

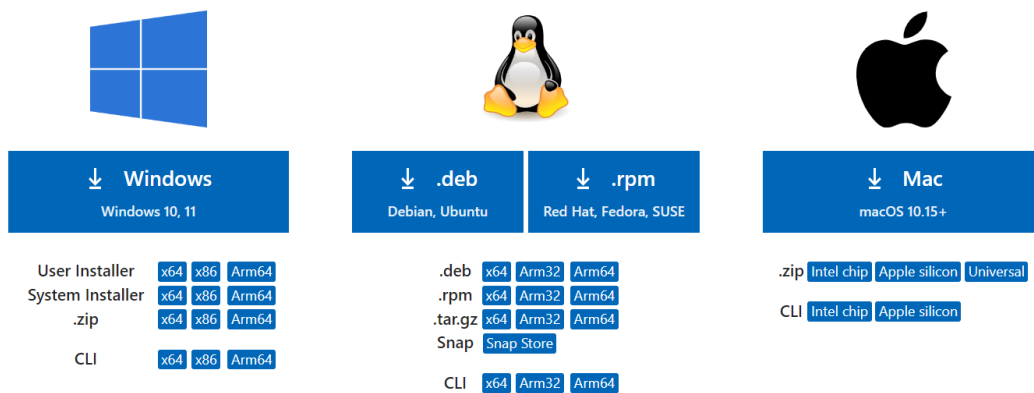
Turing GPU and Environment Tutorial

1. Vscode Installation:

- Download and setup Vscode on your laptop using this link:
<https://code.visualstudio.com/download>.
- Choose the version appropriate for your system (Windows or Mac).

