Introduction to Midway2

Alina Gafanova, Linghui Wu

BFI

References:

- https://rcc.uchicago.edu/docs/
- Clindaniel MACS-30123

Contents

What is and why Midway2?

- High-performance computing platform at the UChicago's Research Computing. Other computing resources such as Acropolis Cluster.
- Necessary and important when running out of computing powers on your local machine
  - Storing and working with heavy data files
  - Parallel computing of computationally expensive tasks

Requesting an account

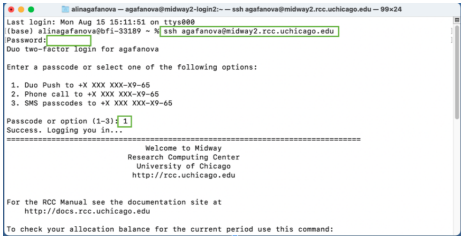
- RPs request a General User Account associated with their PIs account via https://rcc.uchicago.edu/accounts-allocations/general-user-account-request.
- Your PI might or might not have resources allocated to their account. Allocation request can be submitted through https://rcc.uchicago.edu/accounts-allocations/research-i-allocation-request.

Connecting to Midway2: terminal

- Same credential as CNetID login and passwords
- Open terminal and enter

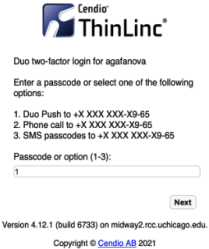
  `ssh <CNetID@midway2.rcc.uchicago.edu>`

- It will prompt you to enter password and proceed with the regular two-factor authentication (2FA).



Connecting to Midway2: graphical interface

- You can also connect and use Midway2 without terminal (though less efficient).



Checking resources

Once logged in, you can check your allocation, storage and etc.
Enter `rcchelp` in terminal and you will see a list of commands.

- `rcchelp balance`: remaining and total allocation
- `rcchelp quota`: remaining and total space of data storage
- When working with large data, check on your storage from time to time.

Data storage on Midway has a hard and a soft limit.
Once you go over a soft limit, you can still store new data but only during a grace period.
After a grace period, you won't be able to store new data on server anymore.

Navigating Midway2 folders

- Two main commands
  - cd for changing directory
  - `ls` for listing files
- For work, we would use PI's directory since it is a shared folder and generally has more space.
  - `cd /project2/<PI CNetID>`
  - `cd /project/<PI CNetID>`
- Enter `cd ~` to get back to your home directory `/home/<CNetID>`.

Transferring files

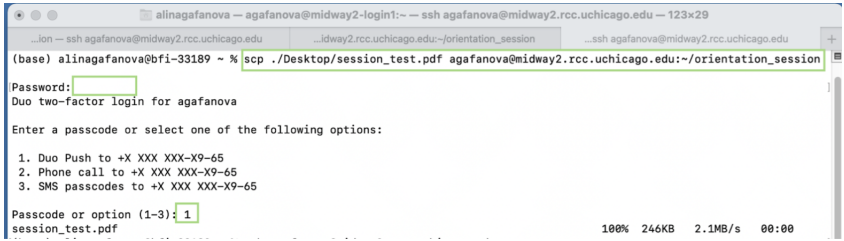
- Transfer a file from local machine to your home directory

  `$ scp <some file> <CNetID>@midway2.rcc.uchicago.edu:<path>`

- Transfer a directory to your home directory on Midway2

  `$ scp -r <some dir> <CNetID>@midway2.rcc.uchicago.edu:<path>`

- You will be prompted to enter your password and 2FA



Cloning git repository

- You may need to clone git repository to a specific directory on Midway2 to access files and code, pull updates and push changes.

```
$ git clone https://github.com/linghui-wu/RP-orientation-2022.git
```

- If any adjustments to file before/after running the jobs on Midway is needed, you can do so with command text editors such as nano, vim, and etc.

```
$ nano <file_name>
$ vim <file_name>
```

Loading modules

- Before running jobs, you need to make sure appropriate software modules have been loaded

```
$ module avail <module_name>
$ module load <modle_name>
$ module list
```

Run interactive jobs

- Intensive computational tasks are not recommended on the login nodes.
- You can specify memory allocation, runnning time, partition and etc by passing arguments.

```
$ sinteractive --time=00:15:00 --ntasks=4 --partition=broadwl --nodes=2
```

Run parallel batch jobs with Slurm scripts

- Simple julia script: `hello_world.jl`
- Slurm script used for running parallel jobs: `julia.sbatch`
- To run the Julia script, simply submit the jobs to the cluster

```
$ sbatch julia.slurm
```

- After the jobs complete, you can check the results in `.out` or debug in `.err` files
- Monitoring or canceling your job status

```
$ squeue --user=$USER
$ scancel <your_job_ID>
```