

# Getting started on subMIT: How to Interact with subMIT

**subMIT Annual Workshop 2025**

<https://indico.mit.edu/event/1276/>

**Matt Heine 1/30/2025**



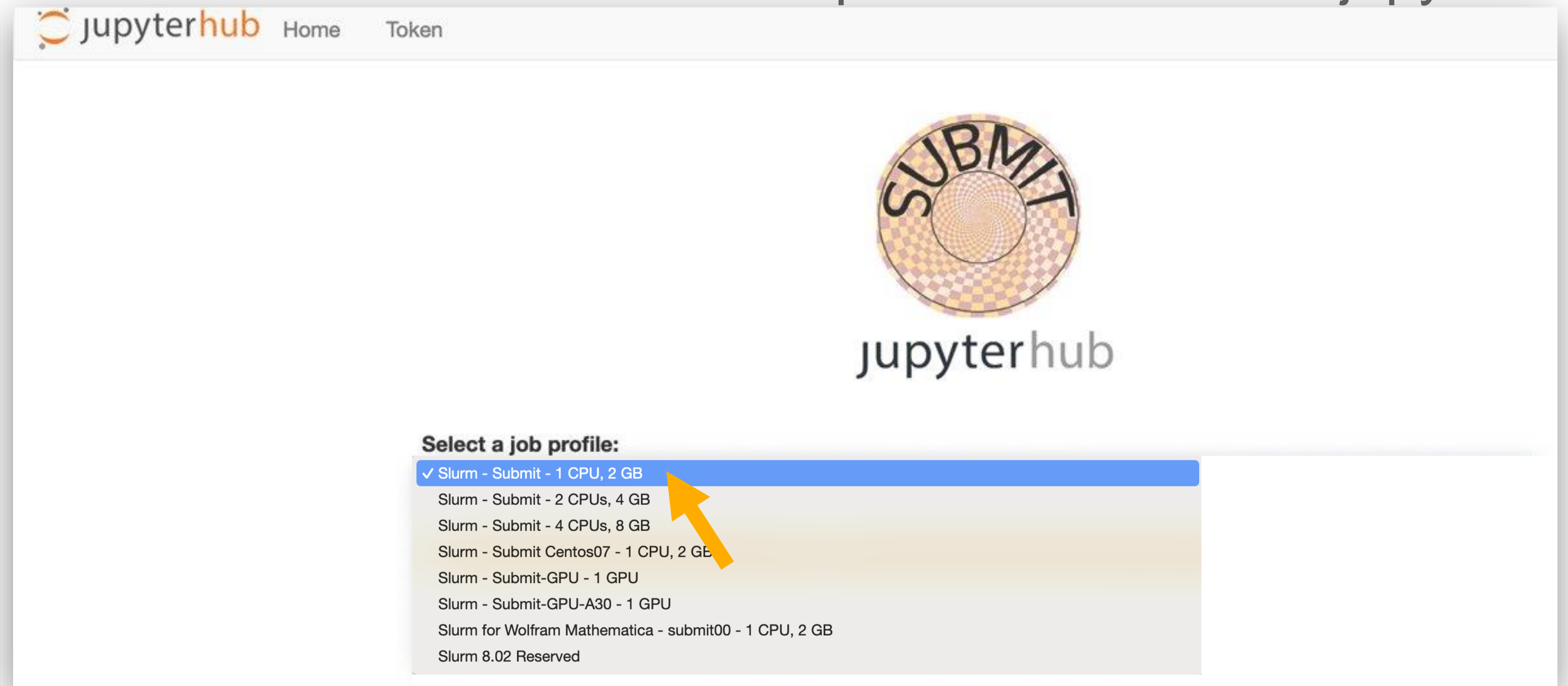
# Outline

- JupyterHub
- Visual Studio Code (VSCoDe)
- terminal / ssh
- X2Go

# JupyterHUB

- Access subMIT from a web browser
- Create/Run Jupyter Notebooks
- Graphical Interface + built-in terminal (in web browser)
- Easy interactive access to compute node resources, including GPUs  
(Not just login nodes)

