Install OpenMC on Windows using WSL

keeperofhoney.neocities.com, 2025

If you want to get started playing with the more technical side of nuclear engineering, a great way to start is by using the open-source Monte Carlo code OpenMC. OpenMC only runs on Linux, but through the help of Windows Subsystem for Linux, you can run it on your windows machine without the need for dual-booting or virtual machines. This took me forever to figure out as a novice Linux user, so I figured I would write a tutorial for how I did it to help anyone else out. This is adapted from:

https://docs.openmc.org/en/stable/quickinstall.html

https://www.researchgate.net/publication/370252055_Installation_tutorial_of_OpenMC_via_Windows_using_Linux_subsystem

https://www.youtube.com/watch?v=bC8LEx1BORA&t=303s

https://code.visualstudio.com/docs/remote/wsl

All of which you should check out for further information or if you are running into issues.

Setting up Windows Subsystem for Linux (WSL)

Installing WSL is pretty straightforward:

1. In the performance tab of Task Manager, Check if Virtualization is enabled in the CPU section. This should say enabled, but if not, follow the steps in this paper to enable it in Bios. https://www.researchgate.net/publication/370252055_Installation_tutorial_of_OpenMC_via_Windows_using_Linux_subsystem
2. Open the Windows PowerShell or Command Prompt in administrator mode
3. Install WSL using the “wsl --install” command. This will install Ubuntu by default, but your distribution can be changed after.
4. Get a list of the available Linux distributions (distros) using “wsl --list --online”. You can’t go wrong with Ubuntu, others like Arch or Debian, but I chose to use AlmaLinux because it’s the preferred distro of CERN and Fermilab and I think that’s wicked.
5. If you want to change your distro, you can run the command “wsl --install -d (Distro name here)”. You can remove distros by running “wsl --unregister (name of distro)”

6. You can now access WSL by typing “wsl” into the PowerShell or command line, or just typing WSL into the search bar.

Installing Conda

Now that we have a Linux terminal on our PC, we will use it to install Conda (or a smaller version, Miniconda or Miniforge). With Conda we will be able to install OpenMC. I am using Miniforge for this, but I believe that Miniconda will work the same and can be installed with the same instructions.

7. Go to the CONDA page and download the most recent version of Miniconda for Linux

   https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html

8. Find the folder where the file was downloaded to (probably downloads) and type “wsl” into the navigation bar. This will open a Linux terminal for that directory.
9. To install this file, type “bash (filename)” with the filename being the name of the Miniconda file you downloaded. For example, “bash Miniconda3-latest-Linux-x86_64.sh”
10. According to the tutorial I followed by Willian Vieira de Abreu and Artur Silva, you don’t want to be working in the “base” environment. You can tell you’re in the base when it says (base) at the beginning of the line. To prevent this, you can run “conda config --set auto_activate_base False” and open a new terminal instance and it should be good.

Installing OpenMC

We have made it! Using our Linux terminal and Conda, we are now able to install OpenMC.

11. Run “conda config --add channels conda-forge” and “conda config --set channel_priority strict”
12. You can then run conda create --name openmc-env openmc . This creates an OpenMC environment called openmc-env, although you can name it whatever you want. This also installs OpenMC.
13. To open this environment, just run “conda activate openmc-env”.

Now OpenMC is installed in this environment, and we must set up a way to write code in it.

Using VS Code & Jupyter Notebook

To use VS Code to make files using OpenMC, you need a little bit of setup to connect to the WSL.

14. Install VSCode on your Windows machine
15. Install the WSL Extension in VSCode
16. In the bottom left corner, click the little box with the >< symbols for to open a remote window. From here, you can connect to WSL, or choose the distro on which you want to connect to. You now can run Linux projects inside VS Code! Yippee!
17. To access your OpenMC files and work on them in VS Code, all you need to do is open your OpenMC environment inside the WSL command line, go to your files directory and run “code .”

To run a Jupyter Notebook in VS Code, the steps are mostly the same.

