Docker on Windows with WSL2

Table of Contents

Overview

Corporates now can ONLY utilized a licensed version of Docker Desktop Client on Windows.

Hence, this page details an alternative way to continue to use windows for development environment while working on docker.

The procedure involves two broad steps:

1. Configure windows to have Ubuntu as a sub system.
2. Docker installation on Ubuntu sub-system.

Detailed steps are described below.

Pre-requisite

1. Windows 10 version 2004 or higher and specific build

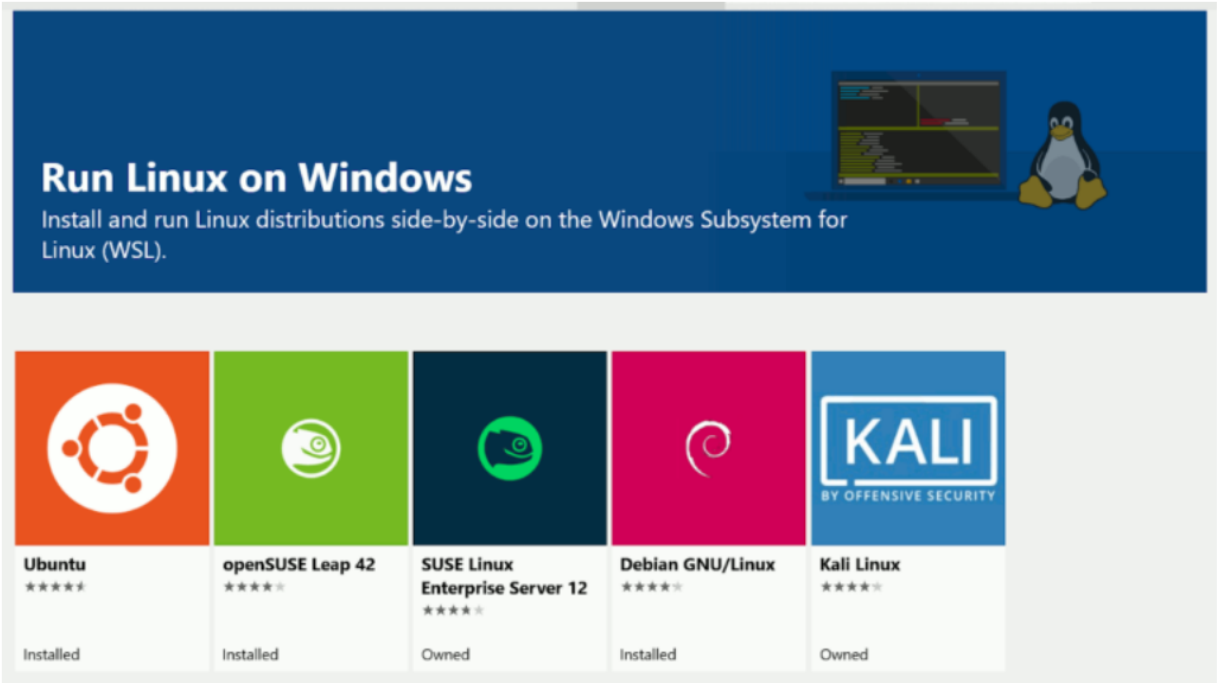
- Specific build details are: **Windows 11 or Windows 10, Version 1903, Build 18362 or later**.
- To check which version is installed press Windows logo key + R and press Enter to view Windows build details.

NOTE: Windows upgrade would be required if it does not meet these version and build details for WSL2 to work,.

2. Disconnect VPN

What is Windows Subsystem for Linux (WSL and WSL2)? — Linux in Windows

https://fossbytes.com/what-is-windows-subsystem-for-linux-wsl/

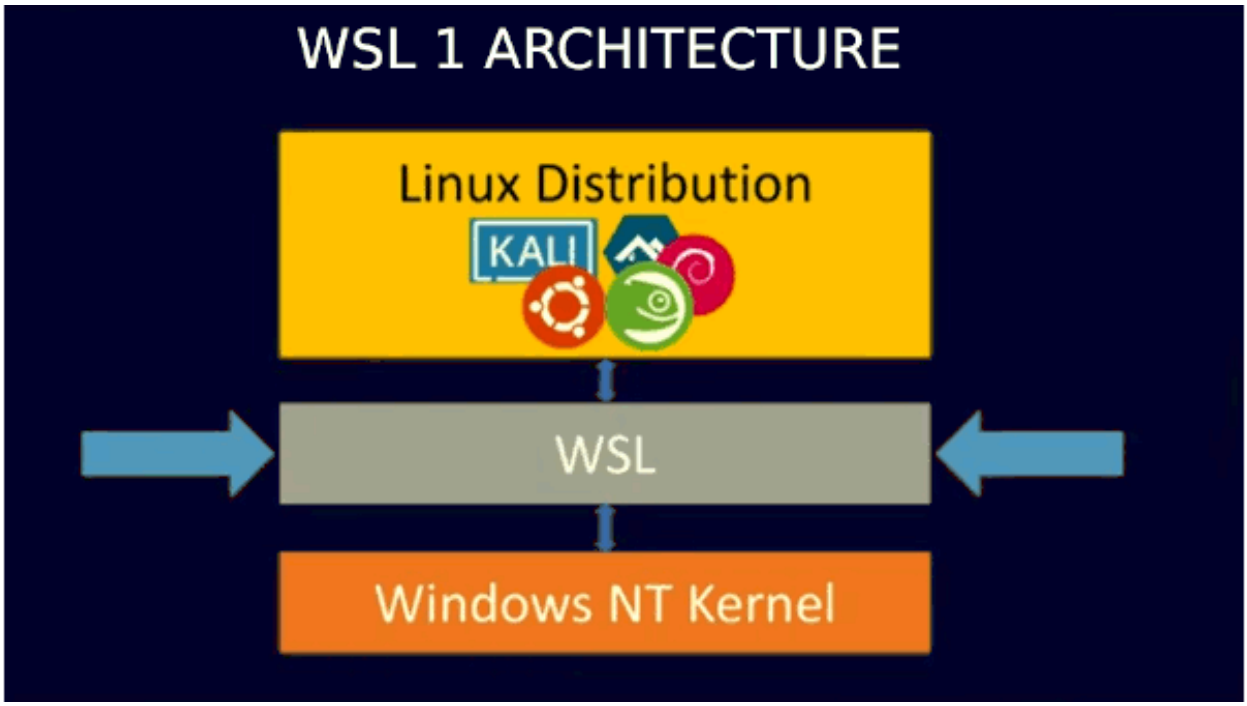


What is WSL (Windows Subsystem for Linux)?

Windows Subsystem for Linux has gained a lot of popularity over the last few Windows 10 updates. Among the barrage of features and additions, WSL stands out as a welcome change. While most of us depended on Virtual Machines to get the job done, WSL seems to be a remarkably lighter alternative to handle the necessary tasks. The latest addition is the improved WSL2.

So what is WSL? Let's have a look at what WSL is. Also, check out the differences between WSL and WSL2. Plus, we'll compare each of them with Virtual Machines, so that we can have a better understanding of the use case scenarios.

WSL was the initial release added to the Windows 10 version 1607 update. WSL is an **optional feature** in Windows 10 that provides the users with a **Lightweight Linux console environment** within the Windows Operating System. Windows Subsystem for Linux adds a **compatibility layer,** which allows you to run the Linux binaries to the native Windows 10 OS—thus eliminating the need for the installation of Virtual Machine to run most command-line tasks.



How does WSL work?

Well, WSL translates the Linux system calls returned by the process to the Windows kernel calls while a Linux Distro sits on top. It is thus eliminating the need for the Linux kernel in the process. So basically, it is **Linux inside Windows**. Furthermore, there is **no emulation** involved whatsoever.

This feature follows suit with Microsoft's ambitions to bring the **Linux experience to Windows 10** natively. One of the reasons why Linux operating systems are highly popular is due to its superior dev environment. With the implementation of the Windows subsystem for Linux, Redmond giant plans to bring the best of both worlds to the Developers.

Why use WSL?

- **Linux Command line tools**
- **Windows 10 Filesystem access**
- **No VM needed**

The Windows Subsystem for Linux allowed the developers and system administrators to use core **command-line tools** and services. But most importantly, the **bash shell**. The availability of package tools such as **apt** and **dpkg** in of itself affords developers a plethora of command-line GNU applications.

Additionally, Windows 10 **filesystem** and drives can be accessed and handled within the WSL console interface itself. The drive partitions are mounted at launch within the path of /mnt/. For example, the logical drive C:\ mount at /mnt/c.

Talking about the interface in the WSL, we are limited to the **command-line interface**. GUI (Graphical User Interface) does not exist. Although we can run a minimal number of GUI applications with the help of small tweaks, the terminal console is all we would be working with.

Moreover, the list of **available distributions** is sufficient. Although Ubuntu was the only one distro, which was available initially, the list keeps growing continuously. Also, we can install more distros unavailable in the list with the use of some workarounds.

Why not Use WSL?

