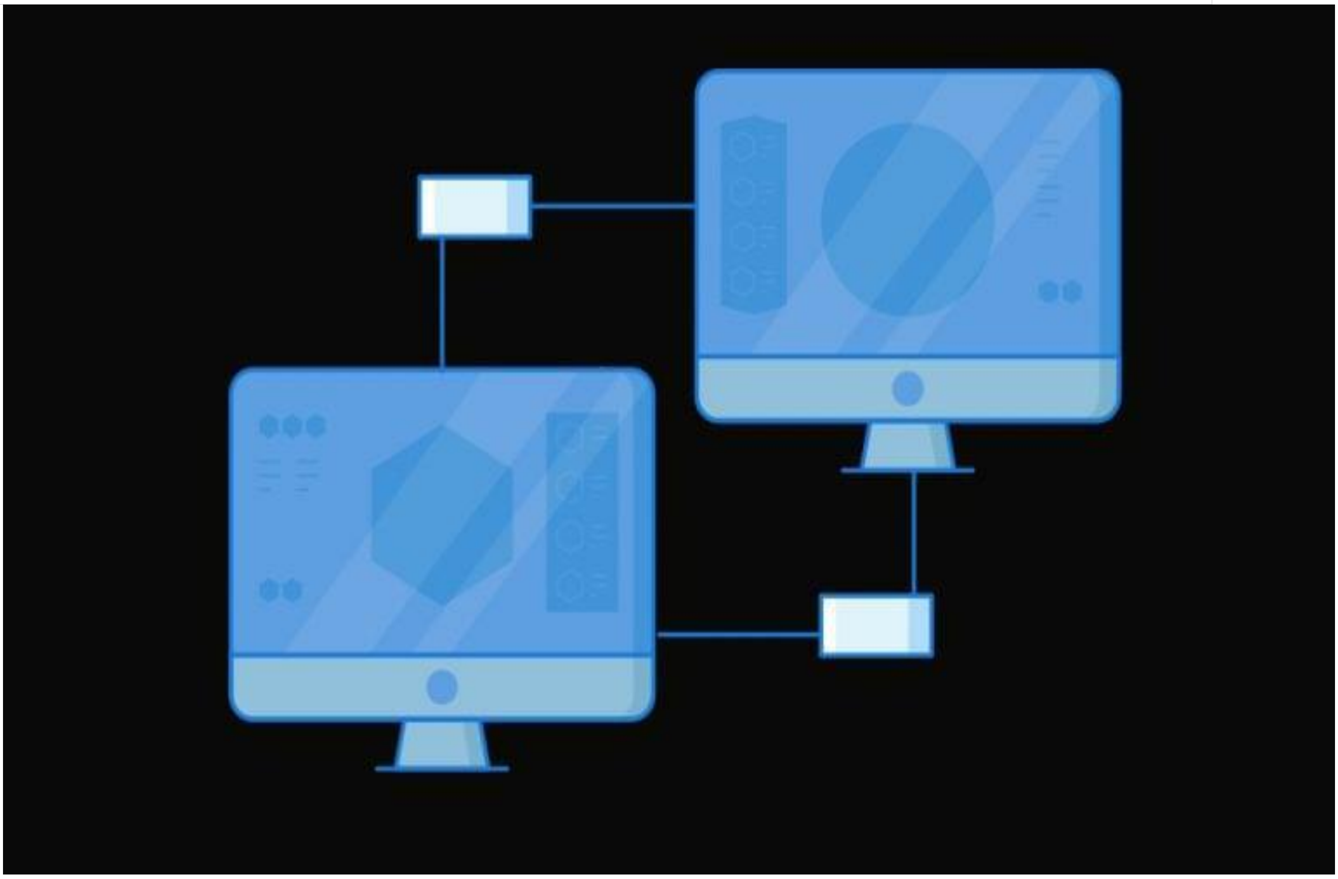
Windows Subsystem for Linux 2 (WSL2): A Complete Guide for Windows 10 & 11



This WSL2 Windows 10 & 11 tutorial demonstrates how to install, manage, and use a full Linux environment on your Windows 10 or Windows 11 PC with WSL2. You'll run a Linux kernel using the Windows subsystem for Linux and execute native applications while seamlessly sharing files and services with Windows code editors and browsers.

Key Takeaways

- WSL2 allows seamless integration of a Linux environment on Windows 10 and 11, enabling the use of Linux applications and files alongside Windows resources.

- Installation and setup of WSL2 are more streamlined on Windows 11 compared to Windows 10, which requires enabling virtual machine features manually.
- Linux GUI applications can be run directly in Windows with WSL2, enhancing usability without the need for additional X server software.
- Docker can be efficiently utilized within WSL2, benefiting from faster startup times and better performance without needing dedicated Hyper-V.
- Access to Windows files from Linux and vice versa is facilitated through mounting and symbolic links, allowing easy file sharing and management across systems.
- VS Code integration with WSL2 provides a robust development environment, enabling direct access to Linux files and terminals from within the editor.

Why Use Linux?

Your web host almost certainly uses Linux. It's fast, open source, and runs the majority of web software, including:

- Servers (Apache, NGINX, LiteSpeed, etc.)
- Language runtimes (PHP, Python, Node.js, Ruby, etc.)
- Utility libraries (image manipulation, ZIP creation, emailers, etc.)
- Databases (MySQL, MongoDB, Redis, etc.)
- Other dependencies (Elasticsearch, RabbitMQ, proxies, etc.)

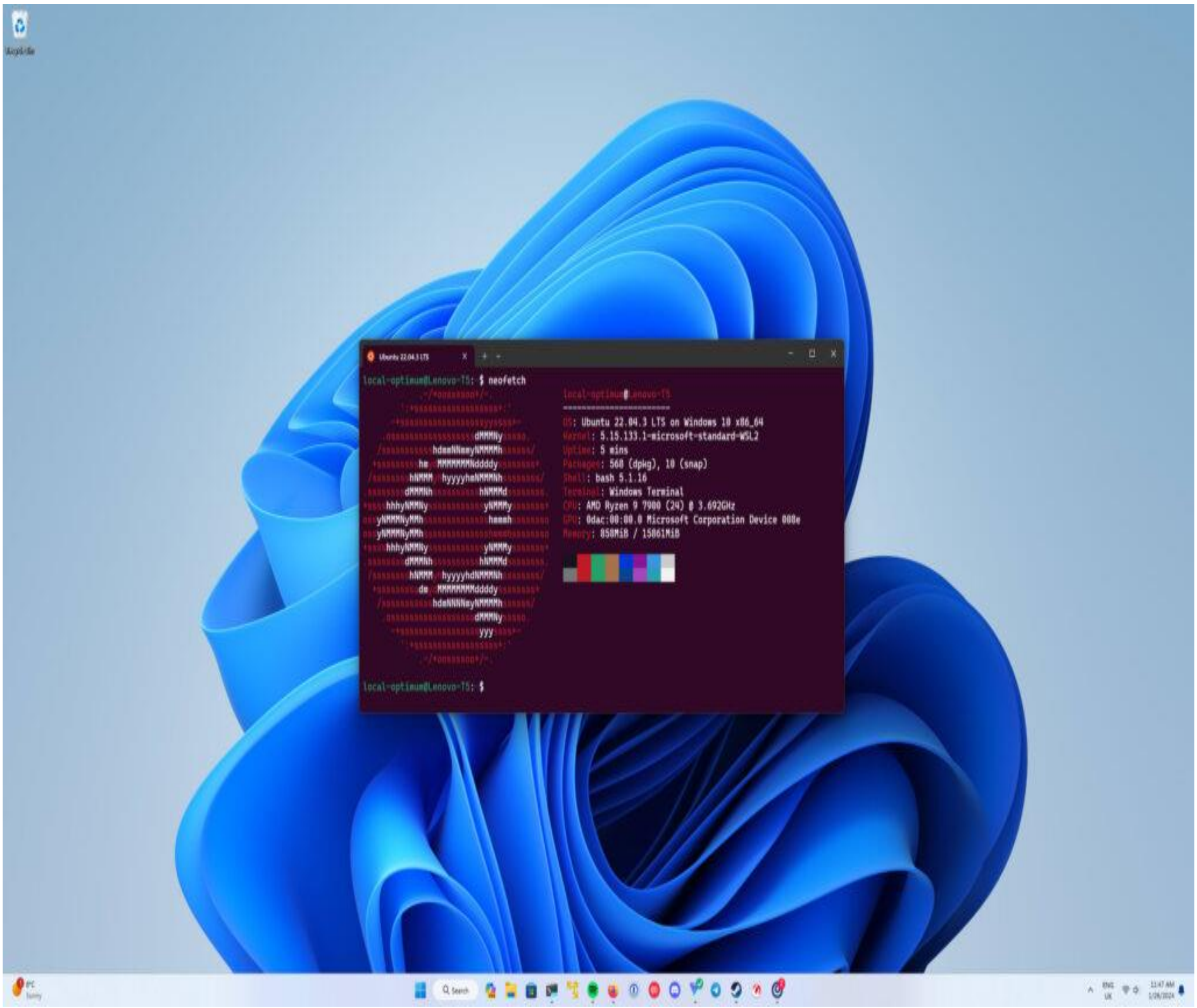
Some of these dependencies may be available on Windows and macOS, but installing and maintaining identical versions is difficult. They're often slower and you may encounter subtle differences which will not become apparent until you deploy.

