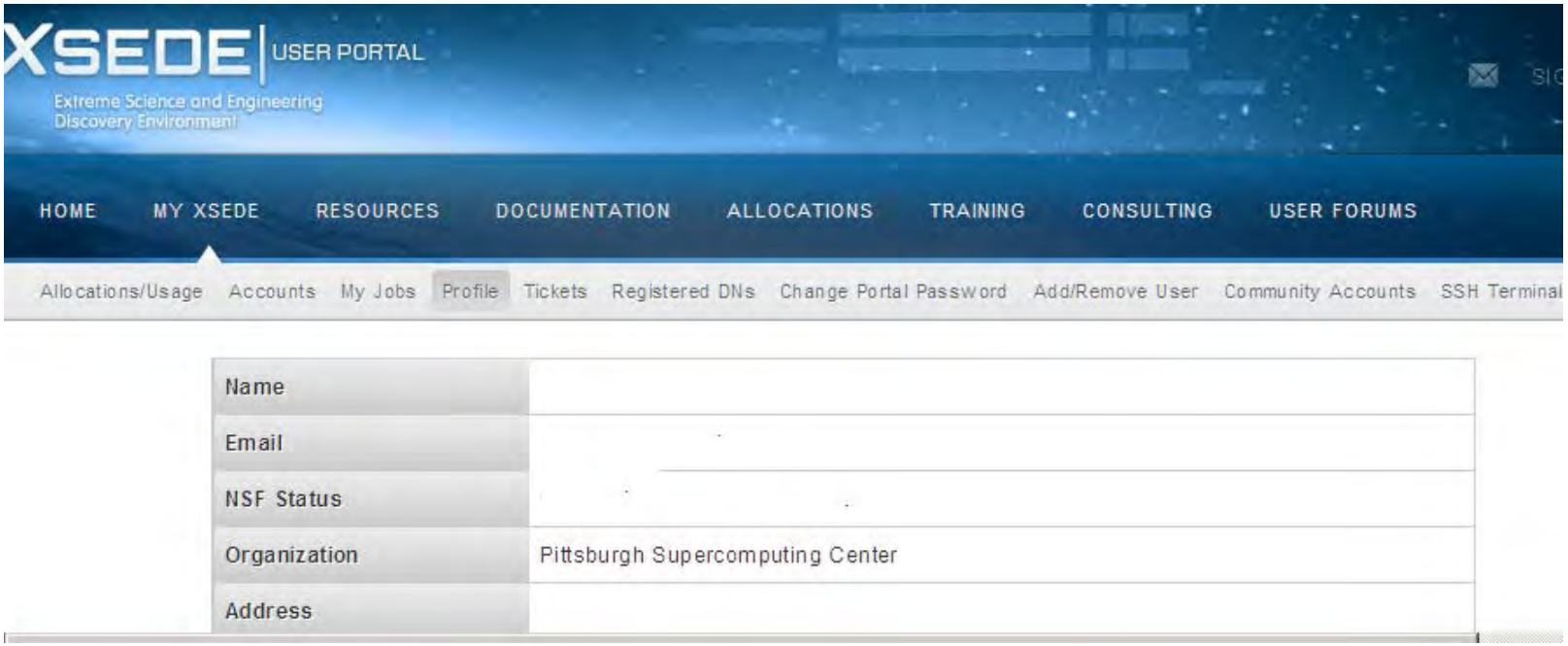
MY ACTIVE ALLOCATIONS

- Summary of the active projects for which you are either a PI or member.

Update your XUP User Profile

MY XSEDE→Profile

- View and or change your user information (organization, address).
- Make sure your email address is correct. XSEDE staff will use it to communicate with you regarding your allocation.

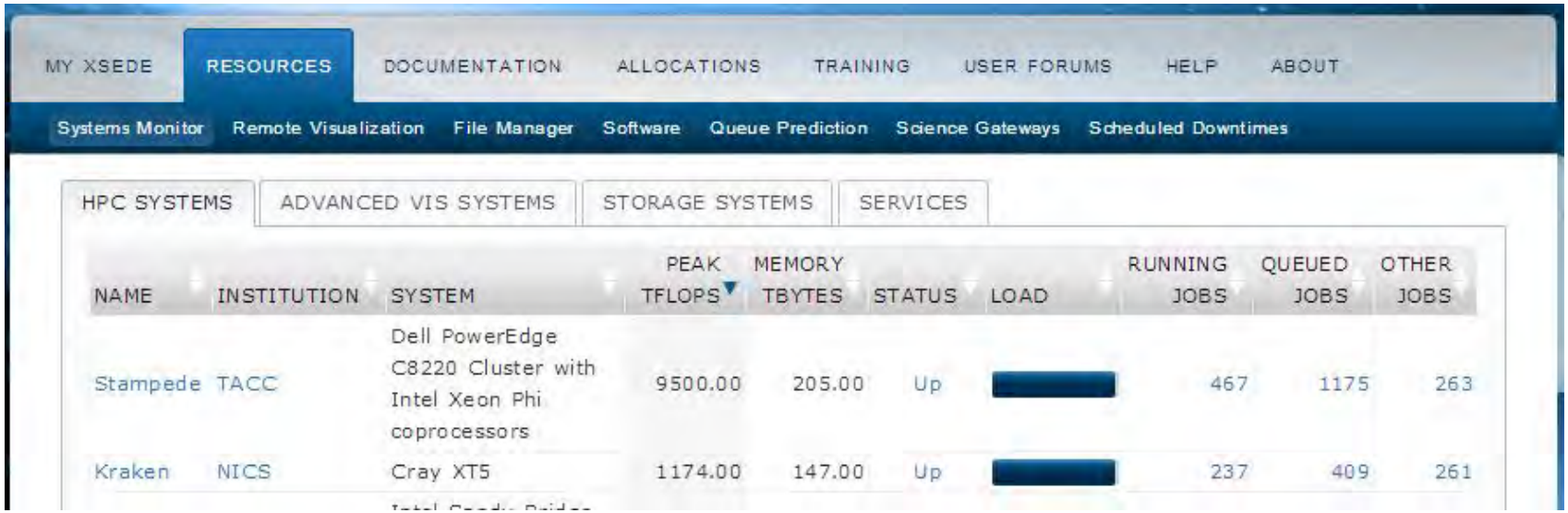


Navigating the XUP

- My XSEDE
- Resources
- Documentation
- Allocations
- Training
- User Forums
- Help
- About