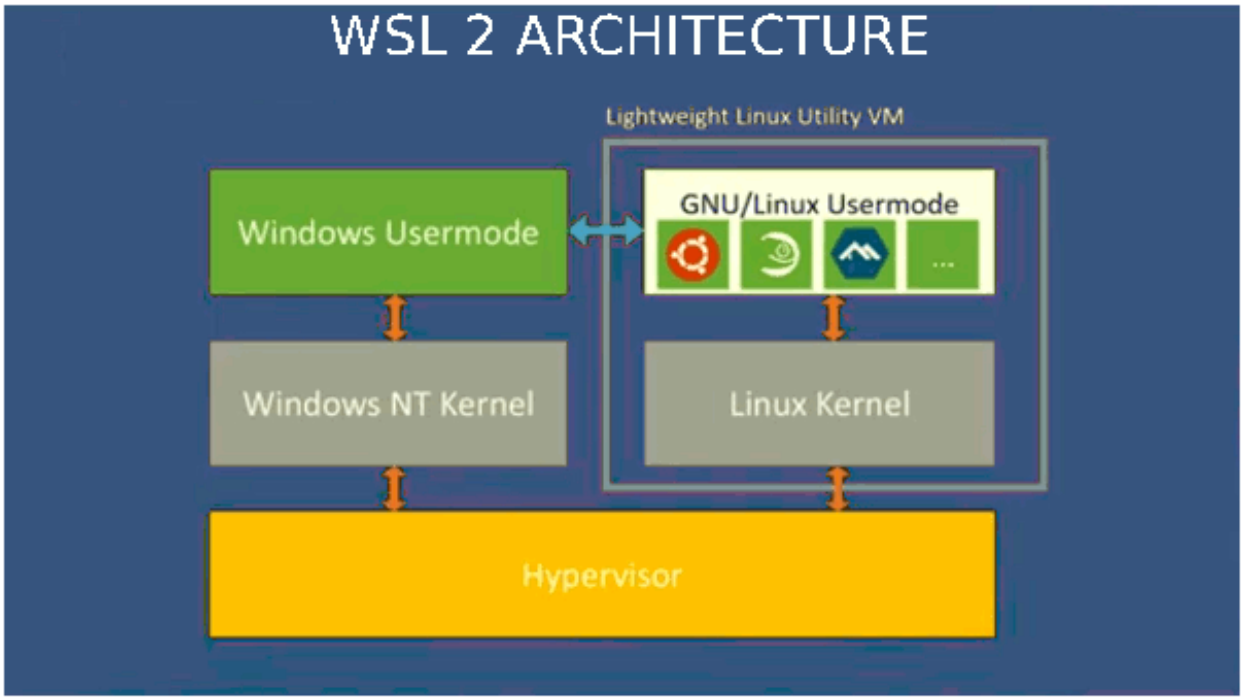
First of all, there's the obvious problem of scalability. As there is **no native Linux kernel support**, comprehensive processes and tasks are limited.

Services like **Docker are not supported** in the first generation release of the Windows Subsystem for Linux owing to the unavailability of kernel-level support for Linux containers.

Network communication (although possible) requires several layers to pass through, making the WSL **not ideal for low-level networking** processes.

For tasks requiring more peripheral access like graphics, sound, and high-speed file I/O, WSL may not be the best option.

WSL2: How is the architecture different from WSL1?



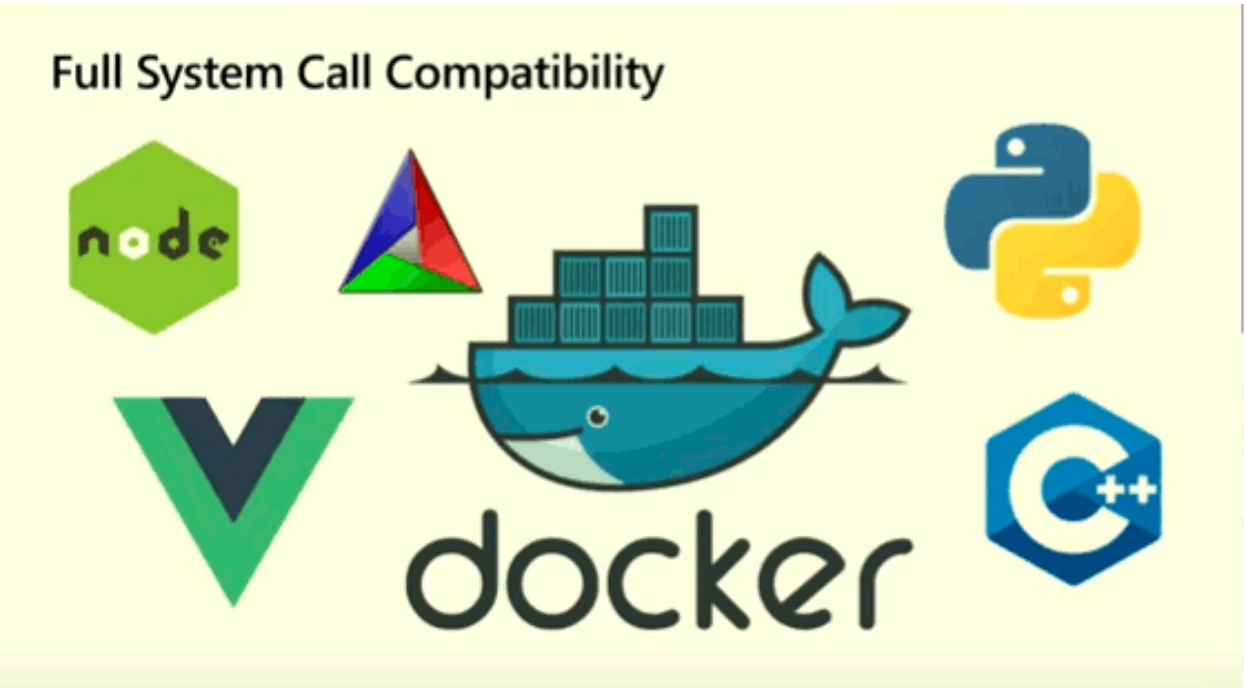
The new and improved WSL2 is the apparent successor to WSL, which is available from the Windows 10 version 2004. WSL2 architecture is entirely different from WSL1. Unlike WSL1, which runs as a translation layer, the Windows subsystem for Linux 2 has a **virtual machine** with an actual Linux Kernel that can immediately react to system calls. Such an architecture enables **Full System call** capabilities.

Additionally, the Linux Kernel used in the WSL 2 architecture, is a Microsoft developed open-source Linux Kernal. Along with a significant overhaul from its predecessor, WSL2 has under the hood changes that help alleviate some of the drawbacks of the original version. Let's review these improvements a bit more —

WSL2 vs. WSL1: How is WSL2 better?

- **An Actual Linux Kernel**
- **Full System Call Capability**
- **Socket/Networking Performance**
- **File System Performance**

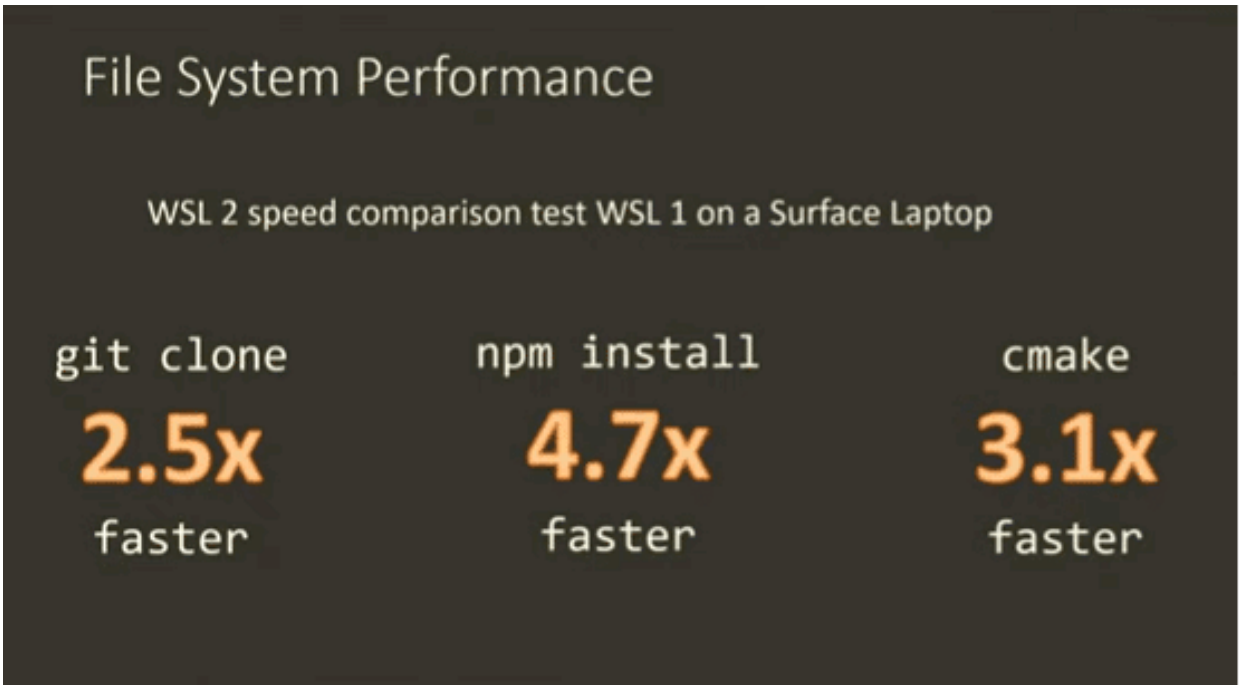
As we said before, the whole architecture changed dramatically with the new Windows Subsystem for Linux 2 release. WSL2 is a genuinely **lightweight virtual machine** based on Microsoft's hardware virtualization service Hyper-V. Having a native **hypervisor** helps the WSL2 implementation to pull clear of any disadvantages that WSL1 had when it first launched.



For example, the availability of a real **Linux Kernal** enables the user to execute tasks with full system call compatibility. So, the platform becomes much more potent for handling Linux-based applications. Also, services like Docker and fuse will be compatible.

File I/O performance is much improved. Therefore, operations such as apt-installs, npm installs will be snappier. Furthermore, this should be the case for all read and write and services as well.