However, we can't always use Linux as the primary OS on our laptops. For example, if you are using non-Linux software such as Microsoft or Adobe products, you must use Windows or macOS. But there are a few alternatives we can use:

- Use two PCs or dual boot on a single device. Switching between systems can be cumbersome.
- Run a Linux virtual machine (VM) using Hyper-V, VirtualBox, Parallels, WMware, or Vagrant. This requires considerable OS resources and reasonable IT knowledge. Sharing files between systems can be awkward.
- Run web applications in Docker containers. It's another topic to learn and Docker itself runs natively on Linux. Both the Windows and macOS editions

use Linux below the surface and the Windows edition of Docker Desktop requires WSL2.

What is the Windows Subsystem for Linux?



```
local-optimum@Lenovo-T5: $ neofetch
local-optimum@Lenovo-T5
------------------------
OS: Ubuntu 22.04.3 LTS on Windows 10 x86_64
Kernel: 5.15.133.1-microsoft-standard-WSL2
Uptime: 5 mins
Packages: 568 (dpkg), 10 (snap)
Shell: bash 5.1.16
Terminal: Windows Terminal
CPU: AMD Ryzen 9 7900 (24) @ 3.692GHz
GPU: 0dac:00:00.0 Microsoft Corporation Device 008e
Memory: 858MiB / 15861MiB

local-optimum@Lenovo-T5: $
```

The Windows WSL2 lets developers run a Linux environment directly on Windows. It’s a full Linux OS running inside Windows, so you can use the same apps and files seamlessly.

WSL1 was groundbreaking and translated Linux system calls to Windows equivalents. But it was slow, and some features weren’t available. WSL2 uses

Hyper-V virtual machine technology, so Linux compatibility is excellent, and file access is up to 20x faster than before.

Windows 10 and Windows 11 Differences

Microsoft introduced WSL 2 in 2020, so it was fully integrated into the Windows 11 release in October 2021.

Windows 10 requires a little more configuration effort. If you're considering an upgrade to Windows 11, do it before installing WSL 2. Windows 11 is not an improvement in some respects (I still miss the taskbar icon ungrouping and dragging files!), but WSL 2 seems more stable, and I rarely encounter issues.

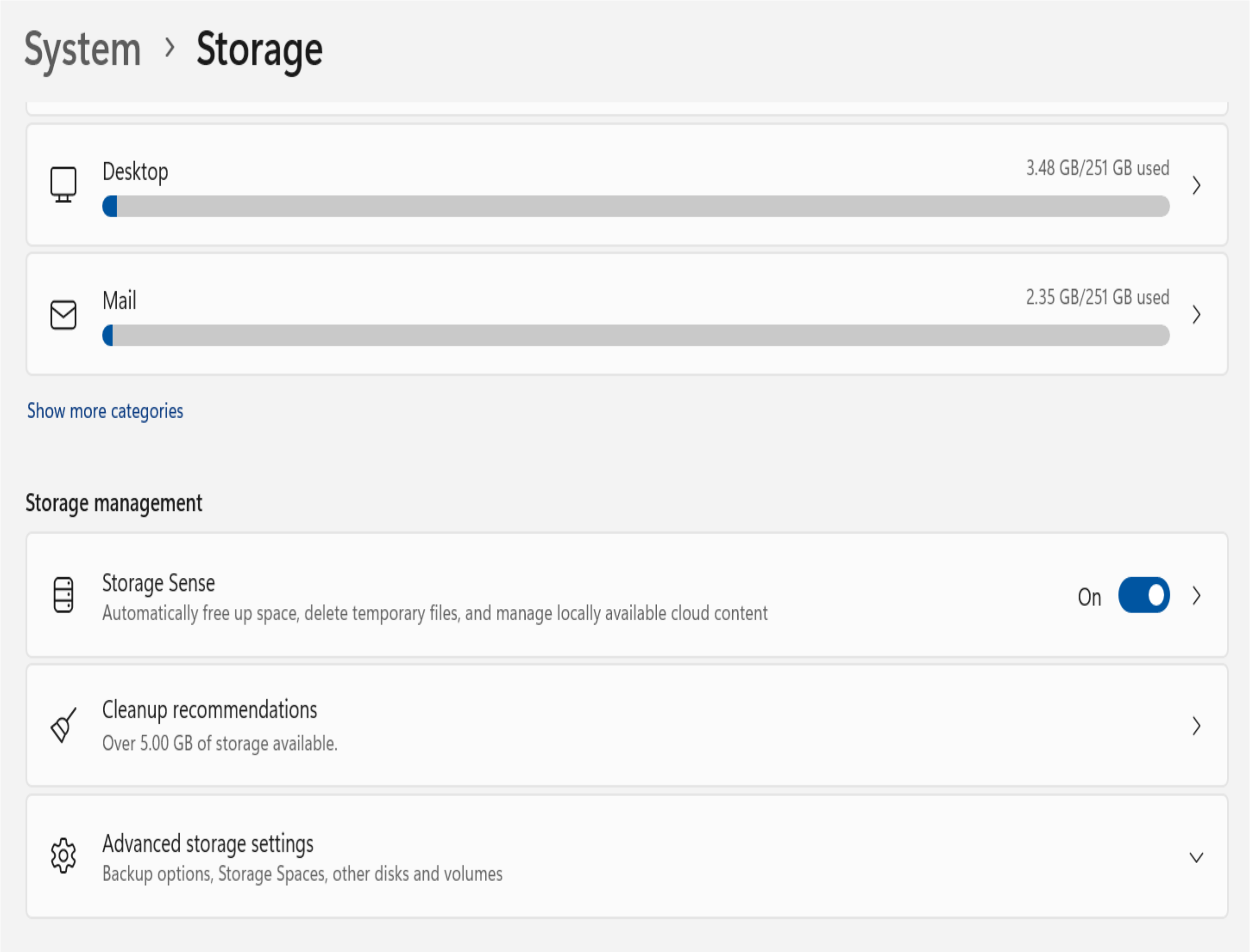
PC Requirements

WSL 2 uses Hyper-V. This requires hardware virtualization support, which you must enable in your BIOS.

*WARNING: **Fiddling with your BIOS settings can trash your PC!** Be careful when making changes. Consult your motherboard manufacturer's help pages or search for online advice about your specific make and model.*

Hardware virtualization is available on most modern devices, but you can check by rebooting and opening the BIOS panels — typically by hitting the DEL, F2, or F10 key as the system starts. Look for **Virtualization Technology**, **VTx**, **Secure Virtual Machine (SVM)**, or similar options. Ensure they are enabled, and save and reboot the machine.

Finally, ensure your C: drive has 1GB of spare disk space — ideally more. You may be able to free up some space in **Settings**, **System**, **Storage**, or using the **Disk clean-up** tool in **Windows Administrative Tools**.



System › Storage

Desktop 3.48 GB/251 GB used

Mail 2.35 GB/251 GB used

Show more categories

Storage management

Storage Sense
Automatically free up space, delete temporary files, and manage locally available cloud content
On

Cleanup recommendations
Over 5.00 GB of storage available.

