

# SDSC Summer Institute, Day 1

## Prep Day: Accounts, Login, Environments,

June 22, 2022

EXPANSE  
COMPUTING WITHOUT BOUNDARIES

Mary Thomas

San Diego Supercomputer Center

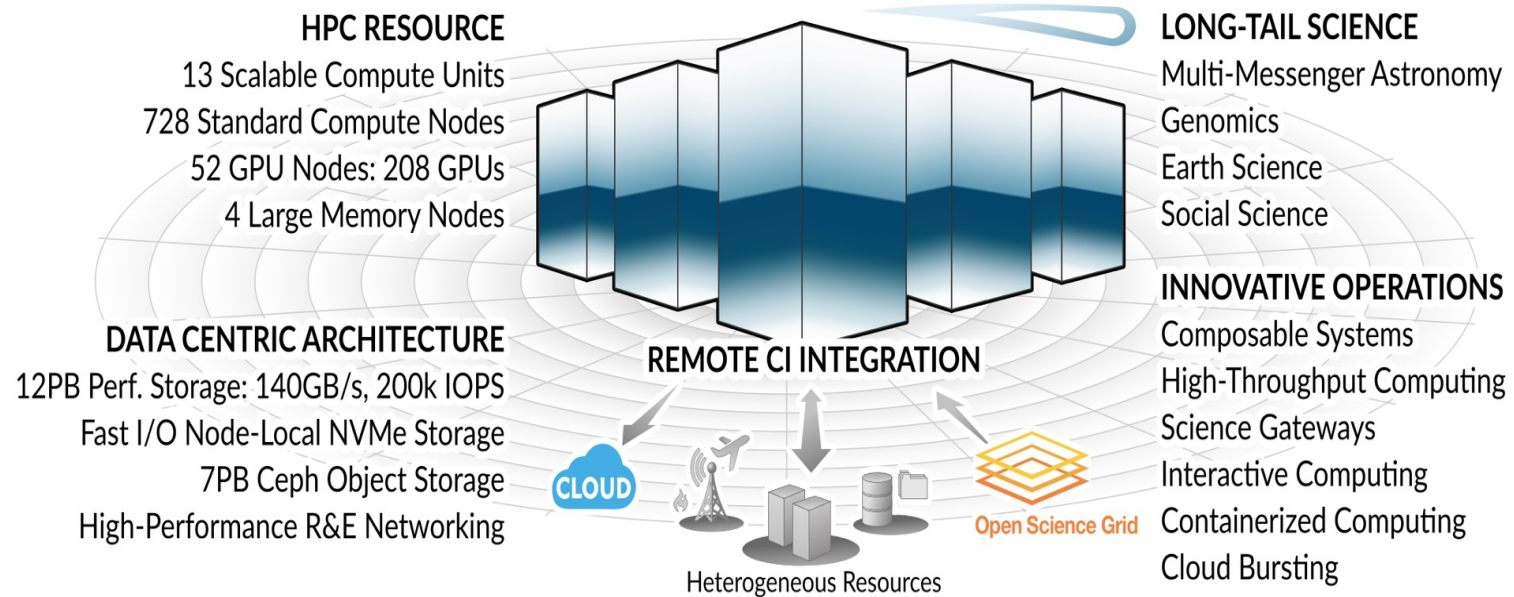
# Outline

- Introduction
- Connecting to Expanse
- Modules
- Account Management



# EXPANSE

COMPUTING WITHOUT BOUNDARIES  
5 PETAFL0P/S HPC and DATA RESOURCE



For more details see the Expanse user guide @ [https://www.sdsc.edu/support/user\\_guides/expanse.html](https://www.sdsc.edu/support/user_guides/expanse.html)  
and the "Introduction to Expanse" webinar @ [https://www.sdsc.edu/event\\_items/202006\\_Introduction\\_to\\_Expanse.html](https://www.sdsc.edu/event_items/202006_Introduction_to_Expanse.html)