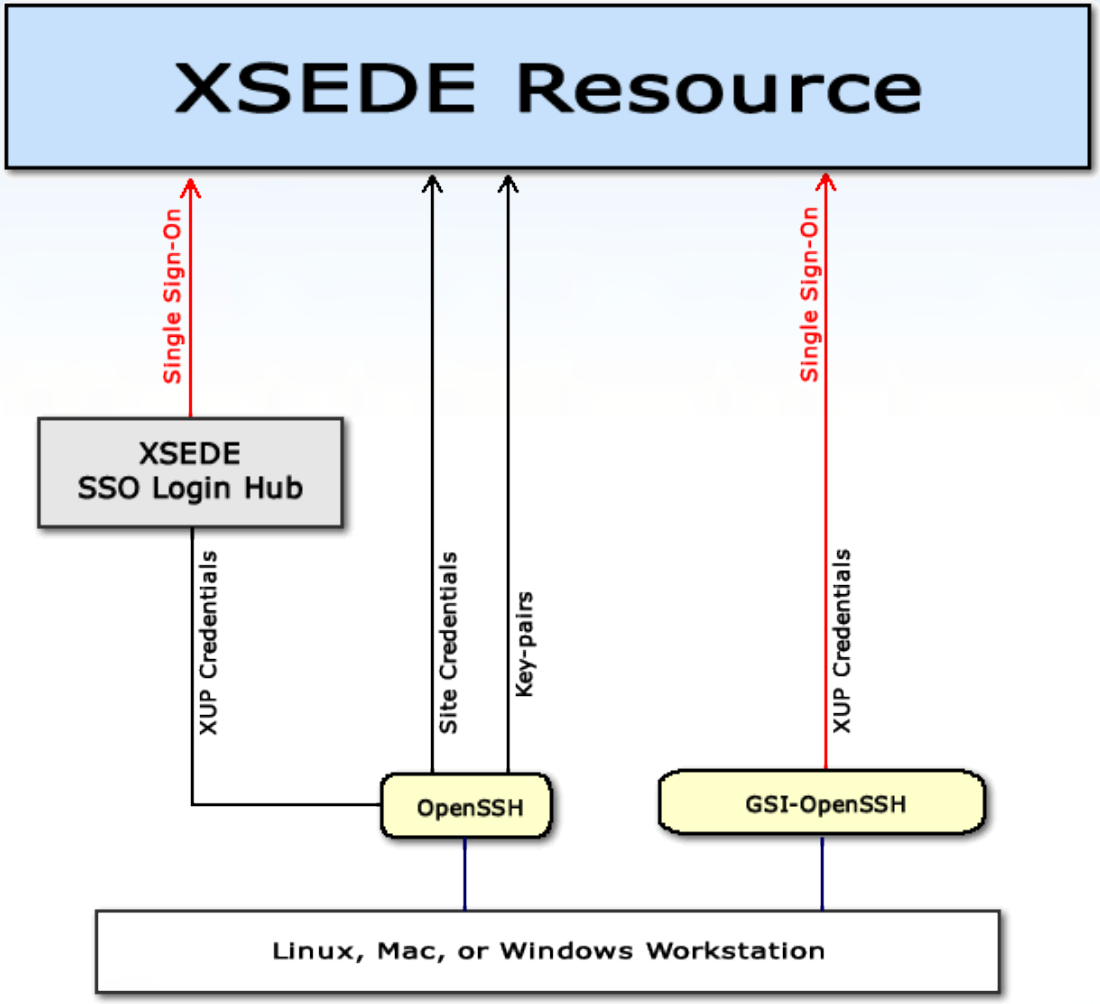
View the XSEDE Systems Monitor

- **Resources -> Systems Monitor**
  - Provides technical and status information for all of XSEDE's resources.
  - The STATUS column indicates whether the system is up or down. If down, can click on status to find when the machine is expected to come back up.



NAME	INSTITUTION	SYSTEM	PEAK TFLOPS	MEMORY TBYTES	STATUS	LOAD	RUNNING JOBS	QUEUED JOBS	OTHER JOBS
Stampede	TACC	Dell PowerEdge C8220 Cluster with Intel Xeon Phi coprocessors	9500.00	205.00	Up		467	1175	263
Kraken	NICS	Cray XT5	1174.00	147.00	Up		237	409	261

Accessing XSEDE Resources



Authentication Methods

1. Password
   - XUP credentials
   - Site-password
   - One-time password
2. Key-based

Single Sign-On

- Enables logging in once to access all of your allocated resources

Connection Methods

1. GSI-OpenSSH
2. OpenSSH

XSEDE SSO Login Hub

An SSO enabled connection point to XSEDE resources

- Move among resources using **gsissh** command

- SSH to **login.xsede.org** using your XUP credentials

Following along with today's tutorial:

- Verify that everyone has an ssh client on their laptop!
- For ssh to XSEDE SSO login hub (today!)

  ssh –l *username* login.xsede.org

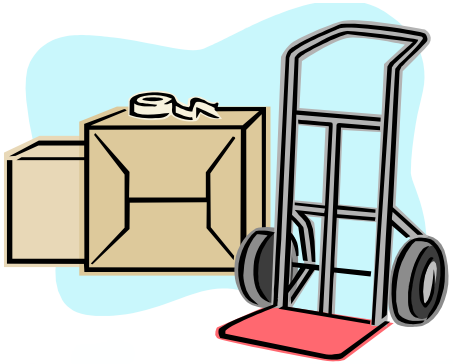
  username on handout
- And from there go to your XSEDE resource, for example:

  gsissh comet.sdsc.edu

Managing your XSEDE files

1. **Where to store files**
   - Home directory
   - Scratch directory
   - Archival storage

2. **How to move files**
   - Command line using globus-url-copy, uberftp, scp, or sftp
   - **Globus Online**