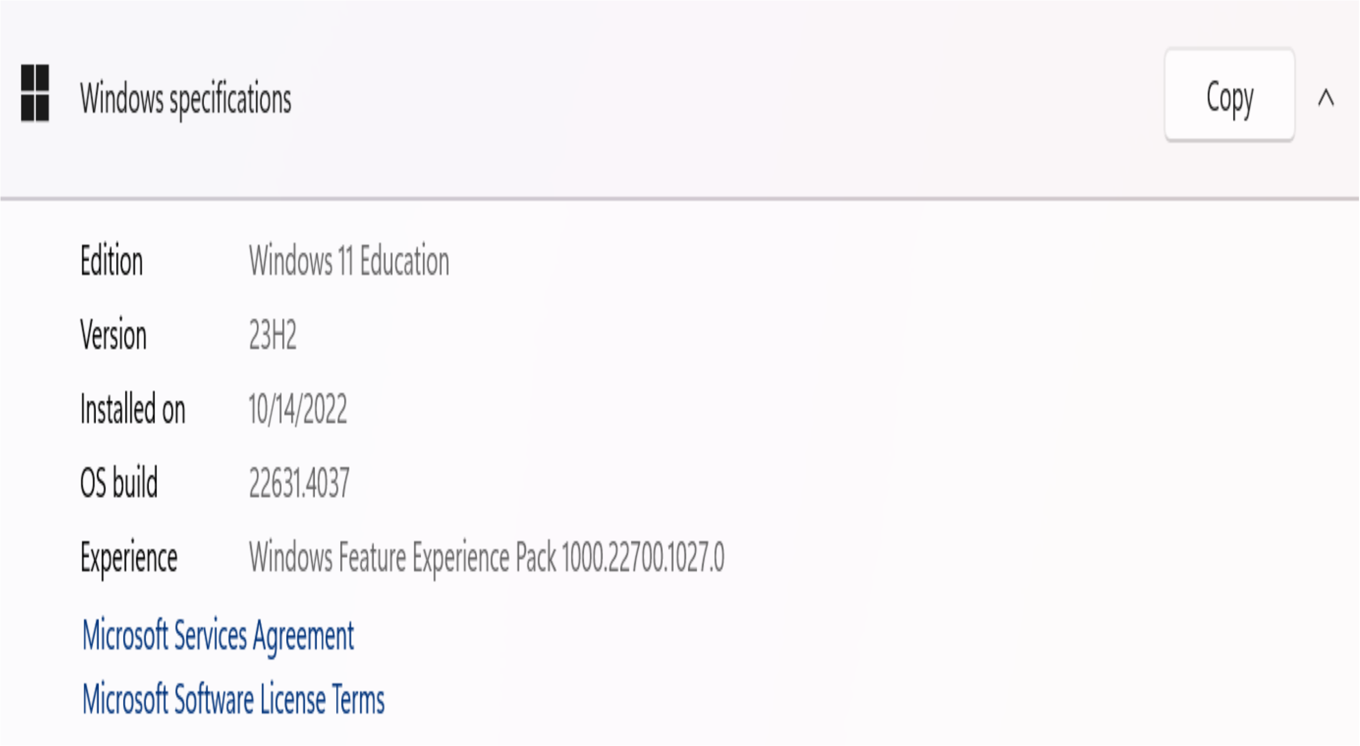
Advanced storage settings
Backup options, Storage Spaces, other disks and volumes

WSL 2 will install your Linux distribution within a hidden folder on your C: drive by default. It's possible to move it elsewhere after installation.

Windows Requirements

WSL2 is available with the Home, Pro, or Server editions of Windows but *not* Windows 10/11 S. S mode only permits apps from the Microsoft Store to "improve" security. You can switch from S mode to Windows Home/Pro using the **Switch to Windows Home** option in **Settings** > **Update & Security** > **Activation**.

Windows 10 requires the May 2020 update —version 2004 or above. To update, choose **Settings** from the Start menu, then select **System**, followed by **About**. The **Windows specifications** are shown at the bottom of the panel.



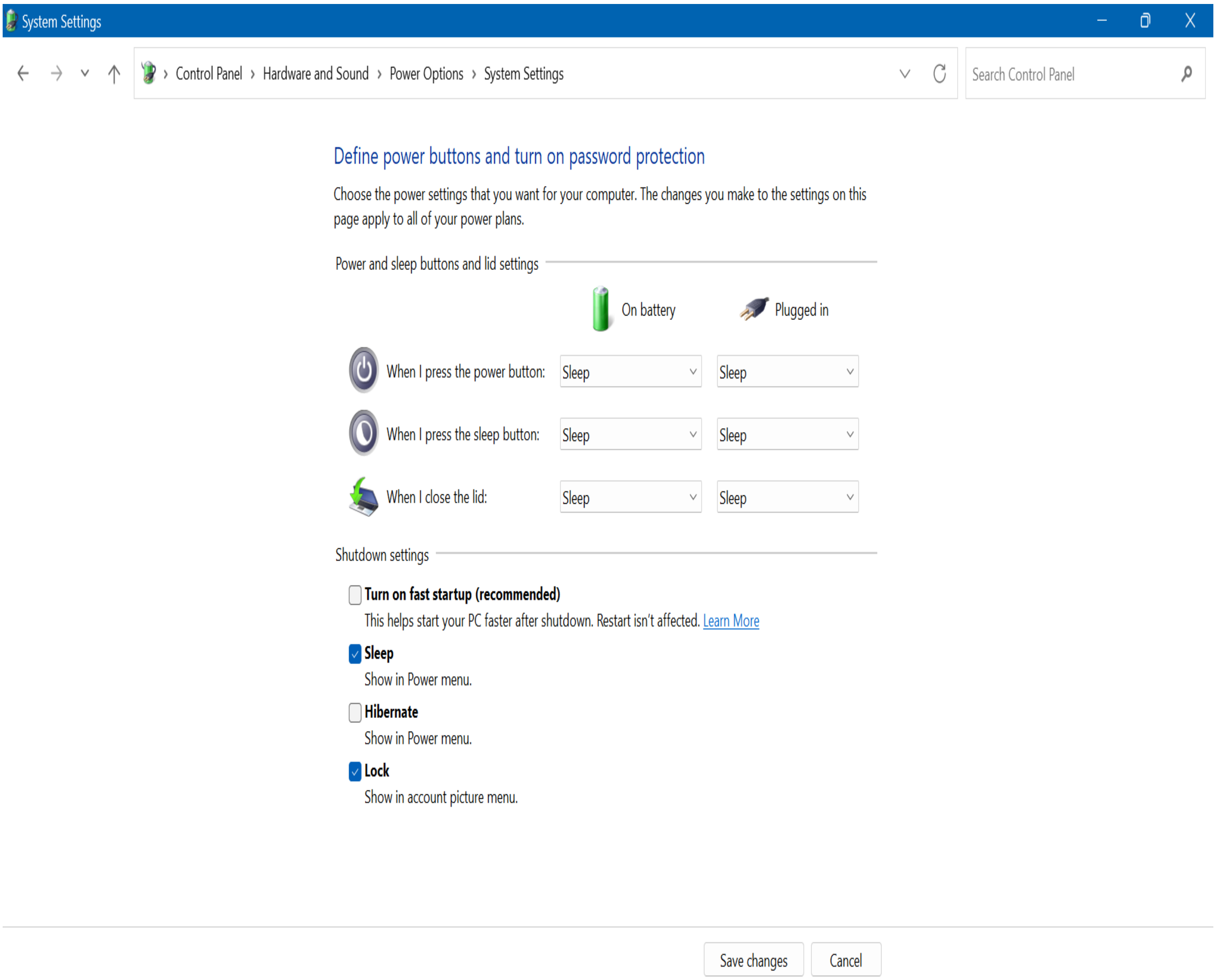
Edition	Windows 11 Education
Version	23H2
Installed on	10/14/2022
OS build	22631.4037
Experience	Windows Feature Experience Pack 1000.22700.1027.0

Microsoft Services Agreement
Microsoft Software License Terms

If it’s not installed, use the **Check for Updates** button in **Settings**, then **Update & Security**. However, some older Windows machines may never be compatible, and Microsoft will unlikely offer a patch.

Disable Fast Start-up (Optional)

Fast start-up saves the Windows session and device drivers to a file so the next boot is faster. This can cause problems for the Linux kernel, which becomes unresponsive on the next boot. If you encounter problems, disable fast start-up from the classic **Control Panel** > **Power Options** > **Choose what the power buttons do**:



System Settings

Control Panel › Hardware and Sound › Power Options › System Settings

Define power buttons and turn on password protection

Choose the power settings that you want for your computer. The changes you make to the settings on this page apply to all of your power plans.

Power and sleep buttons and lid settings

	On battery	Plugged in
When I press the power button:	Sleep	Sleep
When I press the sleep button:	Sleep	Sleep
When I close the lid:	Sleep	Sleep

Shutdown settings

☐ **Turn on fast startup (recommended)**
This helps start your PC faster after shutdown. Restart isn't affected. Learn More

☑ **Sleep**
Show in Power menu.

☐ **Hibernate**
Show in Power menu.

☑ **Lock**
Show in account picture menu.

