Mandatory

cluster setup + basic linux commands

- Start here: intro to grid PDF

scripting

- https://missing.csail.mit.edu/
  If you have gaps in scripting knowledge, this is a good reference.

scratch space

- For e.g. datasets and experiment outputs, use the group's drive `/shared/share_mala`
- Keep your code in your home directory ~/

github: keep track of your (and others') code

- Why github?
  - It helps you keep track of changes you made. For example, if you made a mistake, you can go back to an earlier version.
  - It lets you collaborate with others in a sensible way.
  - Github is how people in the ML community share their code.
- Cheatsheet

Conda environments

Optional but really good

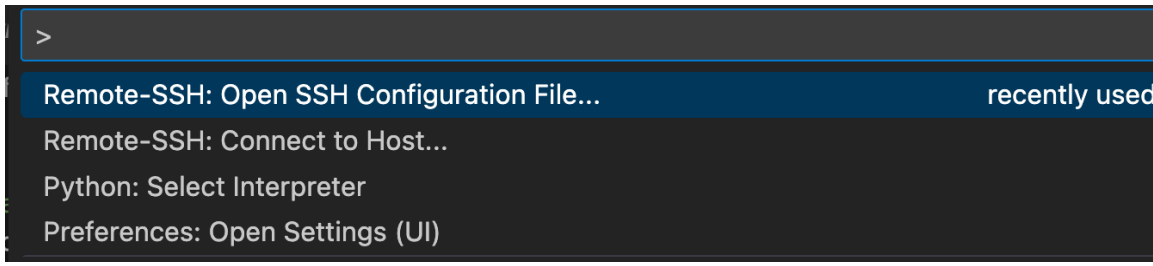
tmux: windows in the command line

- Why tmux?
  a. Just like how you have multiple tabs open in a browser to switch between, you'll probably want multiple windows open in your command line. (You can also have panes to switch between in a window, and also sessions to switch between that have windows: sessions -> windows -> panes)
     - Put this in your ~/.tmux.conf for an easy way to manage splitting a window into panes.
  b. A tmux session continues running even when you log off. This lets you run things even when you're not around.
- Cheatsheet

jupyter notebook: interactive python in your browser

- Why interactive python?
  - It's easier to tell what's going on with interactive python than running a script where you can't intervene.
- Why jupyter notebook?

    - You can use interactive python in the command line, but jupyter notebook is more convenient for e.g. plotting and saving what you've done already.
- To run on gsb: `grid_run --grid_mem=50G /user/tc3100/.conda/envs/testenv/bin/jupyter-notebook --ip=\$\(hostname\).gsb.columbia.edu`
    - Except replace `/user/tc3100/.conda/envs/testenv/bin/jupyter-notebook` with your version
- To run on gsb using VSCode
    - set up ssh configuration files
        - Command + shift + P (mac) -> Remote-SSH: Open SSH Configuration File...