# Expanse



# Outline

- Introduction
- Connecting to Expanse
- Modules
- Account Management

# Basic Skills Needed to Use HPC/CI Systems

[https://github.com/sdsc-hpc-training-org/basic\\_skills](https://github.com/sdsc-hpc-training-org/basic_skills)

- Overview/Understanding of HPC systems
  - See presentations on Day 2
- Understanding security and authentication
- Using Github
- Unix/Linux
- Understand the HPC system environment (accounts, modules, etc)

# Common Ways to Connect to Expanse

- Connect via a **Terminal application**
  - "Command Line" environment
  - Access **Interactive compute nodes**
- Connect Using the **Expanse User Portal**
  - Web Based Connection
  - built-in applications (Jupyter Notebooks, Matlab, R)
- Connect Using **Jupyter Notebooks/Lab**

Expanse User Guide: [https://www.sdsc.edu/support/user\\_guides/expanse.html](https://www.sdsc.edu/support/user_guides/expanse.html)



# Logging onto Expanse

- Expanse supports Single Sign-On through the XSEDE User Portal
- From the command line using an XSEDE-wide password,
- From the Expanse User Portal (<https://portal.expanse.sdsc.edu>).
- Note that CPU and GPU resources are **allocated separately**, but the login nodes are the same.
- To log in to Expanse from the command line
  - hostname: login.expanse.sdsc.edu
  - Secure shell (SSH) command examples:

```
ssh <user>@login.expanse.sdsc.edu  
ssh -l < user > login.expanse.sdsc.edu
```

- When you log onto *login.expanse.sdsc.edu*, you will be assigned one of the two login nodes login0[1-2]-expanse.sdsc.edu. Both systems are identical.

