

Follow the link :

[itsfoss.com/install-bash-on-windows](https://itsfoss.com/install-bash-on-windows) and then install gcc with "sudo apt install gcc". Or "sudo apt-get install gcc"

Source: <https://docs.microsoft.com/en-us/windows/wsl/faq#:~:text=The%20Windows%20Subsystem%20for%20Linux,about%20page%20for%20more%20details>.

## **What is Windows Subsystem for Linux (WSL)?**

The Windows Subsystem for Linux (WSL) is a feature of the Windows operating system that enables you to run a Linux file system, along with Linux command-line tools and GUI apps, directly on Windows, alongside your traditional Windows desktop and apps.

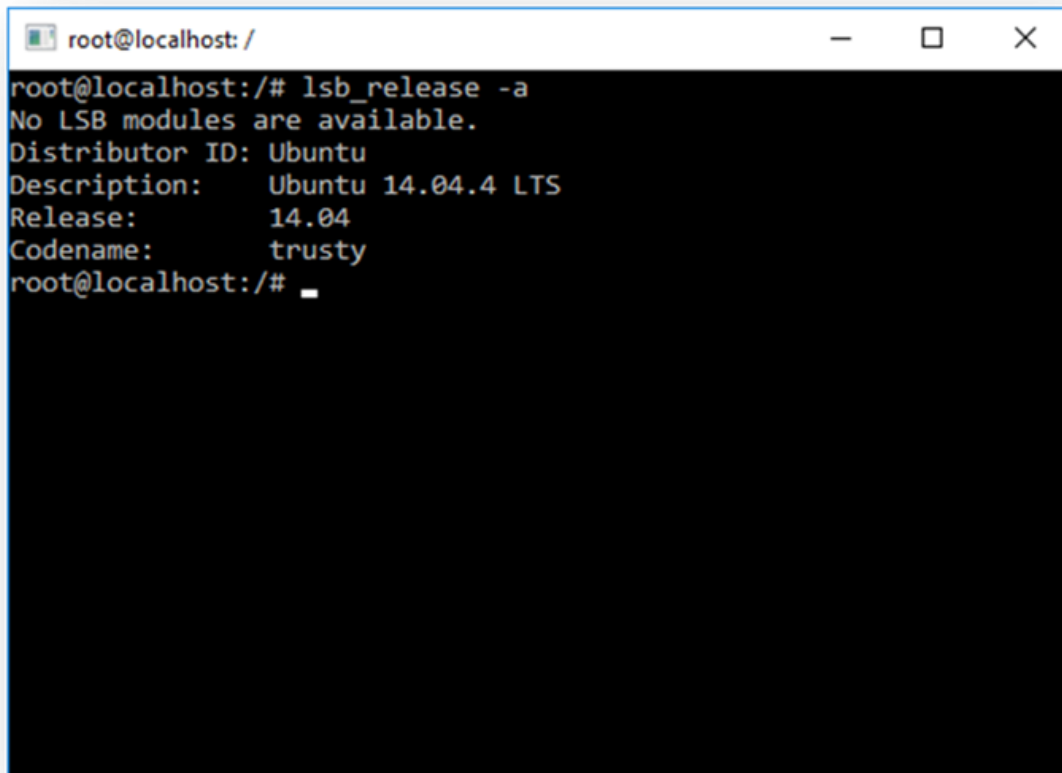
See the [about page](#) for more details.

## **Who is WSL for?**

This is primarily a tool for developers, especially web developers, those working on open source projects, or deploying to Linux server environments. WSL is for anyone who likes using Bash, common Linux tools (*sed*, *awk*, etc.) and Linux-first frameworks (Ruby, Python, etc.) but also enjoys using Windows productivity tools

## **What can I do with WSL?**

WSL enables you to run Linux in a Bash shell with your choice of distribution (Ubuntu, Debian, OpenSUSE, Kali, Alpine, etc). Using Bash, you can run command-line Linux tools and apps. For example, type `lsb_release -a` and hit enter; you'll see details of the Linux distro currently running:

A terminal window titled 'root@localhost: /' with standard window controls (minimize, maximize, close). The terminal shows the command 'lsb\_release -a' being executed. The output is as follows:

```
root@localhost:/# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 14.04.4 LTS
Release:        14.04
Codename:       trusty
root@localhost:/# _
```