XSEDE File Systems

- **Home directory**
  - Location specified in the environment variable $HOME.
  - Use to store project files you want to keep long term such as source code, scripts, and input data sets.
  - Not backed up regularly and not purged.
  - Quotas typically set to limit amount of disk space available.
- **Scratch directory**
  - Location specified in environment variable varies among resources but will include the term SCRATCH, e.g. $SCRATCH_DIR.
  - Use to temporarily store files produced during application runs.
  - Not backed up and routinely purged.
  - No quotas. Available space depends on cumulative use by all users.
- Archival storage
  - Must request through allocation process

Your XSEDE Compute Environment

- Your default XSEDE compute environment provides access to the compilers, directories, and software you will need to efficiently use your XSEDE resources.
  - Environment: *An area of a computer's memory used by the operating system and some programs to store certain variables to which they need frequent access*
- Customize environment using **Modules**

XSEDE Customizing Environment Tutorial

https://portal.xsede.org/web/xup/online-training

Modules Package

- A command line interface used to configure the shell for an application. Two components:
  1. Modulefiles - contain configuration information
  2. Module command - interprets modulefiles
- Pre-written modulefiles available for compilers, mpi implementations
- Pre-written modulefiles available for common software, e.g. NAMD, GAMESS

Module Commands

Module command	Description
module avail [path...]	List all modulefiles available on the system.
module list	List the modulefiles currently loaded in the shell environment.
module help modulefile	Print help information for the **modulefile** specified in the argument.
module display modulefile	Display the changes made to the environment when the specified modulefile is loaded.
module load modulefile	Interpret the commands contained within the specified modulefile.
module switch modulefile1 modulefile2	Remove the environment changes made by **modulefile1** and make the changes specified in **modulefile2**.
module unload modulefile	Remove the environment changes made by **modulefile**.

Module Commands Example

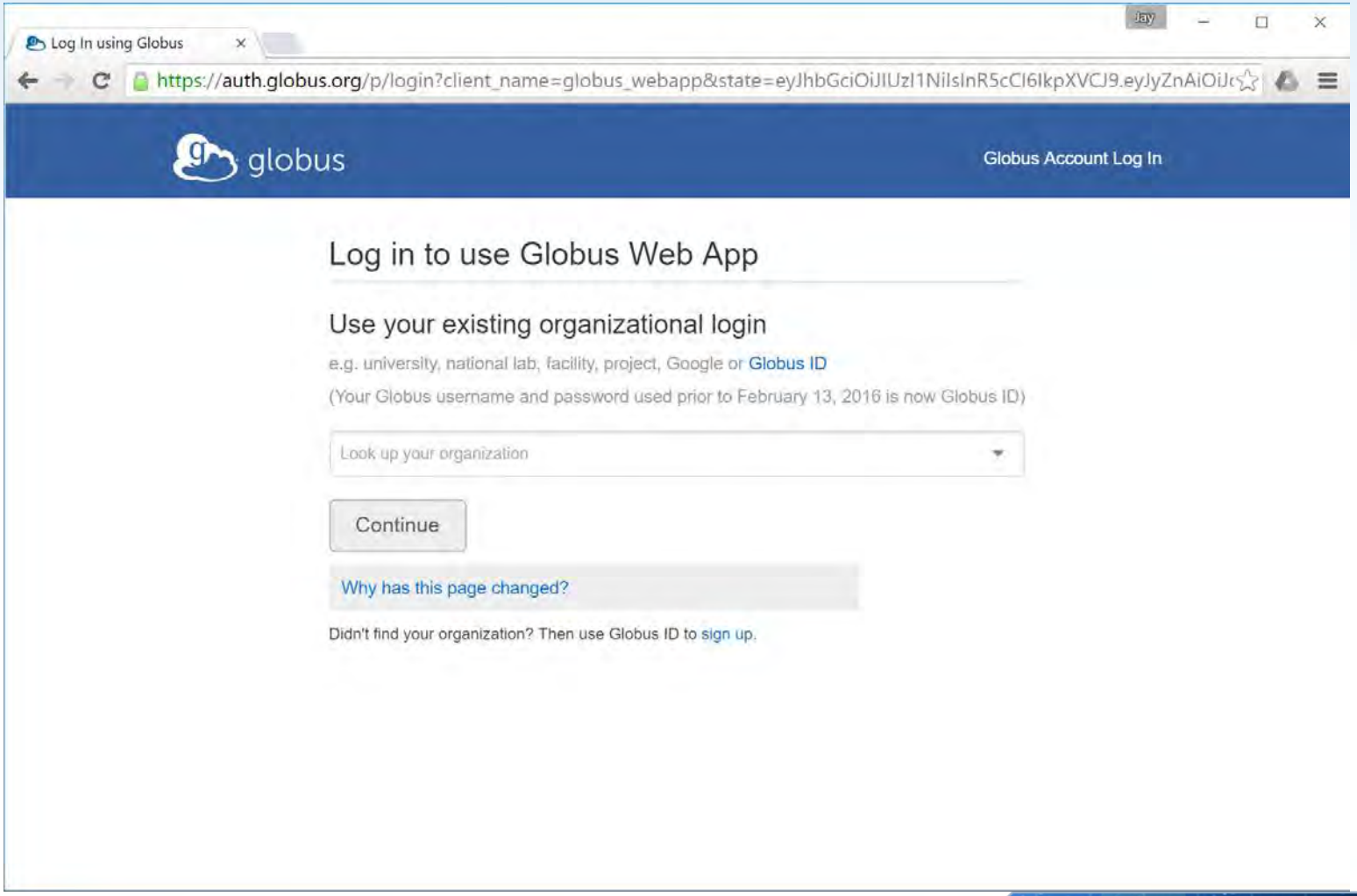
```
% module list
Currently Loaded Modulefiles:
  1) torque/2.3.13_psc   4) icc/14.0.0          7) globus/5.2.2
  2) mpt/2.04            5) imkl/10.3.3         8) xdusage/1.0-r7
  3) ifort/14.0.0        6) psc_path/1.0
% module avail gcc
------------------------- /usr/local/opt/modulefiles --------------------------
gcc/4.3.5 gcc/4.4.6 gcc/4.5.3 gcc/4.6.0 gcc/4.7.2 gcc/4.8.0 gcc/4.8.1
% module load gcc/4.8.1
% module list
Currently Loaded Modulefiles:
  1) torque/2.3.13_psc   5) imkl/10.3.3         9) mpfr/3.1.0
  2) mpt/2.04            6) psc_path/1.0       10) gmp/5.0.5
  3) ifort/14.0.0        7) globus/5.2.2       11) mpc/0.8.2
  4) icc/14.0.0          8) xdusage/1.0-r7     12) gcc/4.8.1
% module unload gcc
% module list
Currently Loaded Modulefiles:
  1) torque/2.3.13_psc   4) icc/14.0.0          7) globus/5.2.2
  2) mpt/2.04            5) imkl/10.3.3         8) xdusage/1.0-r7
  3) ifort/14.0.0        6) psc_path/1.0
```