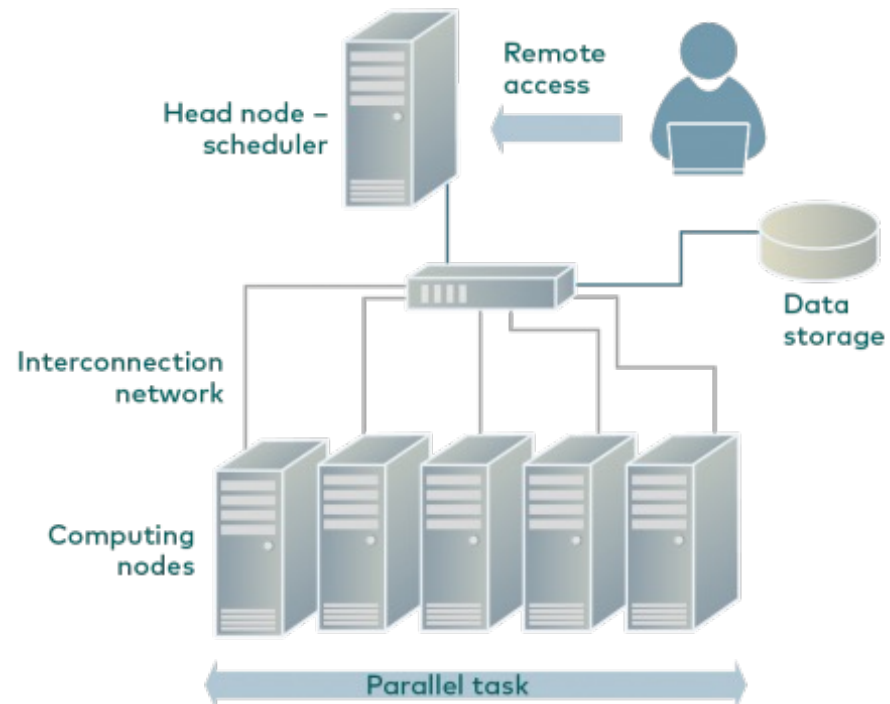


# Using SSH Keys

- You can append your public key (e.g. from your laptop) to your ~/.ssh/authorized\_keys file to enable access from authorized hosts without having to enter your password.
- RSA, ECDSA and ed25519 keys are accepted.
- Make sure you have a **strong passphrase** on the private key on your local machine.
- You can use **ssh-agent** forwarding to avoid repeatedly typing the private key password:
  - [https://github.com/sdsc-hpc-training-org/hpc-security/blob/master/ssh\\_methods/connect-using-ssh-agent.md](https://github.com/sdsc-hpc-training-org/hpc-security/blob/master/ssh_methods/connect-using-ssh-agent.md)
- Hosts which try to connect to SSH more frequently than ten times per minute may get blocked for a short period of time
- See the SDSC webinar: Indispensable Security: Tips to Use SDSC's HPC Resources Securely:
  - **2021 Webinar:** [https://www.sdsc.edu/event\\_items/202007\\_CometWebinar.html](https://www.sdsc.edu/event_items/202007_CometWebinar.html)
  - 2022 Webinar scheduled for April, 2022. Watch the training pages:
    - [https://www.sdsc.edu/education\\_and\\_training/training\\_hpc.html](https://www.sdsc.edu/education_and_training/training_hpc.html)

# System Access: Clients

- Linux/Mac –
  - use terminal + installed ssh app
- Windows:
  - Win10 terminal app + installed ssh app
  - Older Windows OS's: ssh clients apps Putty, Cygwin
- Login hostname for SDSC Expanse:
  - [login.expense.sdsc.edu](https://login.expense.sdsc.edu)
  - 198.202.113.252



Source: <https://hpc.rtu.lv/hpc/introduction-to-hpc/?lang=en>

For more on SDSC security, see: <https://github.com/sdsc-hpc-training-org/hpc-security>

# Connecting via Terminal Application

```
Welcome to Bright release      9.0

Based on Rocky Linux
8
ID: #000002

-----

WELCOME TO

  _____
 /___/ // __ V | /// ___//___/
/_/ | // // // // // ^__ V _/
//___/ V ___/ ___ // ^ /___//___/
/____// ^// / / / | // /___/___/

-----

Use the following commands to adjust your environment:

'module avail'      - show available modules
'module add <module>' - adds a module to your environment for this session
'module initadd <module>' - configure module to be loaded at every login

-----

Last login: Tue Jun 21 17:07:20 2022 from 76.176.117.51
```

# Terminal Connection: Command Line

```
[username@login02 ~]$ date
Tue Jul 26 23:14:42 PDT 2022
[username @login02 ~]$ hostname
login02
[username @login02 ~]$ squeue
  JOBID PARTITION  NAME   USER ST  TIME  NODES NODELIST(REASON)
  13574113 compute jobname username PD   0:00    2 (MaxMemPerLimit)
  12668967 compute jobname username PD   0:00    1 (MaxMemPerLimit)
[SNIP]
  14743360 shared jobname username  R  23:26:07    1 exp-6-33
  14743363 shared jobname username  R  23:26:07    1 exp-6-35
[username @login02 ~]$ squeue | wc
4394  35204  354029
[username @login02 ~]$ ls -al /cm/shared/examples/sdsc/openmp
total 23
drwxr-xr-x  2 mahidhar use300  6 Oct 13  2020 .
drwxrwxr-x 46 root    use300  44 Jul 25 16:58 ..
-rwxr-xr-x  1 mahidhar use300 19640 Oct 13  2020 hello_openmp
-rw-r--r--  1 mahidhar use300  236 Oct 13  2020 hello_openmp.f90
-rw-r--r--  1 mahidhar use300  672 Oct 13  2020 hello_openmp_shared.108737.exp-6-56.out
-rw-r--r--  1 mahidhar use300  442 Oct 13  2020 openmp-slurm-shared.sb
-rw-r--r--  1 mahidhar use300  168 Oct 13  2020 README.txt
-rw-r--r--  1 mahidhar use300  253 Oct 13  2020 test.sb
```