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

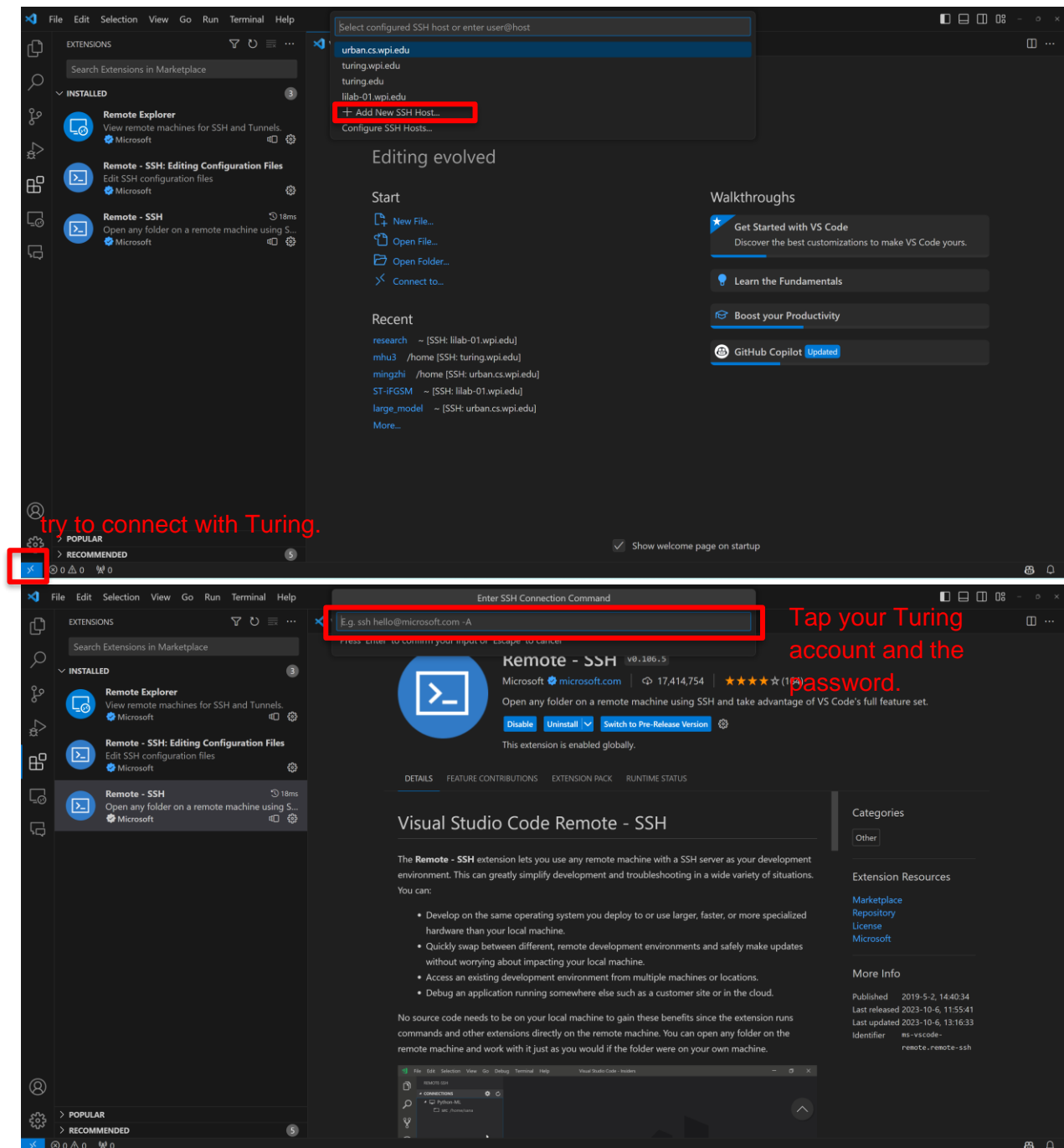


2. Connect to the WPI Network

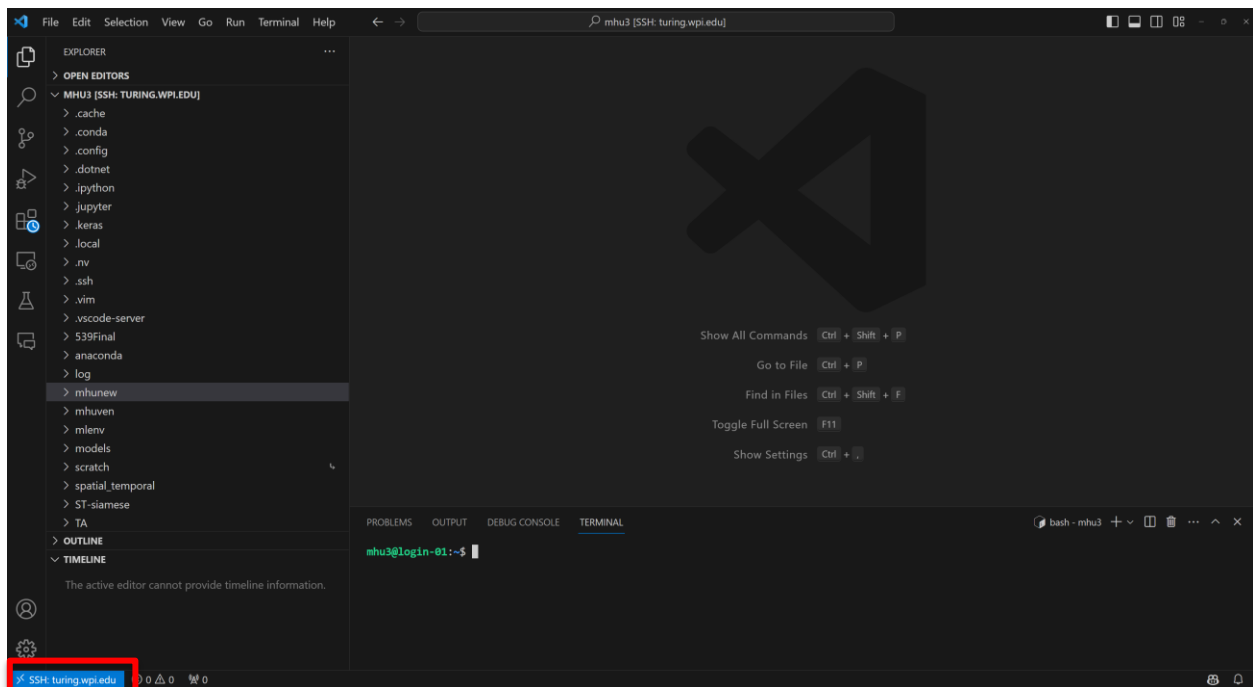
- Connect directly if you're on campus.
- Use the VPN following (<https://hub.wpi.edu/article/444/globalprotect-vpn-client-configuration>).

3. Open Vscode and try to connect to the Turing:

Vscode will automatically install the extensions of SSH for you.