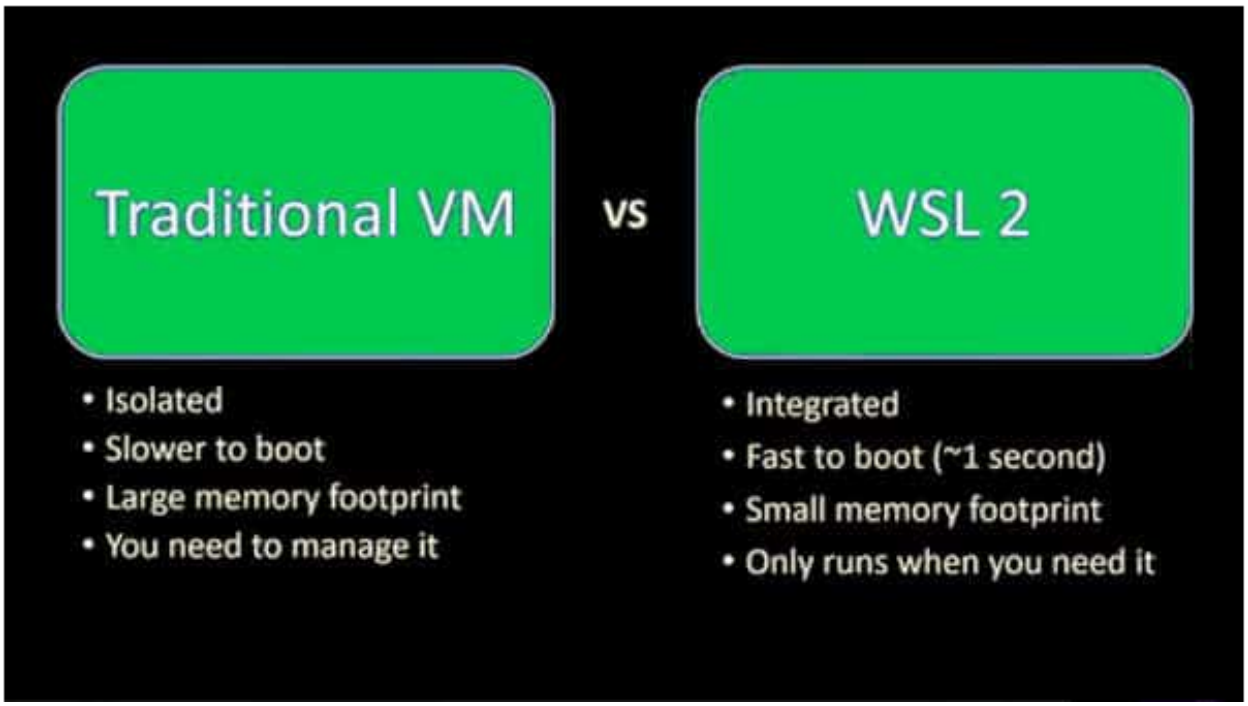
*(**Note:** Microsoft did mention slower cross OS file speeds in the initial builds.)*



Finally, Version **Switching** is as simple of a process there is. Along with this, the networking and socket performances has had considerable improvements.

WSL/WSL2 vs. VirtualBox/ Virtual Machines?

So how does it stand out from the traditional Virtual Machines? After all, VM, such as Virtual BOX, remains to be the **most popular option** that users have for a similar cross-platform experience.



A Virtual Machine creates an environment inside your system that acts as a separate virtual system with all the hardware components like CPU, RAM, Storage with the help of Hyper Visors. Such an architecture enables the user to have a proper and complete system within the operating system that you are running. Unlike WSL/WSL2, Virtual Machines can run full graphical interfaces as the environment itself is a full-fledged Linux workspace. But as usual, there are caveats.

A Virtual Machine is **Isolated**. What this means is that you won't be able to access files outside the Virtual System (*a.k.a. your actual system files*).

Slower boot and performance: Virtual machines typically run inside an operating system. So, they can access the actual hardware only through the OS, which, in turn, results in performance issues. To put it simply, Virtual Machines are only as powerful as your hardware allows it to be. That said, most modern hardware is powerful enough for such heavy-duty tasks.

Memory Usage: Whenever we require the need for a VM, a large chunk of memory must be dedicated to it at all times.

Management: Virtual Machines have operating systems that require management.

Which Should You Use?

Now that we have learned how each of these services works, let's look at the use case scenarios:

Why Choose a Virtual Machine?

- You require all the power and features of Linux.
- For the Larger scale, Production level works.
- GUI (Graphical User interface)
- Graphical or Network-intensive workloads.

When to choose WSL/WSL2?

- For using basic Linux commands.
- You have less powerful hardware.
- You need memory/storage efficiency.
- When you need faster I/O.
- Don’t want to deal with the hassle of installation/management/overheads of a VM.

Finally, it is safe to say that developers are the clear winners as they are the biggest beneficiaries of these new features that Windows 10 brings.

WSL 1 creates a Linux environment on Windows through the use of a Linux binary translation layer. WSL 2 does so with a lightweight virtualization platform based on Hyper-V.

Kernel Drivers

When WSL is enabled on Windows 10, two NT core kernel drivers are loaded by Windows 10 (Figure 1-1). These drivers are Lxss.sys, a stub driver loaded early in the boot process, and LxCore.sys, the full WSL driver, which is loaded later in the boot process.

Pico Processes

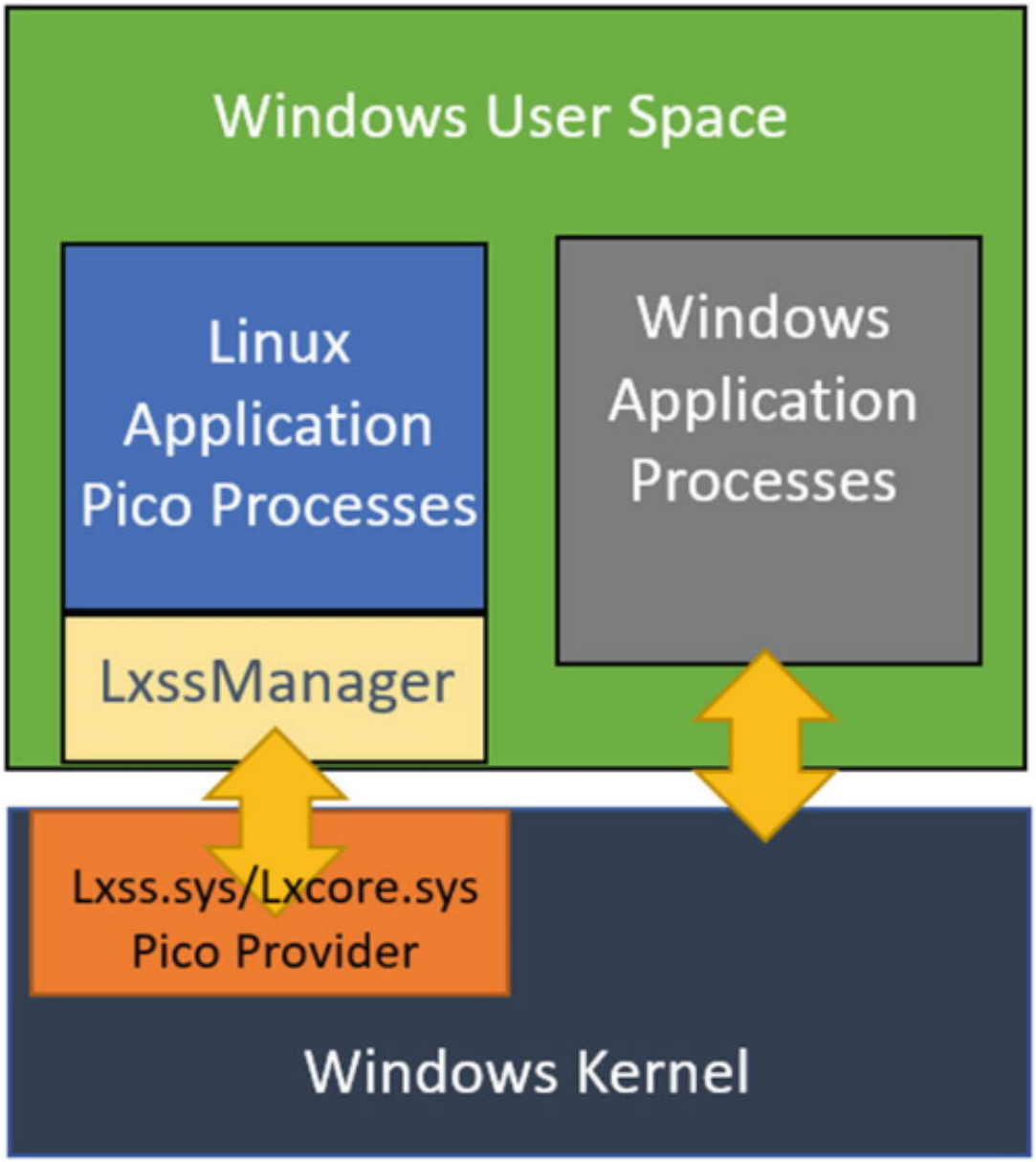


Figure 1-1 Diagram of WSL 1 architecture

WSL 2

WSL 2 is a vastly different architectural approach from WSL 1. By leveraging a Linux kernel and a lightweight Hyper-V container, WSL 2 addresses many of the issues users encountered with WSL 1, such as syscall incompatibility.

Hyper-V

WSL 2 addresses the challenge of implementing complete system call translation support for every possible Linux system call by implementing a true Linux kernel in a lightweight virtualization platform built on Hyper-V (Figure 1-5).

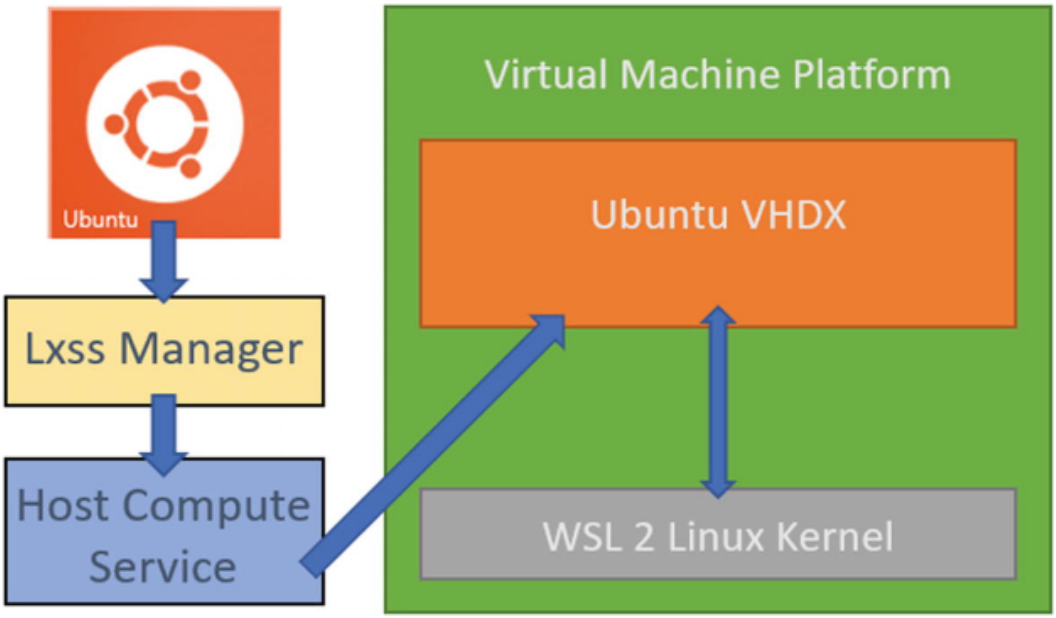


Figure 1-5 Diagram of WSL 2 architecture

Hyper-V is the native virtualization technology built into the Windows NT kernel, equivalent to the native virtualization implementations such as KVM on Linux or Hypervisor.Framework on macOS.

