Logging into Engaging

1. In the terminal:
    - `ssh orcd-login`
    - duo push authentication
    - `orcd-login` is an alias for the `orcd-login001` hostname I set in my ssh config file (`C:\Users\munib.ssh.config)
    - in `settings.json` file, you can specify that the login node is linux so you don't have to specify it each time (`C:\Users\munib\AppData\Roaming\Code\User\settings.json)
2. allocate resources
    - check which partitions you have access to with `sinfo` and see a summary of each node's resources here
        - `mit_normal` for cpu-only and longer batch jobs
        - `mit_quicktest` for short batch jobs and interactive jobs that only need a cpu
        - `mit_normal_gpu` for batch and interactive jobs that need a gpu
        - Flavell lab has access to `bcs` partitions. see here for resources
            - d
    - Requesting a single core for an interactive job for 1 hour
        - `salloc -t 01:00:00 -p mit_normal`
    - Check current job status with `squeue --me`
    - Stop a job with `scancel [jobid]`
        - stop multiple with `scancel [jobid], [jobid2], ..`
        - stop all user's jobs with `scancel [username]`
    - Retrieve job stats with `sacct`
    - See here about info for requesting resources
        - tasks, nodes, tasks per node, memory, cores
        - a guide to finding how much memory your job will need
        - GPU requests
        - another nice resource on cores and such
3. ssh into compute node in VS code (terminal will automatically switch from login to compute if you set `ProxyJump` to `orcd-login` (or whatever the login node `HostName` is set to in your ssh config file).
    - `ssh orcd-compute` **OR**
    - `Connect to remote host` in VS code

Using python

- `module load miniforge`
- create a conda environment or load one

Using jupyter notebooks

Data storage

- we have 60TB available at `/orcd/data/flavell/001`
    - unclear if backed up
- each user has !TB available at `/orcd/home/002/munib/orcd/pool`
    - not backed up

- each user's home directory at `/orcd/home/002/munib` has 200GB
    - backed up
- each user's scratch directory at `/orcd/home/002/munib/orcd/pool` has 1TB
    - not backed up
- check how much available storage in a directory `df -h /orcd/data/flavell/001`

File transfer

The most recommended method to transfer data is Globus because it is safe and efficient. You can do all operations on a web browser on your computer. Refer to Globus user guide for details. Use the endpoint mithpc#openmind for OpenMind and use the collection MIT ORCD Home Collection for Engaging. (Might also be MIT Engaging Collection)

You can also consider using `rclone` to transfer data. Refer to the `rclone` section on this page.

It is less recommended to use `rsync` to transfer data because the transfer speed is relatively slow. If you want to use rsync for some reason, refer to this page for details.

Git and github

1. Follow these instructions to make an SSH key
2. Initialize a git repo:
    - `git init`
3. Switch your remote to SSH:
    - `git remote set-url origin git@github.com:MunibH/g5ht-pipeline.git`
4. Test:
    - `ssh -T git@github.com`
    - Should say **successfully authenticated**
5. Can you now stage, commit, and push

OTHER

- max jobs you can submit on Engaging is 500
    - in practice I found it was only 440 (I had one job open which was the original allocation on the login node)
        - `sacctmgr show user $USER withassoc`