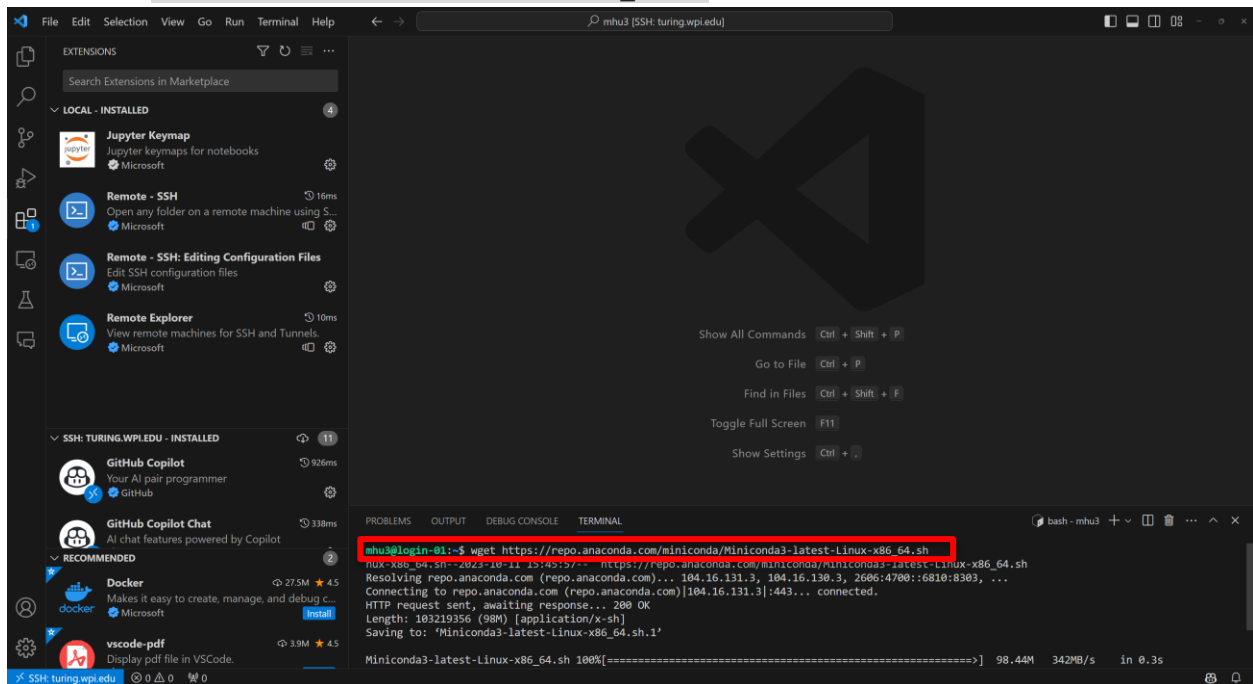


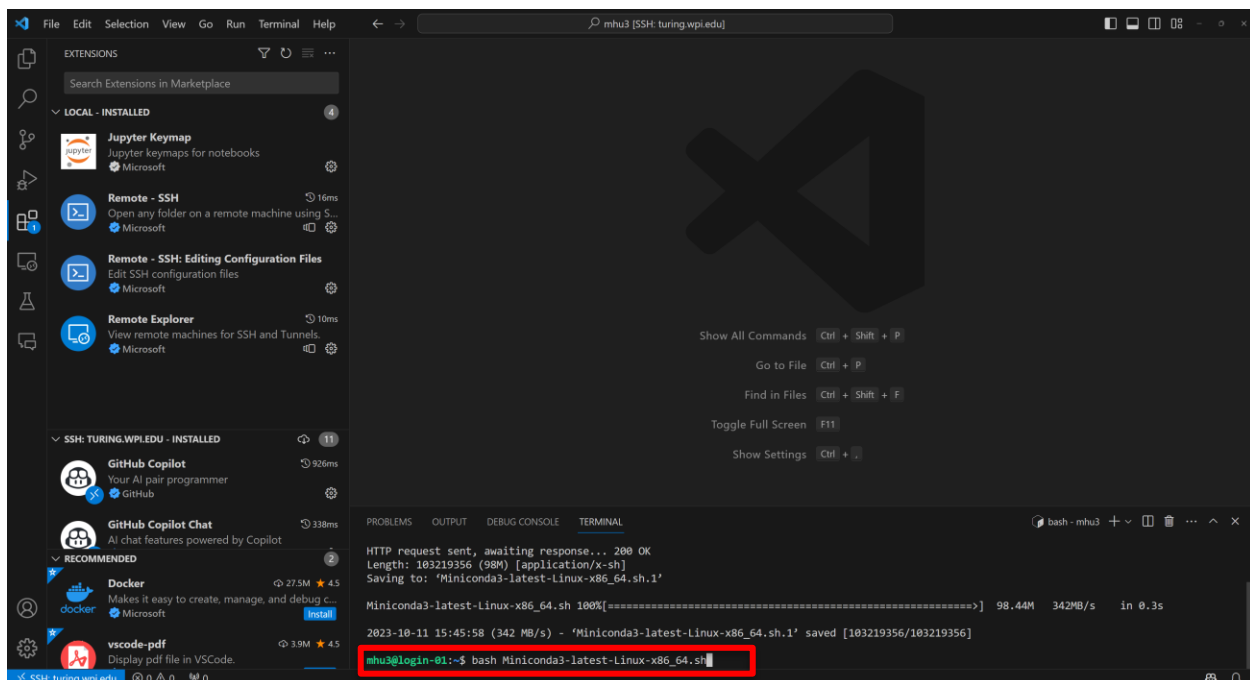
Now you are connected to Turing.



