PACE-ICE GUI Guide for CS6476

CS6476

Spring 2025

Note: All links are also at the end of the document as full URLs.

1 Enable the Georgia Tech VPN

To access PACE ICE, you'll need to be connected to the Georgia Tech VPN. You can download the appropriate GlobalProtect version from this page:

GlobalProtect VPN Download.

After your download, use **vpn.gatech.edu** as the portal on your VPN widget, and enter your GT login (and use Duo 2FA) to connect.

Alternatively, you can log into the online GT VPN found here:

GlobalProtect VPN Online.

We recommend downloading the VPN instead since it saves your information and keeps you logged in.

2 Unzip and upload your project

- Unzip the project folder you downloaded locally on your laptop first.
- Go to **PACE ICE OnDemand link**, click 'Files' and then 'Home Directory'.
- Open the 'scratch' directory (see Fig. 7) and upload the unzipped project folder using the blue 'Upload' button.

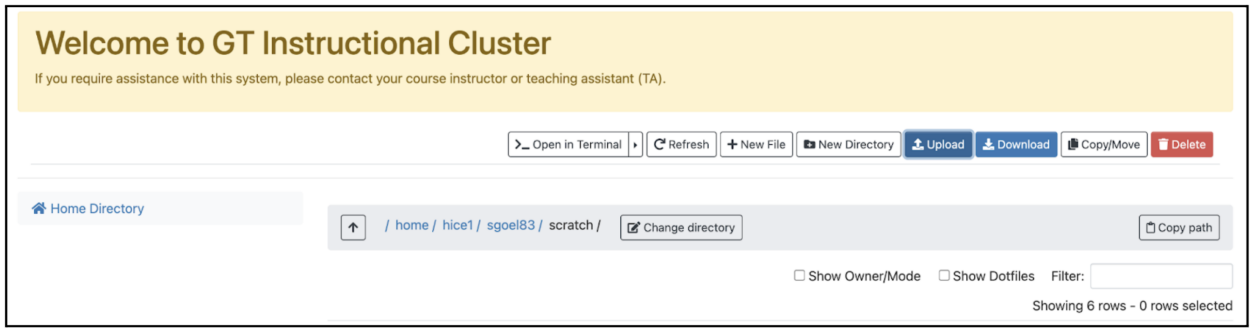


Figure 1: Scratch directory and Upload

We recommend you keep all your working files in the scratch directory.

3 Create and configure VSCode session

Use the **Interactive Apps** dropdown to select **VSCode** (see Fig. 7).