Alternatively, create a **wsl.bat** file in your Startup folder (C:\Users\<user-name>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup) with the content:

```
wsl --shutdown
```

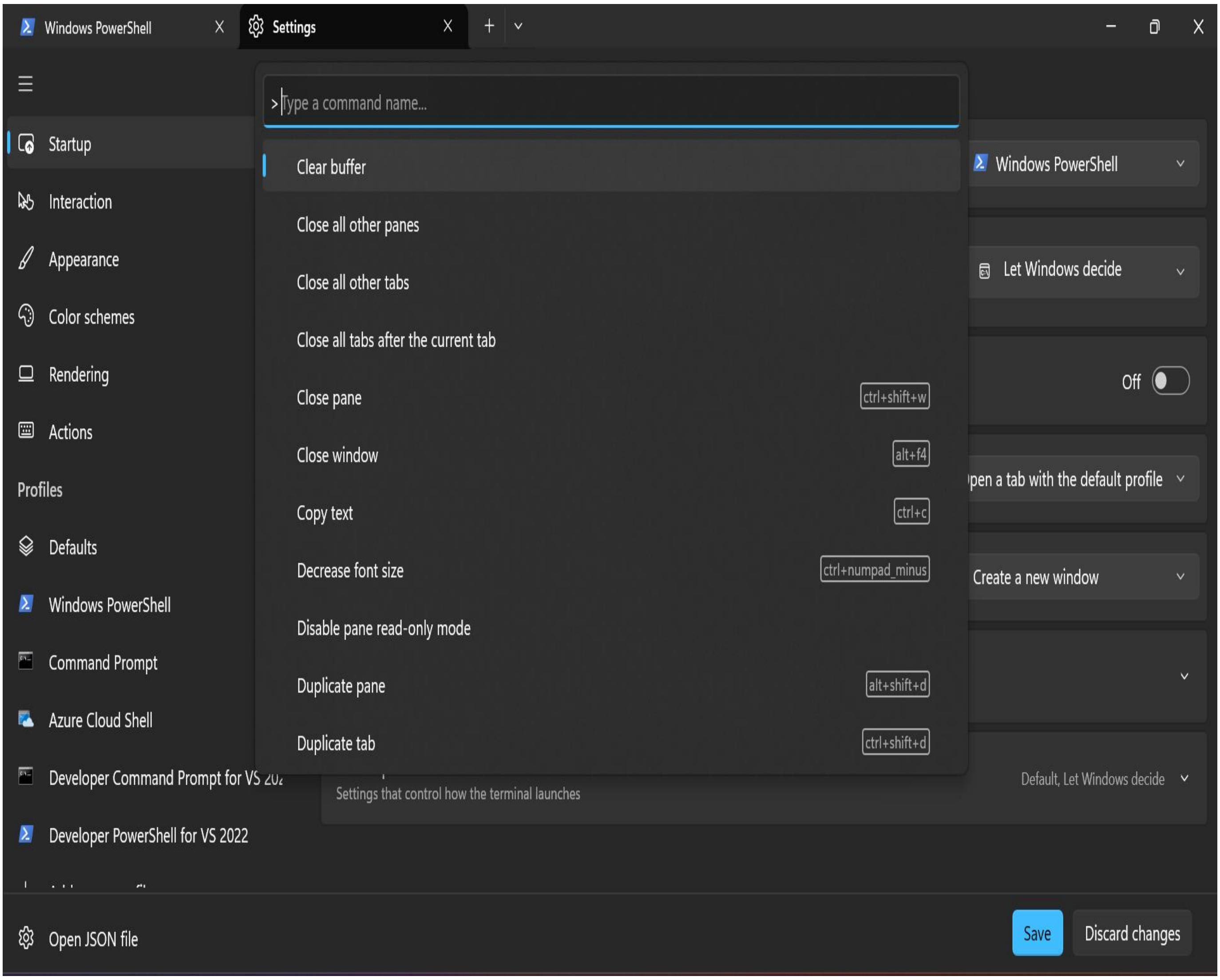
This will shut down WSL after Windows boots but it will restart as soon as you access a Linux distro.

Install Windows Terminal (Optional but Recommended)

Windows Terminal is a great terminal application available from the **Microsoft Store** or its repository at github.com/microsoft/terminal/.

Windows Terminal automatically adds WSL2 Linux distros when they’re installed and offers a configurable range of options including tabs, split views, themes, transparency, and key bindings.

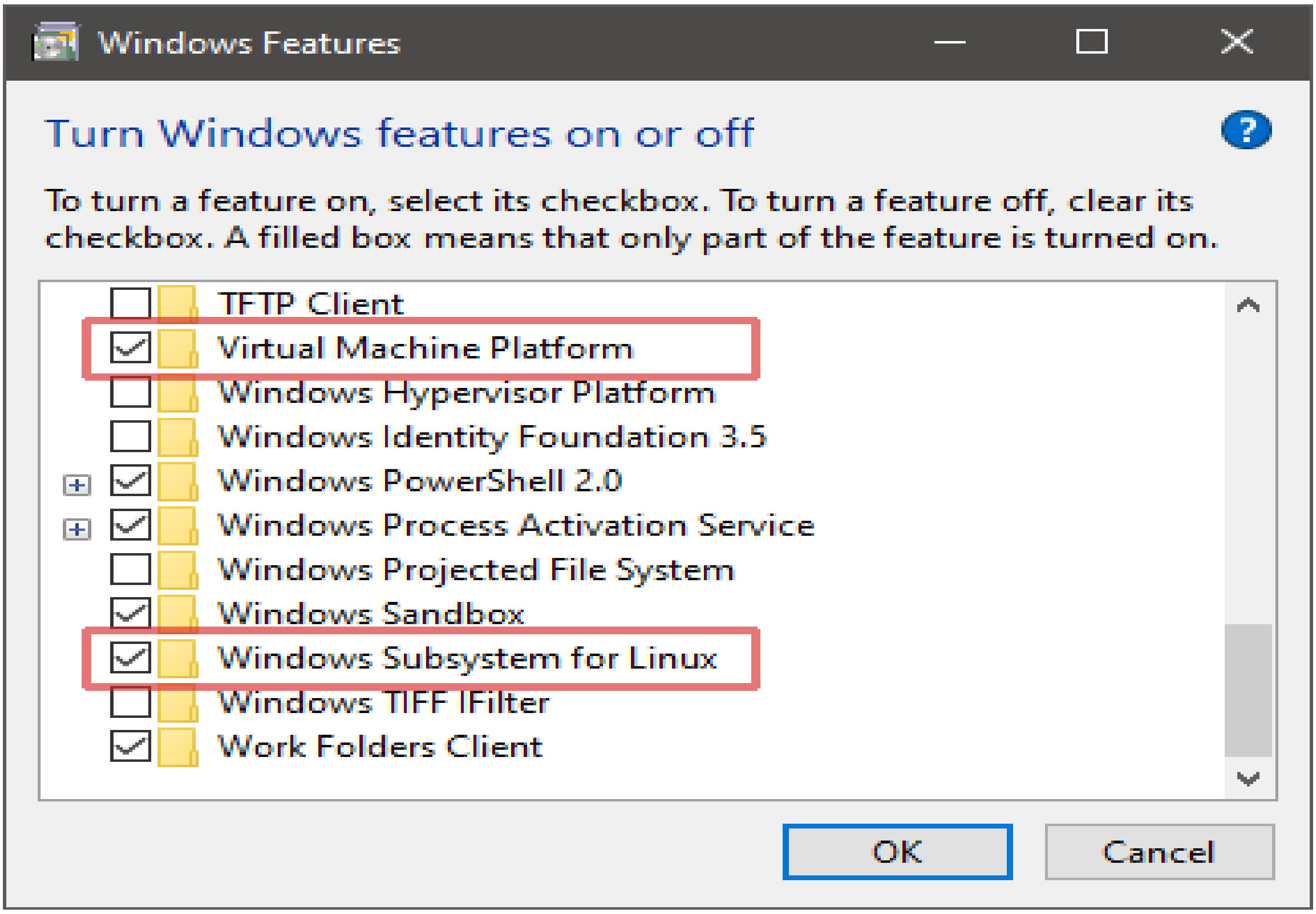
Configuration is available from the **Settings** menu item (Ctrl + ,), the **Command Palette** (Ctrl + Shift + P), or the settings.json accessed at the bottom of the **Settings** panel.



Enable WSL2 (Windows 10)

The steps in this WSL2 tutorial apply to WSL2 Windows 10. However, they should also work fine in Windows 11, although you should double-check if you performed a Windows 10 to 11 upgrade.

The **Virtual Machine Platform** and **Windows Subsystem for Linux** features must be enabled in **Turn Windows features on or off** option. The panel is accessed by hitting Start and typing “features” or from the **Programs and Features** icon in the classic **Control Panel**.