4. Download conda to the server and install it.

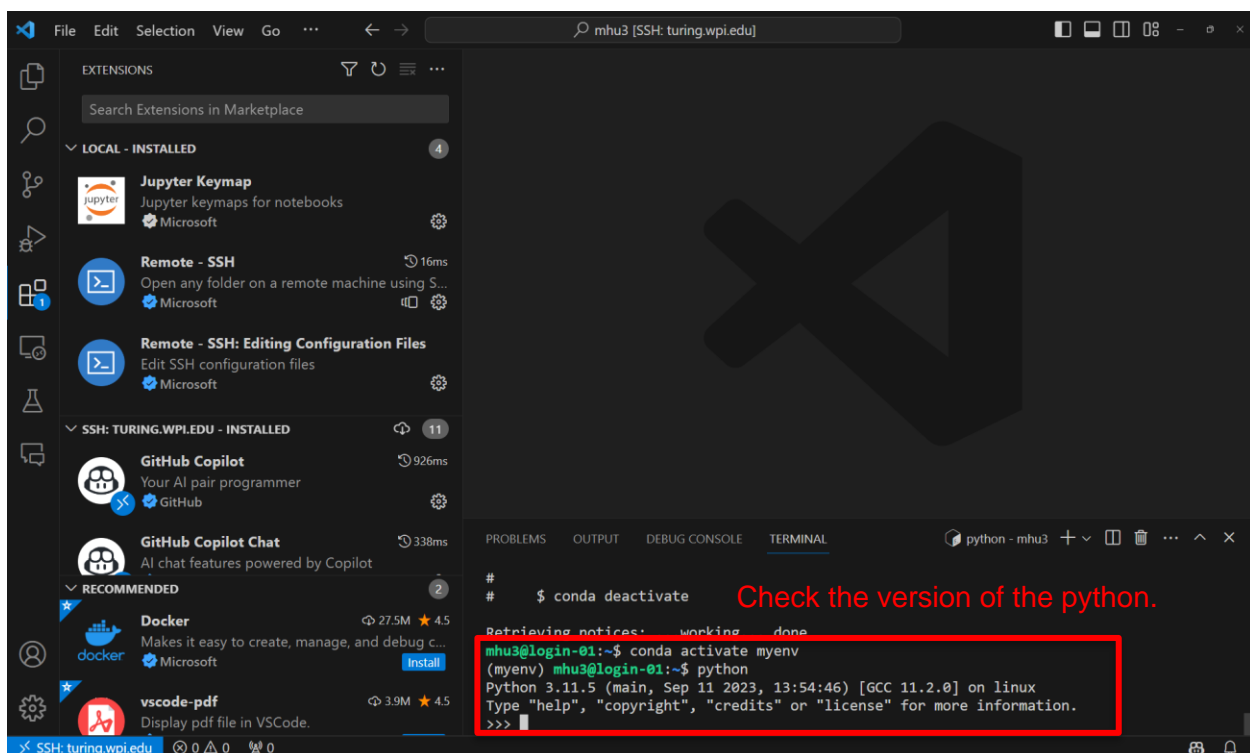
- `wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh`
- `bash Miniconda3-latest-Linux-x86_64.sh`