PD == Pending

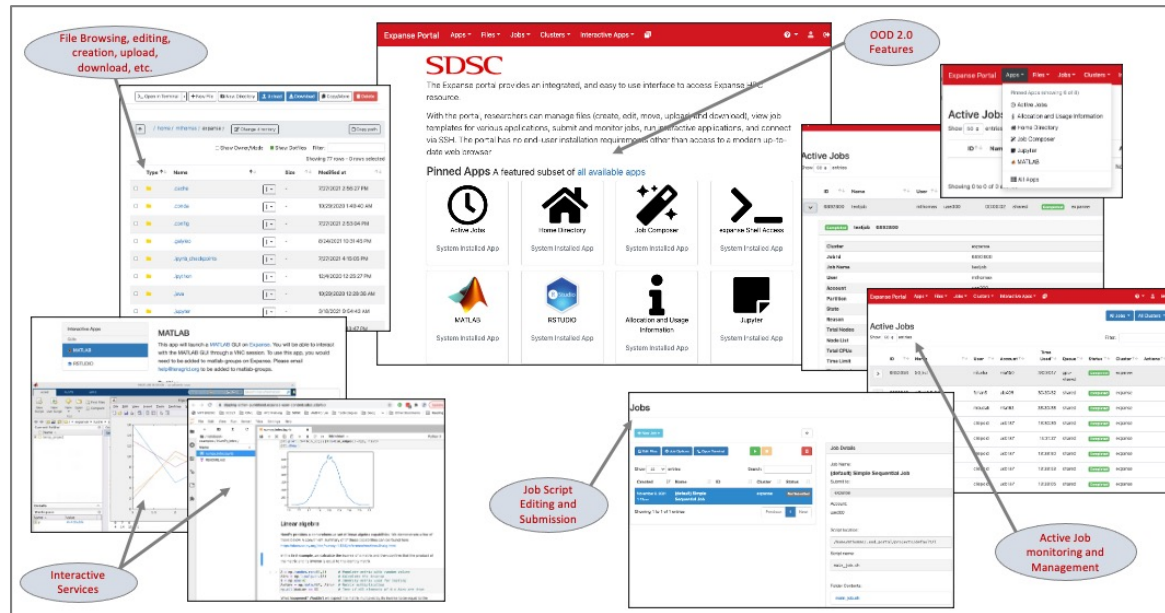
R == Running



# Using Login Nodes Properly

- The login nodes are meant for file editing, simple data analysis, & tasks that use minimal compute resources.
- All computationally demanding jobs should be submitted and run through the batch queuing system.
- **Do not use the login nodes for:**
  - computationally intensive processes,
  - as hosts for running workflow management tools
  - as primary data transfer nodes for large or numerous data transfers
  - as servers providing other services accessible to the Internet.
  - running Jupyter notebooks
- **Login nodes are not the same as the batch nodes.**
  - Users should request an interactive sessions to compile ;arge programs.