Turn Windows features on or off

To turn a feature on, select its checkbox. To turn a feature off, clear its checkbox. 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

Alternatively, you can execute the following commands in a Windows Powershell terminal run as an Administrator:

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

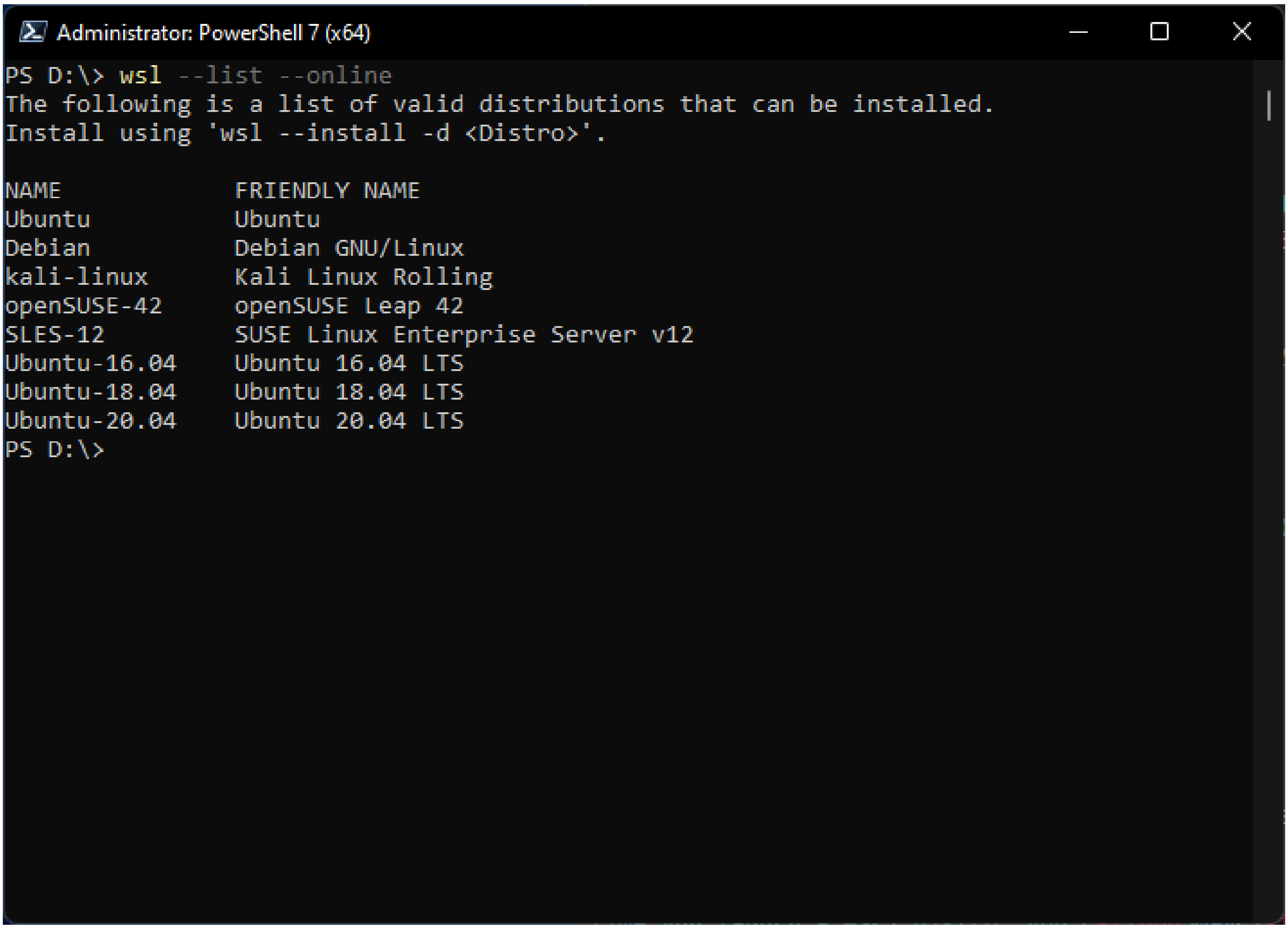
Reboot Windows, then start WSL2 as the default by entering the following command in a Windows PowerShell or command prompt:

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

Download Linux

Search for or locate **PowerShell** in the Start menu, right-click the icon, and choose **Run as administrator**. Display a list of available Linux distros by entering:

```
wsl --list –online
```



```
PS D:\> wsl --list --online
The following is a list of valid distributions that can be installed.
Install using 'wsl --install -d <Distro>'.

NAME               FRIENDLY NAME
Ubuntu             Ubuntu
Debian             Debian GNU/Linux
kali-linux         Kali Linux Rolling
openSUSE-42        openSUSE Leap 42
SLES-12            SUSE Linux Enterprise Server v12
Ubuntu-16.04       Ubuntu 16.04 LTS
Ubuntu-18.04       Ubuntu 18.04 LTS
Ubuntu-20.04       Ubuntu 20.04 LTS
PS D:\>
```

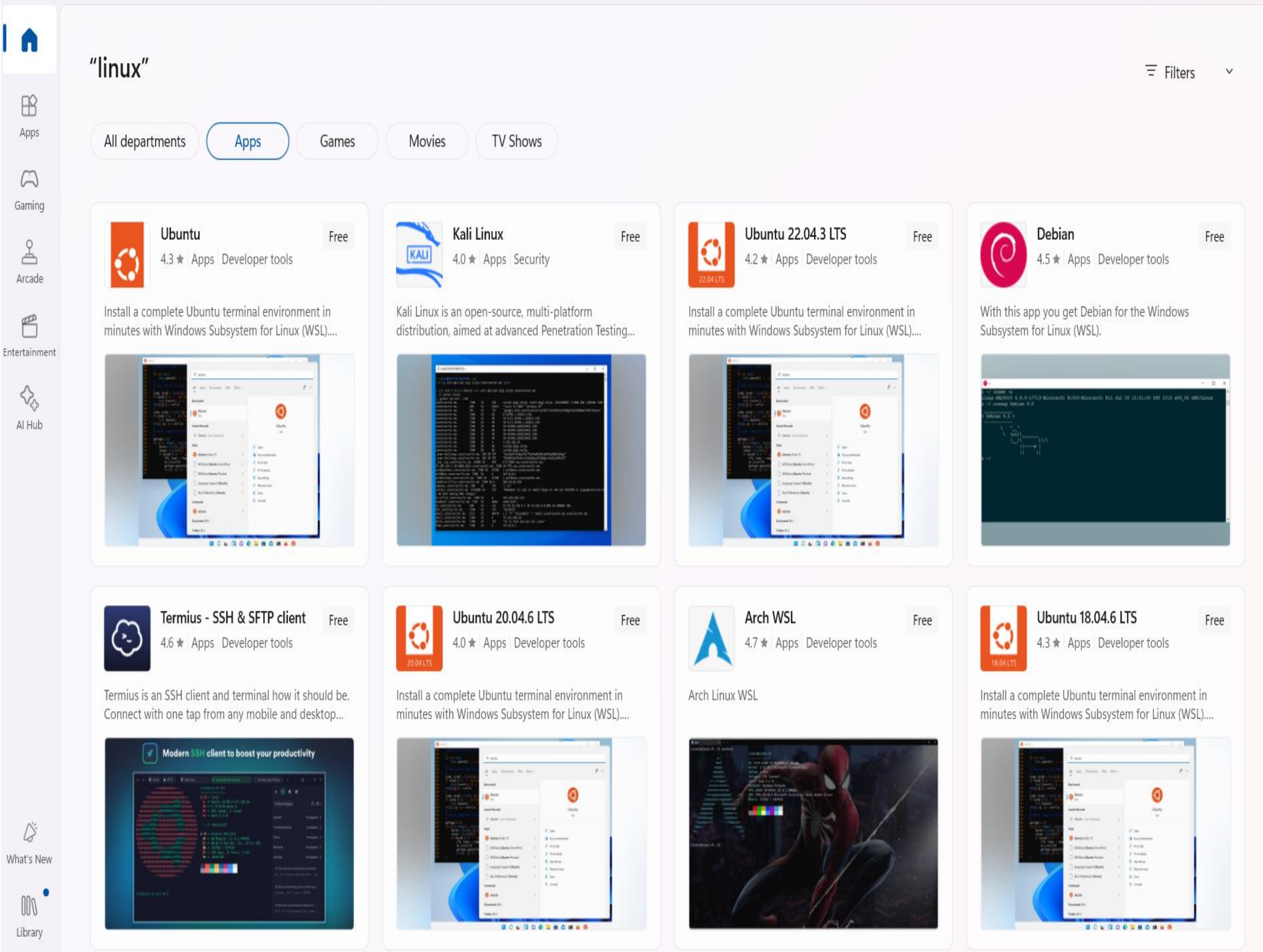
To install the default Ubuntu 20.04 distro, enter:

```
wsl --install
```

To install a specific distro by name, such as Debian, enter:

```
wsl --install -d Debian
```

Alternatively, you can install Linux distros from the **Microsoft Store**, which can be accessed from the Start menu. Enter “Linux” in the search box (be wary that software other than WSL distros may be available).



Ubuntu Free
4.3 ★ Apps Developer tools
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL)....

Kali Linux Free
4.0 ★ Apps Security
Kali Linux is an open-source, multi-platform distribution, aimed at advanced Penetration Testing...

Ubuntu 22.04.3 LTS Free
4.2 ★ Apps Developer tools
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL)....

Debian Free
4.5 ★ Apps Developer tools
With this app you get Debian for the Windows Subsystem for Linux (WSL).

Termius - SSH & SFTP client Free
4.6 ★ Apps Developer tools
Termius is an SSH client and terminal how it should be. Connect with one tap from any mobile and desktop...

Ubuntu 20.04.6 LTS Free
4.0 ★ Apps Developer tools
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL)....

Arch WSL Free
4.7 ★ Apps Developer tools
Arch Linux WSL

Ubuntu 18.04.6 LTS Free
4.3 ★ Apps Developer tools
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL)....

Ubuntu is a good choice unless you have specific Linux requirements. The latest edition is installed after you click the **Ubuntu** icon and follow the **Get** button.