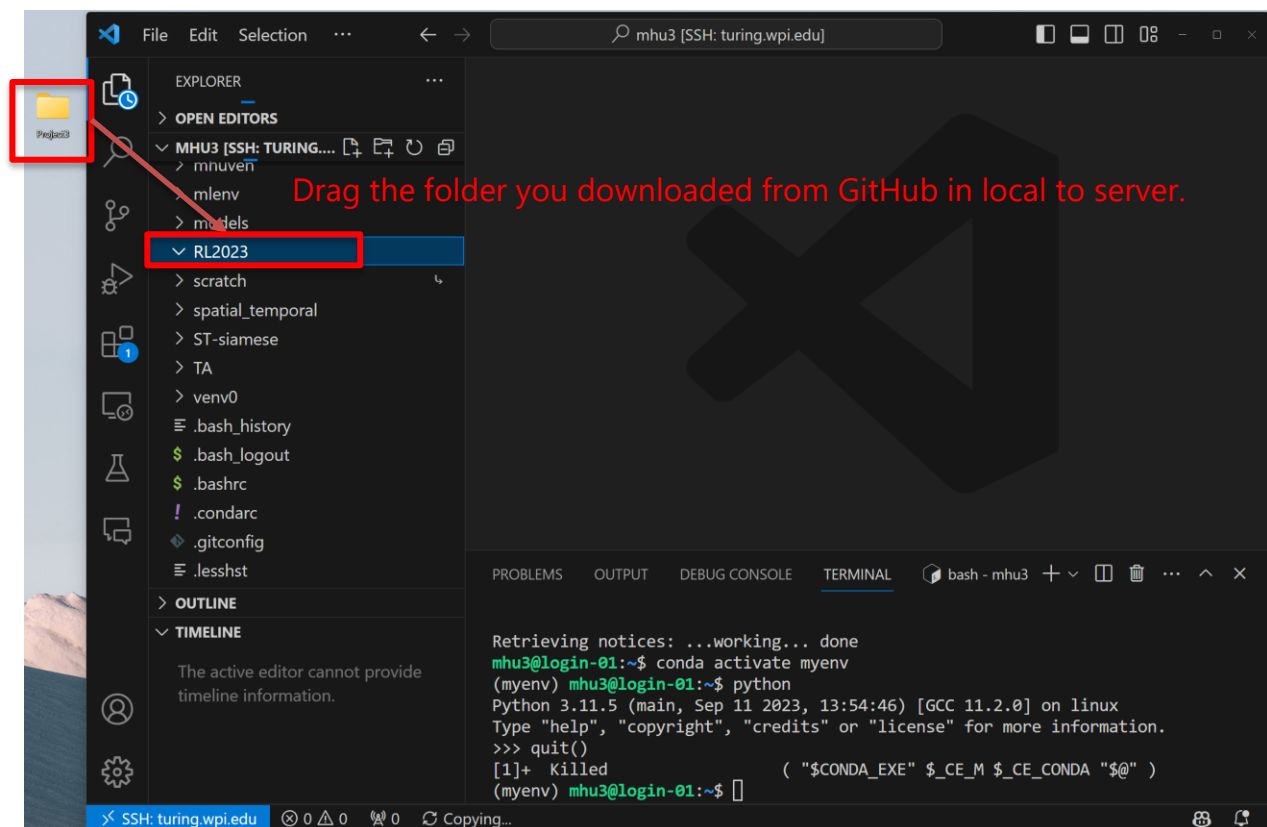
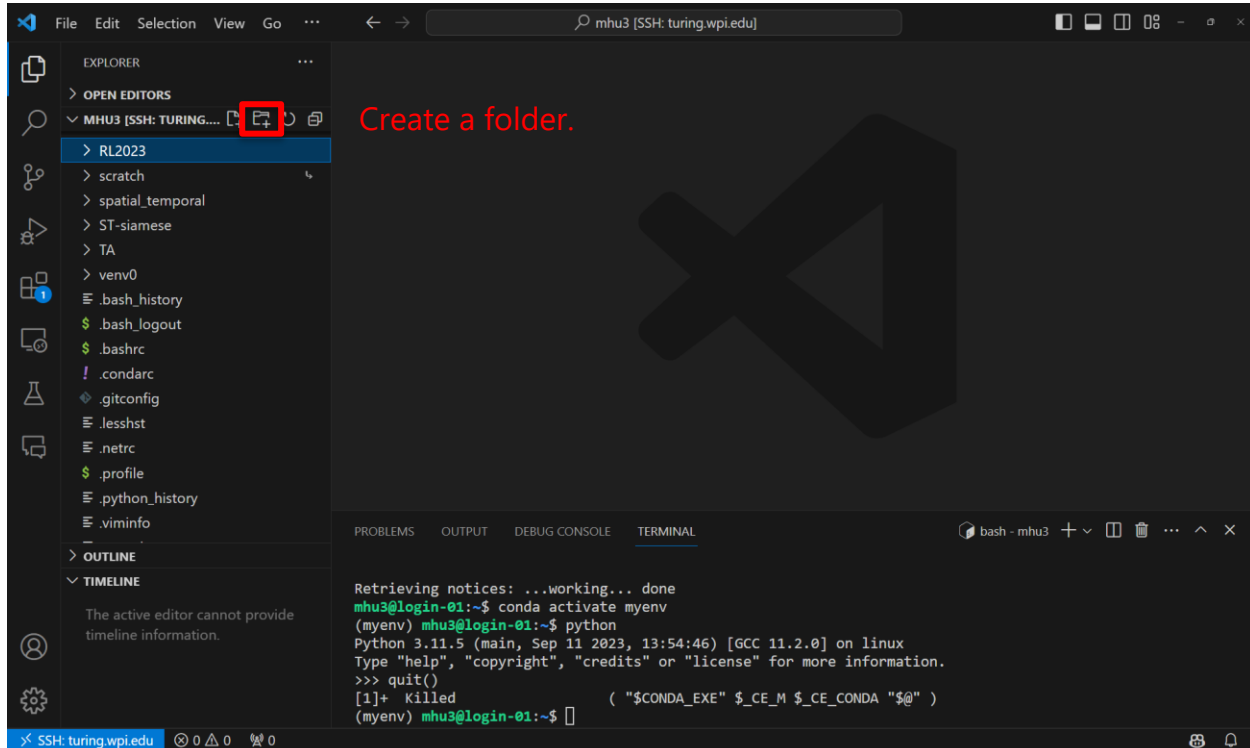