# Expanse User Portal



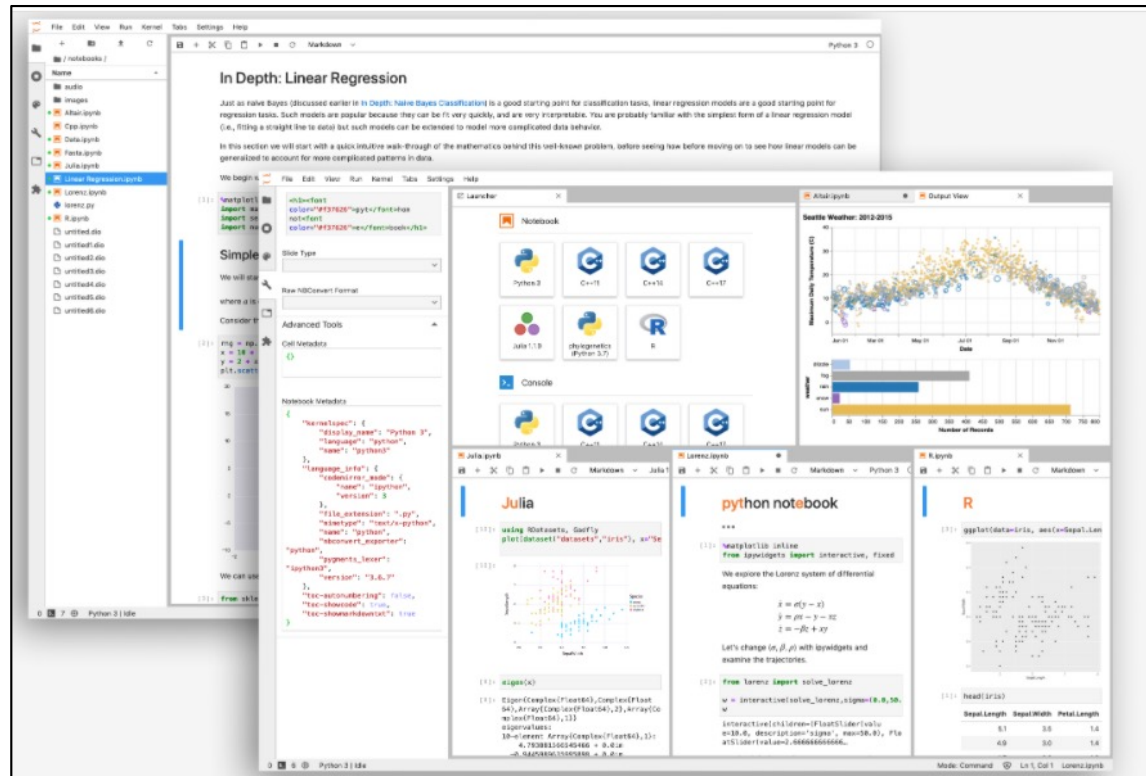
- <https://portal.expanse.sdsc.edu>
- Access using XSEDE credentials
- Securely hosts batch job submission & monitoring, and interactive applications

Sivagnanam, S., Irving, C., Kandes, M., Mishin, D., Sakai, S., Strande, S., Tatini, M., Thomas, M., Norman, M. (2021). Experiences in building a user portal for Expanse supercomputer. *PEARC21: Practice and Experience in Advanced Research Computing*, Accepted. ACM. <https://doi.org/10.1145/3437359.3465590>

# Jupyter Notebooks

## Common Jupyter Services:

- Jupyter Notebooks (single user)
- JupyterLab: advanced version of notebook
- JupyterHub: multiuser Jupyter service



# Outline

- Introduction
- Connecting to Expanse
- Modules
- Account Management



# Expanse Environment Modules

- Expanse uses *Lmod*, a *Lua* based module system.
  - [https://lmod.readthedocs.io/en/latest/010\\_user.html](https://lmod.readthedocs.io/en/latest/010_user.html)
- Users setup custom environments by loading available modules into the shell environment, *including needed compilers and libraries* and the batch scheduler.
- What modules let you do:
  - Dynamic modification of your shell environment
  - User can set, change, or delete environment variables
  - User chooses between different versions of the same software or different combinations of related codes.

# Modules on Expanse

- Users will need to load the scheduler (e.g. slurm)
- Users will *not* see all available modules when they run command "module available" *without loading a compiler*.
- Use the command "**module spider**" option to see if a particular package exists and can be loaded, run command
  - `module spider <package>`
  - `module keywords <term>`
- For additional details, and to identify module dependencies modules, use the command: `module spider <application_name>`
- The **module paths are different** for the CPU and GPU nodes. Users can enable the paths by loading the following modules:
  - `module load cpu` (for cpu nodes)
  - `module load gpu` (for gpu nodes)
  - avoid loading both modules

