

Sep 13, 2021

Install WSL and VSCode on Windows 10

Kenneth Schackart¹¹University of Arizona

1 Works for me

Share

dx.doi.org/10.17504/protocols.io.bx63prgnKenneth Schackart
University of Arizona

ABSTRACT

This protocol describes how to install Ubuntu, WSL, and connect to WSL with VS Code on Windows 10. By the end you should have a working version of Ubuntu and WSL. VS Code will be able to work with your Ubuntu installation. You will also have Python3 since Ubuntu comes packaged with it.

Most of the information presented here is given in the attached Microsoft Docs link.

DOI

dx.doi.org/10.17504/protocols.io.bx63prgn

EXTERNAL LINK

<https://docs.microsoft.com/en-us/windows/wsl/install-win10#manual-installation-steps>

PROTOCOL CITATION

Kenneth Schackart 2021. Install WSL and VSCode on Windows 10 . **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.bx63prgn>

KEYWORDS

windows 10, install wsl, install wsl2, install ubuntu, VS Code wsl

LICENSE

———— This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

IMAGE ATTRIBUTION

lewing@isc.tamu.edu Larry Ewing

CREATED

Sep 13, 2021

LAST MODIFIED

Sep 13, 2021

PROTOCOL INTEGER ID

53179

GUIDELINES

This protocol only pertains to machines running Windows 10. Your results may vary for other Windows operating systems.

MATERIALS TEXT

Windows 10 machine

SAFETY WARNINGS

Getting WSL and Python running smoothly on windows can be frustrating, take a break before you want to beak something.

BEFORE STARTING

To update to WSL 2, you must be running Windows 10.

- For x64 systems: **Version 1903** or higher, with **Build 18362** or higher.
- For ARM64 systems: **Version 2004** or higher, with **Build 19041** or higher.
- Builds lower than 18362 do not support WSL 2. Use the [Windows Update Assistant](#) to update your version of Windows.

To check your version and build number, select **Windows logo key + R**, type **winver**, select **OK**. [Update to the latest Windows version](#) in the Settings menu.

Enable WSL

- 1 Open Windows Powershell **as Administrator**.

To run as Administrator, you can hit the Windows key, search for powershell, and on the right select "Run as Administrator". This allows you to make changes to your system.

- 2 In Powershell, enter the following command to enable WSL:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

- 3 In Powershell, enter the following to enable virtualization:

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

- 4 Restart your machine to make the changes take effect.

- 5 Download the Linux kernel update package:

https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi

Run the package.

Set WSL2 as Default

- 6 Open Powershell as administrator and run:

```
wsl --set-default-version 2
```

Install Ubuntu

- 7 Go to the Microsoft Store and install a version of Ubuntu.

Hit Windows key, type "Store", select Microsoft Store. Search for Ubuntu. Installing a version that ends with LTS

means it is *Long term stable*. I recommend one of those versions.

Check Installation

- 8 Open Ubuntu by hitting the Windows key, and type "Ubuntu", and open it.

If an Ubuntu window opens but is quickly closed, continue to the next step.

If Ubuntu opens properly, go to step 13 to set up VS Code.

- 9 If WSL has failed to open by briefly flashing and then closing, it means your machine is still blocking virtualization. This must be resolved by entering BIOS. How to do this varies by laptop type. It will generally involve shutting the machine off, powering it on, and while it boots you need to hit a certain key to pause boot up.

Search the web for "How to enter BIOS <insert computer name here>"

You will not be able to see this protocol in BIOS, so open this page on another device.

- 10 Once in BIOS, use the arrow keys to navigate the menus and search for any settings regarding "virtualization" and enable them.

This may be something like "Virtualization Technology".

Refer to the menu at the bottom of the BIOS screen for instructions on how to navigate, change settings, and save changes.

- 11 Save the changes you have made to the BIOS settings and restart.

Check Installation Again

- 12 Try starting Ubuntu again.

If it still gets closed, hit the Windows key, type features, and select "Turn Windows features on or off". If there is the feature "Windows Hypervisor Platform", enable that. Also make sure that "Virtual Machine Platform" and "Windows Subsystem for Linux" are enabled.

Set up VS Code

- 13 Install Visual Studio Code.

<https://code.visualstudio.com/download>

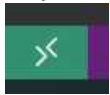
Open it once installed.

- 14 Open the Extensions tab by clicking on the icon with 4 squares (3 together, 1 detached).



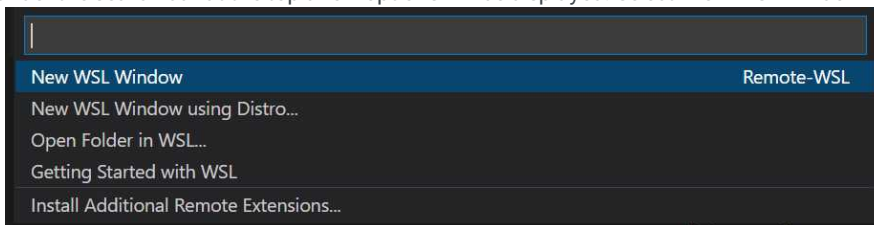
Screen capture of sidebar, with Extensions highlighted

- 15 Search for "Remote - WSL" and install it. You will now have a green square in the bottom left corner of VS Code. This is how you access WSL from VS Code.



Displaying no current remote connection

- 16 Click the green square in the bottom left corner. Under the search bar at the top a few options will be displayed. Select "New WSL window"



Remote options menu, with first option highlighted to open new WSL window

You can then close the old window.

- 17 Now VS Code is accessing your WSL instance. The bottom left corner should now look something like this:



Displaying active remote connection to WSL Ubuntu

Update

18 From the top menu, select Terminal -> New Terminal

19 In the new terminal run the following to update

```
sudo apt-get update
```

20 You should now have a working version of WSL and access to it through VS Code.

Some things to keep in mind:

- You can see if your current VS Code session is running in WSL by looking at the green rectangle at the bottom left corner
- Since WSL has its own file system, if you plan to interact with GitHub using SSH keys, you will need to generate new ones using the WSL terminal, and add them to your GitHub
- Whatever repositories you want to run in WSL should be cloned using the WSL terminal
- WSL can be run without VS Code to have a nice linux terminal