Step1: Install Windows Subsystem for Linux

This documented process will help you to install the Ubuntu Sub - System on your windows machine side by side while u are using windows and will allow you to install docker in the end

https://docs.microsoft.com/en-us/windows/wsl/about

The Windows Subsystem for Linux lets developers run a GNU/Linux environment -- including most command-line tools, utilities, and applications -- directly on Windows, unmodified, without the overhead of a traditional virtual machine or dualboot setup.

You can:

- Choose your favorite GNU/Linux distributions from the Microsoft Store.
- Run common command-line tools such as `grep`, `sed`, `awk`, or other ELF-64 binaries.
- Run Bash shell scripts and GNU/Linux command-line applications including:
    - Tools: vim, emacs, tmux
    - Languages: NodeJS, Javascript, Python, Ruby, C/C++, C# & F#, Rust, Go, etc.
    - Services: SSHD, MySQL, Apache, lighttpd, MongoDB, PostgreSQL.
- Install additional software using your own GNU/Linux distribution package manager.
- Invoke Windows applications using a Unix-like command-line shell.
- Invoke GNU/Linux applications on Windows.

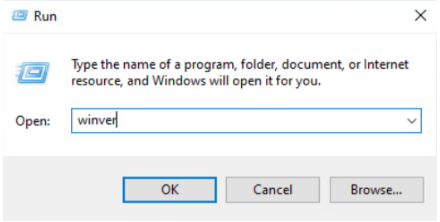
https://docs.microsoft.com/en-us/windows/wsl/install

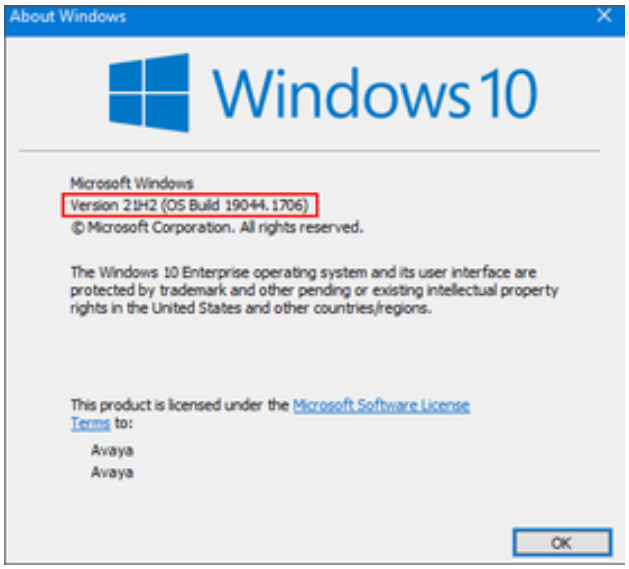
Install Ubuntu Sub System

Hardware & Software Requirements:

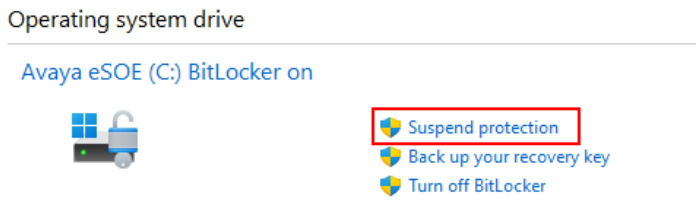
- Virtualization is enabled at BIOS level
- A computer with Hyper-V support
- Windows OS with the following version:
    - Windows 10 May 2020 (2004)
    - Windows 10 May 2019 (1903)
    - Windows 10 November 2019 (1909) or later

a. Use ***winver*** to check your windows version:

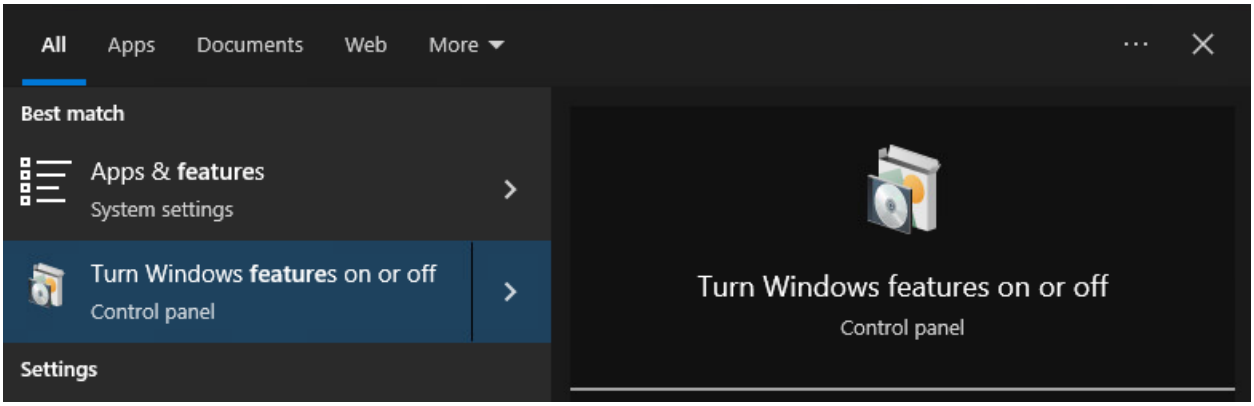




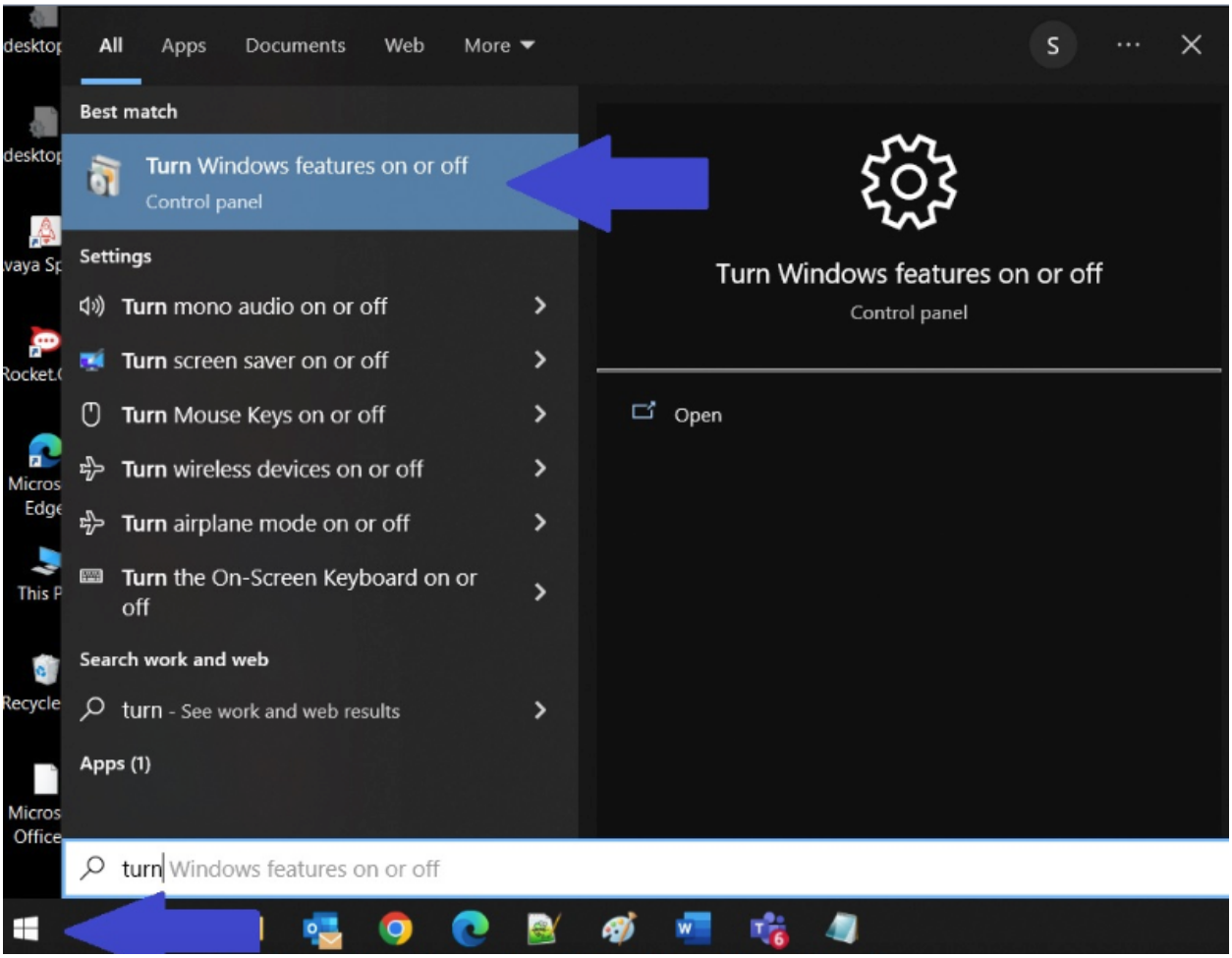
b. Before starting the feature installation, first **suspend the BitLocker:**