Moving Files - Globus

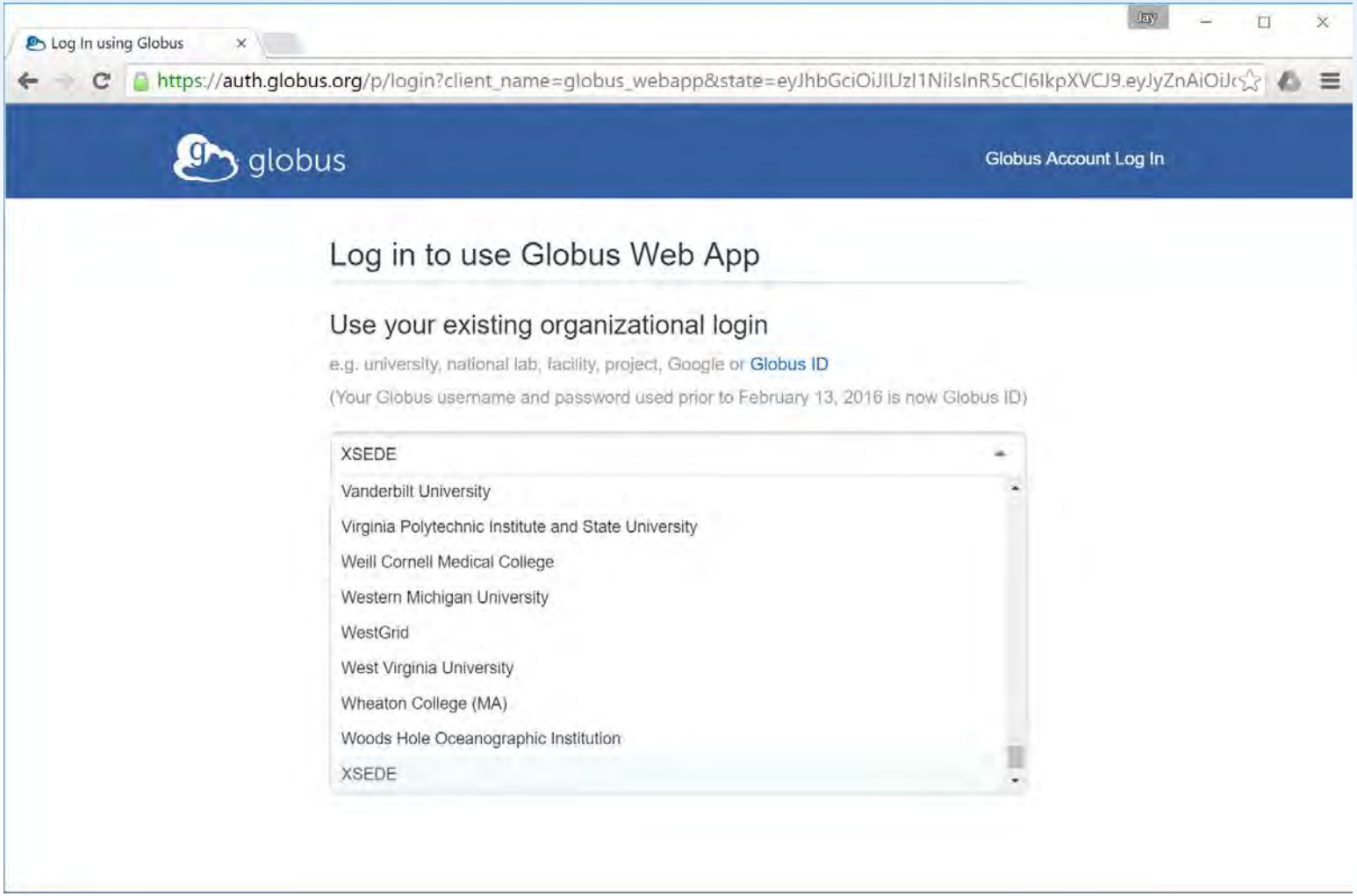
- A fast, reliable, and secure file transfer service geared to the big data needs of the research community.
- Moves terabytes of data in thousands of files
- Automatic fault recovery
- Easy to use
- No client software installation
- Consolidated support and troubleshooting
- Supports file transfer to any machine
- Accounts are free - https://www.globus.org/

