

# GATS Companion Installing WSL

Author: Garth Santor

Editors: Trinh Hân

Version/Copyright: 1.0.0 (2023-09-24)

## Overview

How to install Windows Subsystem for Linux (WSL) on Windows 11.

## Installation

[Install WSL | Microsoft Learn](#)

Open a PowerShell console in Windows 11, then enter the command: `wsl --install`

WSL will install ubuntu Linux by default.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\gsantor> wsl --install
The requested operation requires elevation.
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Installing: Ubuntu
Ubuntu has been installed.
The requested operation is successful. Changes will not be effective until the system is rebooted.
PS C:\Users\gsantor>
```

Reboot your computer.

You should see:

```
Ubuntu
Ubuntu is already installed.
Launching Ubuntu...
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:
```

## Adding a user account

Enter an account name and password for your new Linux account. Note that you will not see anything reflected in the console when you type the password (Linux doesn't even want to give up the number of characters in your password).

```
gbs@DESKTOP-9Q9E7PB: ~ x + v
Ubuntu is already installed.
Launching Ubuntu..
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: gbs
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/gbs/.hushlogin file.
gbs@DESKTOP-9Q9E7PB:~$
```

## Verify the installation

We can verify the installation using the command `lsb_release`. LSB stands for Linux Standard Base, a project to standardize the system structure of multiple Linux distributions, to support open standards for Linux binary applications.

At the console prompt, type: `lsb_release --all`

```
gbs@DESKTOP-9Q9E7PB: ~ x + v
Description:      Ubuntu 22.04.2 LTS
gbs@DESKTOP-9Q9E7PB:~$ lsb_release --all
No LSB modules are available.
Distributor ID:  Ubuntu
Description:     Ubuntu 22.04.2 LTS
Release:         22.04
Codename:        jammy
gbs@DESKTOP-9Q9E7PB:~$
```

## Updating your Installation

At the console prompt, type: `sudo apt update`

You will see something like this:

```
gbs@DESKTOP-9Q9E7PB: ~$ sudo apt update
[sudo] password for gbs:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [802 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [169 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.3 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [882 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [142 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [536 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [785 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [143 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.7 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Get:16 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1012 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [227 kB]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [898 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [145 kB]
Get:28 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 B]
Get:29 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [984 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [215 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [21.8 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:34 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [476 B]
Get:35 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:38 http://archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:39 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.4 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [640 B]
Get:42 http://archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 27.4 MB in 5s (5962 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
102 packages can be upgraded. Run 'apt list --upgradable' to see them.
gbs@DESKTOP-9Q9E7PB: ~$
```

## Leaving WSL

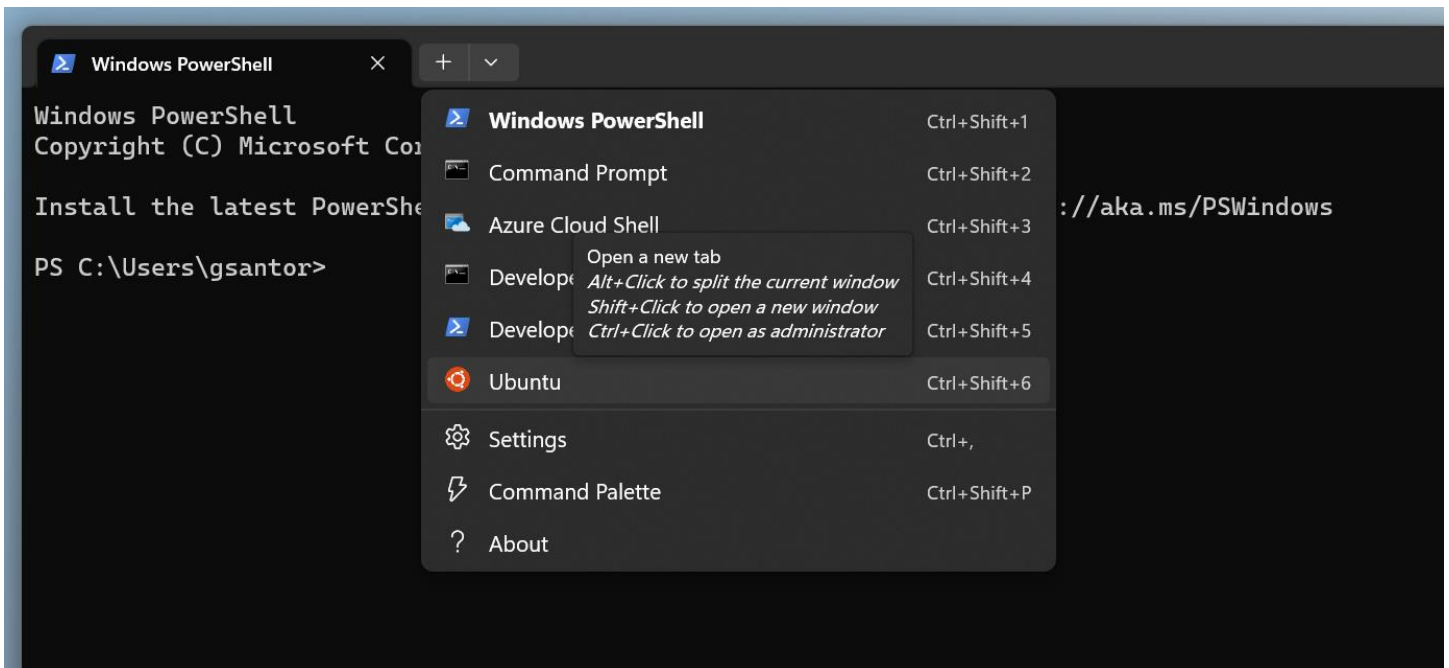
At the console prompt, type: `exit`↵

## Using WSL

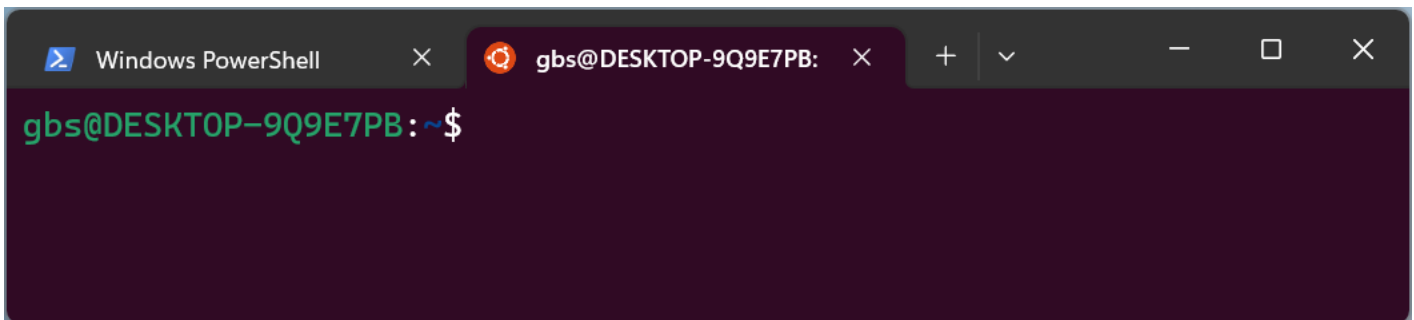
The most common way to use WSL is through windows terminal.

Launch windows terminal.

With WSL installed you'll see an option for Ubuntu on the drop menu.



Select 'Ubuntu' and you will be taken to your WSL account.

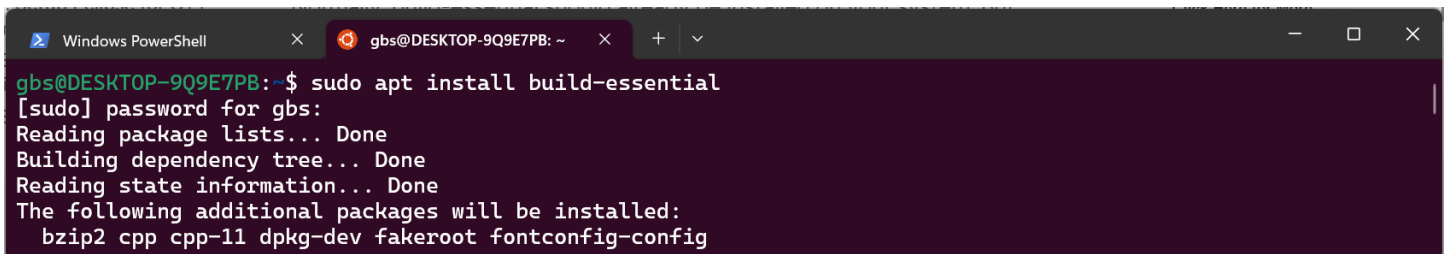


## Installing development tools

The easiest way to get a development environment for Linux is to install the build-essential package.

To do so, type: `sudo apt install build-essential` at the Linux command prompt.

The beginning of the installation will look like this:



You'll be prompted for additional installation files, respond with 'Y' to add them.

It should finish up with:

```
Windows PowerShell x gbs@DESKTOP-9Q9E7PB: ~ x + v
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
/sbin/ldconfig.real: /usr/lib/wsl/lib/libcuda.so.1 is not a symbolic link

gbs@DESKTOP-9Q9E7PB: ~$
```

## Testing a .C program

Enter the following program with your favour editor (I used vim).

```
hello.c
#include <stdio.h>

int main() {
    printf("hello, C-world\n");
    return 0;
}
```

At the prompt, type: `cc hello.c -o helloc`

Execute the program by typing at the prompt: `./helloc`

```
Windows PowerShell x gbs@DESKTOP-9Q9E7PB: ~ x + v
gbs@DESKTOP-9Q9E7PB: ~$ cc hello.c -o helloc
gbs@DESKTOP-9Q9E7PB: ~$ ls
hello.c helloc
gbs@DESKTOP-9Q9E7PB: ~$ ./helloc
hello, C-world
gbs@DESKTOP-9Q9E7PB: ~$
```

## Testing a .C++ program

Enter the following program with your favour editor (I used vim).

```
hello.cpp
#include <iostream>

int main() {
    std::cout << "Hello, C++-world\n";
}
```

At the prompt, type: `g++ hello.cpp -o helloc++`

Execute the program by typing at the prompt: `./helloc++`

```
Windows PowerShell x gbs@DESKTOP-9Q9E7PB: ~ x + v
gbs@DESKTOP-9Q9E7PB:~$ g++ hello.cpp -o helloc++
gbs@DESKTOP-9Q9E7PB:~$ ls
hello.c hello.cpp helloc helloc++
gbs@DESKTOP-9Q9E7PB:~$ ./helloc++
Hello, C++-world
gbs@DESKTOP-9Q9E7PB:~$
```

## References

### Installation

[Running Visual Studio Code on Linux](#)

## Document History

Version	Date	Activity
1.0.0	2023-09-24	Document created.