<https://submit.mit.edu/jupyter/>



The screenshot shows the JupyterHub web interface. At the top left, there is a navigation bar with the JupyterHub logo, "Home", and "Token". In the center, there is a large circular logo with the word "SUBMIT" inside, and the text "jupyterhub" below it. At the bottom, there is a "Select a job profile:" dropdown menu. The first option, "✓ Slurm - Submit - 1 CPU, 2 GB", is selected and highlighted in blue. A yellow arrow points to this option. Other options in the menu include "Slurm - Submit - 2 CPUs, 4 GB", "Slurm - Submit - 4 CPUs, 8 GB", "Slurm - Submit Centos07 - 1 CPU, 2 GB", "Slurm - Submit-GPU - 1 GPU", "Slurm - Submit-GPU-A30 - 1 GPU", "Slurm for Wolfram Mathematica - submit00 - 1 CPU, 2 GB", and "Slurm 8.02 Reserved".

# JupyterHUB

- File Browser
- Jupyter Notebooks: Self-Contained
  - Code
  - Results / Visualization
  - Documentation (Markdown, LaTeX)
  - Easily shared
- Kernels = sets of software / packages used to run code in your notebook
  - Use your conda environments as kernels (automatic setup)
  - Use singularity images (containers) as kernels
  - Change kernels w/ a click
- Mathematica accessible via JupyterHub
- Many languages (even w/in same notebook)
- Many extensions
  - Debugging
  - Source Control (git / GitHub)

subMIT User's Guide:

<https://submit.mit.edu/submit-users-guide/program.html#jupyterhub>

<https://submit.mit.edu/submit-users-guide/program.html#jupyterhub-for-mathematica>

## A Quick Example

In [1]:

```
%%latex
```

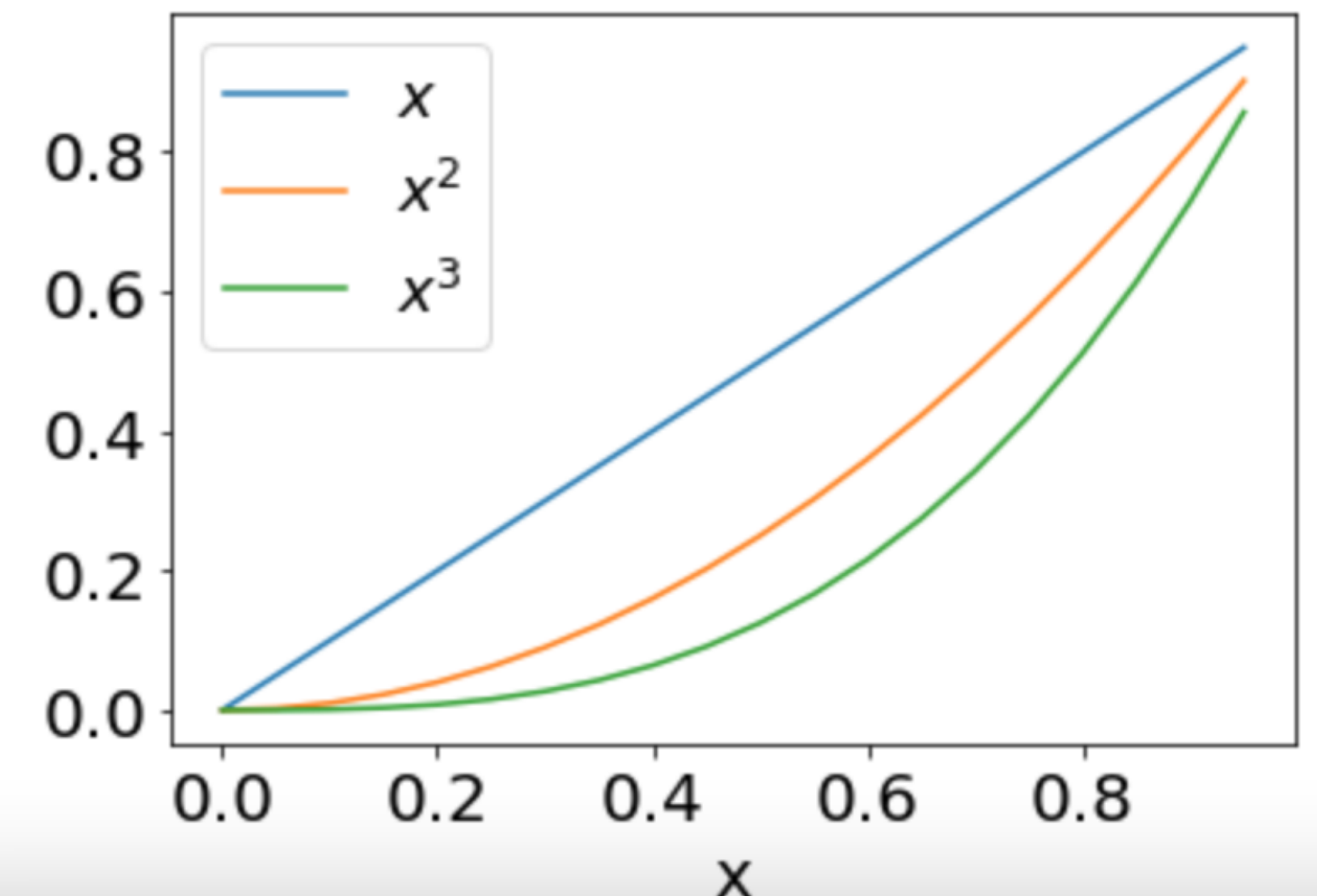
Here we will plot  $x$ ,  $x^2$ , and  $x^3$

