

Editing code in your Workspace

TABLE OF CONTENTS

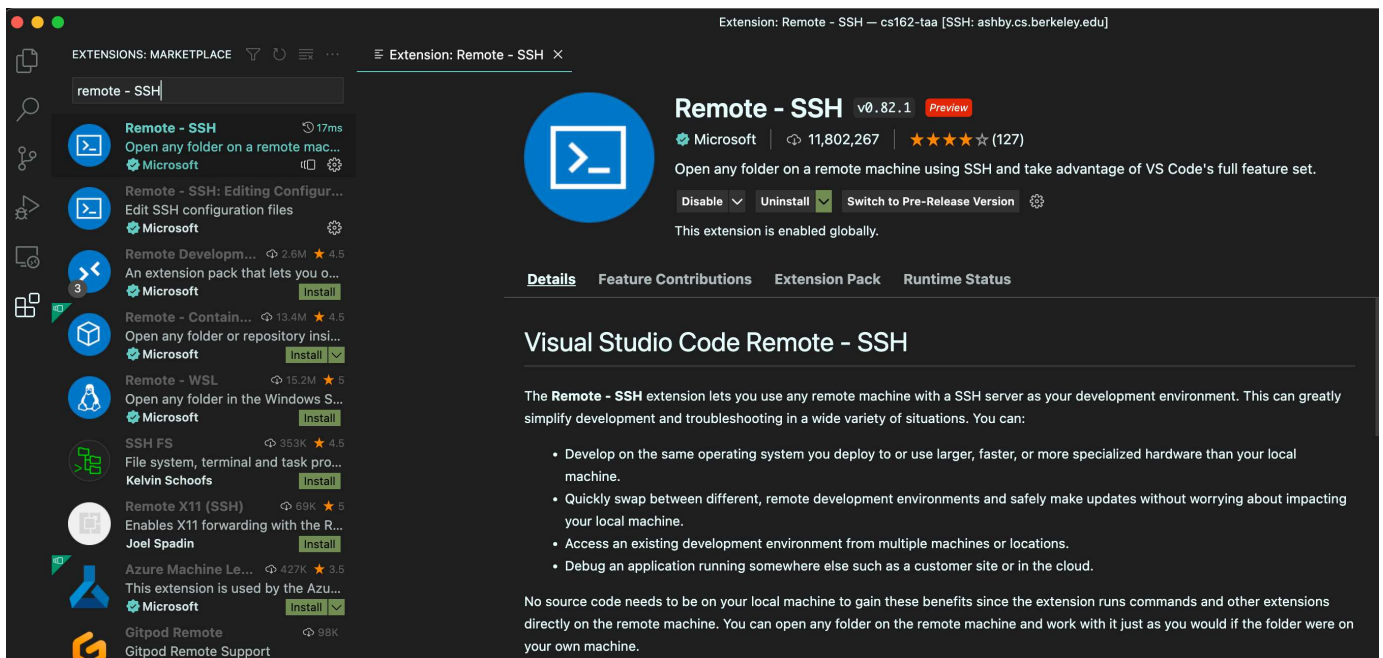
- 1 [VSCode C/C++ Extension Config](#)
 - 2 [\(NOT RECOMMENDED\) Using the local files directly from the Workspace](#)
-

The Workspace gives you access to all the local files that you will need. This will allow you to edit code using text editors on your host and run git commands from inside the Workspace. These files are stored within the `.workspace` directory in the Workspace repo. We don't recommend editing the files directly from your host as you may run into inconsistencies.