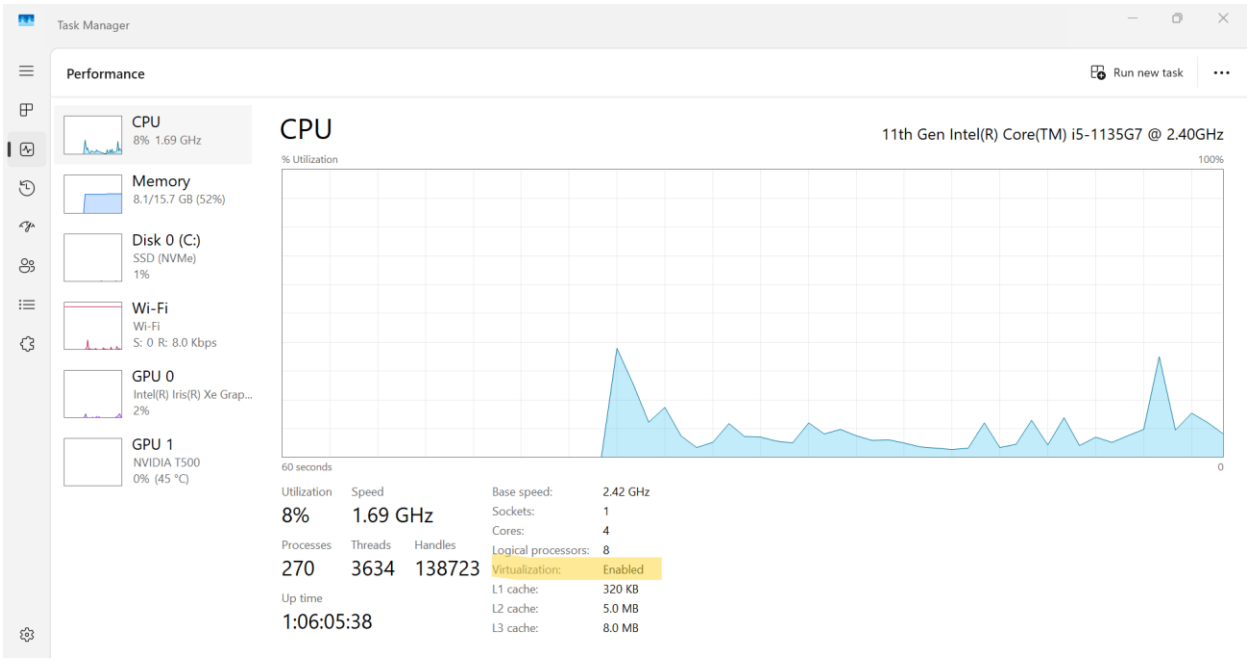
18. In your OpenMC Environment, run “pip install Notebook” to install Jupyter Notebook.
19. After running “code .”, create a new Jupyter Notebook in VS Code.
20. In the top right corner, click the kernel select, then select python environments. There should be your OpenMC environment.

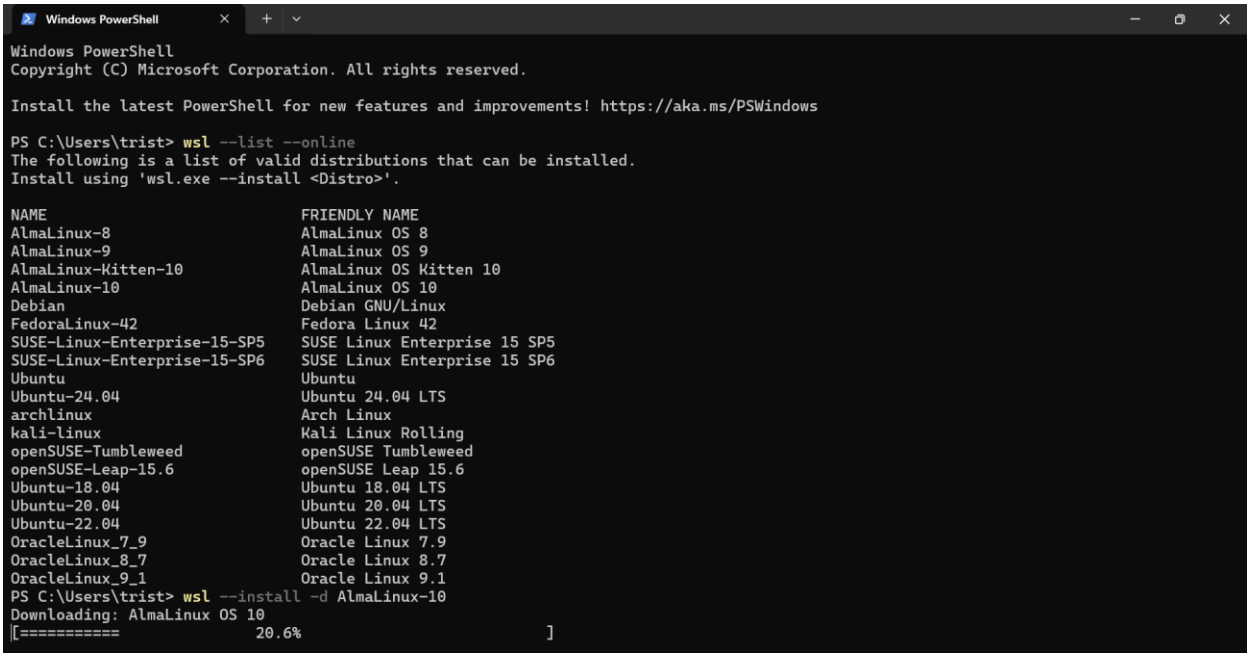
You can also run Jupyter Notebook in browser, without the use of VS Code.