You can also access your local machine's file system from within the Linux Bash shell – you'll find your local drives mounted under the `/mnt` folder. For example, your `c:` drive is mounted under `/mnt/c:`

```

root@localhost: /
root@localhost:/# ll /mnt/c
ls: cannot access /mnt/c/Documents and Settings: Input/output error
ls: cannot access /mnt/c/pagefile.sys: Permission denied
ls: cannot access /mnt/c/swapfile.sys: Permission denied
total 452
drwxrwxrwx 2 root root      0 Mar 15 22:26 $Recycle.Bin/
drwxrwxrwx 2 root root      0 Mar 15 23:20 ./
drwxrwxr-x 2 root root 1000  0 Jan 1 1970 ../
-rwxrwxrwx 1 root root      1 Mar 15 16:13 BOOTNXT*
d????????? ? ?      ?      ?      ? Documents and Settings/
drwxrwxrwx 2 root root      0 Mar 15 16:22 PerfLogs/
drwxrwxrwx 2 root root      0 Mar 15 22:18 Program Files/
drwxrwxrwx 2 root root      0 Mar 15 23:18 Program Files (x86)/
drwxrwxrwx 2 root root      0 Mar 15 22:27 ProgramData/
drwxrwxrwx 2 root root      0 Mar 15 22:19 Recovery/
drwxrwxrwx 2 root root      0 Mar 15 22:42 System Volume Information/drw
xrwxrwx 2 root root      0 Mar 15 22:38 Users/
drwxrwxrwx 2 root root      0 Mar 15 22:37 windows/
-rwxrwxrwx 1 root root 403762 Mar 15 16:13 bootmgr*
drwxrwxrwx 2 root root      0 Mar 15 22:37 conedg/
-????????? ? ?      ?      ?      ? pagefile.sys
-????????? ? ?      ?      ?      ? swapfile.sys
root@localhost:/#

```

## How does this work?

Check out this article on the Windows Command Line blog: [A Deep Dive Into How WSL Allows Windows to Access Linux Files](#) which goes into detail about the underlying technology.

## Why would I use WSL rather than Linux in a VM?

WSL requires fewer resources (CPU, memory, and storage) than a full virtual machine. WSL also allows you to run Linux command-line tools and apps alongside your Windows command-line, desktop and store apps, and to access your Windows files from within Linux. This enables you to use Windows apps and Linux command-line tools on the same set of files if you wish.

## Why would I use, for example, Ruby on Linux instead of on Windows?

Some cross-platform tools were built assuming that the environment in which they run behaves like Linux. For example, some tools assume that they are able to access very long file paths or that specific files/folders exist. This often causes problems on Windows which often behaves differently from Linux.

Many languages like Ruby and Node.js are often ported to, and run great, on Windows. However, not all of the Ruby Gem or node/NPM library owners port their libraries to support Windows, and many have Linux-specific dependencies. This can often result in systems built using such tools and libraries suffering from build and sometimes runtime errors or unwanted behaviors on Windows.

These are just some of issues that caused many people to ask Microsoft to improve Windows' command-line tools and what drove us to partner with Canonical to enable native Bash and Linux command-line tools to run on Windows.

## What does this mean for PowerShell?

While working with OSS projects, there are numerous scenarios where it's immensely useful to drop into Bash from a PowerShell prompt. Bash support is complementary and strengthens the value of the command-line on Windows, allowing PowerShell and the PowerShell community to leverage other popular technologies.

Read more on the PowerShell team blog -- [Bash for Windows: Why it's awesome and what it means for PowerShell](#)

## What processors does WSL support?

WSL supports x64 and ARM CPUs.

## How do I access my C: drive?

Mount points for hard drives on the local machine are automatically created and provide easy access to the Windows file system.

**/mnt/<drive letter>/**

Example usage would be `cd /mnt/c` to access `c:\`

## How do I set up Git Credential Manager? (How do I use my Windows Git permissions in WSL?)

See the tutorial [Get started using Git on Windows Subsystem for Linux](#), which features a section on setting up Git Credential Manager and storing authentication tokens in Windows Credential Manager.

## How do I use a Windows file with a Linux app?

One of the benefits of WSL is being able to access your files via both Windows and Linux apps or tools.

WSL mounts your machine's fixed drives under the `/mnt/<drive>` folder in your Linux distros. For example, your `c:` drive is mounted under `/mnt/c/`

Using your mounted drives, you can edit code in, for example, `C:\dev\myproj\` using [Visual Studio](#) / or [VS Code](#), and build/test that code in Linux by accessing the same files via `/mnt/c/dev/myproj`.

Learn more in [Working across Windows and Linux file systems](#) article.