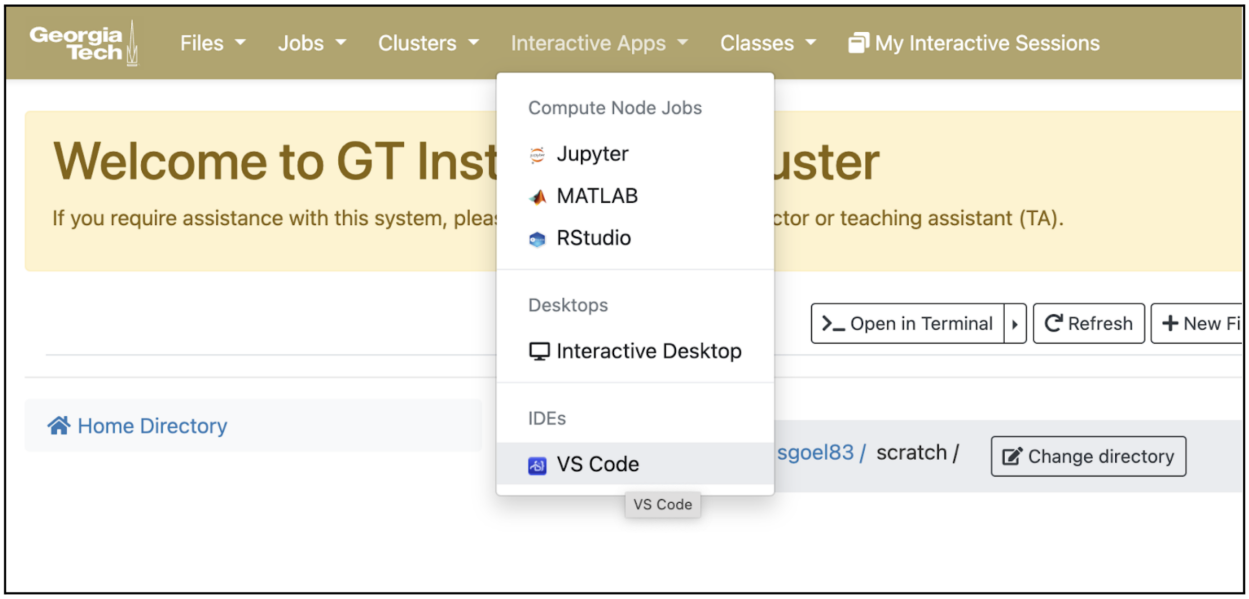


Figure 2: Interactive Apps dropdown

Configure your session as shown in Fig. 7 and described here below.

Under Modules, please pick **Anaconda 3 - 2023.03** or the latest Anaconda version available.

Under Quality of Service, if you plan to use Slurm jobs, pick **coc-ice**, else you can pick **Default (none)**.

Under Node Type, we strongly recommend choosing **NVIDIA GPU (first avail)** to have a short waiting time when being assigned a node. There isn't a need to specify GPU type for CS6476, and using the first available one helps waiting times. If you know that you don't need GPU capability in that work session (for example, if you will only be working on data pre-processing), you can pick **CPU (first avail)** instead, since you will be quickly assigned a CPU.

For Cores, PACE recommends **4**.

For Memory, we recommend a minimum of **16 GB**. If you are using very large files and will store large amounts of data in processing (for deep learning projects where it is necessary), your kernel may need 32 GB or higher instead. You'll know this is the case if your kernel crashes exactly when you're generating/populating data every time.

For GPUs, we recommend you enter **1** when training and testing your models. If you know that you don't need GPU capability in that work session (for example, if you will only be working on data pre-processing), you can pick **0** instead, since you will be quickly assigned a CPU.

You can also use your node for up to **8 hours** at a time. If your work extends beyond that, simply create a new VSCode session with the same configuration once your previous session runs out.

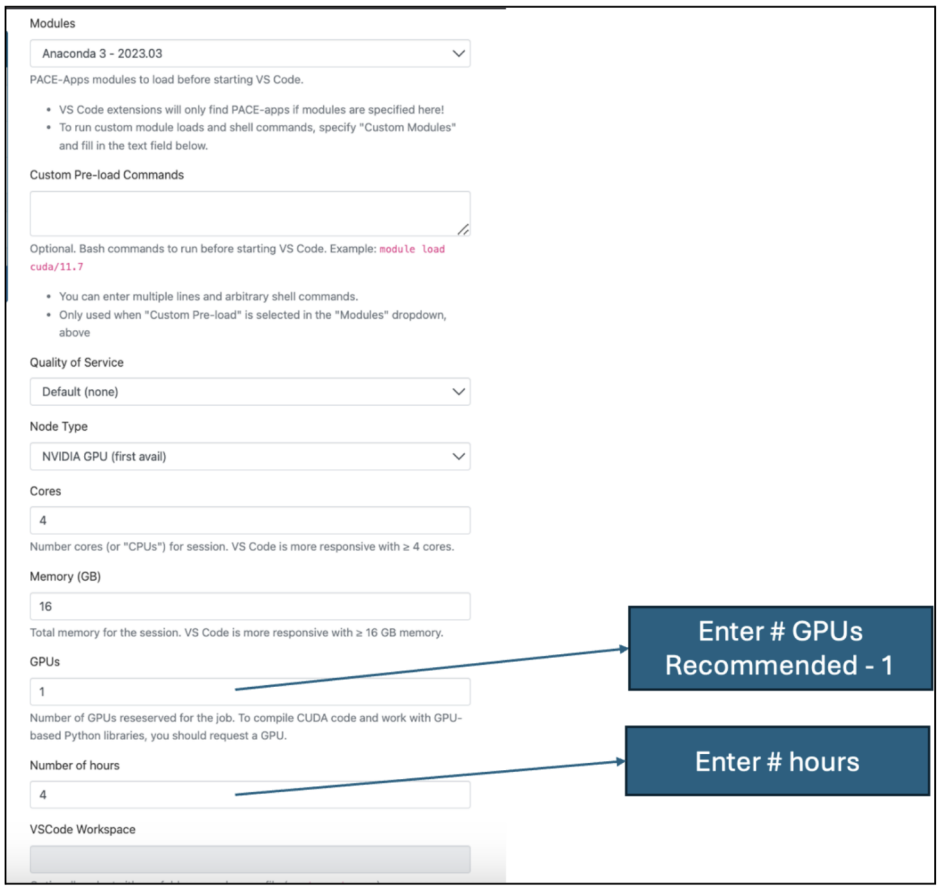
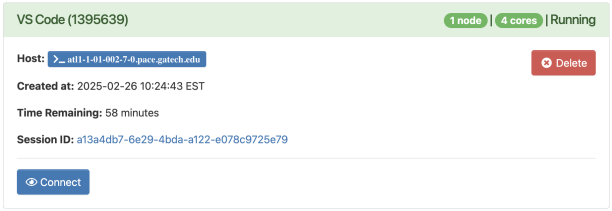


