

This document is for collecting solutions to problems with setting up docker for students to work on CS 225 programming problems.

Windows Home: WSL2 required for Docker Desktop

According to the [official documentation](#), WSL2 (Windows Subsystem for Linux 2) is now listed as a system requirement for Windows Home users. Without WSL2, Windows Home users won't be able to use Docker Desktop as WSL2 is the backend.

To install WSL2, you can follow either of the two official installation instructions released by Microsoft. There is this [newer one](#), and this [older one](#). The newer installation method is rather easy, a single "wsl --install" command would take care of everything for you. However, this command is only available on newer builds of the Windows OS. For those of you that are on older builds, you should instead follow the older method. I have helped a student install WSL2 using the older method, and it was a rather smooth process, so I think it shouldn't be a problem. If you are following the older instructions, I would recommend you to install Ubuntu 20.04 when you reach "Step 6 - Install your Linux distribution of choice", as this is also the default distribution to be installed if you were to use the newer method instead.

After you set up WSL2, you can then proceed to install Docker Desktop. The installation wizard for Docker Desktop will include an option to "**Install required Windows components for WSL 2**", and it is vital that you check this box. The details to install Docker Desktop can be found in this [official documentation](#).

EDIT 1: To check the WSL version, use `wsl -l -v` to list the Linux distro(s) in your Windows. If one of the version number(s) is `2`, then you have WSL2. Otherwise, to install WSL2, first remove the desired distro (i.e. Ubuntu 20.04) using `wsl --unregister <distro name>` and then follow the steps in the older instructions to upgrade to WSL2.

Missing "Remote-Containers: Open Folder in Container..." Option

In some cases, restarting (exiting and re-opening) VS Code refreshes the newly installed bextension and shows the option correctly afterward. In some cases, the option is shown as "Open Folder in Container..." instead, without the extension prefix.

Docker version (something) or later required.

When trying to “Open Folder in Container” in VS Code.

If you are on macOS, after you have installed Docker by dragging the app into your Applications folder, you must also **open the Docker application at least once** before you continue setting up VS Code. Docker will prompt you for an admin account and password for your computer, followed by a Terms of Service agreement that you must accept. Following that, the Docker main window will open. You can close this window and continue with the set up. **You do not have to interact with anything within the Docker app.**

Note: If you are following the “On Your Own Machine” guide, this should be done at some point before step 4.

Missing system headers (such as string).

When compiling code through VS Code and the bottom left corner does **NOT** say “Dev Container: Existing Dockerfile.”

(Especially if it says “Dev Container: C++” or something similar)

The Docker VM was not set up correctly and instead is running a VS Code-generated Dockerfile that runs Debian bullseye. **This is not a supported Docker VM configuration.** To fix, delete the “.devcontainers” folder (this will delete the Remote Containers setup for the folder), make sure the Dockerfile is correct, and set up the VM again in VS Code, **making sure to use the Dockerfile.**

Finally, verify it is correctly installed by ensuring the bottom left corner of VS Code shows “Dev Container: Existing Dockerfile.”

Problem with the SSL CA cert (path? access rights?)

When trying to use git/curl in the Docker VM on a M1 Mac host.

This is an issue we do not yet have a proper solution for. A band-aid fix includes asking students to add a ‘-c http.sslVerify=false’ to their git commands. This would look like ‘git -c http.sslVerify=false push’. **The flag must be in between git and the action.** More information for debugging and potential fixes are available in the Discord server in #vm.

Also, if you own a M1 Mac, please check if you have this issue. Currently, reproduced **5** times.

“Command “cmake” not found,” but inside container

Make sure you have the right Dockerfile. If you have an outdated Dockerfile, you may have built an image without cmake included. Docker caches everything, so even if you change the Dockerfile it might not build a new container. After changing the Dockerfile to the correct contents, you may need to click the bottom left corner of VS Code where it says “Dev Container: Existing Dockerfile.” From there, another pop up will show up in the middle. Click “reopen container”. If it still doesn’t work after this, open up that pop up again with the same steps and try “rebuild container”.