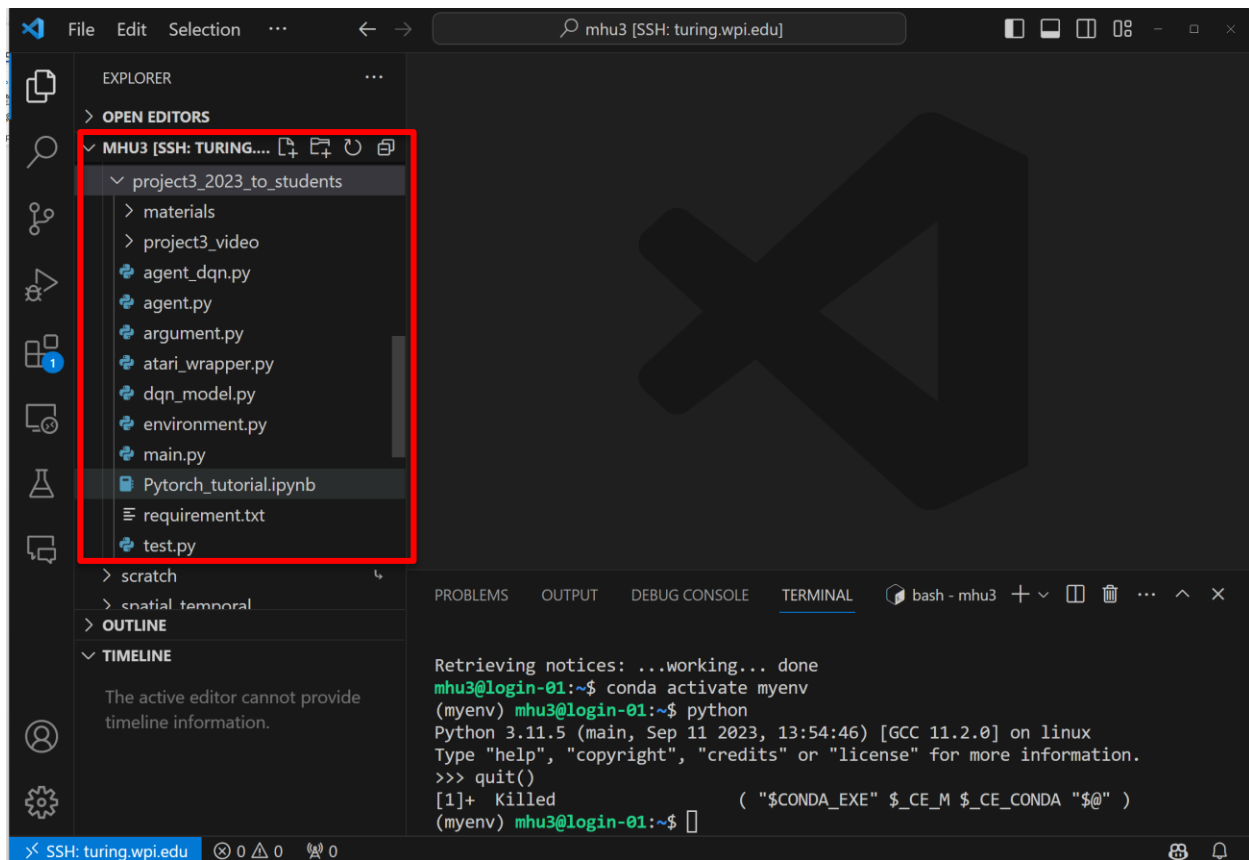
After the installation is done, restart your bash by typing **bash** if needed. conda will create an environment named "base" for you.

5. Create virtual environment: `conda create -n myenv python==3.11.5.`
6. After creating the virtual environment, activate it: `conda activate myenv`. Please install the required packages in the environment.