Figure 3: VSCode Session configuration

4 Open the VSCode workspace

For your reference, your request for a node will be in the queue (see Fig. 4a), move to the 'starting' stage (not shown here) and finally begin running, where you can click the blue 'Connect' button (see Fig. 4b).

(a) Queued state



(b) Running state

Figure 4: Session queued and running.

Once open, use 'Add Folder to Workspace' to navigate to the project directory in your scratch folder (see Fig. 7).

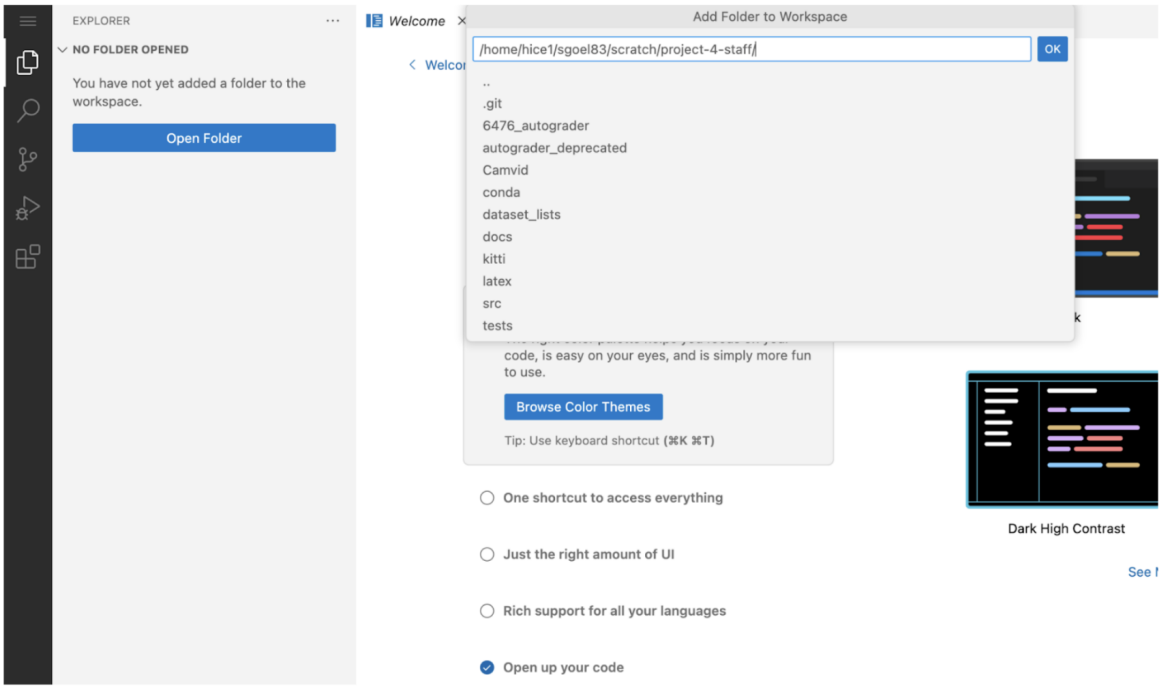


Figure 5: Add your project folder to the VSCode workspace.

5 Setup a virtual environment using Miniconda

The first time you enter a session, you will need to set up a virtual environment as described here. On subsequent VSCode sessions, you will simply need to launch a terminal and activate your environment before beginning your work.