Globus Dashboard

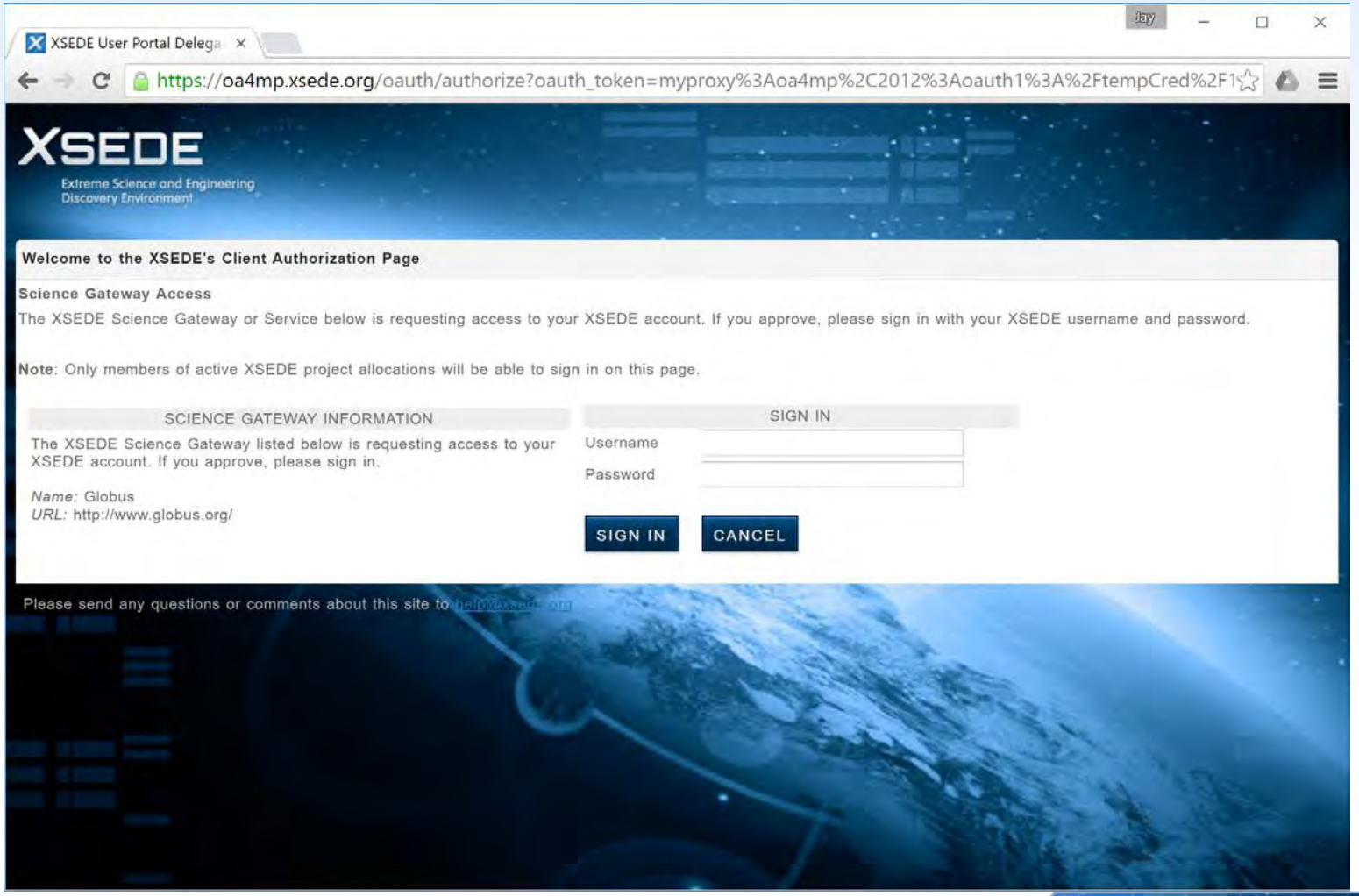
Login to use Globus Web App



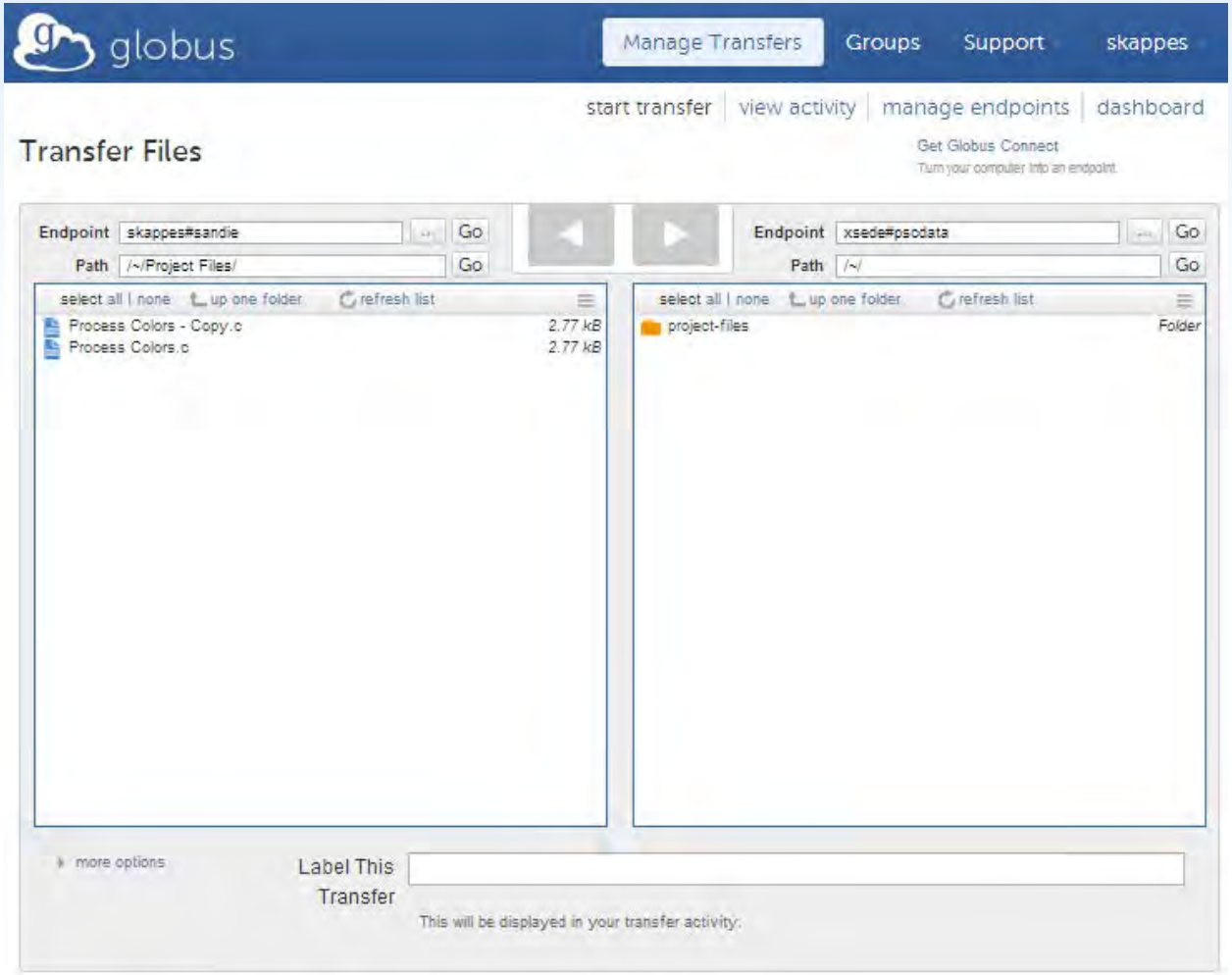
Use XSEDE Identity Provider



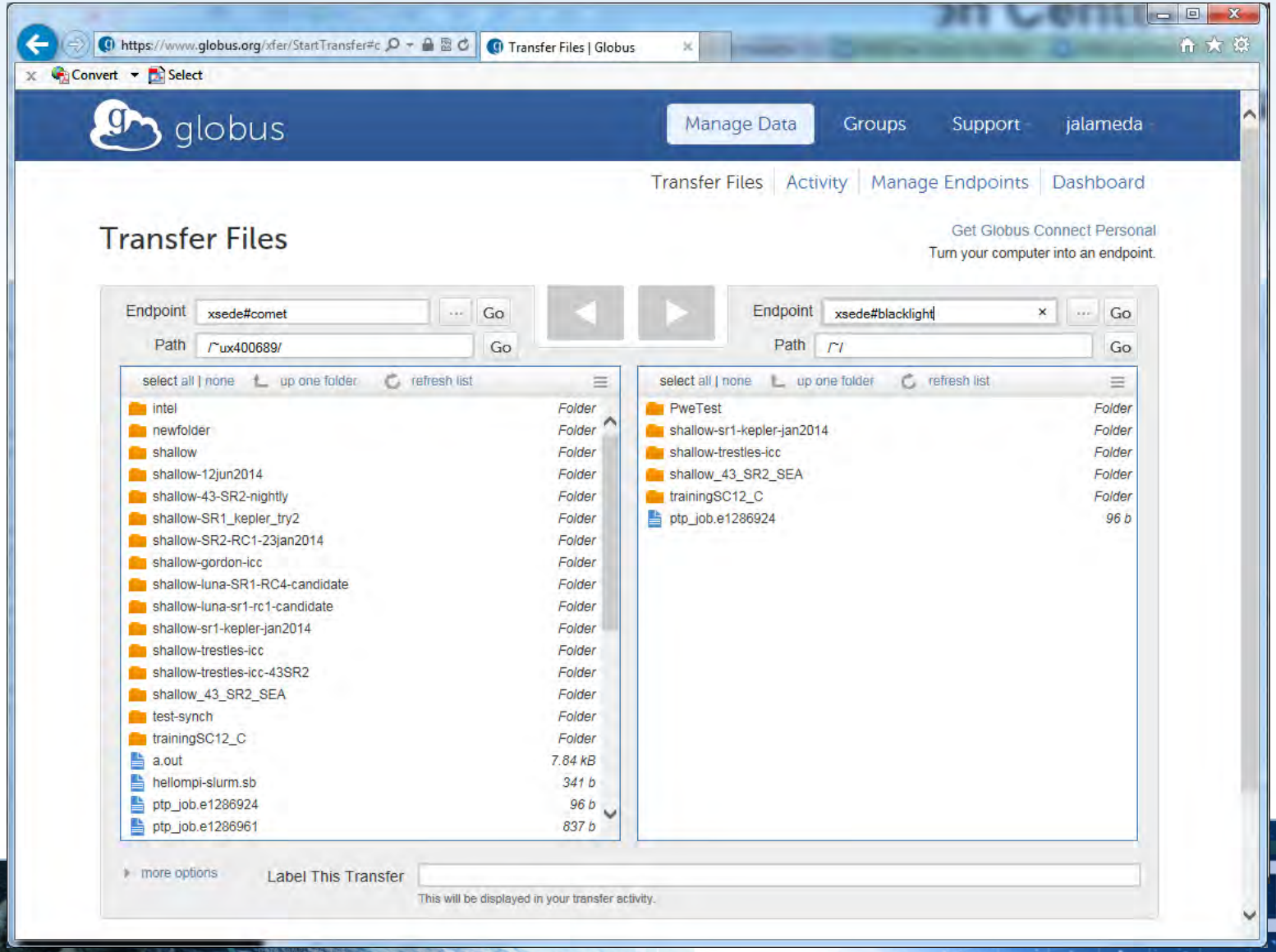
Sign in with XSEDE credentials



Globus Online File Transfer



Choosing a file to move…



Running Jobs Overview

Researcher

Login

Command Line

When you connect to a resource, you are on a login node shared by many users.

Batch Script

Commands for code execution, copy input files to scratch,… Specify number/type of nodes, length of run, output directory, …

Data

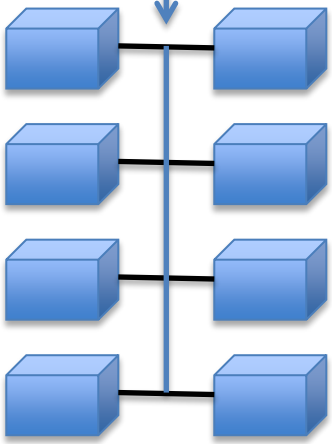
Job

Login Nodes

Use for tasks such as file editing, code compilation, data backup, and job submission.

Store project files Such as source code, scripts, and input data sets to Home directory.

Run jobs by submitting your batch script to the compute nodes using the "qsub" command.



Home

Scratch

File System

Data

Read/write data from compute nodes to Scratch directory.

Compute Nodes

Your job is submitted to a queue and will wait in line until nodes are available. Queues are managed by a job scheduler that allows jobs to run efficiently.

XSEDE

Login Nodes

- When you login to an XSEDE resource, you connect to a login node.
- Use login nodes for basic tasks such as file editing, code compilation, data backup, and job submission.
- Do not run compute jobs on the login nodes.

Running Compute Jobs

- Jobs are run on the compute nodes by submitting a **batch script** on a login node
- All jobs are placed in a batch queue after they are submitted.
- All XSEDE compute resources use a batch scheduler for running jobs.
- Resource User Guides on the XUP have details on your system's scheduler.