Here we will plot  $x$ ,  $x^2$ , and  $x^3$

In [7]:

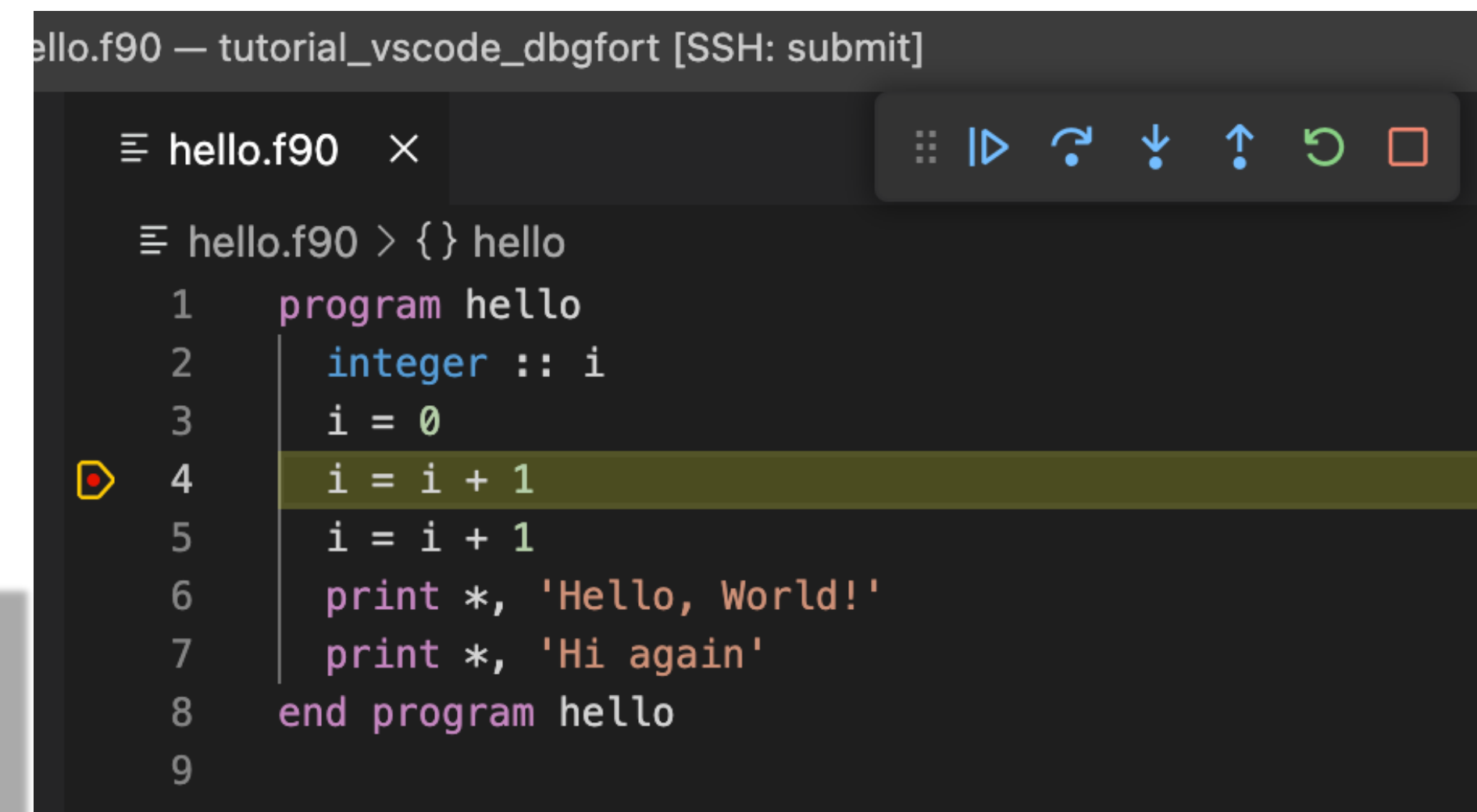
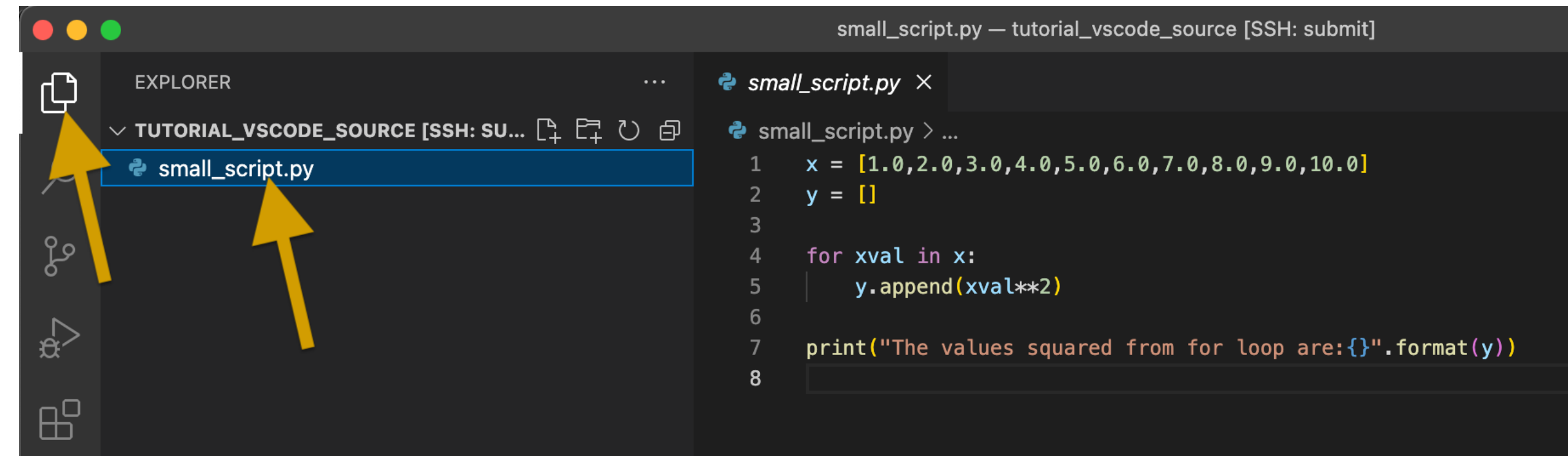
```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 1, 0.05);
plt.plot(x, x, label='$x$');
plt.plot(x, x**2, label='$x^2$');
plt.plot(x, x**3, label='$x^3$');
plt.xlabel('x');
plt.rc('font', size=18);
plt.legend();
```



# Visual Studio Code: Remote Development

- code runs on subMIT, GUI runs on your laptop
- File Browser
  - GUI to navigate/view your subMIT files/directories
- Many languages / extensions
  - Python, C/C++, Java, Julia, Fortran, ...
  - LaTeX, HTML/CSS, Markdown, rst, ...
- Code navigation
- Debugging (code runs on subMIT cluster)
  - breakpoints, variable inspection/watch, stack navigation
- Source Control (Integrated / GUI)
- Automatic Code Completion
  - Intellisense
  - Snippets, AI-assisted development
- subMIT User's Guide:
  - <https://submit.mit.edu/submit-users-guide/program.html#vscode>
  - Tutorials: <https://submit.mit.edu/submit-users-guide/#tutorials-submit>