The download and installation may take some time, depending on your network and PC speed. You may be prompted to reboot again.

Launch Linux

A new Linux (**Ubuntu**) option/icon will now be available in:

- The Start menu
- Windows Terminal (if installed)
- Powershell — enter wsl or bash at the prompt

Third-party terminal software such as Cmder, ConEmu, or Hyper can be used by creating profiles which launch %windir%\system32\bash.exe ~.

The installation process will be complete the first time you run the WSL2 Linux OS.

Note: You ***may*** *see a message in the terminal about a kernel update. Copy the link into your browser, then download and install the update. You'll need to launch the Linux distro again to continue the installation.*

You'll be prompted to enter a username and password. These are the credentials for Linux administration: *they're completely separate from your Windows username and password* (although choosing the same ones may be practical).

Linux will eventually be ready, and your terminal will show content similar to the following:

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.10.102.1-microsoft-standard-WSL2 x86_64)

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

  System information as of Tue Aug 16 10:19:36 BST 2022

  System load:  0.0                Processes:             8
  Usage of /:   1.6% of 250.98GB   Users logged in:       0
  Memory usage: 1%                 IPv4 address for eth0: 172.21.6.83
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

Update Linux

There are likely to be several Linux updates. To update Ubuntu applications, enter the following commands in the Linux terminal:

```
sudo apt update
sudo apt upgrade
```

Other Linux distros will have a similar process — so check the documentation for assistance.

You can also check for Linux kernal updates from Powershell using:

```
wsl --update
```

Switch Between WSL1 and WSL2

Existing WSL1 distros can be converted to WSL2. For that, enter the following command in a Windows Powershell terminal to display the installed Linux distros and their WSL version:

```
PS C:\> wsl --list --verbose
```

```
  NAME    STATE    VERSION
  * Ubuntu  Running  1
```

(The * asterisk highlights the default Linux installation.)

To switch Ubuntu to WSL2, enter:

```
wsl --set-version Ubuntu 2
```

Similarly, to switch back to WSL1, enter:

```
wsl --set-version Ubuntu 1
```

Set a Default Linux Distribution

One Linux distribution must be set as the default when you have more than one installed. This will be used when wsl is entered in a Windows Powershell terminal.

To set the default distro, list your Linux installations:

```
wsl --list
```

and set a default with:

```
wsl --setdefault <DistributionName>
```

Ubuntu example: wsl –setdefault Ubuntu.

Run Linux as a Specific User

To run your default distribution as a specific user, enter the following command in a Windows Powershell terminal:

```
wsl --user <username>
```

A default user is defined during installation but you can create others. A root user is also created, but you should avoid using it under normal circumstances. *It's too easy to enter a dangerous command!*

Move or Clone Your Linux Disk Image

Linux disk images are installed on your C: drive. Optionally, you can either:

- Move it to another drive to free up space on C:, or
- Use the same image to create multiple Linux installations (which can be useful if you need different applications and setups for different projects — although Docker may be more practical)

The following instructions presume you're moving the **Ubuntu** Linux distro to D:\wsl.

In a Windows Powershell terminal (*not* the Ubuntu terminal), enter wsl –list to view your Linux distributions, then export one by name to a backup .tar file — such as D:\backup\ubuntu.tar:

```
mkdir D:\backup
wsl --export Ubuntu D:\backup\ubuntu.tar
```

Unregister that distro to remove it from the C: drive:

```
wsl --unregister Ubuntu
```

Enter wsl –list to verify the distro has been removed.

Import the backup into a new WSL2 distro at another location, such as D:\wsl:

```
mkdir D:\wsl
wsl --import Ubuntu D:\wsl\ D:\backup\ubuntu.tar
```

You can make any number of named clones from the same back-up. For example:

```
wsl --import Clone01 D:\wsl\ D:\backup\ubuntu.tar
wsl --import Another D:\wsl\ D:\backup\ubuntu.tar
```

Verify the WSL distro has been successfully created by entering wsl –list and launching **Ubuntu** from the Start menu or Windows Terminal.

At this point, Ubuntu will use root as the default user. To revert to your own account, enter the following command:

```
ubuntu config --default-user <yourname>
```

Here, <yourname> is the username you defined during installation.

For other distros that aren't the WSL Windows default, define a user by logging on to the distro and creating/editing /etc/wsl.conf:

```
touch /etc/wsl.conf
nano /etc/wsl.conf
```

Add the following lines to the file:

```
[user]
default=<yourname>
```

Save (Ctrl + O) and exit (Ctrl + X), then restart the distro in a Powershell terminal with:

```
wsl --terminate <distroname>
```

Presuming all has gone well, you can delete the backup file (D:\backup\ubuntu.tar).

Access Linux Files from Windows

You can access WSL2 files from Windows by navigating to \\wsl$\. Enter it in the File Explorer address bar or any file open dialog.

Your installed Linux distros are listed, so you can access the Ubuntu root directory at \\wsl$\Ubuntu. Your personal Linux files are typically be stored at:

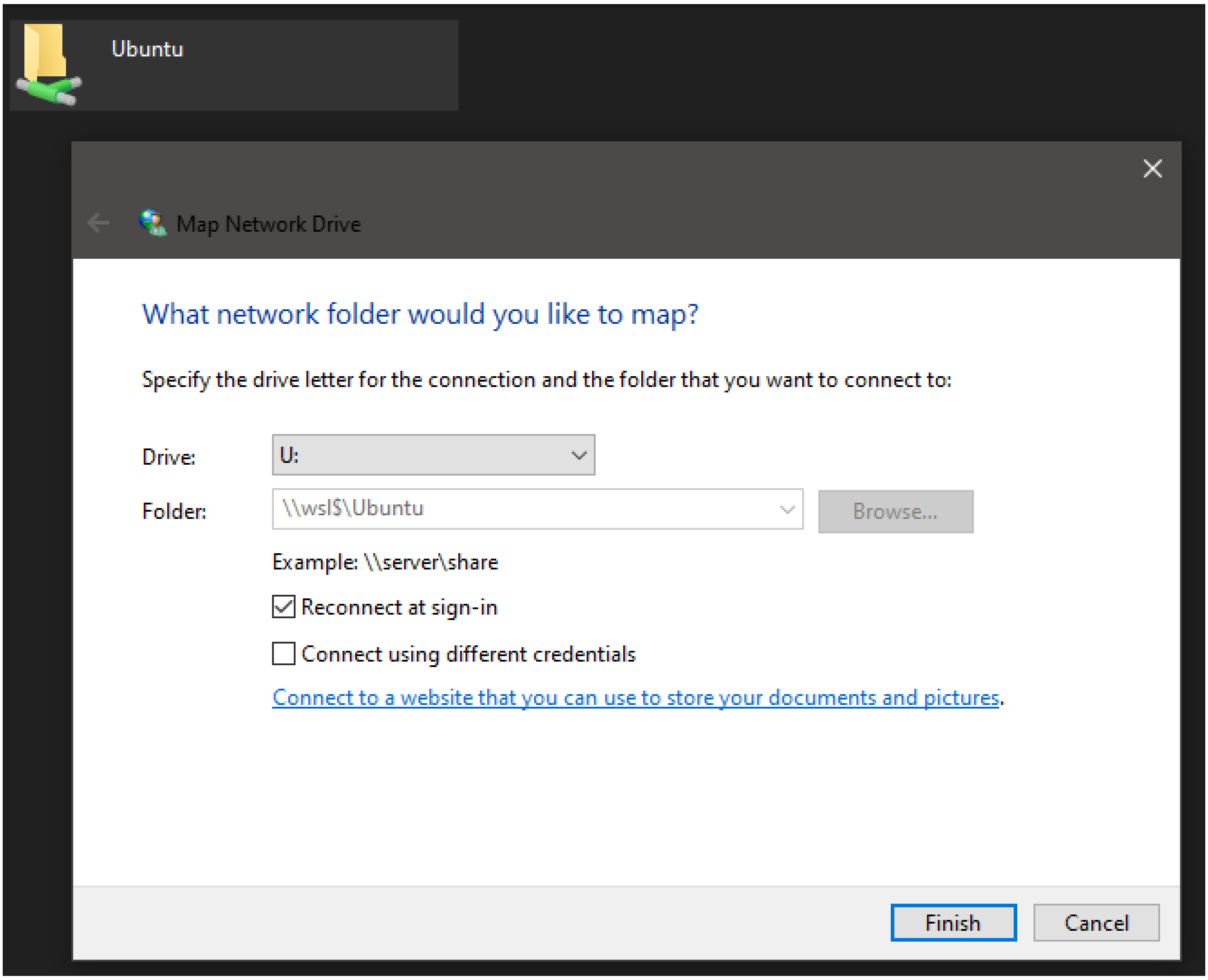
```
\\wsl$\Ubuntu\home\<yourname>
```

Here, <yourname> is the username you defined during installation.