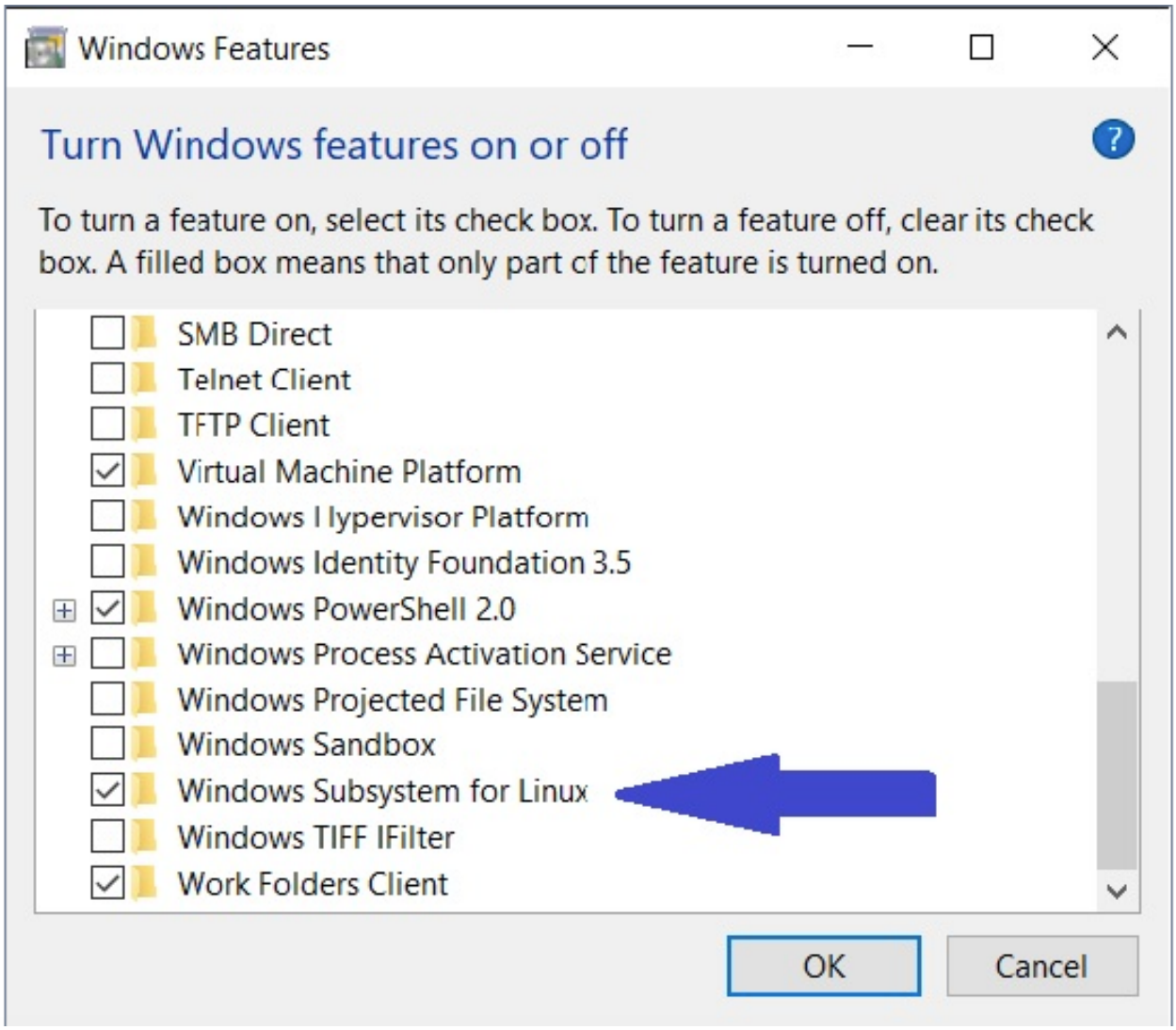


To enable the Hyper-V feature, go to the '**Turn Windows features on or off**':

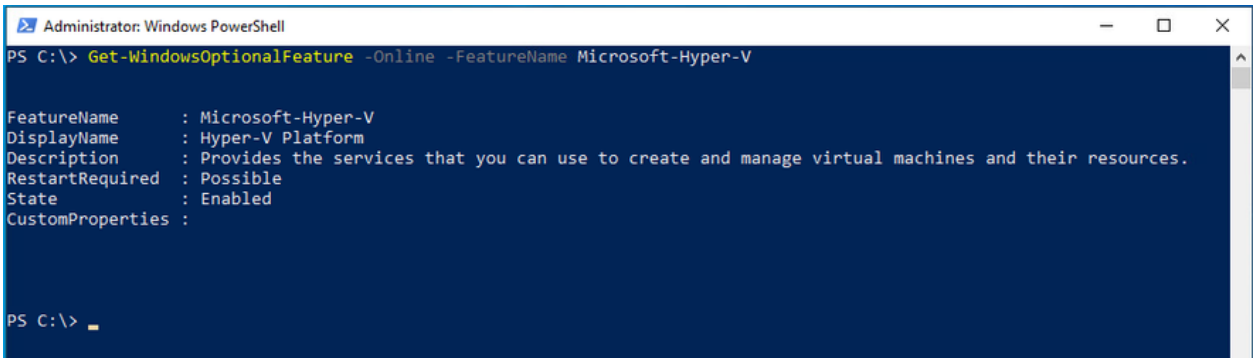


Then enable the **Hyper-V** features as shown in the below image followed by a system reboot:

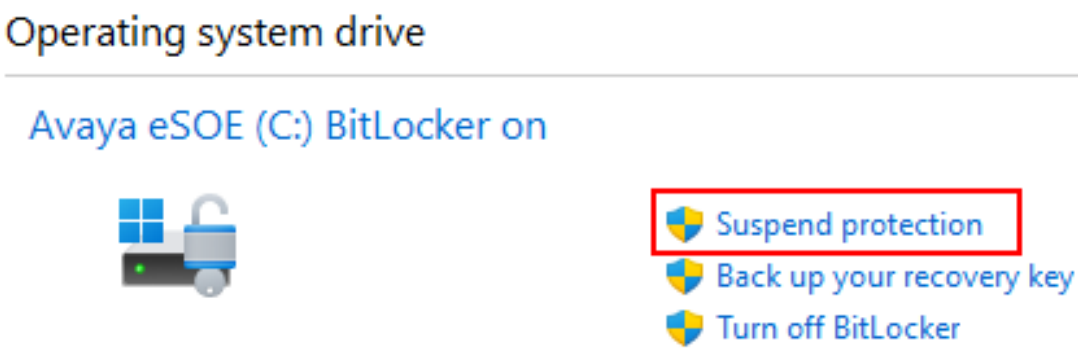




Once the system is back online, login and start the PowerShell in Administrator mode and check the status of Hyper-V:



c. Before starting the WSL v2 installation, first **suspend the BitLocker:**



Now install the **WSL v2** using the below command in the Windows Command Prompt which has to be opened in the Administrator mode:

Use this Guide

https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-10#2-install-wsl

https://documentation.ubuntu.com/wsl/en/latest/guides/install-ubuntu-wsl2/

Restart your Laptop.

This concludes the WSL2 setup.

Installing Ubuntu on WSL2

1. Open a PowerShell with administrative privileges.
2. Execute the below command to set the default version of wsl2:

   wsl --set-default-version 2

3. Install Ubuntu in WSL2 with the following command:

   wsl --install -d Ubuntu-20.04

NOTE: Ubuntu-20.04 is to be installed.

----------Sample output----------

PS C:\WINDOWS\system32> wsl --install -d Ubuntu

Downloading: Ubuntu

Installing: Ubuntu

Ubuntu has been installed.

Launching Ubuntu...

******A sperate window for Ubuntu would be launched ********

Installing, this may take a few minutes...

Please create a default UNIX user account. The username does not need to match your Windows username.

For more information visit: https://aka.ms/wslusers

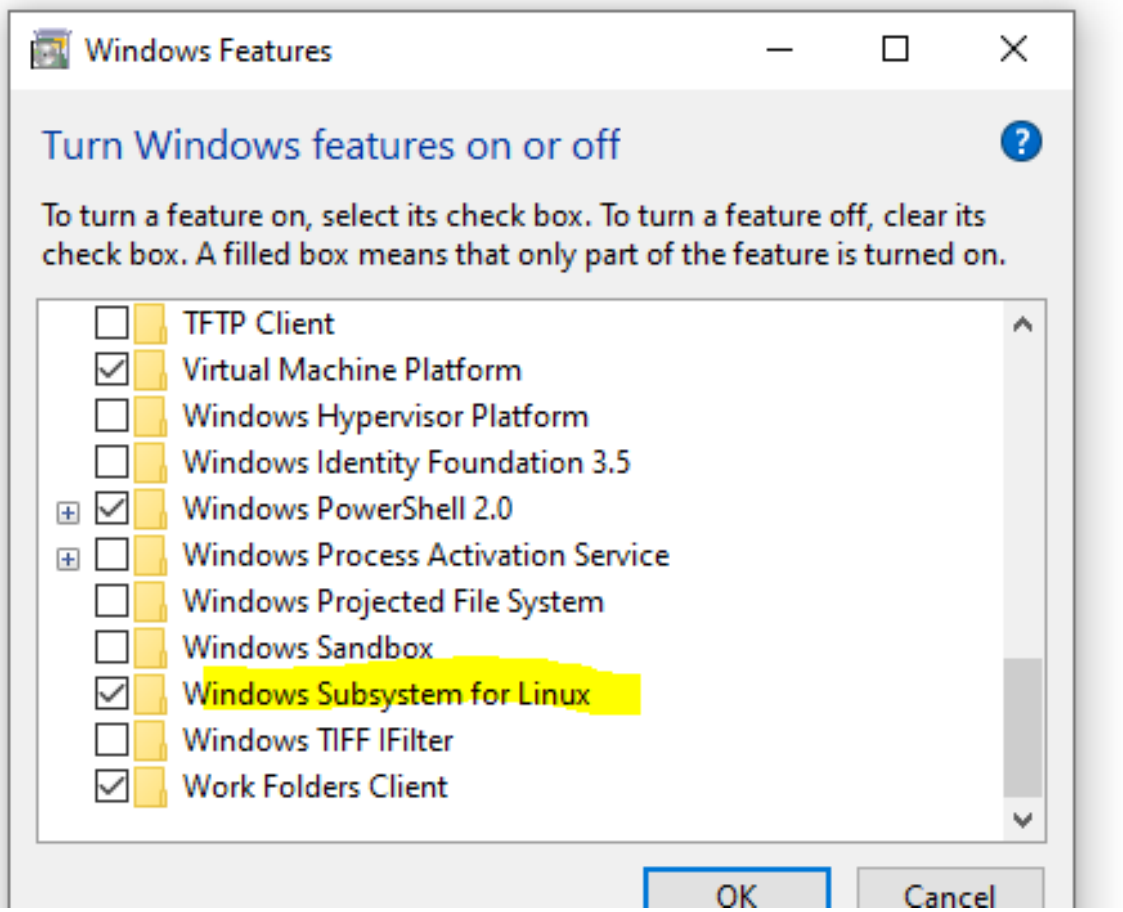
- Enter new UNIX username: <Give a username>
- New password:<Set a password>
- Retype new password: <Retype password>
- passwd: password updated successfully

Installation successful!