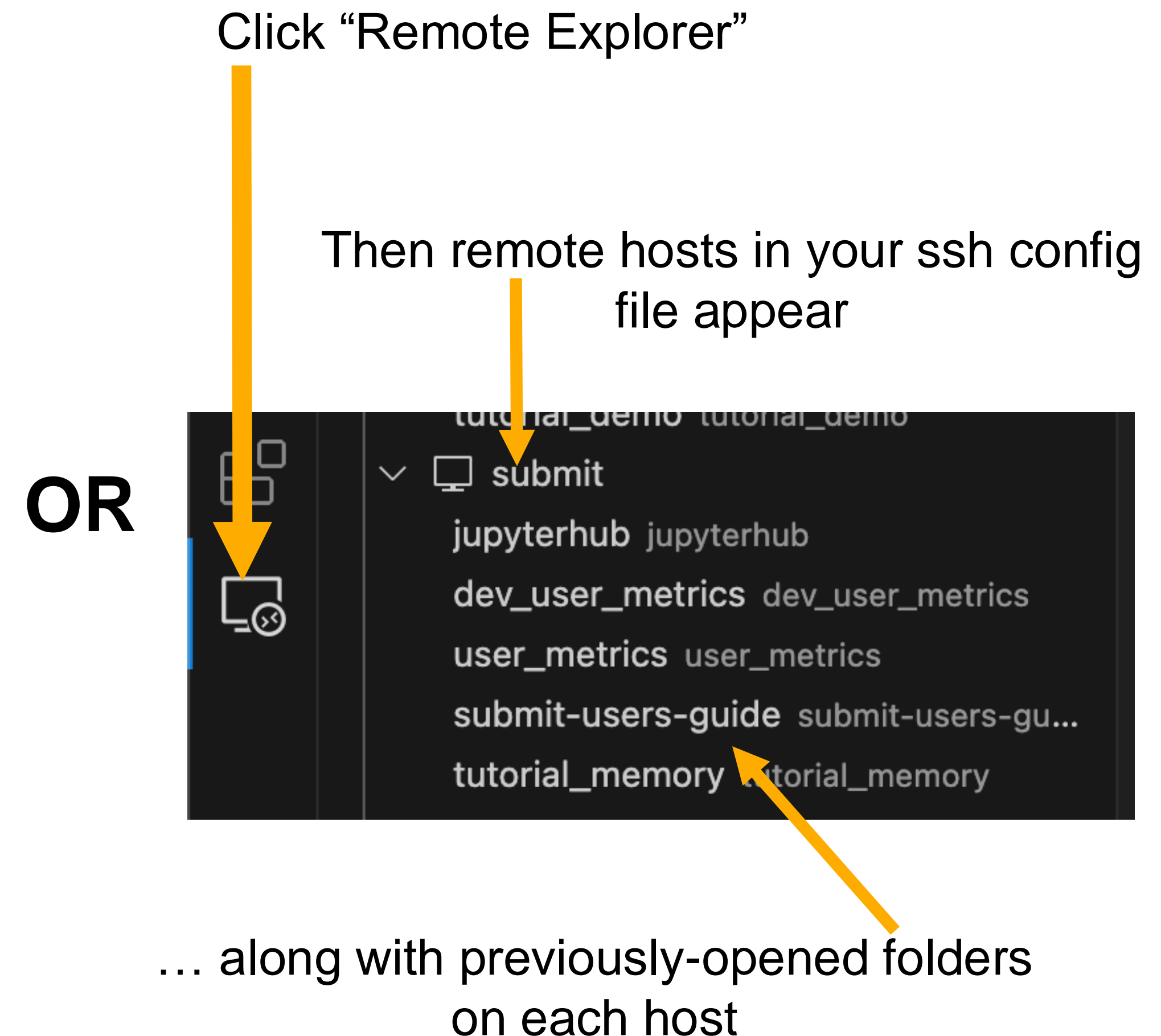
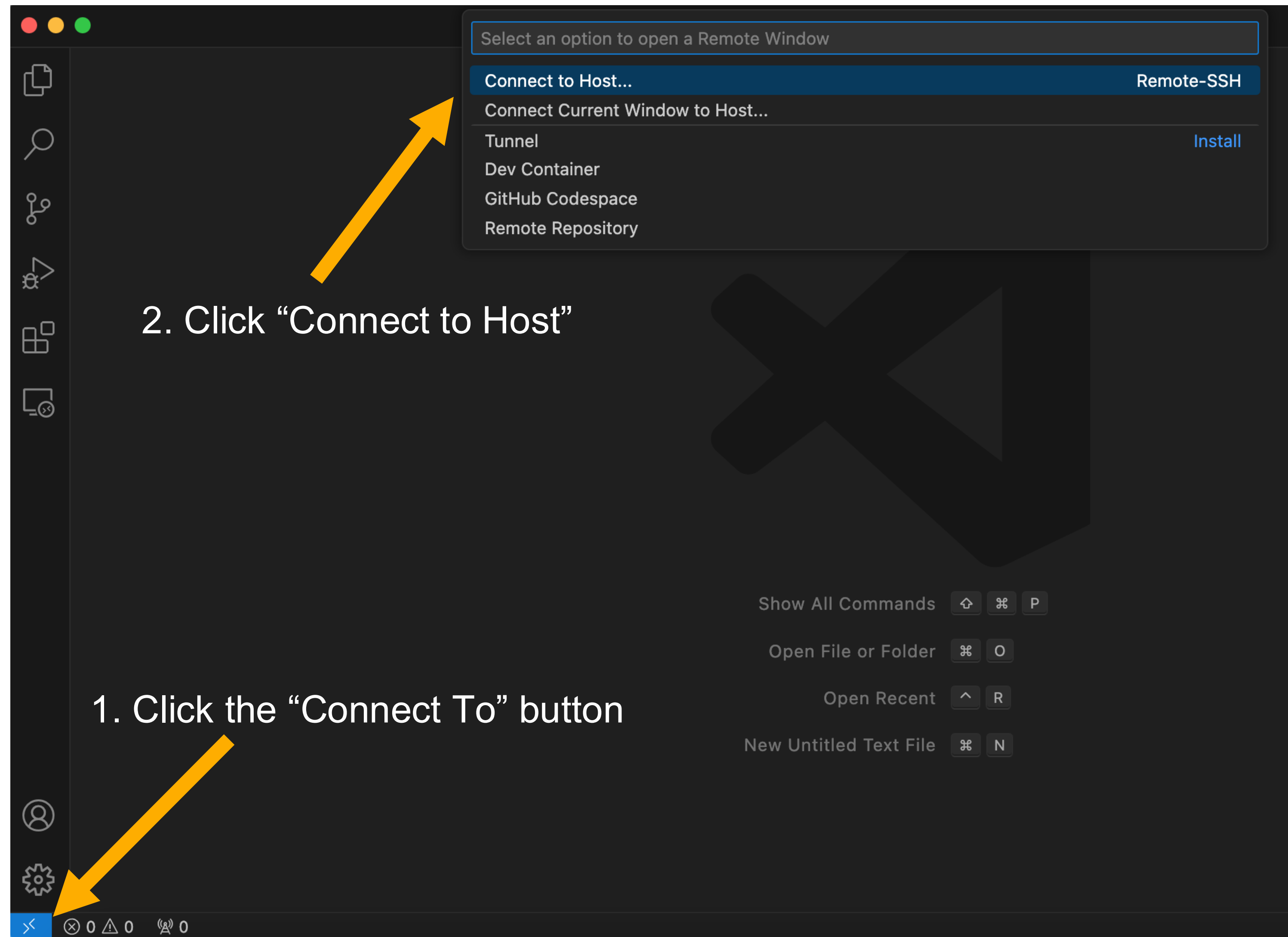
Change conda environments (on subMIT) for python code w/ a click