It’s best to set this as the starting folder for the distro in Windows Terminal. Open the **Settings**, click a profile, then change the **Starting directory** option.

Map a Network Drive

You can mount a network drive to \\wsl$\Ubuntu\ by opening \\wsl$\ in File Explorer, right-clicking the Ubuntu folder, and choosing **Map network drive…**.



What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: U:

Folder: \\wsl$\Ubuntu

Example: \\server\share

☑ Reconnect at sign-in

□ Connect using different credentials

Connect to a website that you can use to store your documents and pictures.

Unfortunately, it’s not currently possible to map deeper subdirectories. You may also encounter mapped-drive issues with some applications.

Accessing Windows Files from Linux

The WSL2 access Windows files capability allows seamless interaction with files stored on your Windows drives through the /mnt/ directory. For example, your personal Users folder at C:\Users\<yourname> is available at:

```
/mnt/c/Users/<yourname>
```

Here, <yourname> is your Windows username.

For ease of access, you can create a Linux symbolic link to any Windows folder from the terminal. For example, for C:\projects\code\:

```
cd ~
ln -s /mnt/c/projects/code/
```

A code folder will appear in your Linux home directory. Navigate to it using cd ~/code and you’ll actually be in /mnt/c/projects/code/, which maps directly to C:\projects\code\.

Accessing Windows files from Linux is considerably slower than using the native Linux file system. Where possible, create projects in the Linux file space, typically in your home folder (/home/<yourname>/ or ~).

Run Linux Commands from Windows

Any Linux (bash) shell command can be run from a Windows Powershell or command-line terminal using wsl:

```
wsl <linux-command>
```

For example, wsl ls -la lists all files in a Windows folder.

Run Windows Applications from Linux

Through Windows WSL2, you can launch Windows applications like Notepad or VS Code directly from your Linux environment.

```
explorer.exe .
```

Or edit .bashrc in Notepad:

```
notepad.exe ~/.bashrc
```

Or open the a specific project directory in VS Code:

```
code ~/projects/mywebsite
```

Install Applications

Always remember you're running **two** operating systems. They may be tightly integrated, but there are situations when you want an application installed everywhere.

For example, Git is useful in both Windows and Linux. The Windows edition is installed by downloading an executable. It's best to ensure it doesn't convert line endings:

```
git config --global core.autocrlf input
```

Git on Ubuntu is installed using:

```
sudo apt update
sudo apt install git-all
```

Similarly, you may want to test Node.js applications on both Windows and Linux. Again, Windows has a runtime installer while Node.js runtime is installed on Ubuntu using commands such as:

```
sudo apt-get install build-essential
```

```
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs
```

git, node and npm commands will now work in either OS. Be wary that they may be different versions.

Installing Graphical Linux Applications

WSL Windows allows you to run graphical X11 and Wayland Linux applications directly on Windows. You can launch Linux apps from the Start menu, pin them to the task bar, and use features such as cut and paste across applications. There are some downsides:

- Not all graphical apps will work as expected.
- Some offer Linux features — such as file dialogs — that could confuse Windows users.
- There’s little benefit in installing applications with native Windows editions such as Firefox, VLC, or GIMP.

To proceed, install the correct GPU driver for your Windows machine:

- Intel GPU driver for WSL
- AMD GPU driver for WSL
- NVIDIA GPU driver for WSL

Then, install any graphical application from the Linux terminal. For example, the GNOME Gedit text editor:

```
sudo apt install gedit
```

Launch it with:

```
gedit
```

Pin the app to your Start menu or task bar like any other Windows program.

Gedit is a small application with a download size of less than 2MB. However, it requires GNOME and other Linux libraries, which incur several hundred megabytes of additional code on top of the chunky GPU drivers (1.1GB for Intel). Using graphical Linux apps in WSL2 only makes sense if you need to install several applications or there's no native Windows alternative.

Installing Docker

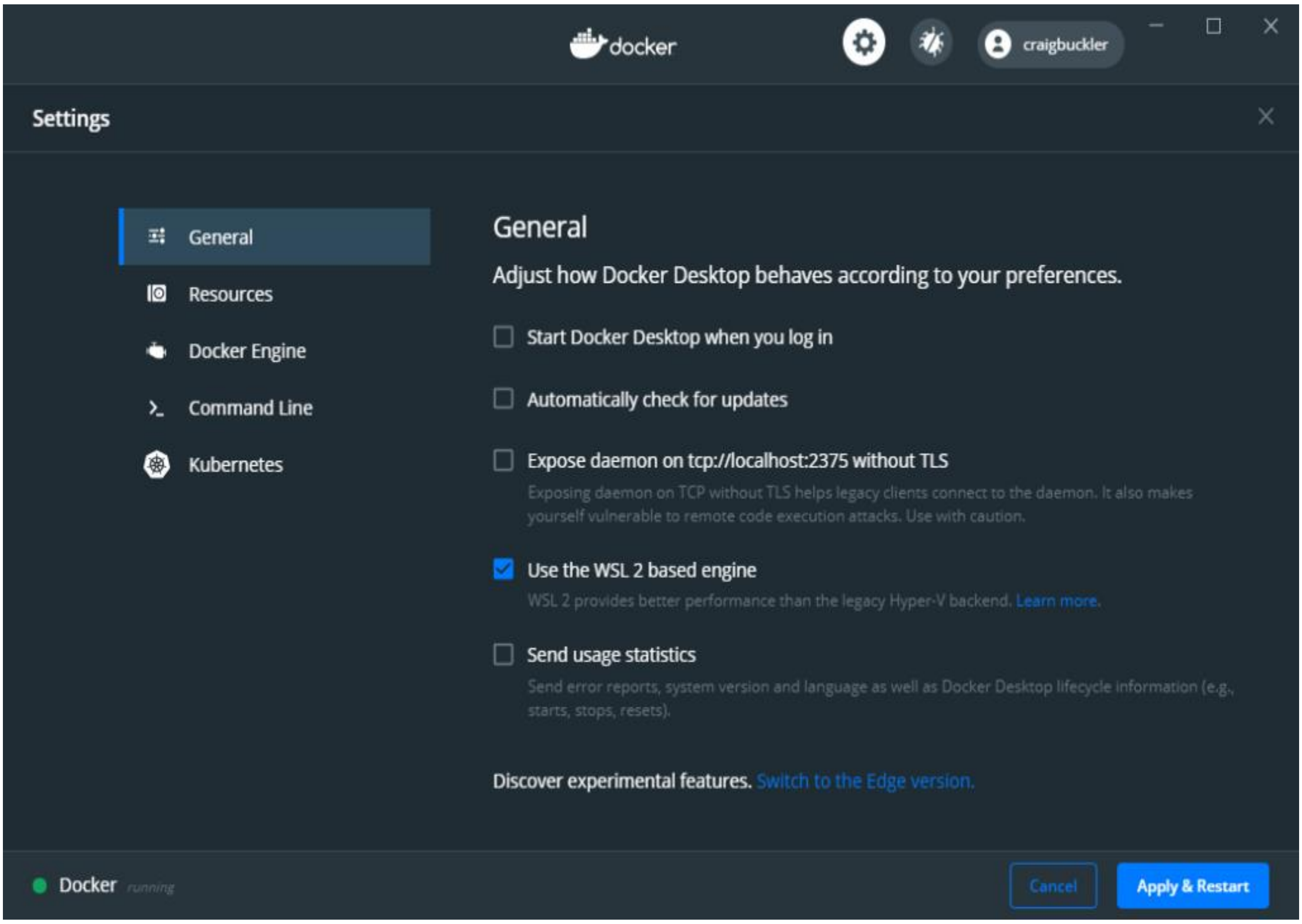
One exception to the *"install on both OSes"* rule is Docker!

Docker runs applications in isolated Linux containers that are conceptually similar to lightweight virtual machines with an OS, single application, and its dependencies. Using Docker for web development provides a consistent environment that's easy to install and works on any device.

Docker Desktop for Windows is fully WSL2-enabled:

- there's no need to use Hyper-V, so Docker can be installed on Windows Home
- Docker starts and runs considerably faster when configured to use WSL2
- installing Docker Desktop on Windows enables docker and docker-compose in both Windows and Linux

Docker Desktop suggests you use WSL2 when it's first launched. Alternatively, you can select **Settings** from the Docker system tray icon menu, then choose the **General** tab, check **Use the WSL 2 based engine**, and hit **Apply & Restart**.



General

Adjust how Docker Desktop behaves according to your preferences.

☐ Start Docker Desktop when you log in

☐ Automatically check for updates

☐ Expose daemon on tcp://localhost:2375 without TLS
Exposing daemon on TCP without TLS helps legacy clients connect to the daemon. It also makes yourself vulnerable to remote code execution attacks. Use with caution.

☑ Use the WSL 2 based engine
WSL 2 provides better performance than the legacy Hyper-V backend. Learn more.

☐ Send usage statistics
Send error reports, system version and language as well as Docker Desktop lifecycle information (e.g., starts, stops, resets).