4. Validate the version of Ubuntu installed by executing the below command at Power Shell.

```
wsl -l -v
```

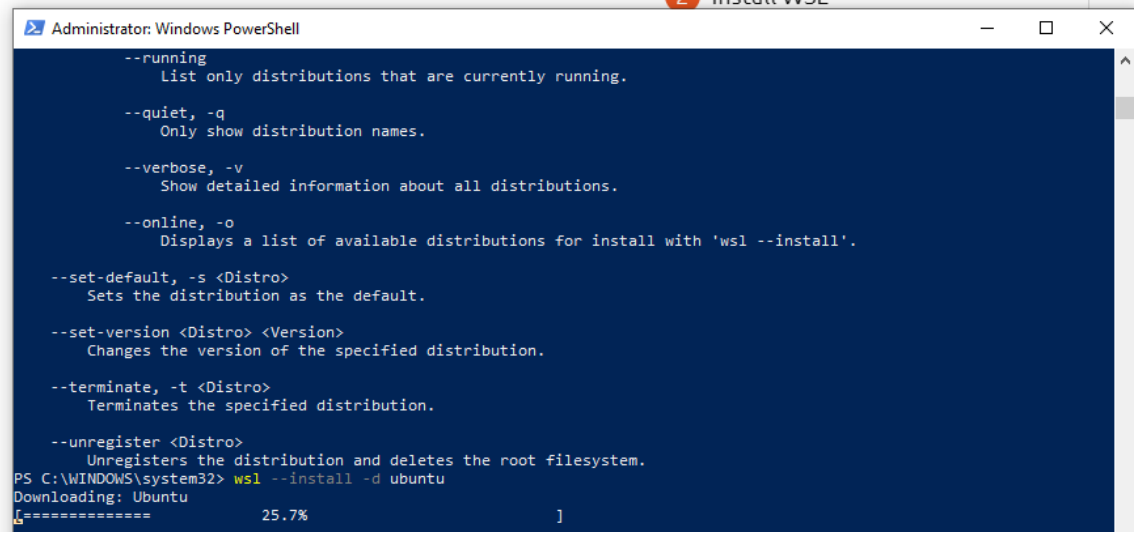
5. Every time to get the Ubuntu sub system up, you may locate for it here at your windows laptop as show in the below image:

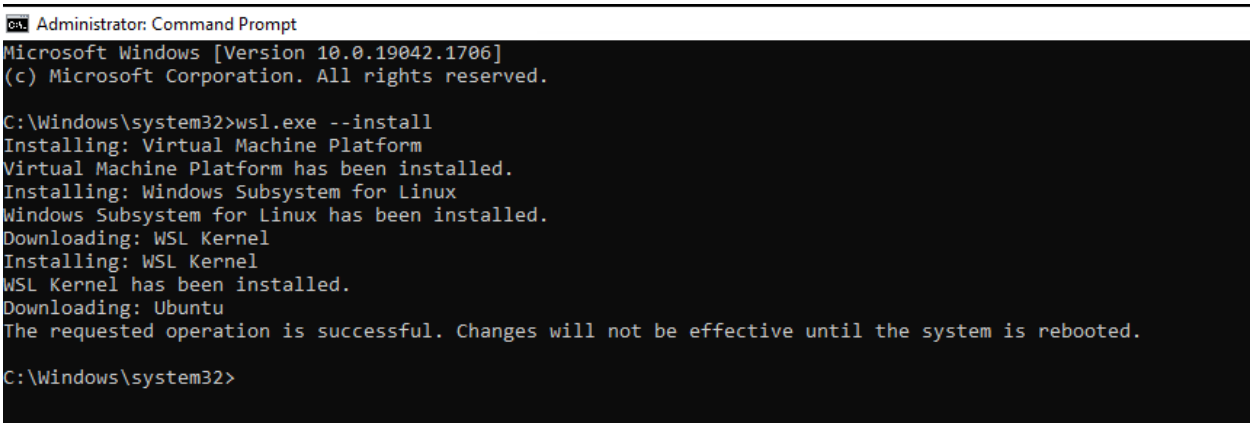

Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.
TFTP Client
Virtual Machine Platform
Windows Hypervisor Platform
Windows Identity Foundation 3.5
Windows PowerShell 2.0
Windows Process Activation Service
Windows Projected File System
Windows Sandbox
Windows Subsystem for Linux
Windows TIFF IFilter
Work Folders Client
OK
Cancel

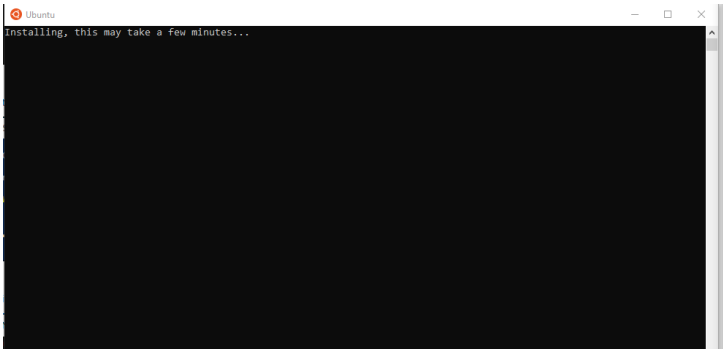
ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-10#2-install-wsl

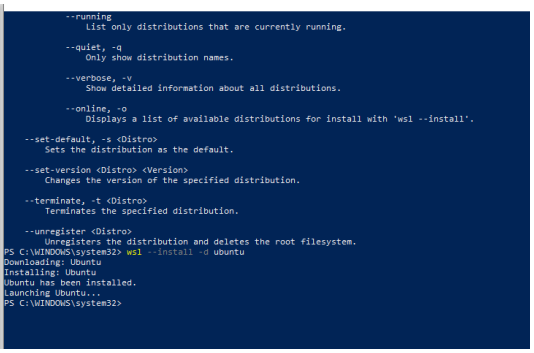
1 Overview

2 Install WSL



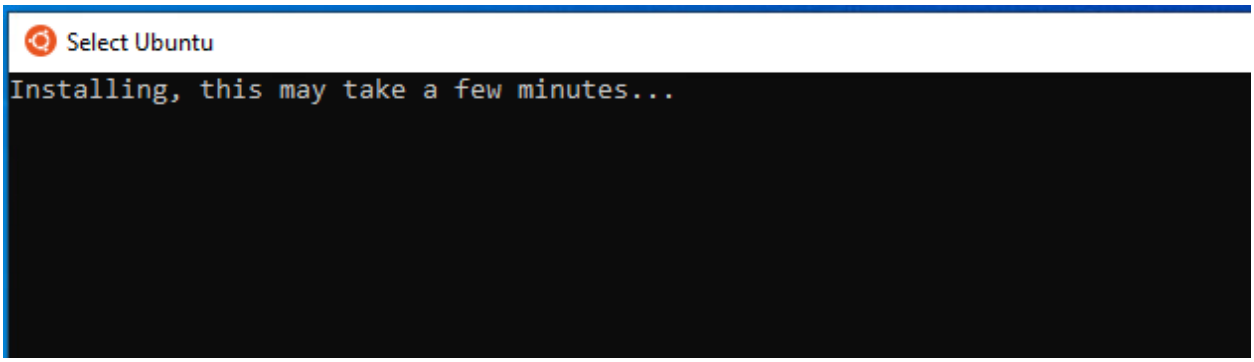




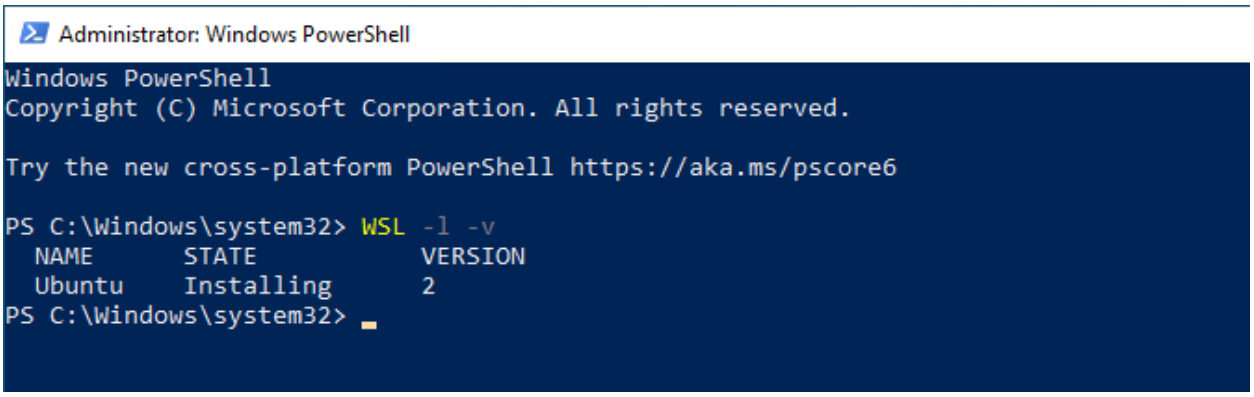


Reboot the system to complete the installation process. It will take some time to apply the changes.