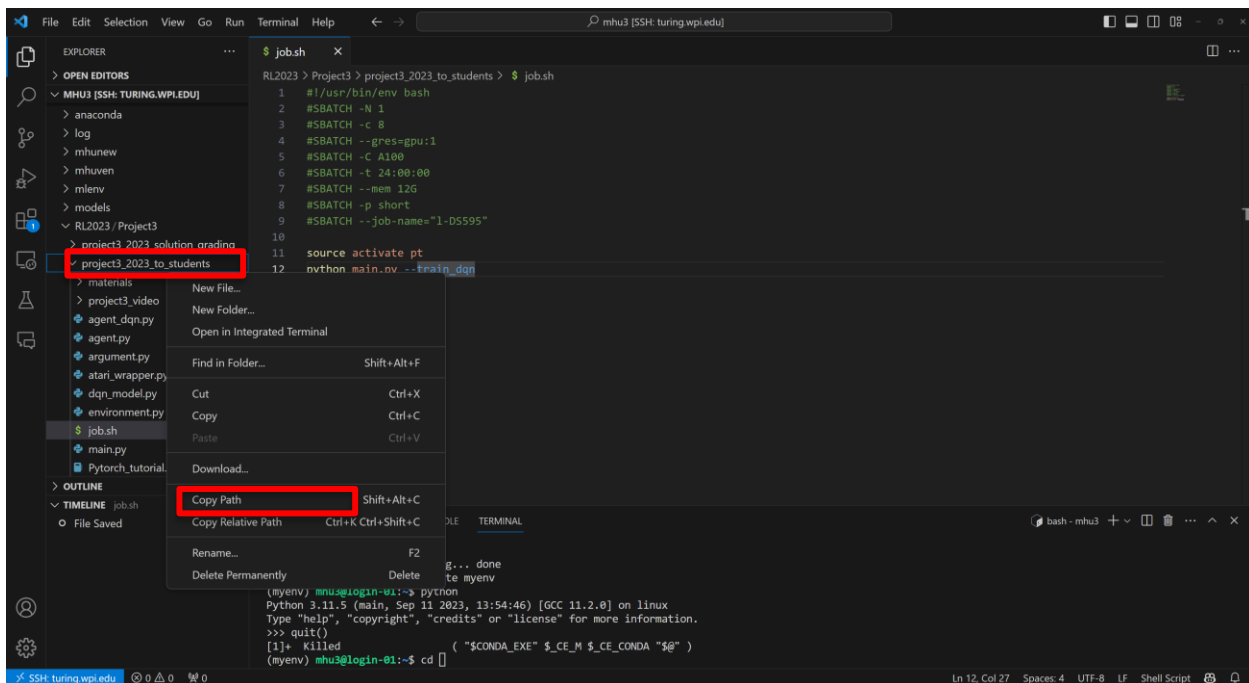
7. Project Directory Setup. Create a folder for the project.





8. Navigate to the Given Directory:

```
cd /home/mhu3/RL2023/Project3/project3_2023_to_students
```



The screenshot shows a VS Code interface with a terminal window. The terminal is running a script named `job.sh` in the directory `RL2023 > Project3 > project3_2023_to_students`. The script content is as follows:

```
1 #!/usr/bin/env bash
2 #SBATCH -N 1
3 #SBATCH -c 8
4 #SBATCH --gres=gpu:1
5 #SBATCH -C A100
6 #SBATCH -t 24:00:00
7 #SBATCH --mem 12G
8 #SBATCH -p short
9 #SBATCH --job-name="1-D5595"
10
11 source activate pt
12 python main.py --train_dqn
```

The terminal output shows the script being executed successfully. The output includes the following lines:

```
Retrieving notices: ...working... done
mhu3@login-01:~$ conda activate myenv
(myenv) mhu3@login-01:~$ python
Python 3.11.5 (main, Sep 11 2023, 13:54:46) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> quit()
(myenv) mhu3@login-01:~$ cd /home/mhu3/RL2023/Project3/project3_2023_to_students
(myenv) mhu3@login-01:~/RL2023/Project3/project3_2023_to_students$
```

9. Prepare Your Job:

Ensure you have a script (e.g., `job.sh`) ready. This script should specify the necessary resources (like CPU cores, RAM, GPU) and contain the instructions for running your job.

The screenshot shows the VS Code Explorer view with the file `job.sh` highlighted in the `RL2023 > Project3 > project3_2023_to_students` directory. The terminal window below shows the same script content as in the previous screenshot.

Create a .sh file for running your code with GPU.