21. In the WSL terminal set to your OpenMC environment, just run “jupyter lab”. This will give you two links, each of which will give you a Jupyter Notebook instance when put into a browser or the kernel of VS Code.

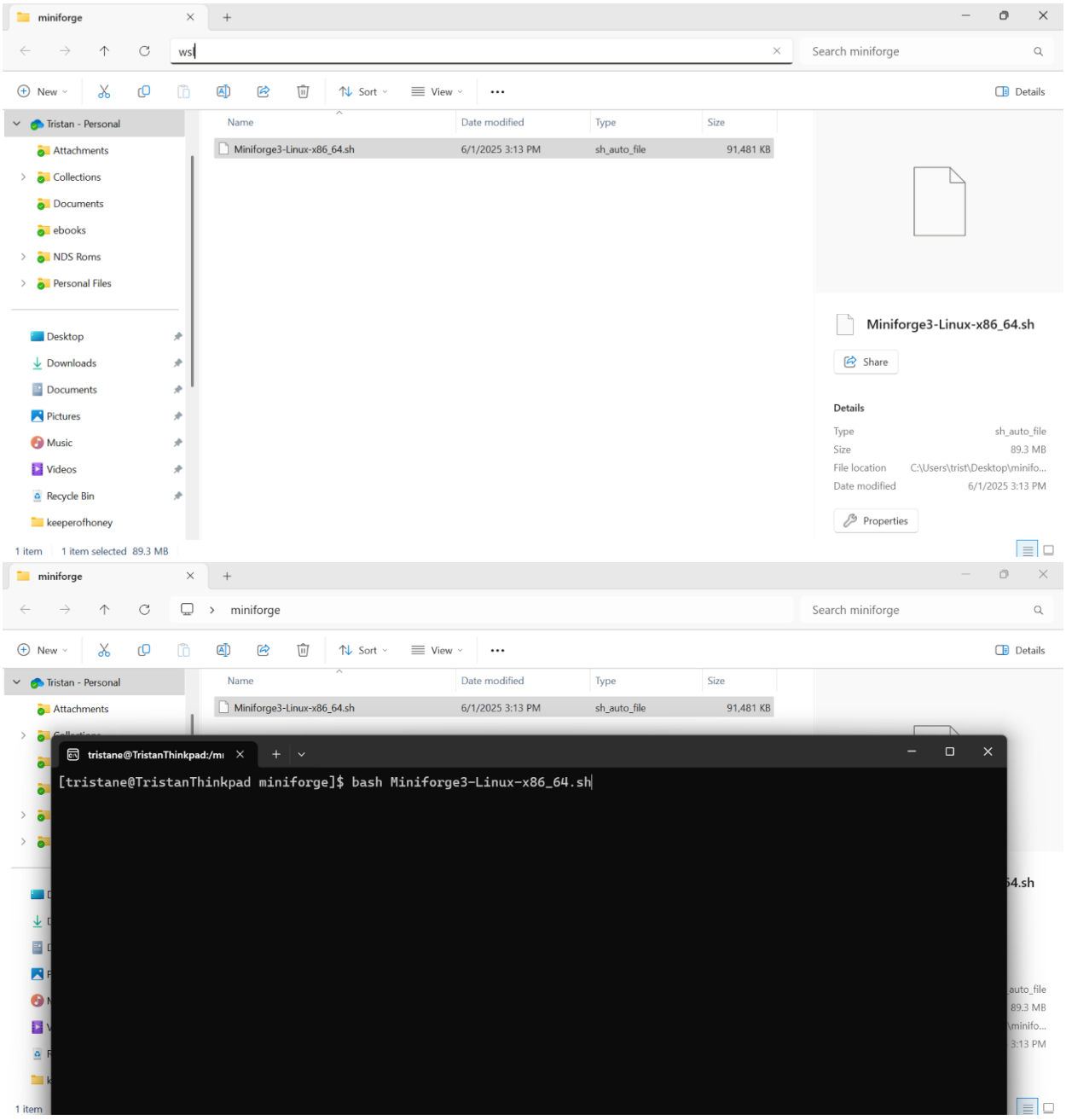
Images:

Installing WSL:

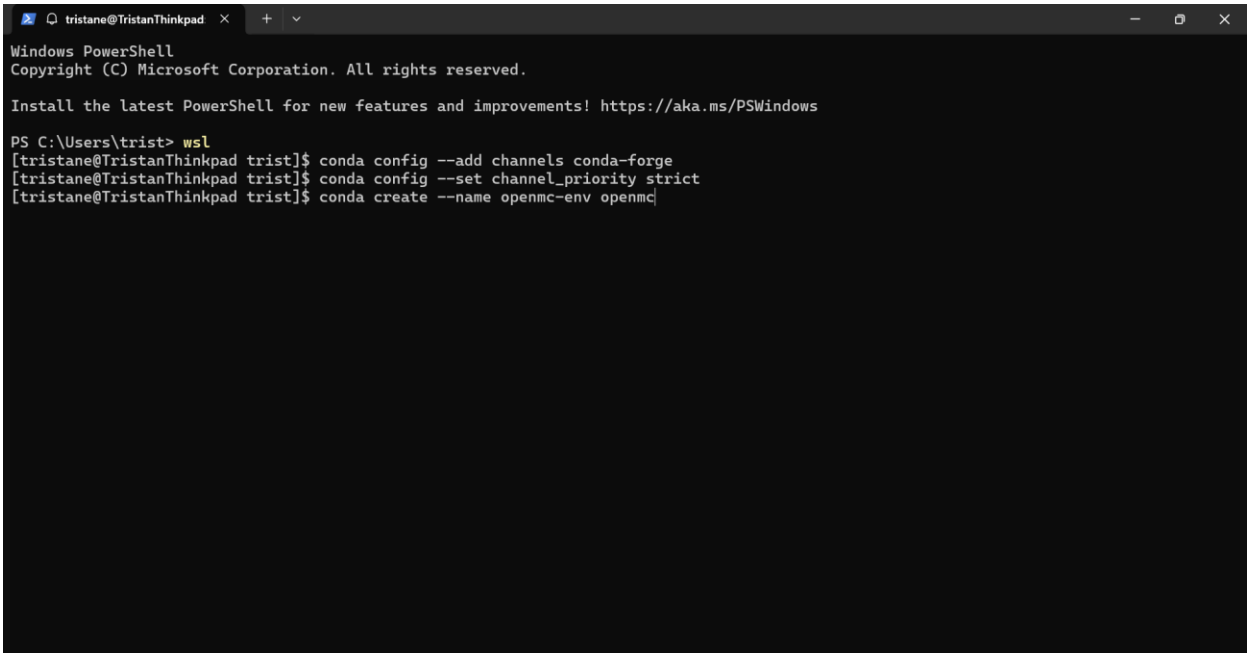




Installing Conda:

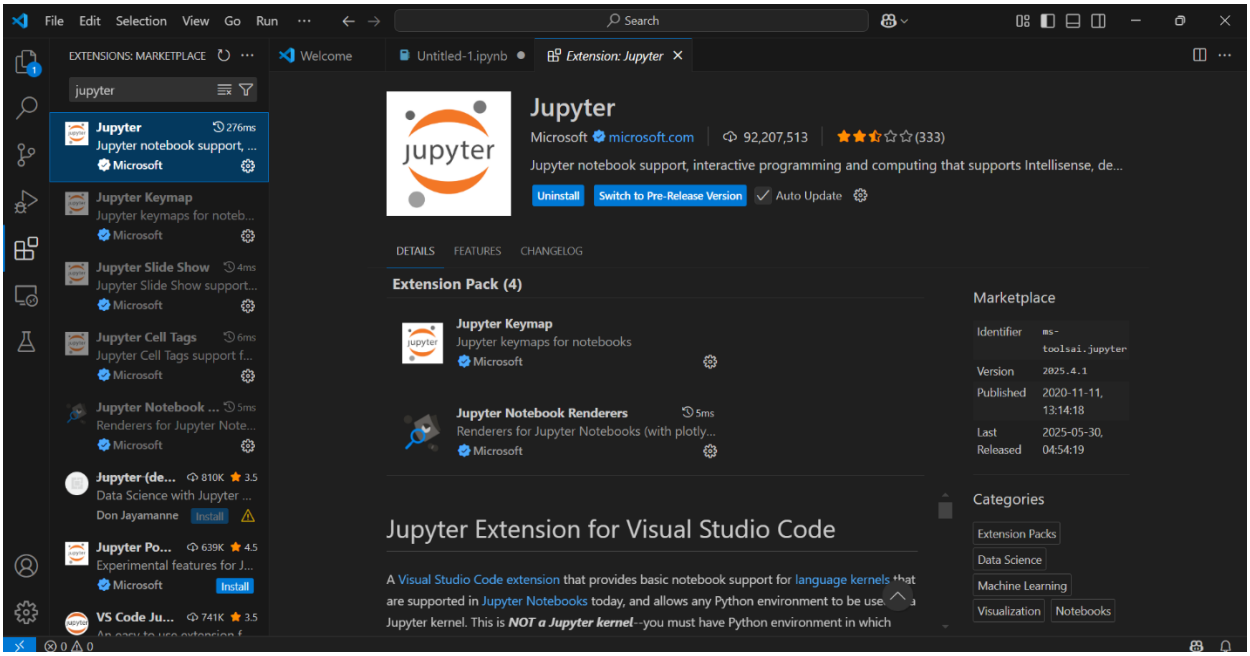


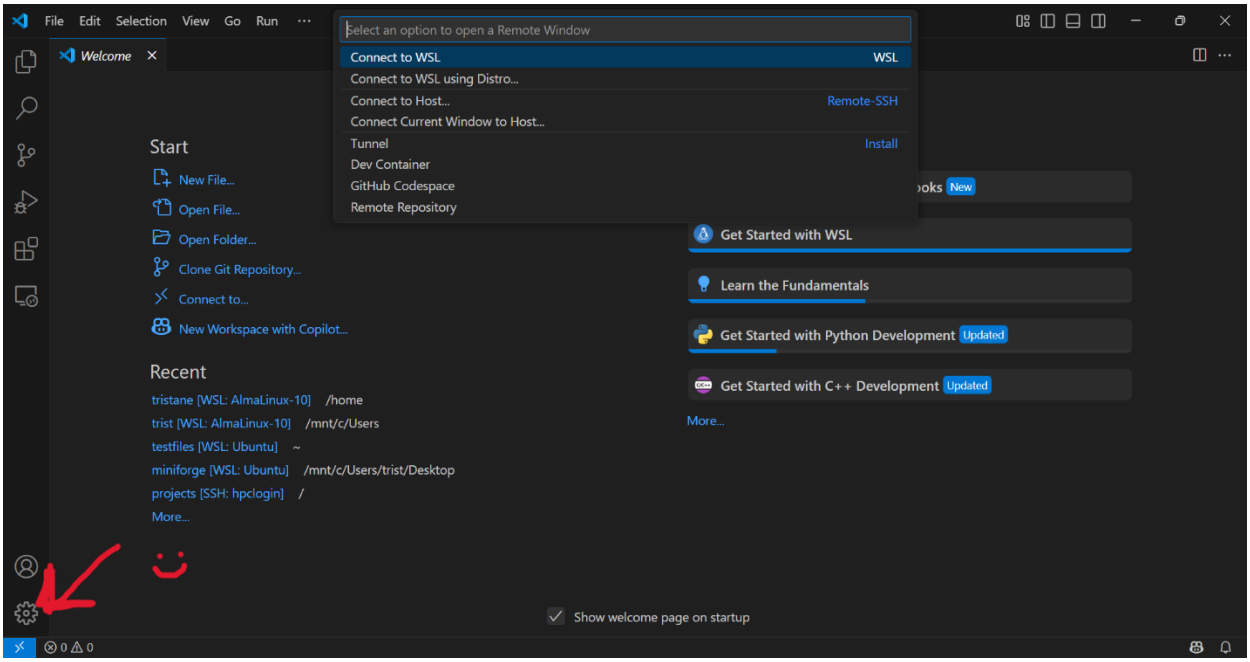
Installing OpenMC:

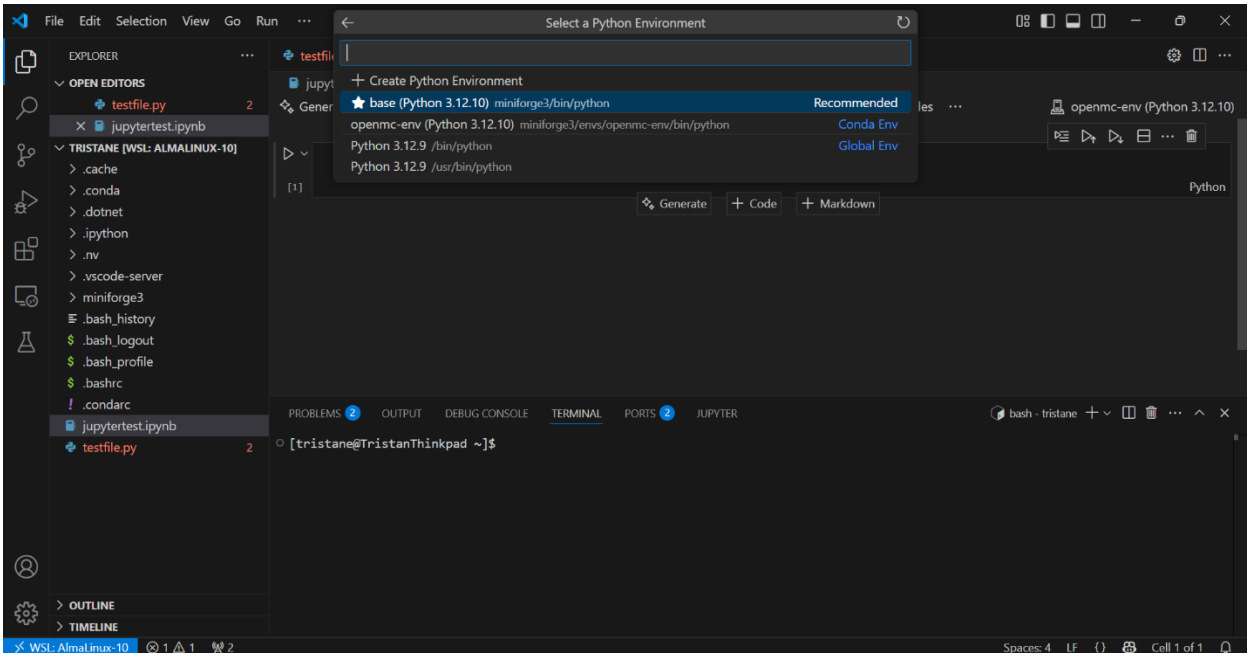


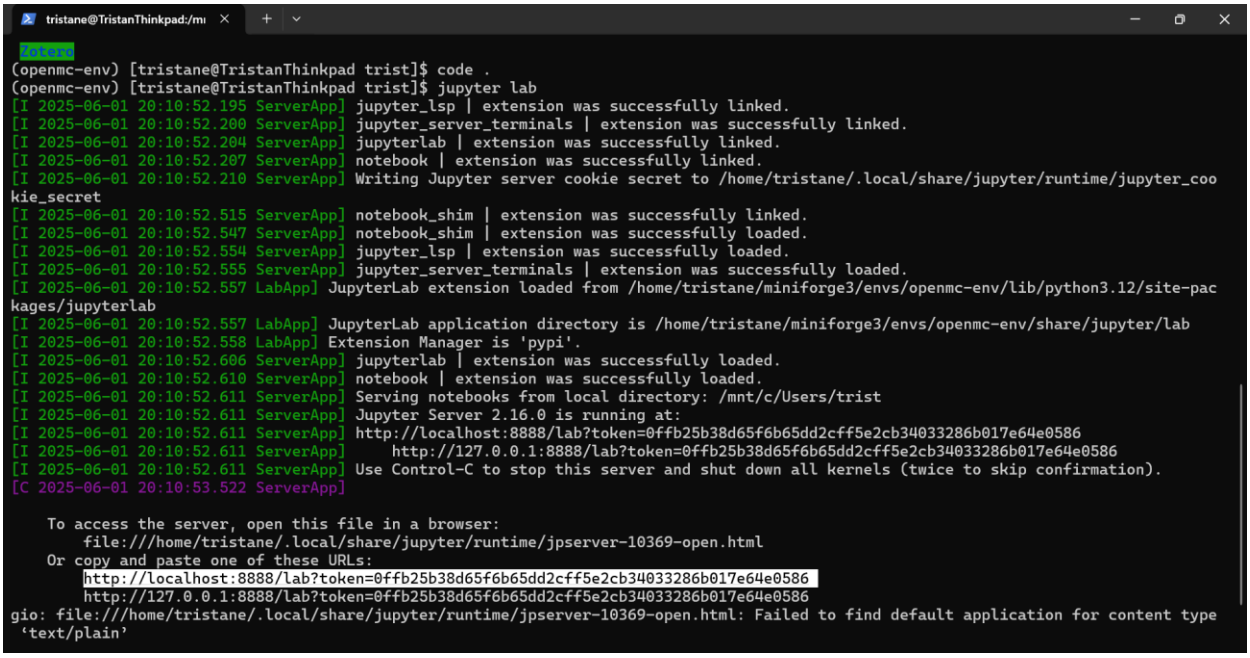
VS Code & Jupyter:


File Edit Selection View Go Run
Search
EXTENSIONS
Search Extensions in ...
LOCAL - INSTALLED 32
Material Theme Icons — ...
Material Theme Icons, the ...
Migrate
Remote - SSH
Open any folder on a remo...
Microsoft
WSL 227ms
Open any folder in the Win...
Microsoft
Atom One Dark Theme
One Dark Theme based on...
Mahmoud Ali
C/C++
C/C++ IntelliSense, debug...
WSL: ALMALINUX-10 - I... 0
RECOMMENDED 4
GitHub Copi... 39.2M 4
Your AI pair programmer
GitHub
Install
Docker 45.6M 4.5
Makes it easy to create, ma...
Welcome
Extension: WSL
WSL
Microsoft microsoft.com 35,047,977 (82)
Open any folder in the Windows Subsystem for Linux (WSL) and take advantage of Visual Studio C...
Disable
Uninstall
Auto Update
DETAILS FEATURES CHANGELOG
Visual Studio Code WSL
The WSL extension lets you use VS Code on Windows to build Linux applications that run on the Windows Subsystem for Linux (WSL). You get all the productivity of Windows while developing with Linux-based tools, runtimes, and utilities.
The WSL extension lets you use VS Code in WSL just as you would from Windows.
Why do I need the WSL extension?
Why WSL?
WSL lets you run a Linux environment -- including command-line tools and applications -- directly on Windows, without the overhead of a traditional virtual machine or dualboot setup. WSL especially helps web developers and those working with Bash and Linux-first tools (i... Ruby, Python) to use their toolchain on Windows and ensure consistency between
Installation
Identifier ms-vscode-remote.remote-wsl
Version 0.99.0
Last Updated 2025-06-01, 15:58:59
Size 2.49MB
Marketplace
Published 2019-05-02, 13:40:03
Last Released 2025-04-09, 10:58:25
WSL: AlmaLinux-10


File Edit Selection View Go Run
Search
EXTENSIONS: MARKETPLACE
jupyter
Jupyter 276ms
Jupyter notebook support, ...
Microsoft
Jupyter Keymap
Jupyter keymaps for noteb...
Jupyter Slide Show 4ms
Jupyter Slide Show support...
Jupyter Cell Tags 6ms
Jupyter Cell Tags support f...
Jupyter Notebook ... 5ms
Renderers for Jupyter Note...
Jupyter (de... 810K 3.5
Data Science with Jupyter ...
Don Jayamanne
Install
Jupyter Po... 639K 4.5
Experimental features for J...
VS Code Ju... 741K 3.5
Welcome
Untitled-1.ipynb
Extension: Jupyter
Jupyter
Microsoft microsoft.com 92,207,513 (333)
Jupyter notebook support, interactive programming and computing that supports Intellisense, de...
Uninstall
Switch to Pre-Release Version
Auto Update
DETAILS FEATURES CHANGELOG
Extension Pack (4)
Jupyter Keymap
Jupyter keymaps for notebooks
Microsoft
Jupyter Notebook Renderers 5ms
Renderers for Jupyter Notebooks (with plotly...
Microsoft
Jupyter Extension for Visual Studio Code
A Visual Studio Code extension that provides basic notebook support for language kernels that are supported in Jupyter Notebooks today, and allows any Python environment to be use... Jupyter kernel. This is NOT a Jupyter kernel--you must have Python environment in which
Marketplace
Identifier ms-toolsai.jupyter
Version 2025.4.1
Published 2020-11-11, 13:14:18
Last Released 2025-05-30, 04:54:19
Categories
Extension Packs
Data Science
Machine Learning
Visualization
Notebooks