# Visual Studio Code: How to Connect

**Note:** This puts you on a login node  
Only for relatively light computation use (debugging)

subMIT User's Guide:

<https://submit.mit.edu/submit-users-guide/program.html#getting-started-with-vscode-on-submit>



# Terminal / SSH

- Classic method of interaction

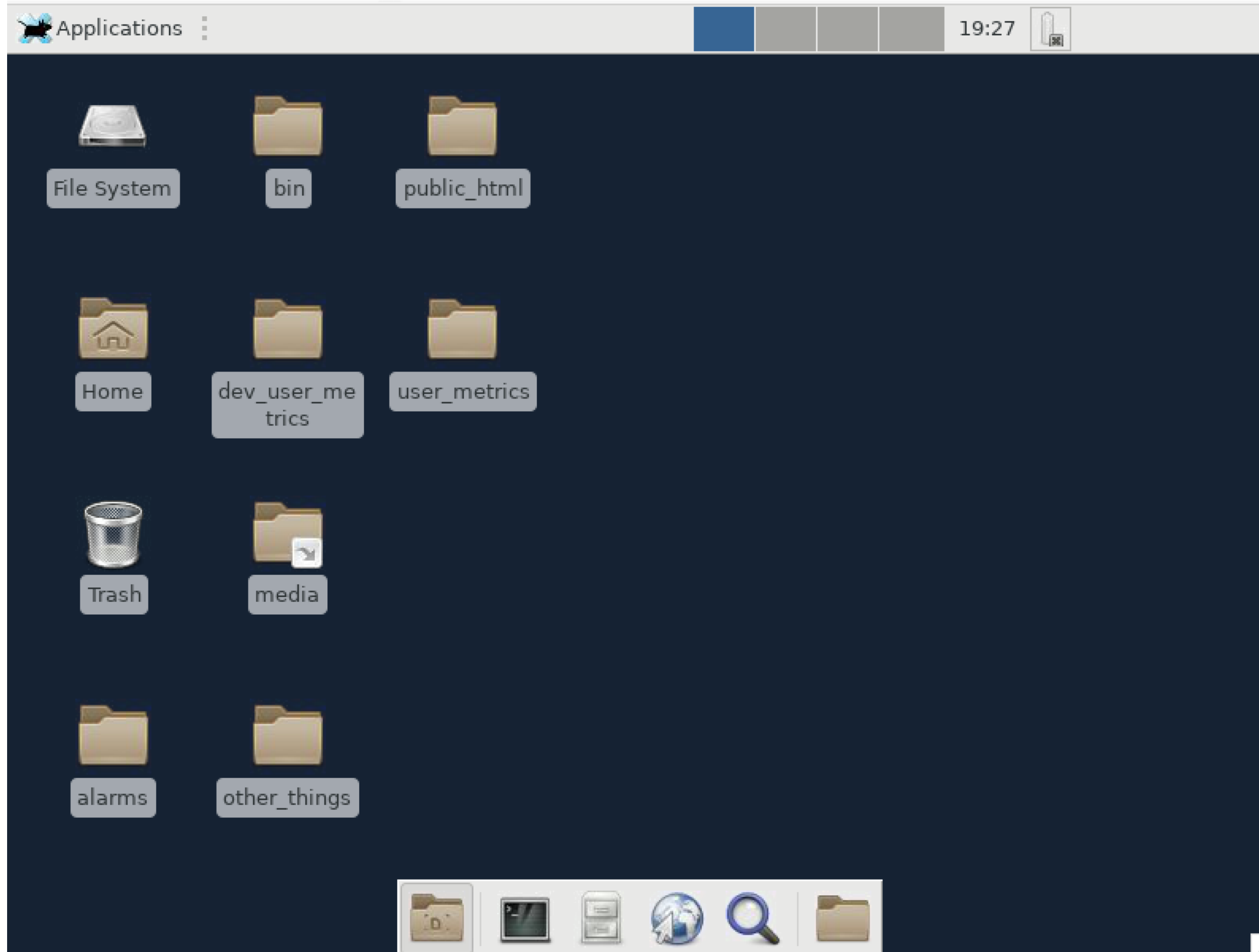
```
ssh <username>@submit.mit.edu
```

**Note:** This puts you on a login node  
Only for relatively light computation use (debugging)

Use `salloc` to get interactive session on compute nodes for heavy use

- MacOS / Linux : built-in. Windows : Windows Subsystem for Linux (WSL)
- Handy tip: set up ssh config file:
  - <https://submit.mit.edu/submit-users-guide/starting.html#common-issues-with-keys>
- subMIT User's Guide
  - Intro to terminal: [https://submit.mit.edu/submit-users-guide/tutorials/tutorial\\_0.html](https://submit.mit.edu/submit-users-guide/tutorials/tutorial_0.html)