# Modules: Popular commands

Command	Description
module list	List the modules that are currently loaded
module avail	List the modules that are available in environment
module spider	List of the modules and extensions currently available
module display <module_name>	Show the environment variables used by <module name> and how they are affected
module unload <module name>	Remove <module name> from the environment
module load <module name>	Load <module name> into the environment
module swap <module one> <module two>	Replace <module one> with <module two> in the environment
module help	get a list of all the commands that module knows about do:
Shorthand notation: ml foo ml -bar	"ml" == module load foo "ml -bar" == module unload bar

*SDSC Guidance: add module calls to your environment and batch scripts*

# Module Command Examples

```
[user@login02 ~]$ module reset
```

Resetting modules to system default. Resetting \$MODULEPATH back to system default. All extra directories will be removed from \$MODULEPATH.

```
[user@login02 ~]$ module list
```

Currently Loaded Modules:

1) shared 2) cpu/0.15.4 3) slurm/expense/21.08.8 4) sdsc/1.0 5) DefaultModules

List Current  
environment

```
user@login02 ~]$ module avail
```

```
----- /cm/shared/apps/spack/cpu/lmod/linux-centos8-x86_64/intel/19.1.1.217 -----
bamtools/2.5.1      grace/5.1.25      libpng/1.6.37      openmpi/4.0.4 (D)
bedtools2/2.27.1    gsl/2.5           libtirpc/1.2.6      papi/6.0.0.1
[SNIP]
eigen/3.3.7         jasper/2.0.16     openmpi/3.1.6

----- /cm/shared/apps/spack/cpu/lmod/linux-centos8-x86_64/Core -----
abaqus/2018          emboss/6.6.0      gmp/6.1.2           parallel/20200822
anaconda3/2020.11    freesurfer/7.1.1  go/1.15.1           pciutils/3.7.0
[SNIP]
cmake/3.18.2         gcc/9.2.0         mpfr/4.0.2          zstd/1.4.5
curl/7.72.0          gcc/10.2.0        nbo/7.0-openblas
doxygen/1.8.17       gh/1.13.1         openjdk/11.0.2

----- /cm/local/modulefiles -----
boost/1.71.0  cmjob  lua/5.3.5  shared  singularitypro/3.5  slurm/expense/20.02.3 (L)
----- /cm/shared/apps/xsede/modulefiles -----
cue-login-env  xdinfo/1.5-1  xdusage/2.1-1
----- /usr/share/modulefiles -----
DefaultModules  cpu/0.15.4 (L)  gct/6.2  globus/6.0  gpu/0.15.4  nostack/0.15.4
----- /cm/shared/modulefiles -----
AMDuProf/3.4.475  default-environment  sdsc/1.0
```

Show  
available  
modules

List will  
change  
as you  
change  
the  
modules  
you are  
using

Where:

L: Module is loaded

D: Default Module



# Module Command Examples

```
[user@login02 ~]$ module show matlab
```

```
-----  
/cm/shared/apps/spack/cpu/lmod/linux-centos8-x86_64/Core/matlab/2020b.lua:  
-----
```

```
whatis("Name : matlab")  
whatis("Version : 2020b")  
whatis("Target : zen")  
whatis("Short description : MATLAB (MATrix LABoratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.")  
help([[MATLAB (MATrix LABoratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#,  
[SNIP]prepend_path("PATH","/cm/shared/apps/spack/cpu/opt/spack/linux-centos8-zen/gcc-8.3.1/matlab-2020b-o5lrpkierrdeih7cfjaxm3yofipxbff6/bin")  
prepend_path("CMAKE_PREFIX_PATH","/cm/shared/apps/spack/cpu/opt/spack/linux-centos8-zen/gcc-8.3.1/matlab-2020b-o5lrpkierrdeih7cfjaxm3yofipxbff6")  
setenv("MATLABHOME","/cm/shared/apps/spack/cpu/opt/spack/linux-centos8-zen/gcc-8.3.1/matlab-2020b-o5lrpkierrdeih7cfjaxm3yofipxbff6")
```

Use "module show" to find out what a particular module will change in the environment

# Outline

- Introduction
- Connecting to Expanse
- Interactive Computing
- Modules
- Account Management