File Edit Selection View Go Run
Welcome
Select an option to open a Remote Window
Connect to WSL WSL
Connect to WSL using Distro...
Connect to Host... Remote-SSH
Connect Current Window to Host...
Tunnel Install
Dev Container
GitHub Codespace
Remote Repository
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
New Workspace with Copilot...
Recent
tristane [WSL: AlmaLinux-10] /home
trist [WSL: AlmaLinux-10] /mnt/c/Users
testfiles [WSL: Ubuntu] ~
miniforge [WSL: Ubuntu] /mnt/c/Users/trist/Desktop
projects [SSH: hpclogin] /
More...
Get Started with WSL
Learn the Fundamentals
Get Started with Python Development Updated
Get Started with C++ Development Updated
More...
Show welcome page on startup


File Edit Selection View Go Run
Select a Python Environment
Create Python Environment
base (Python 3.12.10) miniforge3/bin/python Recommended
openmc-env (Python 3.12.10) miniforge3/envs/openmc-env/bin/python Conda Env
Python 3.12.9 /bin/python Global Env
Python 3.12.9 /usr/bin/python
EXPLORER
OPEN EDITORS
testfile.py 2
jupytertest.ipynb
TRISTANE [WSL: ALMALINUX-10]
.cache
.conda
.dotnet
.ipython
.nv
.vscode-server
miniforge3
.bash_history
.bash_logout
.bash_profile
.bashrc
.condarc
jupytertest.ipynb
testfile.py 2
Generate
Code
Markdown
openmc-env (Python 3.12.10)
Python
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS 2
JUPYTER
bash - tristane
[tristane@TristanThinkpad ~]$
OUTLINE
TIMELINE
WSL: AlmaLinux-10
Spaces: 4
LF
Cell 1 of 1


tristane@TristanThinkpad:/mı
Zotero
(openmc-env) [tristane@TristanThinkpad trist]$ code .
(openmc-env) [tristane@TristanThinkpad trist]$ jupyter lab
[I 2025-06-01 20:10:52.195 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2025-06-01 20:10:52.200 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2025-06-01 20:10:52.204 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-06-01 20:10:52.207 ServerApp] notebook | extension was successfully linked.
[I 2025-06-01 20:10:52.210 ServerApp] Writing Jupyter server cookie secret to /home/tristane/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2025-06-01 20:10:52.515 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-06-01 20:10:52.547 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-06-01 20:10:52.554 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-06-01 20:10:52.555 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-06-01 20:10:52.557 LabApp] JupyterLab extension loaded from /home/tristane/miniforge3/envs/openmc-env/lib/python3.12/site-packages/jupyterlab
[I 2025-06-01 20:10:52.557 LabApp] JupyterLab application directory is /home/tristane/miniforge3/envs/openmc-env/share/jupyter/lab
[I 2025-06-01 20:10:52.558 LabApp] Extension Manager is 'pypi'.
[I 2025-06-01 20:10:52.606 ServerApp] jupyterlab | extension was successfully loaded.
[I 2025-06-01 20:10:52.610 ServerApp] notebook | extension was successfully loaded.
[I 2025-06-01 20:10:52.611 ServerApp] Serving notebooks from local directory: /mnt/c/Users/trist
[I 2025-06-01 20:10:52.611 ServerApp] Jupyter Server 2.16.0 is running at:
[I 2025-06-01 20:10:52.611 ServerApp] http://localhost:8888/lab?token=0ffb25b38d65f6b65dd2cff5e2cb34033286b017e64e0586
[I 2025-06-01 20:10:52.611 ServerApp] http://127.0.0.1:8888/lab?token=0ffb25b38d65f6b65dd2cff5e2cb34033286b017e64e0586
[I 2025-06-01 20:10:52.611 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2025-06-01 20:10:53.522 ServerApp]
To access the server, open this file in a browser:
file:///home/tristane/.local/share/jupyter/runtime/jpserver-10369-open.html
Or copy and paste one of these URLs:
http://localhost:8888/lab?token=0ffb25b38d65f6b65dd2cff5e2cb34033286b017e64e0586
http://127.0.0.1:8888/lab?token=0ffb25b38d65f6b65dd2cff5e2cb34033286b017e64e0586
gio: file:///home/tristane/.local/share/jupyter/runtime/jpserver-10369-open.html: Failed to find default application for content type 'text/plain'