## Are files in the Linux drive different from the mounted Windows drive?

1. Files under the Linux root (i.e. `/`) are controlled by WSL which aligns with Linux behavior, including but not limited to:
  - Files which contain invalid Windows filename characters
  - Symlinks created for non-admin users
  - Changing file attributes through `chmod` and `chown`
  - File/folder case sensitivity
2. Files in mounted drives are controlled by Windows and have the following behaviors:
  - Support case sensitivity
  - All permissions are set to best reflect the Windows permissions

## How do I uninstall a WSL Distribution?

To remove a distribution from WSL and *delete all of the data associated with that Linux distribution*, run `wsl --unregister <distroName>` where `<distroName>` is the name of your Linux distro, which can be seen from the list in the `wsl -l` command.

Additionally, you can then uninstall the Linux distro app on your machine just like any other store application.

To learn more about wsl commands, see the article, [Basic commands for WSL](#).

## How do I run an OpenSSH server?

OpenSSH ships with Windows as an optional feature. See the [Install OpenSSH](#) doc. Administrator privileges in Windows are required to run OpenSSH in WSL. To run an OpenSSH server, run your WSL distribution (ie Ubuntu) or Windows Terminal as an administrator. There are several resources out there covering SSH scenarios with WSL. Check out Scott Hanselman's blog articles: [How to SSH into a Windows 10 Machine from Linux OR Windows OR anywhere](#), [How to SSH into WSL2 on Windows 10 from an external machine](#), [THE EASY WAY how to SSH into Bash and WSL2 on Windows 10 from an external machine](#), and [How to use Windows 10's built-in OpenSSH to automatically SSH into a remote Linux machine](#).

## How do I change the display language of WSL?

WSL install will try to automatically change the Ubuntu locale to match the locale of your Windows install. If you do not want this behavior you can run this command to change the Ubuntu locale after install completes. You will have to relaunch your WSL distribution for this change to take effect.

The below example changes to locale to en-US:

```
BashCopy
sudo update-locale LANG=en_US.UTF8
```

## Why do I not have internet access from WSL?

Some users have reported issues with specific firewall applications blocking internet access in WSL. The firewalls reported are:

1. Kaspersky
2. AVG
3. Avast
4. Symantec Endpoint Protection

## 5. F-Secure

In some cases turning off the firewall allows for access. In some cases simply having the firewall installed looks to block access.

### **How do I access a port from WSL in Windows?**

WSL shares the IP address of Windows, as it is running on Windows. As such you can access any ports on localhost e.g. if you had web content on port 1234 you could `https://localhost:1234` into your Windows browser. For more information, see [Accessing network applications](#).

### **How can I back up my WSL distributions, or move them from one drive to another?**

The best way to backup or move your distributions is via the [export/import commands](#) available in Windows Version 1809 and later. You can export your entire distribution to a tarball using the `wsl --export` command. You can then import this distribution back into WSL using the `wsl --import` command, which can name a new drive location for the import, allowing you to backup and save states of (or move) your WSL distributions.

Please note that traditional backup services that backup files in your AppData folders (like Windows Backup) will not corrupt your Linux files.

## **WSL 2**

### **Does WSL 2 use Hyper-V? Will it be available on Windows 10 Home?**

WSL 2 is available on all Desktop SKUs where WSL is available, including Windows 10 Home.

The newest version of WSL uses Hyper-V architecture to enable its virtualization. This architecture will be available in the 'Virtual Machine Platform' optional component. This optional component will be available on all SKUs. You can expect to see more details about this experience soon as we get closer to the WSL 2 release.

### **What will happen to WSL 1? Will it be abandoned?**

We currently have no plans to deprecate WSL 1. You can run WSL 1 and WSL 2 distros side by side, and can upgrade and downgrade any distro at any time. Adding WSL 2 as a new architecture presents a better platform for the WSL team to deliver features that make WSL an amazing way to run a Linux environment in Windows.

## **Will I be able to run WSL 2 and other 3rd party virtualization tools such as VMware, or VirtualBox?**

Some 3rd party applications cannot work when Hyper-V is in use, which means they will not be able to run when WSL 2 is enabled, such as VMware and VirtualBox. However, recently both VirtualBox and VMware have released versions that support Hyper-V and WSL2. Learn more about [VirtualBox's changes here](#) and [VMware's changes here](#). For troubleshooting issues, take a look at the [VirtualBox issue discussions in the WSL repo on GitHub](#).

We are consistently working on solutions to support third-party integration of Hyper-V. For example, we expose a set of APIs called [Hypervisor Platform](#) that third-party virtualization providers can use to make their software compatible with Hyper-V. This lets applications use the Hyper-V architecture for their emulation such as [the Google Android Emulator](#), and VirtualBox 6 and above which are both now compatible with Hyper-V.