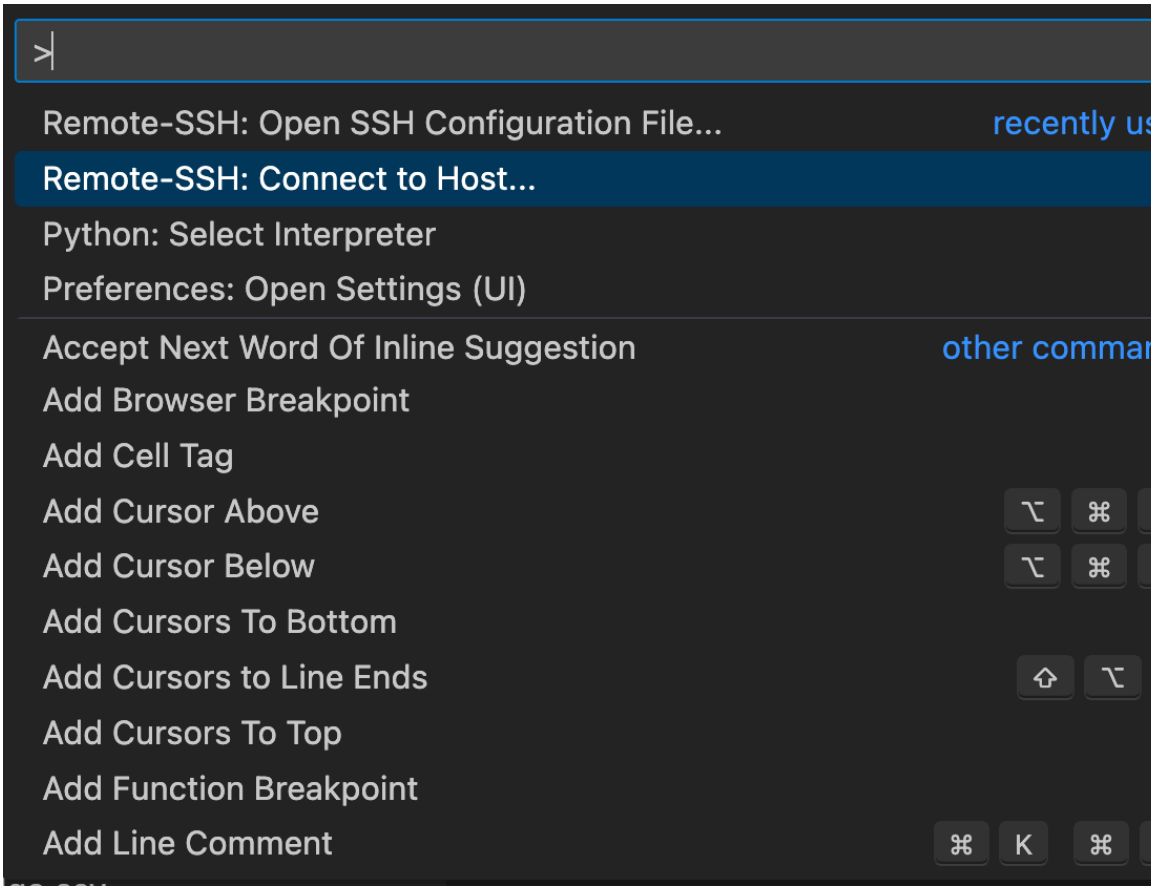


        - Add host name, user, etc
          An example: replace gsb1, gsb2, gsb3, gsb with your preferred abbreviation and replace User by your own uni

    - Connect to the ssh host
        - Command + shift + P (mac) -> Remote-SSH: Connect to Host…
        -

- This is what my .ssh/config file would look like

```
Host gpu01
     HostName researchgpu01.gsb.columbia.edu
     User hn2369

Host gpu02
     HostName researchgpu02.gsb.columbia.edu
     User hn2369

Host gpu03
     HostName researchgpu03.gsb.columbia.edu
     User hn2369

Host gpu04
     HostName researchgpu04.gsb.columbia.edu
     User hn2369

Host gpu05
     HostName researchgpu05.gsb.columbia.edu
     User hn2369

Host login
     HostName research.gsb.columbia.edu
     User hn2369

Host gpu06
    HostName researchgpu06.gsb.columbia.edu
    User hn2369
    ProxyJump hn2369@research.gsb.columbia.edu

Host gpu07
    HostName researchgpu06.gsb.columbia.edu
    User hn2369
    ProxyJump hn2369@research.gsb.columbia.edu
```



- (Optional) create ssh-key https://code.visualstudio.com/docs/remote/ssh-tutorial#_create-an-ssh-key
- choose the host -> open the .ipynb file -> choose the kernel on the upper right corner -> run the jupyter notebook file

How to submit jobs using grid_run

- configurate the python file (e.g., test.py)
  - add this line at the start of test.py
    `#!/user/[uni]/.conda/envs/[env_name]/bin/python`
    replace uni and env_name with your own version
  - in your terminal
    `chmod 700 test.py`
    to change permission of test.py
  - in your terminal, try
    `./test.py`
    to see if it the python file runs as expected
- submit jobs using grid_run
  - ssh to the cluster (instead of a specific GPU)
    `ssh uni@research.gsb.columbia.edu`

- cd to the folder that contains test.py and submit jobs using grid run
  `grid_run [options] ./test.py`
- A typical option for grid_run
  --grid_mem=200G --grid_submit=batch --grid_gpu

Deep learning setup

pytorch

- https://pytorch.org/tutorials/
  - 60 minute pytorch tutorial
- Vision: CS231 at Stanford and code
- NLP: CS224N at Stanford
- This repo has pretty good plug-and-play code for training basic deep nets on CIFAR10.
- **We strongly recommend maintaining a separate conda env for these purposes.**
- Fast ai class may be a reasonable short reference (it's lower quality than other references)

gpus

- To access Hong's GPUs, you can directly ssh into them
  `ssh uni@researchgpu01.gsb.columbia.edu`
- There are 5 GPUs in total (researchgpu01 to researchgpu05). 04 and 05 are very beefy; they're unoccupied now but you should use 01-03 for these exercise level jobs. You can run `nvidia-smi` to see which jobs are running.