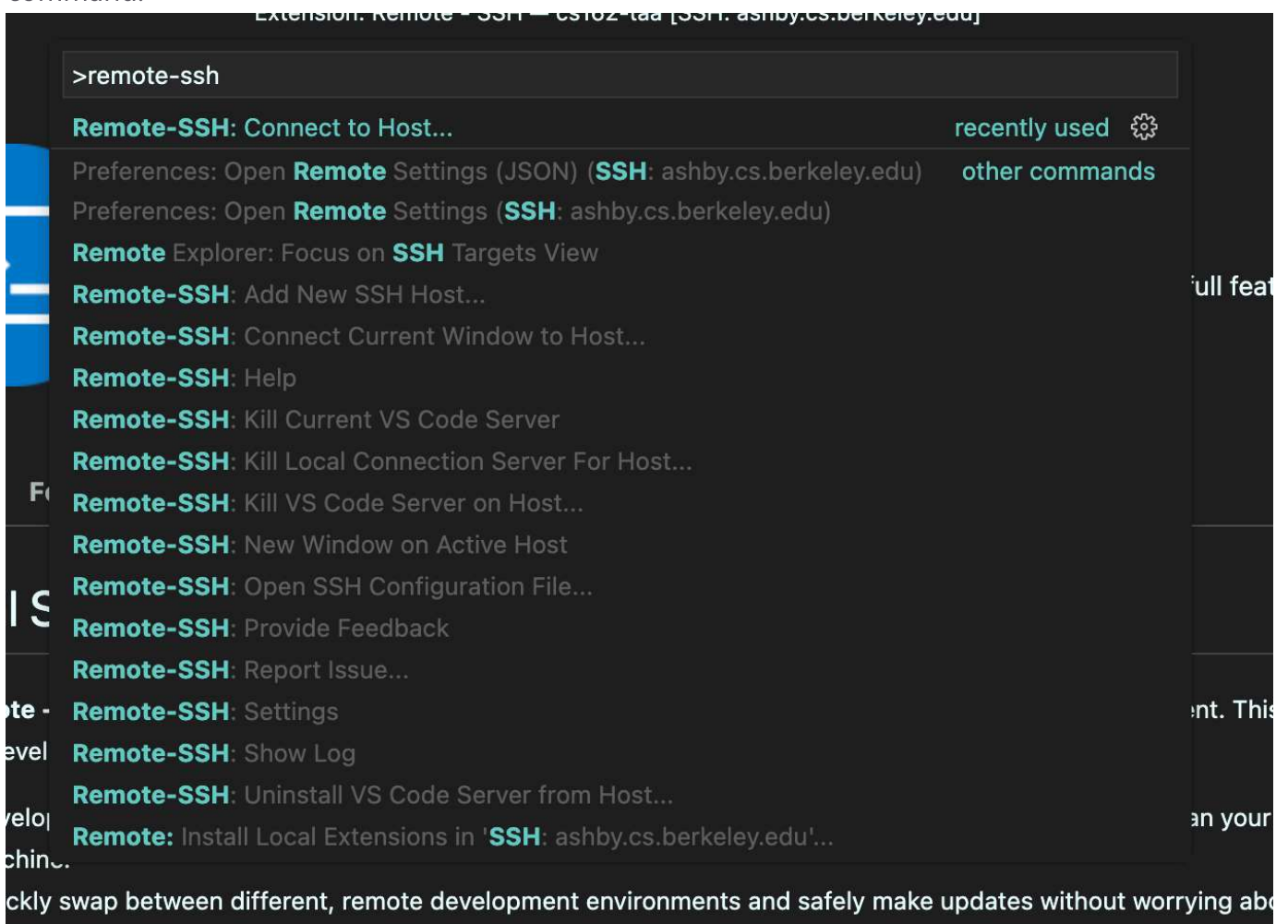
If you use VSCode, you can use its remote ssh feature to connect to your Workspace quite easily. This process is detailed below. **This is the recommended way of working on code for this course.** If you are a vim/emacs poweruser, another possibility is just using a non-graphical text editor in an SSH session. You are free to do whatever suits you best.

To remotely modify code, you will need to `ssh` onto the machines. Visual Studio Code has a helpful [remote development feature](#) that streamlines the process of developing over `ssh`. To set this up, follow the steps below:

- 1 Download and open [Visual Studio Code](#).
- 2 Open the Extensions tab on the left and search for "Remote - SSH". Download the first result, which should have several million downloads.



- 3 Hit `Cmd + Shift + P` and type in "Remote-SSH: Connect to Host", selecting the corresponding command.



- . Select the docker Workspace host that should show up.

- 4 A new window should open up with a bright "Opening Remote" indicator in the bottom left. You may need to enter your password to the workspace ("workspace") in the textbox if you haven't copied your ssh public key to the Workspace.
- 5 You should now be able to edit any file you would like and have it sync with your Workspace! To disconnect, click on the ssh label at the bottom left of the window, which should open up a panel of options including "Close Remote Connection".

VSCode C/C++ Extension Config

To enable useful IDE features like code navigation and warning/error highlighting, first install the [C/C++ Extension Pack](#) within your Remote SSH VSCode instance.

For Pintos-based assignments, the following custom config will enable development that mirrors the build system we use:

- 1 Open the VSCode command palette and run the `C/C++: Edit Configurations (JSON)` command to open the config file.
- 2 Replace the existing contents with:

```
{
  "configurations": [
    {
      "name": "pintos",
      "includePath": [
        "${workspaceFolder}/**"
      ],
      "defines": [
        "USERPROG",
        "THREADS",
        "FILESYS"
      ],
      "compilerPath": "/usr/bin/gcc",
      "cStandard": "c11",
      "intelliSenseMode": "linux-gcc-x64",
      "configurationProvider": "ms-vscode.makefile-tools",
      "compilerArgs": [
        "-nostdinc",
        "-I../..",
```

```
        "-I../lib",
        "-I../lib/kernel",
        "-Wall",
        "-W",
        "-Wstrict-prototypes",
        "-Wmissing-prototypes",
        "-Wsystem-headers"
    ]
}
],
"version": 4
}
```

(NOT RECOMMENDED) Using the local files directly from the Workspace

- 1 Change into the directory where the Workspace is found. (i.e `cd docker`)
- 2 Change into the `.workspace` directory
- 3 Change into the `code` directory within `.workspace`

You should now be able to see the contents of the Workspace's root directory. You can feel free to just edit the appropriate assignment files with any text editor you feel most comfortable with.

- 1 Save the file and enjoy!