# X2Go



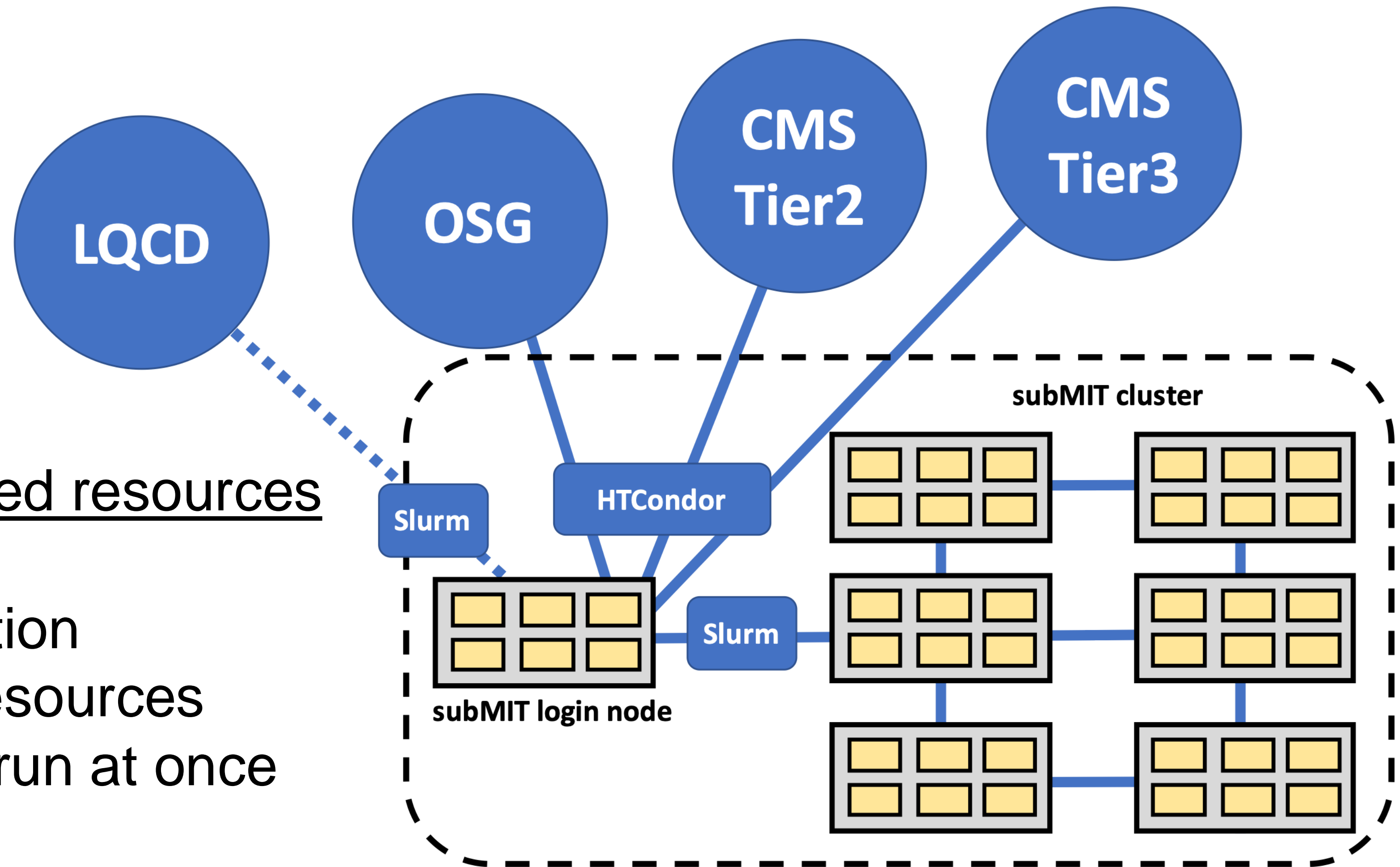
**Note:** This puts you on a login node  
Only for relatively light computation use (debugging)

- Reserved for GUI-heavy needs where the other options do not suffice  
  
(Otherwise, the other options are preferred)
- Familiar GUI interaction w/ the cluster
- Easy alternative to manual X11 Forwarding
- Run GUI applications with little/no setup
- subMIT User's Guide:  
<https://submit.mit.edu/submit-users-guide/program.html#x2go>



# Batch Jobs (overview)

- See the tutorial later this morning for more info!
- What are Batch jobs?
  - request/reserve dedicated resources (nodes, cores, memory)
  - run without user interaction
  - may wait in queue for resources
  - one or many submitted/run at once



- Login nodes are only for light usage
- “Heavy lifting” (significant usage of resources) should be done via batch jobs