Remote Access with Visual Studio Code

This guide is for setting up a Visual Studio Code (VS Code) remote access environment;
then you can write your code locally, and seamlessly execute it remotely. Makes development easier.

Guide

IF you plan on working remotely (not on a campus network), you need to set up a VPN (or ssh-keys).
Guide: https://www.hpc.dtu.dk/?page_id=4317

On campus, you can simply proceed:
0. You need to have a SSH client installed locally. A guide here.
1. Download Visual Studio Code: https://code.visualstudio.com/download
2. Launch VS Code.
3. Download the C/C++ extension pack.
   Note a compiler is not needed, as it is already installed on the remote machine.
4. Download the Remote Development extension pack.
   You may need to restart VS Code after this.

Step 5-9 basically reiterate this guide: Connect to a remote host.

5. On the left-side bar, an icon “Remote Explorer” should appear below “Extensions”.
   Go to the Remote Explorer page.
6. Go Remote > SSH > + and then enter the SSH command to use;
   e.g. “ssh USERNAME@login2.gbar.dtu.dk -A”, where USERNAME is your student number.
   Finally you select an SSH configuration file to save this info to and click connect.
   If you are connected, it should now say “SSH: login2.gbar.dtu.dk” in the little green field on the bottom left.
   If you are not connected go to the remote PC in Remote Access > Remote > SSH > login2.gbar.dtu.dk and click the connect → icon.
   If the remote machine is not there, try clicking the green icon on the lower left and selecting “Connect to Host”.
   It should suggest the host “login2.gbar.dtu.dk”, so choose this.
   If it does not suggest a host (only options are “Add New SSH Host” or “Configure SSH Hosts”), try “Add new SSH Host”.
7. You are now connected! What does this mean? It means basically everything you do now happens remotely.
8. Click “Open…” and open your desired workspace folder on the remote machine.
9. You can now navigate to “Explorer” on the top of the left side bar to navigate/create files in your workspace.

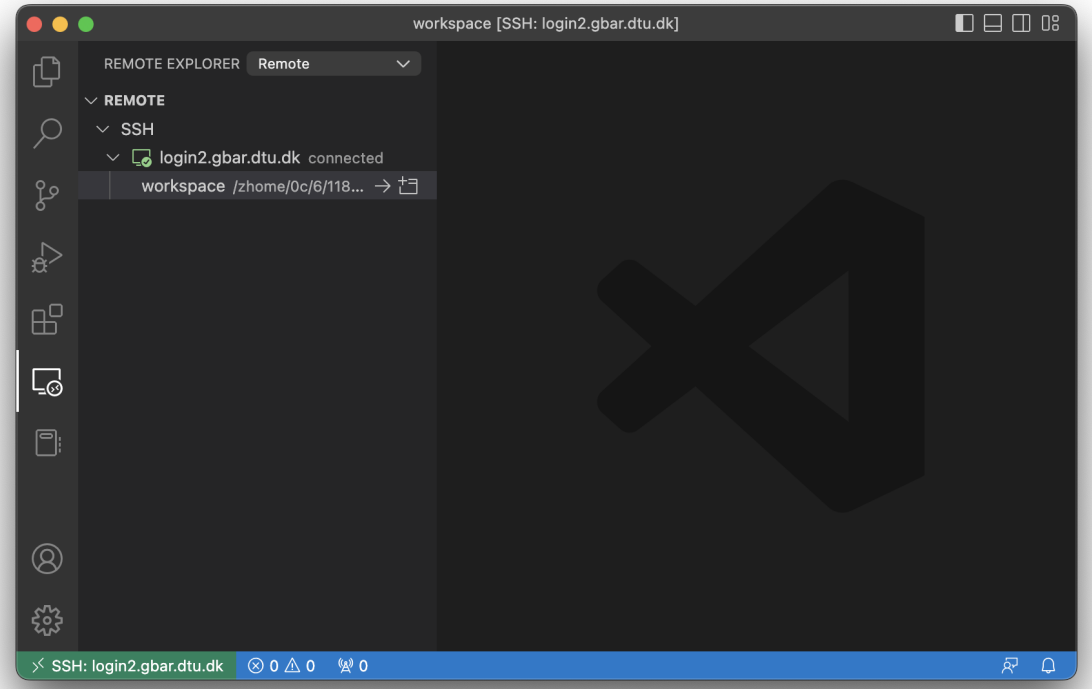
The crucial tools for remote development are basically:
* File navigation in the Explorer, e.g.: Top Bar > View > Explorer
* Use of the Terminal for code execution, e.g.: Top Bar > View > Terminal

Remember not to run jobs on the login node!!
Type “linuxsh” as the first thing in the terminal to get an interactive terminal session where you can debug your code. Or voltash/sxm2sh/a100sh for GPU code.
These nodes are shared. For exclusive access, e.g. for performance measurement purposes, or large jobs, you should submit batch jobs to the hpcintro queue.

Tips:
- When running on CPU, use the OMP_NUM_THREADS environment variable to select the number of cores. (The default may be just 1 core!)
- When profiling, remember to have some warm-up. I.e. a few iterations to ensure you don’t measure the initialisation steps.
- When running GPU jobs on interactive nodes, use e.g. CUDA_VISIBLE_DEVICES=2 to select to use the second GPU. (Run “nvidia-smi” to see which GPU is least used).
- If using GPU with OpenMP and your program crashes with access violation, ensure that you used the appropriate “map” commands to move data to/from the GPU.
  E.g. map(from:array[:array_size]) to transfer from GPU to CPU.

Sebastian Præsius, 2. Jan. 2023

If everything has been successfully set up, next time you open VS Code you can simply connect directly to your workspace by clicking the workspace folder in the Remote Explorer and logging in:



Sebastian Præsius, 2. Jan. 2023