Batch Scripts

- Batch scripts include scheduler specific directives, comments, and executable commands, e.g.:
  - Number and type of nodes needed
  - Time needed to run the job
  - Where to write output files
- Script commands are system specific – see the resource's User Guide on the XUP for details

Batch Script for PSC's Blacklight

```
 1. #!/bin/csh
 2. #PBS -l ncpus=16
 3. #ncpus must be a multiple of 16
 4. #PBS -l walltime=5:00
 5. #PBS -j oe
 6. #PBS -q batch
 7. set echo
 8. ja
 9. #move to my $SCRATCH directory
10. cd $SCRATCH
11. #copy executable to $SCRATCH
12. cp $HOME/mympi
13. #run my executable
14. mpirun -np $PBS_NCPUS ./mympi
15. ja -chlst
```

Blacklight uses the Portable Batch System (PBS) scheduler. Lines 2,4,5, and 6 are PBS directives.

Submitting Batch Scripts

- Commands are machine specific, but follow general principles
- With PBS batch scripts, use the qsub command

```
qsub myscript.job
```

- Can also specify PBS directives as command-line options:

```
qsub -l ncpus=16 -l walltime=5:00 -j oe -q batch myscript.job
```

- Command-line directives override directives in your scripts.

More PBS commands

- qstat - displays the status of batch jobs.

-a	gives the status of all jobs on the system.
qstat -n	lists nodes allocated to a running job in addition to basic information.
qstat -f PBS_JOBID	gives detailed information on a particular job.
-q	provides summary information on all the queues.

- qdel – deletes a queued job or kills a running job.
- See the qsub manpage for more

Example Batch Command

```
qsub amber.job
qstat –a
Job ID  Username Queue  Jobname  SessID NDS  Tasks Memory Time  S Time
------- -------- ------ -------- ------ ---- ----- ------ ----- - -----
29668    user1   batch   job2     21909   1    256   --   08:00  R    02:28
29894    user2   batch   run128    --     1    128   --   02:30    Q   --
29895    user3   batch   STDIN    15921   1      1   --   01:00  R 00:10
29896    user2   batch   jobL     21988   1   2048   --   01:00 R 00:09
29897    user4   batch   STDIN    22367   1      2   --   00:30   R 00:06
29898    user1   batch   amber     25188   1      1   --   01:10  R
   00:00

qdel 29668
```

- After job 29898 runs: user1 should get file amber.job.o29898 with output/errors (log file)

Job Scheduling

- All XSEDE compute resources use a batch scheduler for running jobs.
- All jobs are placed in a batch queue after they are submitted.
- Resource User Guides on the XUP have details on your system's scheduler.

Batch Schedulers

- Attempt to balance queue wait times of competing jobs with efficient system utilization.
  - Job prioritization influenced by number of cores and wall clock time requested
  - FIFO queues with fair use mechanisms to keep a single user from dominating the queue
  - Backfilling unused nodes with smaller jobs
- Will not start jobs if they will not finish before scheduled system maintenance.

Common problems encountered when running jobs:

- Invalid number of cores were requested
- Job runs out of CPU time
- Files can’t be found
- Inadequate software permissions

Improving job turnaround

- Request accurate walltime
- Use flexible walltime
- Pack your job
  - Running many small jobs places a great burden on the scheduler and is also inconvenient for you.
  - Pack many executions into a single job, which you then submit to PBS with a single qsub command.

Requesting flexible walltime

```
-l walltime_min=HH:MM:SS
-l walltime_max=HH:MM:SS
```

Example: Your job requests 64 cores and a walltime between 2 and 4 hours. If there is a 64 core slot available for 3 hours, your job could run in this slot. However, if your job had requested a fixed walltime of 4 hours it would not run until the larger time slot becomes available.

Packing Serial Jobs

Run each program execution in the background and place a wait command after each execution.

```
#!/bin/csh
#PBS -l ncpus=96
#PBS -l walltime=5:00
#PBS -q batch
dplace -c 0 ./myserial1 < serial1.dat &
dplace -c 32 ./myserial2 < serial2.dat &
dplace -c 64 ./myserial3 < serial3.dat &
wait
```

Packing OpenMP Jobs

To pack OpenMP executables, replace the dplace command with the omplace command. Sample job to pack OpenMP executables:

```
omplace -nt 32 -c 0 ./myopenmp1 < myopenmp1.dat &

omplace -nt 32 -c 32 ./myopenmp2 < myopenmp2.dat &

omplace -nt 32 -c 64 ./myopenmp3 < myopenmp3.dat &

omplace -nt 32 -c 96 ./myopenmp4 < myopenmp4.dat &

wait
```

Managing Your Environment: **Modules**

- Allows you to manipulate your environment.
- 'module list' shows currently loaded modules.
- 'module avail' shows available modules.
- 'module show' <name> describes module.

http://modules.sourceforge.net/

```
% module load gcc/3.1.1
% which gcc
/usr/local/gcc/3.1.1/linux/bin/gcc

% module switch gcc/3.1.1 gcc/3.2.0
% which gcc
/usr/local/gcc/3.2.0/linux/bin/gcc

% module unload gcc
% which gcc
gcc not found
```

For the following exercise (same steps as before):

- Check to see if connection is still live, if not:
- For ssh to XSEDE SSO login hub (today!)

  ssh –l *username* login.xsede.org

  username on handout
- And from there go to your XSEDE resource, for example:

  gsissh comet.sdsc.edu

SDSC comet Cluster & Modules

- Default environment intel compilers, mvapich2 MPI implementation
- We will swap intel compilers with gnu compilers

  module swap intel gnu

  which gcc
- And then we'll load the openMPI library

  module load openmpi_ib

  which mpicc

Module demo on comet

```
-bash-4.1$ module swap intel gnu
Unloading compiler-dependent module tau/2.23
Need to load an mpi module before loading fftw/2.23
Unloading compiler-dependent module pdt/3.20
Unloading compiler-dependent module papi/5.4.1
Unloading compiler-dependent module tau/2.23
Need to load an mpi module before loading fftw/2.23
-bash-4.1$ module list
Currently Loaded Modulefiles:
  1) gnutools/2.69     2) globus/5.2.5      3) gnu/4.9.2         4) .intel/tau/2.23
-bash-4.1$ module load openmpi_ib
-bash-4.1$ which mpicc
/opt/openmpi/gnu/ib/bin/mpicc
-bash-4.1$
```

Exercise

- Make sure you are on comet.sdsc.edu
- Run the shallow water model code provided
- No input file needed
- Copy batch script from my home directory:

  cp ~ux400689/shallow-slurm.sb .