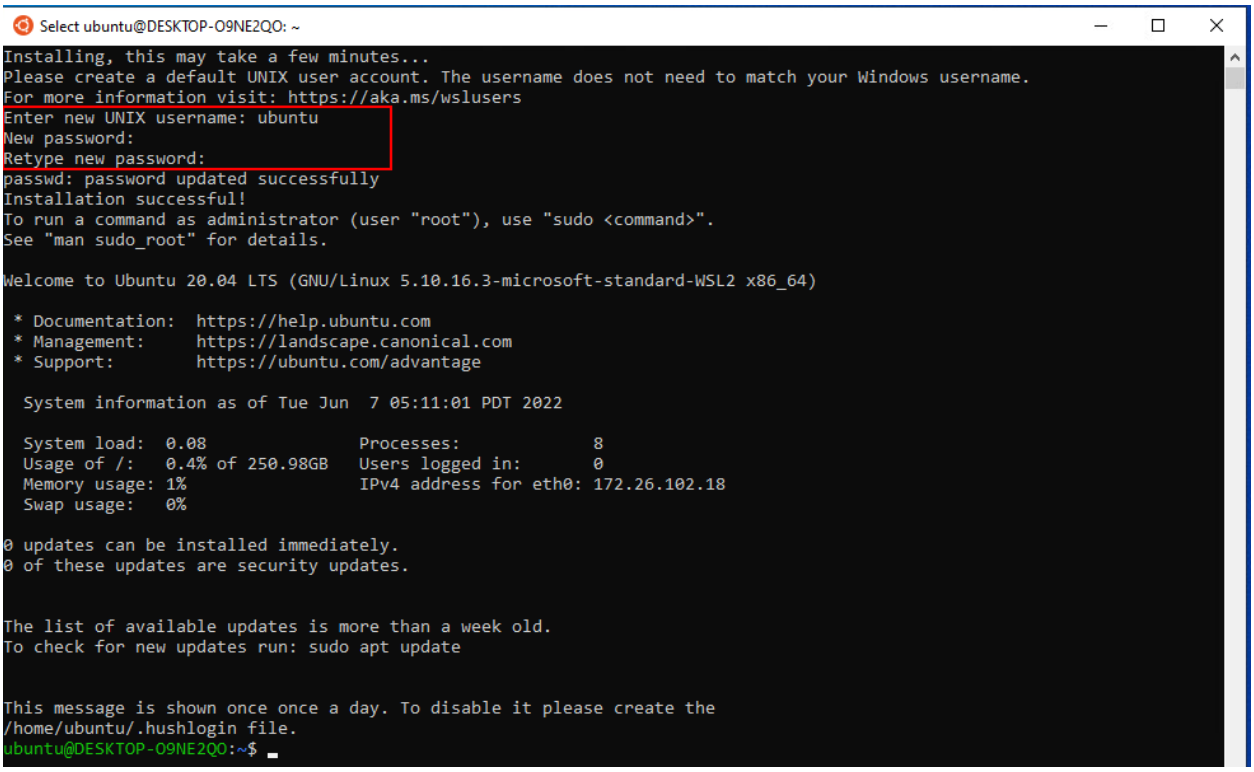
Once the system is back online, login to the system and you can see that its trying to install the ubuntu in the system



Same can be confirmed from the PowerShell:

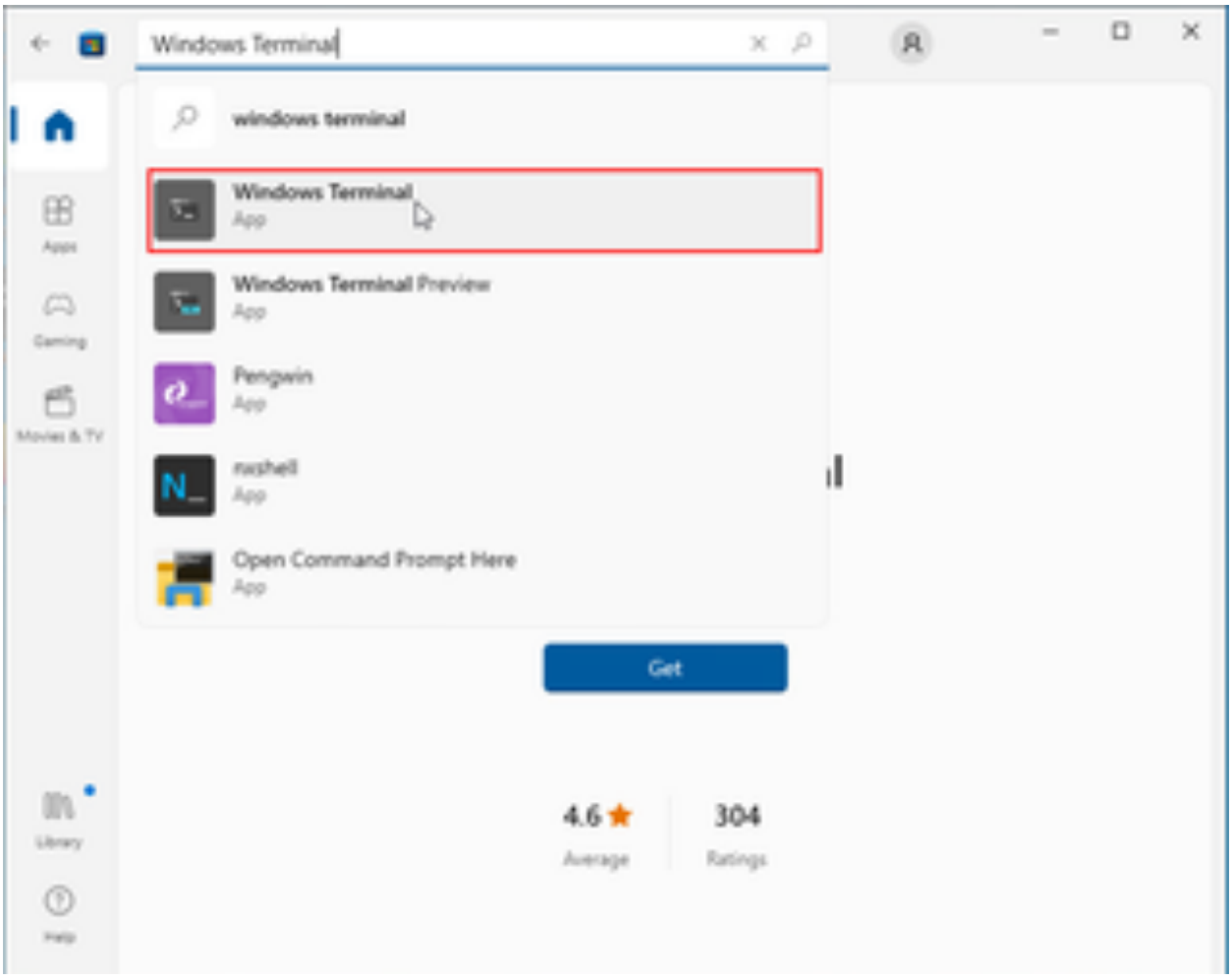


Once the installation is done, it will ask you to create an user and assign password to it:

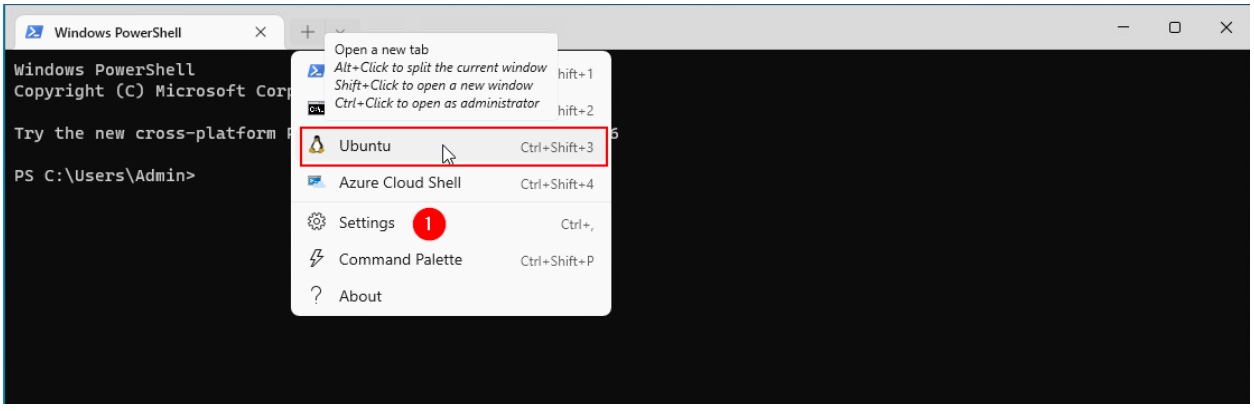


Now the installation is complete. Next step is to confiture the windows terminal to use the linux subsystem.

d. To install the Windows Terminal, start the Microsoft Store and search for **Windows Terminal** and click on **Get**:



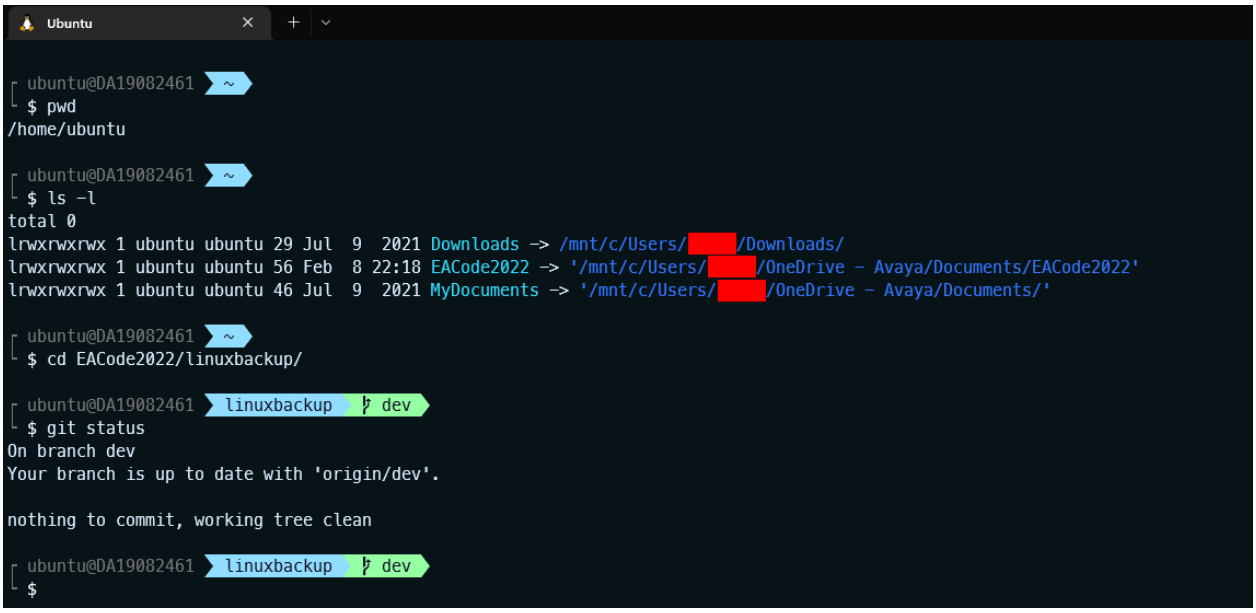
Once the installation is done, you can start it and click on drop-down and select the Ubuntu.