See the WSL issues repo for more background and discussion on [WSL 2 issues with VirtualBox 6.1](#).

\*If you're looking for a Windows 10 virtual machine, VMWare, Hyper-V, VirtualBox, and Parallels VM downloads are [available on the Windows Dev Center](#).

## **Can I access the GPU in WSL 2? Are there plans to increase hardware support?**

We have released support for accessing the GPU inside of WSL 2 distributions! This means you can now use WSL for machine learning, artificial intelligence, and data science scenarios more easily when big data sets are involved. Check out the [get started with GPU support](#) tutorial. As of right now WSL 2 does not include serial support, or USB device support. We are investigating the best way to add these features. However, USB support is now available through the USBIPD-WIN project. See [Connect USB devices](#) for steps to set up USB device support.

## **Will WSL 2 be able to use networking applications?**

Yes, in general networking applications will be faster and work better since we have full system call compatibility. However, the new architecture uses virtualized networking components. This means that in initial preview builds WSL 2 will behave more similarly to a virtual machine, e.g: WSL 2 will have a different IP address than the host machine. We are committed to making WSL 2 feel the same as WSL 1, and that includes improving our networking story.

## **Can I run WSL 2 in a virtual machine?**



Yes! You need to make sure that the virtual machine has nested virtualization enabled. This can be enabled in your parent Hyper-V host by running the following command in a PowerShell window with Administrator privileges:

```
Set-VMProcessor -VMName <VMName> -ExposeVirtualizationExtensions $true
```

Make sure to replace '<VMName>' with the name of your virtual machine.

## Can I use wsl.conf in WSL 2?

WSL 2 supports the same wsl.conf file that WSL 1 uses. This means that any configuration options that you had set in a WSL 1 distro, such as automounting Windows drives, enabling or disabling interop, changing the directory where Windows drives will be mounted, etc. will all work inside of WSL 2. You can learn more about the configuration options in WSL in the [Distribution Management](#) page. Learn more about support for mounting drives, disks, devices, or virtual hard disks (VHDs) in the [Mount a Linux disk in WSL 2](#) article.

## Where can I provide feedback?

The [WSL product repo issues](#) enables you to:

- **Search existing issues** to see if there are any associated with a problem that you are having. Note that in the search bar, you can remove "is:open" to include issues that have already been resolved in your search. Please consider commenting or giving a thumbs up to any open issues that you would like to express your interest in moving forward as a priority.
- **File a new issue.** If you have found a problem with WSL and there does not appear to be an existing issue, you can select the green *New issue* button and then choose *WSL - Bug Report*. You will need to include a title for the issue, your Windows build number (run `cmd.exe /c ver` to see your current build #), whether you're running WSL 1 or 2, your current Linux Kernel version # (run `wsl.exe --status` or `cat /proc/version`), the version # of your distribution (run `lsb_release -r`), any other software versions involved, the repro steps, expected behavior, actual behavior, and diagnostic logs if available and appropriate. For more info, see [contributing to WSL](#).
- **File a feature request** by selecting the green *New issue* button and then select *Feature request*. You will need to address a few questions describing your request.

You can also:

- **File a documentation issue** using the [WSL docs repo](#). To contribute to the WSL docs, see the [Microsoft Docs contributor guide](#).
- **File a Windows Terminal issue** using the [Windows Terminal product repo](#) if your problem is related more to the Windows Terminal, Windows Console, or the command-line UI.

Without options, `ls` displays files names only. The most common options to display additional information are:

- `-l` long format, displaying [Unix file types](#), permissions, number of [hard links](#), owner, group, size, last-modified date and filename
- `-F` append a "/" to directory names and a "\*" to executable files.
- `-g` display group but not owner
- `-o` display owner but not group (when combined with `-g` both group and owner are suppressed)
- `-d` shows information about a symbolic link or directory, rather than about the link's target or listing the contents of a directory.
- `-h` output sizes in human readable format. (e.g., 1K, 234M, 2G, etc.) This option is not part of the POSIX standard, although implemented in several systems, e.g., GNU coreutils in 1997,<sup>[10]</sup> FreeBSD 4.5 in 2002,<sup>[11]</sup> and Solaris 9 in 2002.<sup>[12]</sup>

Additional options controlling how files are displayed include:

- `-f` do not sort. Useful for directories containing large numbers of files.
- `-t` sort the list of files by modification time. (default is alphabetically)
- `-1` (one) force output to be one entry per line.
- `-R` recursively list files in subdirectories and their subdirectories ...
- `--full-time` to show times with seconds and milliseconds instead of down to the minute.