- First, launch a new terminal using the dropdown as seen in Fig. 7 here.

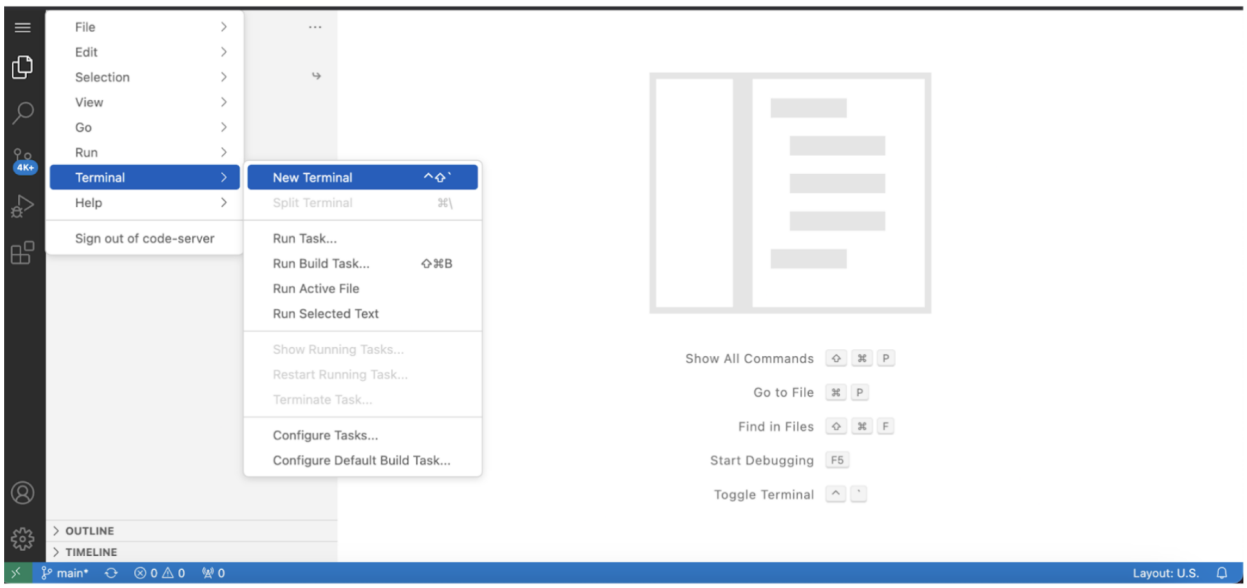


Figure 6: Launching a new terminal

- Navigate to your scratch folder using the terminal.
- Run the following instructions inside the scratch folder:
  - Create a directory for Miniconda:

```
mkdir miniconda
```

  - Download Miniconda installer:

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

  - Install Miniconda:

```
bash miniconda/miniconda.sh -b -u -p miniconda
```

  - Remove the installer:

```
rm miniconda/miniconda.sh
```

  - Activate Miniconda:

```
source miniconda/bin/activate
```

- Navigate to the conda subfolder inside the project folder. Your path should look like this: /home/hice1/sgambhira6/scratch/project-4/conda
  Then run the following instructions:
  - Create the virtual environment:

```
conda env create -f environment.yml
```

  - Activate the environment:

```
conda activate cv_proj4
```

  - If the virtual environment is not active upon launching a new session, activate it manually:

```
conda <miniconda path>/bin/activate cv_proj4
```

  - Here's an example of what that should look like:

```
conda ~/scratch/miniconda/bin/activate cv_proj4
```

- Once activated, your terminal should look like this (see Fig. 7):

Figure 7: Activated environment - shows '(cv_proj4)'

Remember to activate the environment through the terminal when you start a new session - you do not need to set up the virtual environment again. You're now ready to begin coding!

Credits

PACE Guide developed and tested by Sriman Goel and Sirish Gambhira.

References

ICE Cluster Pages

All links as text URLs for access

- GlobalProtect VPN Download: https://vpn.gatech.edu/global-protect/getsoftwarepage.esp.
- GlobalProtect VPN Online: https://vpn.gatech.edu/global-protect/login.esp
- PACE ICE OnDemand link: https://ondemand-ice.pace.gatech.edu/pun/sys/dashboard/
- ICE Cluster Pages: https://gatech.service-now.com/technology?id=kb_article_view&sysparm_article=KB0042102#getting-started-on-ice