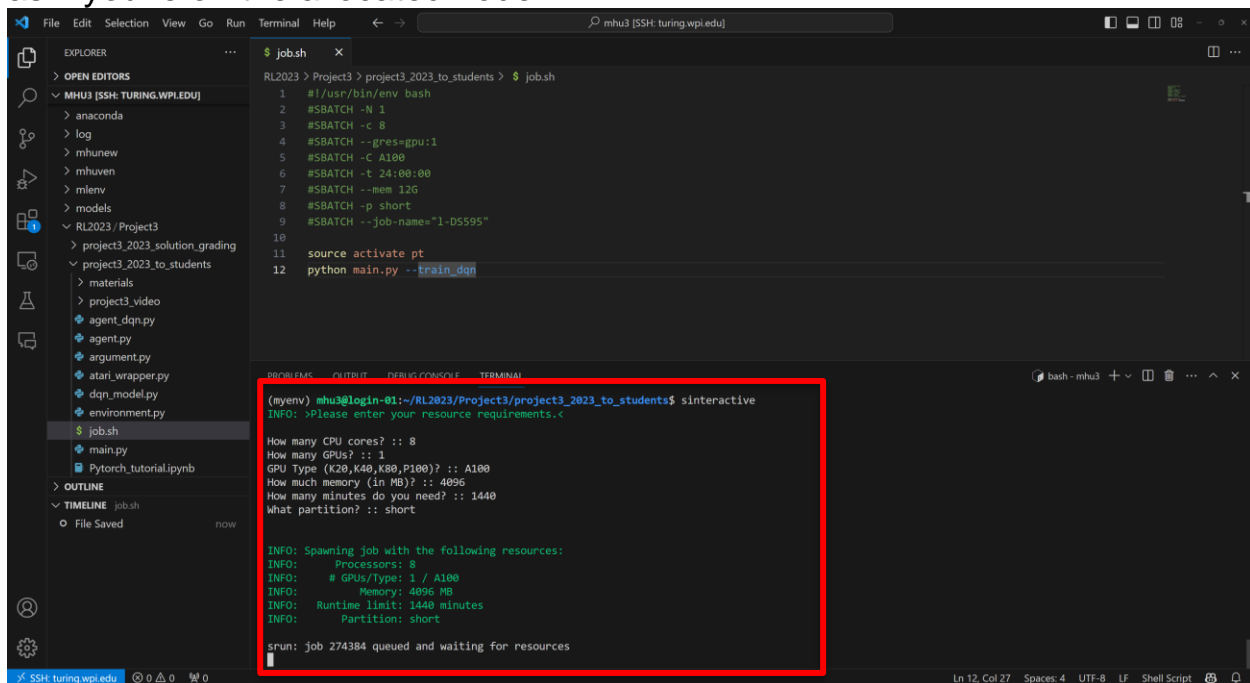
10. Use the following command to submit your job: `sbatch job.sh`.

11. Launch an Interactive Session for Testing: Use `sinteractive`.

This command is particularly useful when you want to test your code on the server's resources before running the actual job. Note that in an interactive session:

- Your terminal session must remain active for the duration of the session.
- Any code you run will execute immediately, unlike the batch mode (sbatch), which queues your job to be run based on the server's schedule and available resources.

After running this command, wait for the server (in this case, Turing) to allocate resources to you, such as a GPU. Once resources are assigned and your prompt changes, you're in an interactive session and can run commands as if you're on the allocated node.



```
File Edit Selection View Go Run Terminal Help
mhu3 [SSH: turing.wpi.edu]

EXPLORER
  OPEN EDITORS
  MHU3 [SSH: TURING.WPI.EDU]
    anaconda
    log
    mhunew
    mhuvn
    mlenv
    models
  RL2023 / Project3
    project3_2023_solution_grading
    project3_2023_to_students
      materials
      project3_video
      agent_dqn.py
      agent.py
      argument.py
      atari_wrapper.py
      dqn_model.py
      environment.py
    job.sh
    main.py
    Pytorch_tutorial.ipynb
  OUTLINE
  TIMELINE
    job.sh
    File Saved
    now

Terminal
  RL2023 > Project3 > project3_2023_to_students > $ job.sh
  1 #!/usr/bin/env bash
  2 #SBATCH -N 1
  3 #SBATCH -c 8
  4 #SBATCH --gres=gpu:1
  5 #SBATCH -C A100
  6 #SBATCH -t 24:00:00
  7 #SBATCH --mem 12G
  8 #SBATCH -p short
  9 #SBATCH --job-name="1-D5595"
 10
 11 source activate pt
 12 python main.py --train_dqn

  (myenv) mhu3@login-01:~/RL2023/Project3/project3_2023_to_students$ sinteractive
  INFO: >Please enter your resource requirements.<
  How many CPU cores? :: 8
  How many GPUs? :: 1
  GPU Type (K20,K40,K80,P100)? :: A100
  How much memory (in MB)? :: 4096
  How many minutes do you need? :: 1440
  What partition? :: short

  INFO: Spawning job with the following resources:
  INFO:   Processors: 8
  INFO:   # GPUs/Type: 1 / A100
  INFO:   Memory: 4096 MB
  INFO:   Runtime limit: 1440 minutes
  INFO:   Partition: short

  srun: job 274384 queued and waiting for resources
```