# Multiple Allocations/Projects

- Many users will have access to multiple accounts and hence *projects*:
  - an allocation for a research project and a separate allocation for classroom or educational use
- Users should verify that the correct *project* is designated for all batch jobs.
- Awards are granted for a specific purposes and should not be used for other *projects*.
- To charge your job to one of your *projects*, replace << project >> with one from your list and put this PBS directive in your job script:
  - `#SBATCH -A << project >>`
  - `srun -account=<< project >>`

# Managing Your User Account

- The **expanse-client** script provides additional details regarding project availability and usage. The script is located at:

```
[username @login02 ~]$ ll /cm/shared/apps/sdsc/current/bin/expanse-client  
-rwxr-xr-x 1 dmishin hpc-admin 9962184 May 19 14:25 /cm/shared/apps/sdsc/current/bin/expanse-client
```

- To use the script you may need to load the 'sdsc' module.

```
[username @login02 ~]$ module load sdsc  
[username @login02 ~]$ module list  
Currently Loaded Modules:  
  1) shared      3) DefaultModules      5) sdsc/1.0  
  2) cpu/0.15.4  4) slurm/expanse/21.08.8
```



# Allocation Information: User

To review your available projects on Expanse resource, use the 'user' parameter

```
[username@login02 ~]$ [mthomas@login02 ~]$ expanse-client user
```

Resource expanse

	NAME	STATE	PROJECT	TG PROJECT	USED	AVAILABLE	USED BY PROJECT
1	username	allow	ddp363		65	50000	4022
2	username	allow	sds173	TG-CCR190013	1	50000	683
3	username	allow	sds184	TG-TRA210003	0	50000	10276
4	username	allow	use300		2209	5050000	3452324

# Allocation Information: Resource

To see full list of available resources, use the 'resource' parameter:

```
[mthomas@login02 ~]$ expanse-client resource  
Available resources:  
expanse  
expanse_gpu  
expanse_industry  
expanse_industry_gpu
```

To review your available projects on a specific Expanse resource, use the 'user -r resource' parameter:

```
[mthomas@login02 ~]$ expanse-client user -r expanse_gpu  
Resource expanse_gpu
```

	NAME	STATE	PROJECT	TG PROJECT	USED	AVAILABLE	USED BY PROJECT
1	username	allow	ddp363		0	2500	61
2	username	allow	sds173	TG-CCR190013	6	2500	116
3	username	allow	sds184	TG-TRA210003	0	2500	697
4	username	allow	use300		80	269000	63637

# Charging

- Charge unit for all SDSC machines, including Expanse, is the Service Unit (SU).
  - 1 CPU core using  $\leq 2\text{G}$  of data for 1 hour
  - 1 GPU using  $< 96\text{G}$  of data for 1 hour
  - 'shared' partitions: based on either # of cores or fraction of memory requested, whichever is larger
- Charges based on resources used by job regardless of use.
- Charges are based on either # of cores or fraction of the memory requested, whichever is larger.
- **Minimum charge for any job is 1 SU:**
  - *Can quickly use up SUs if you run a lot of very short jobs.*
- More details in Expanse user guide

[https://www.sdsc.edu/support/user\\_guides/expanse.html#charging](https://www.sdsc.edu/support/user_guides/expanse.html#charging)

# Resources

- Expanse User Guide & Tutorial
  - [https://www.sdsc.edu/support/user\\_guides/expanse.html](https://www.sdsc.edu/support/user_guides/expanse.html)
  - <https://hpc-training.sdsc.edu/expanse-101/>
- Clone code examples for this tutorial:
  - <https://github.com/sdsc-hpc-training-org/expanse-101>
- SDSC Training Resources
  - [https://www.sdsc.edu/education\\_and\\_training/training\\_hpc.html](https://www.sdsc.edu/education_and_training/training_hpc.html)
- XSEDE Training Resources
  - <https://www.xsede.org/for-users/training>
  - <https://cvw.cac.cornell.edu/expanse/>

# Thank You