You can change the default terminal by selecting the Settings as showing in (1).

Your whole Operating system is mapped under **/mnt/c** drive. You can switch to Users/<User Name> to switch to your home directly.

Now you can refer Custom Prompt Setup to setup your windows terminal. Once the terminal configuration is done for customization, it should see similar to the below one:



At the home directly, you can create the Soft Links to the specific directories for quick access as shown in the above screenshot.

NOTE:

Some time the linux subsystem may crash. In that case you have to stop and start the compute service by running the PowerShell in the Administrator mode:

```
PS C:\> Get-Service vmcompute | Stop-Service
PS C:\> Get-Service vmcompute | Start-Service
```

Install Extension in Visual Studio with WSL

https://ubuntu.com/tutorials/working-with-visual-studio-code-on-ubuntu-on-wsl2#3-install-visual-studio-code-on-windows

Follow these steps to install docker in WSL

https://medium.com/geekculture/run-docker-in-windows-10-11-wsl-without-docker-desktop-a2a7eb90556d

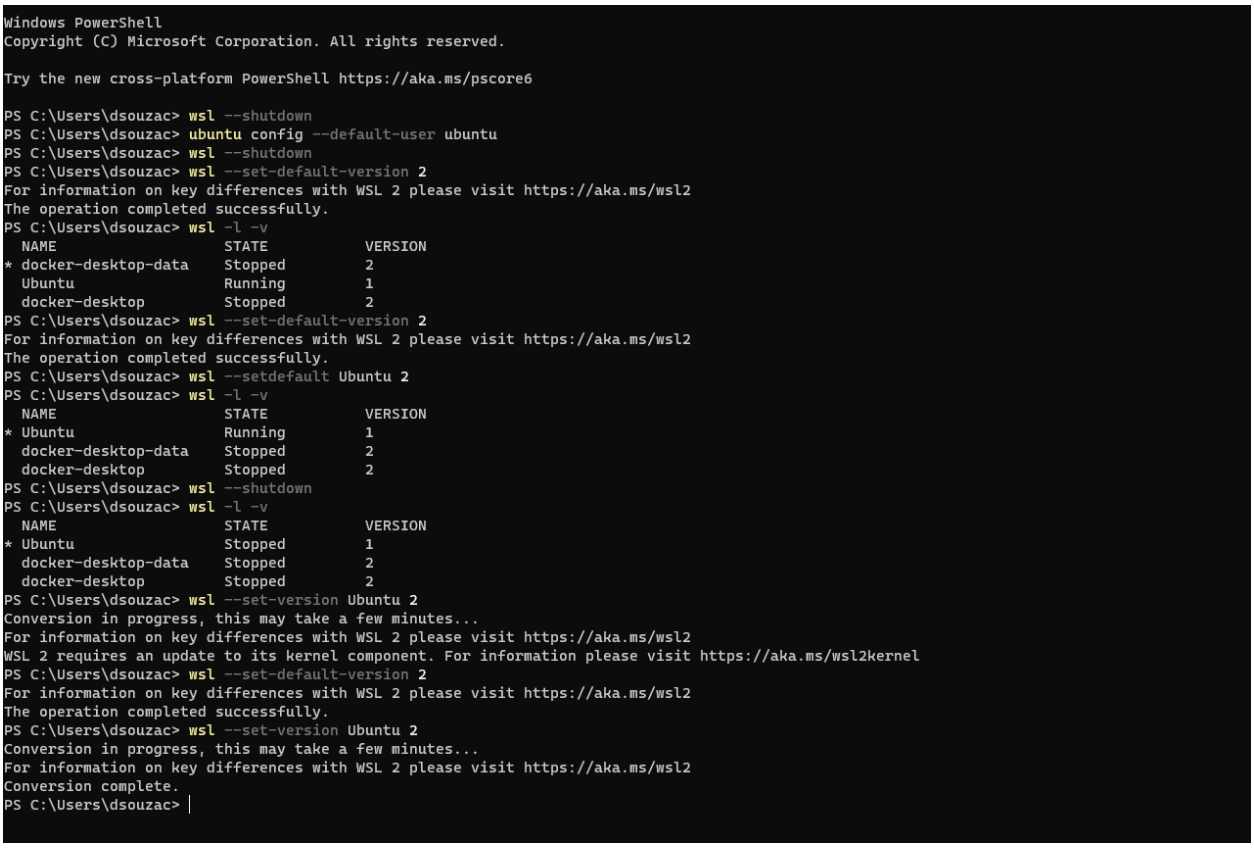
Check version of WSL. Should be Version 2

wsl --list --verbose

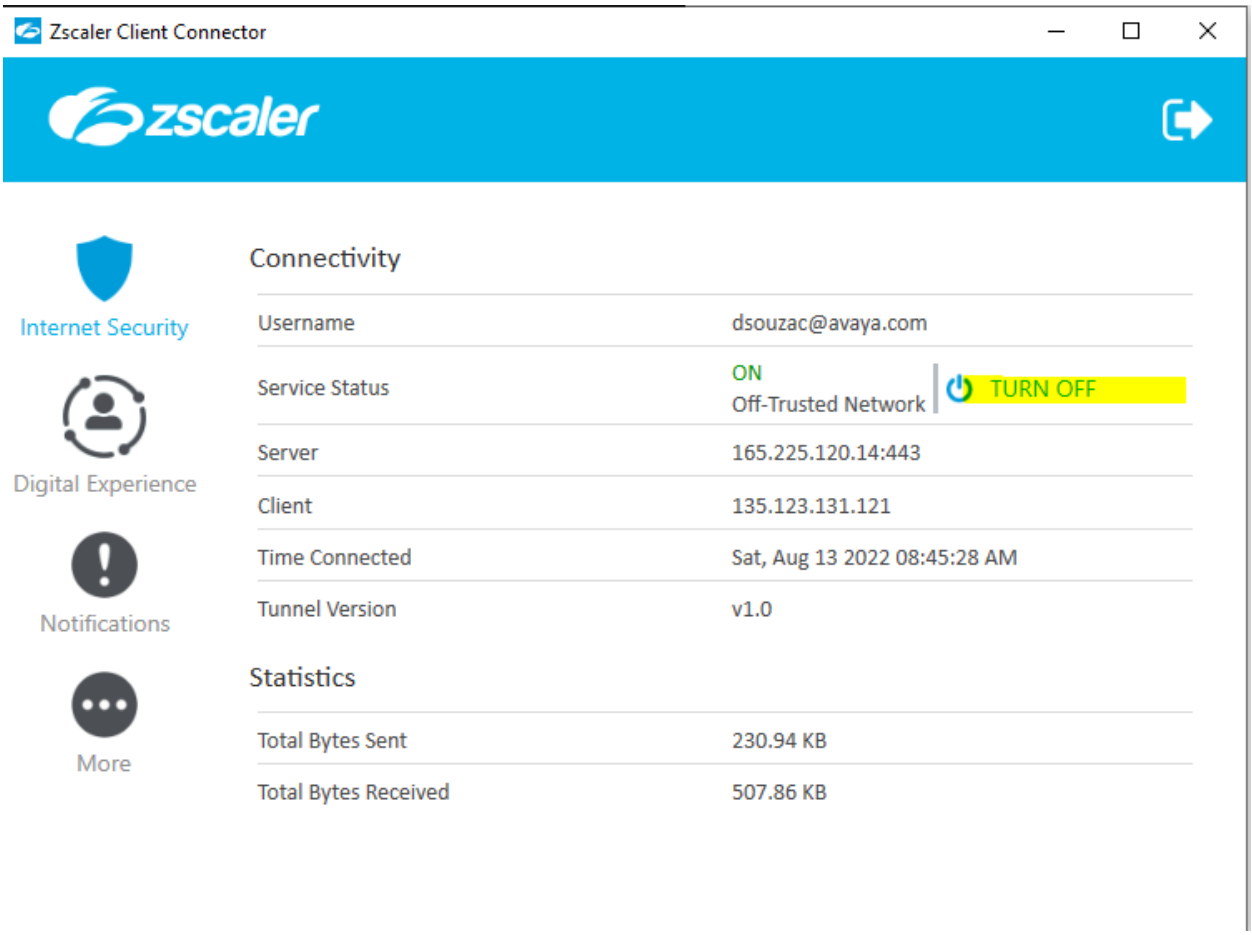
```
PS C:\Users\dsouzac> wsl --list --verbose
  NAME                   STATE           VERSION
* Ubuntu                 Stopped         2
  docker-desktop-data    Stopped         2
  docker-desktop         Stopped         2
PS C:\Users\dsouzac>
```

If not upgrade using the command

wsl --set-version Ubuntu 2



Zscaler will need to be disabled to perform these steps. Click on turn off



Check the version of Ubuntu installed