Job script

```
#!/bin/bash
#SBATCH --job-name="shallow"
#SBATCH --output="shallow.%j.%N.out"
#SBATCH --partition=shared
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=5
#SBATCH --export=ALL
#SBATCH -t 00:30:00

#This job runs with 1 nodes, 5 cores per node for a total of 5 cores.
#ibrun in verbose mode will give binding detail

ibrun -v ~ux400689/shallow/shallow
```

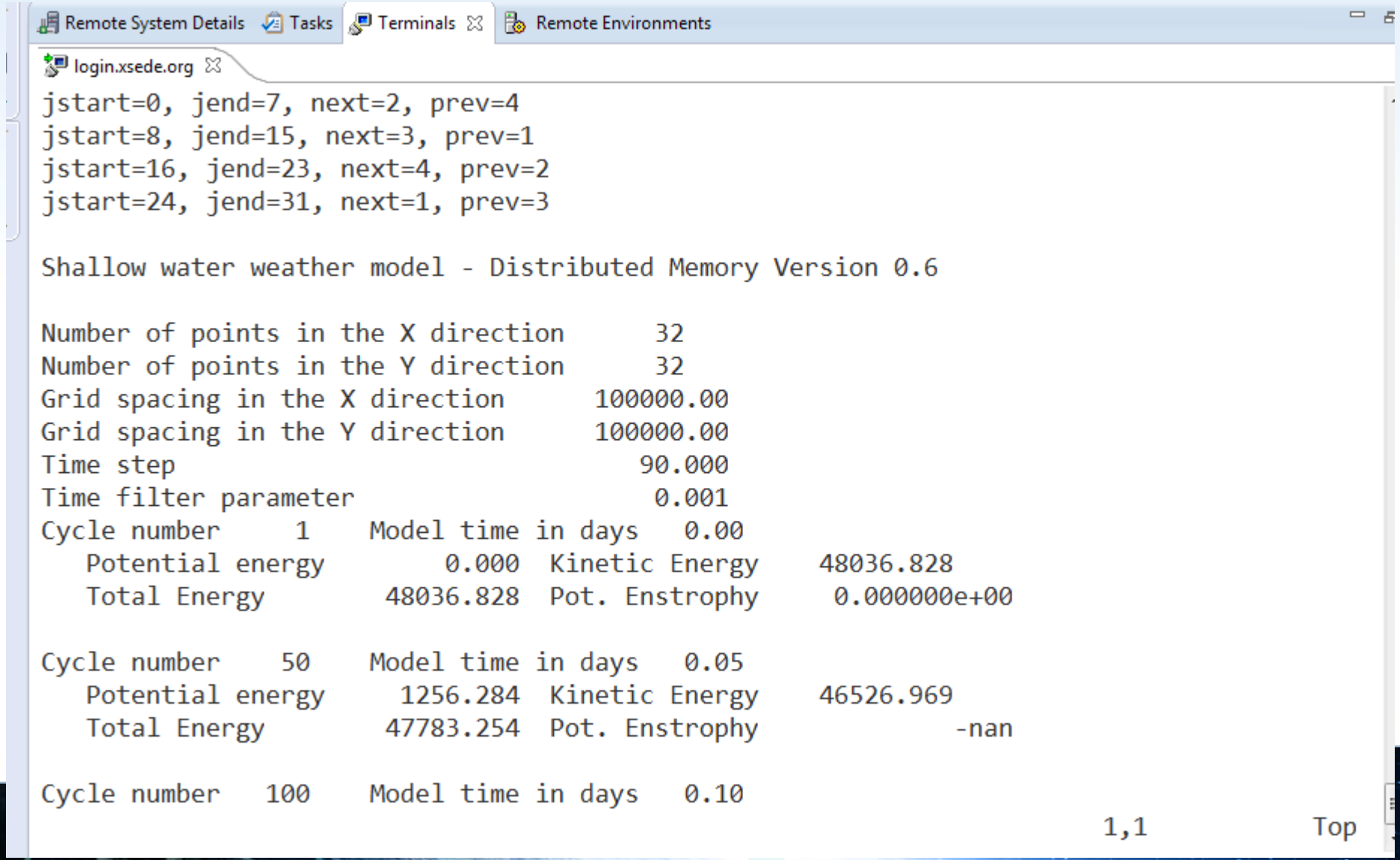
Exercise:

- Submit the job (sbatch shallow-slurm.sb)
- Monitor the job (squeue –u *username*)
- Make sure you have the output files at job completion

```
-bash-4.1$ lscalc.c  decs.h  eclipse.inc  Makefile                    shallow-batch.sh
tstep.f90calc.o  diag.c  init.c       shallow                     shallow-slurm.sb
tstep.ocopy.c  diag.o  init.o       shallow.582135.comet-03-56.out  time.c
worker.ccopy.o  dump.c  main.c       shallow.591445.comet-04-66.out  time.o
worker.oCVS     dump.o  main.o       shallow.591474.comet-04-66.out  tstep.c
-bash-4.1$
```

more shallow*out (for this case, yours will be different!)

Output files: need to show successful completion



Need help? Reporting and Tracking Issues

- portal.xsede.org → Help

  Submit ticket

- portal.xsede.org → My XSEDE → Tickets
  - Submit ticket
  - View past tickets (both open and closed)
- Can also email help@xsede.org or call 1-866-907-2383, at any hour (24/7)

More “helpful” resources

xsede.org→User Services

- Resources available at each Service Provider
  - User Guides describing memory, number of CPUs, file systems, etc.
  - Storage facilities
  - Software (Comprehensive Search)
- Training: portal.xsede.org → Training
  - Course Calendar
  - On-line training
- Get face-to-face help from XSEDE experts at your institution; contact your local Campus Champions.
- Extended Collaborative Support (formerly known as Advanced User Support (AUSS))

XSEDE Training Survey

- Please complete a short on-line survey about this module at http://bit.ly/hamptonxsede. We value your feedback, and will use your feedback to help improve our training offerings.
- Slides from this workshop are available at

  http://hpcuniversity.org/trainingMaterials/219

May 13, 2016

Thanks for listening and welcome to XSEDE!

XSEDE

Extreme Science and Engineering Discovery Environment