Discover experimental features. Switch to the Edge version.

Docker uses the default Linux distro, but you can also enable it in other installed distros from the **WSL Integration** panel in **Settings**, then **Resources**.

Launching Apache and PHP Applications Using Docker

Installing web dependencies such as Apache and PHP directly in your WSL2 Linux environment is possible. However, Docker is safer, quicker, and more configurable: it allows you to run different versions of PHP on the same device simultaneously.

In this example, you'll create a small PHP file and run it using the latest release of PHP 7.4 in a Docker container. Ensure you've installed Docker Desktop for Windows and enabled WSL2 mode, then create a folder in your Linux environment with an index.php file. For example:

```
mkdir ~/php
cd ~/php
   touch index.php
```

Edit index.php, perhaps using Notepad or VS Code as shown above. Add the following content, which outputs PHP status information:

```
<?php
  phpinfo();
```

Run the following docker command to start a PHP 8.1-enabled Apache web server in a container which mounts files in the current host directory to /var/www/html in the container:

```
docker run \
  -it --rm -p 8080:80 --name php \
  -v "$PWD":/var/www/html \
    php:8.1-apache
```

Note: this command also works in Windows, but the reference to $PWD must be replaced by the full file path in Linux notation — such as /c/php/.

Open https://localhost:8080/ in any Windows web browser to view the output of the phpinfo() command.

To finish, press Ctrl + C in your Docker terminal to stop and delete the container.

WSL2 and localhost

Windows, WSL2 Linux, and Docker container applications are always accessed from localhost or 127.0.0.1.

The PHP example above exposed port 8080, which was mapped to port 80 in the container. Similarly, if you have Node.js installed in Linux, you can cd to any directory and start a static file server:

```
npx small-static-server
```

Open http://localhost:8888/ in a browser to view files in that directory (index.html is returned by default).

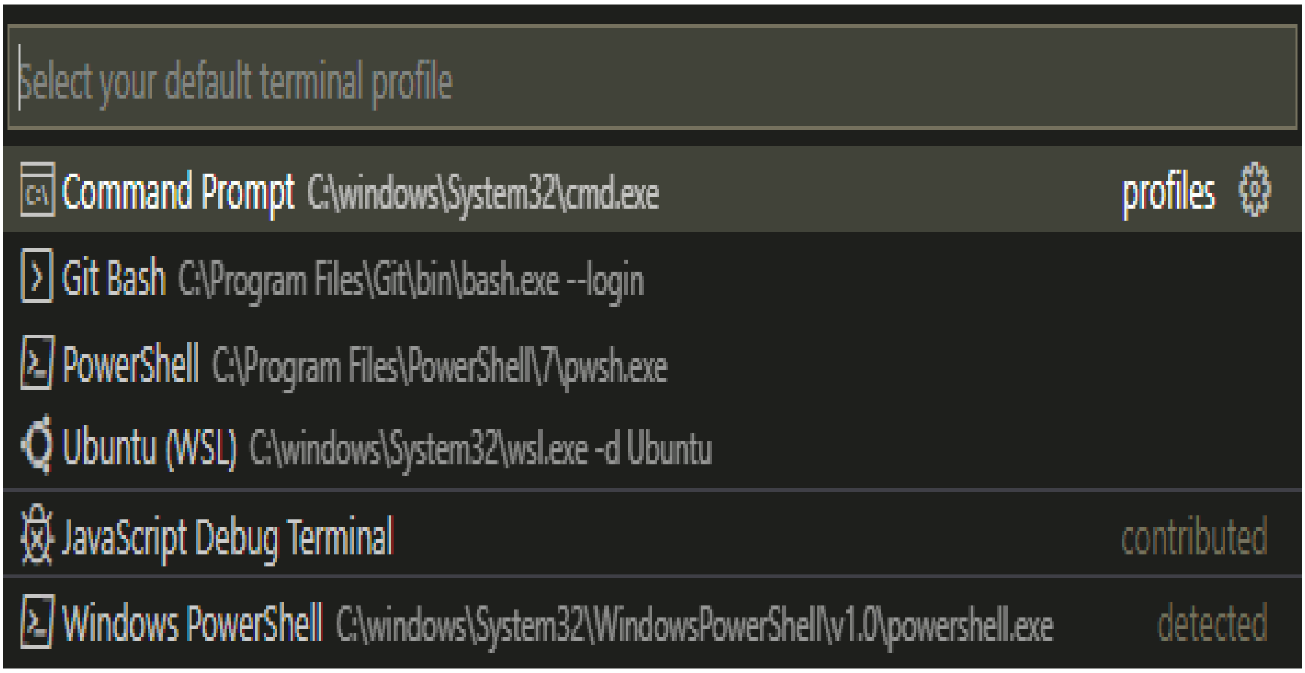
Be aware that ports available in one OS may be unavailable in the other. Port numbers less than 1,000 are only available to Linux super users. The Windows Store

Skype App often reserves port 80 and can’t be changed (although the classic Desktop edition permits it).

SSL certificates that are either self-signed or locally trusted using tools such as mkcert can be used interchangeably in both Windows and Linux environments.

VS Code WSL2 Integration

In VS Code, you can start WSL2 sessions directly from the integrated terminal. Open the terminal with **View** > **Terminal** or Ctrl + ‘ then hit the down arrow next to the **+** icon to launch a specific type. The same menu has a **Select Default Profile** option where you can select the terminal to use by default.


Select your default terminal profile
Command Prompt C:\windows\System32\cmd.exe
profiles
Git Bash C:\Program Files\Git\bin\bash.exe --login
PowerShell C:\Program Files\PowerShell\7\pwsh.exe
Ubuntu (WSL) C:\windows\System32\wsl.exe -d Ubuntu
JavaScript Debug Terminal
contributed
Windows PowerShell C:\windows\System32\WindowsPowerShell\v1.0\powershell.exe
detected

With VS Code, you can directly access WSL2 files from Windows at \\wsl$\ and debuggers running on localhost. There’s no need for further extensions, but the following options permit easier Linux and Docker file management:

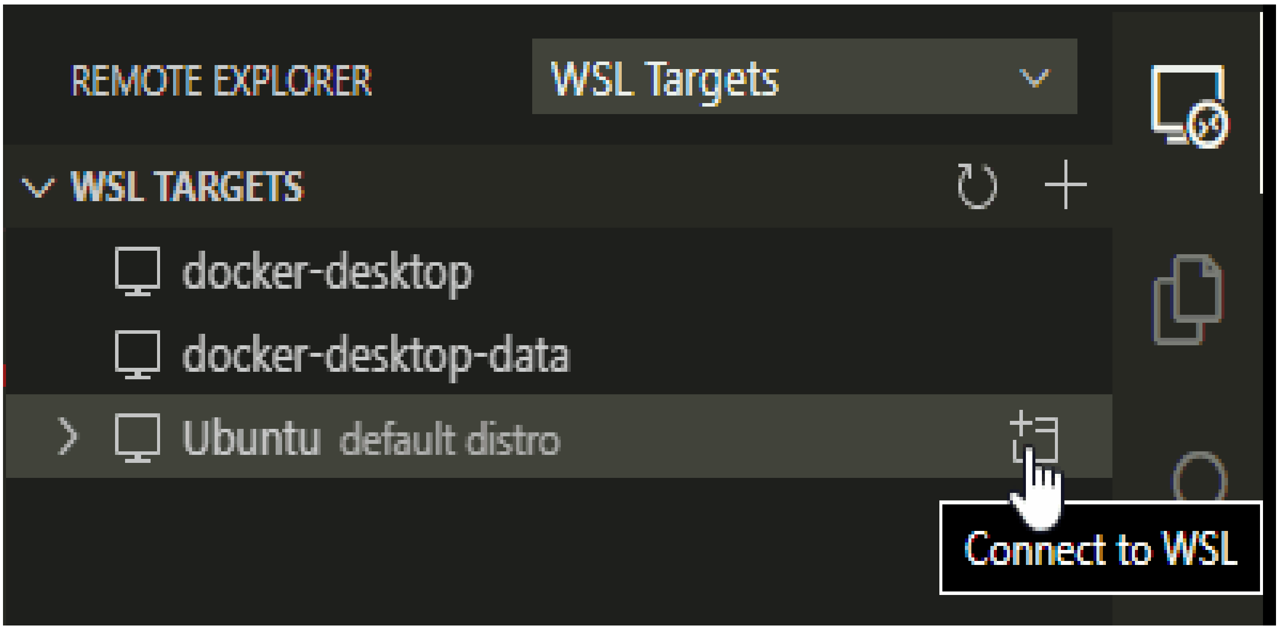
- Remote – WSL: open WSL2 Linux folders
- Remote – Containers: open or attach to a running Docker container
- Remote Development Pack: includes both extensions above as well as Remote – SSH to connect to remote servers

After installing **Remote – WSL**, you can navigate to any Linux directory and launch a WSL-connected instance of VS Code:

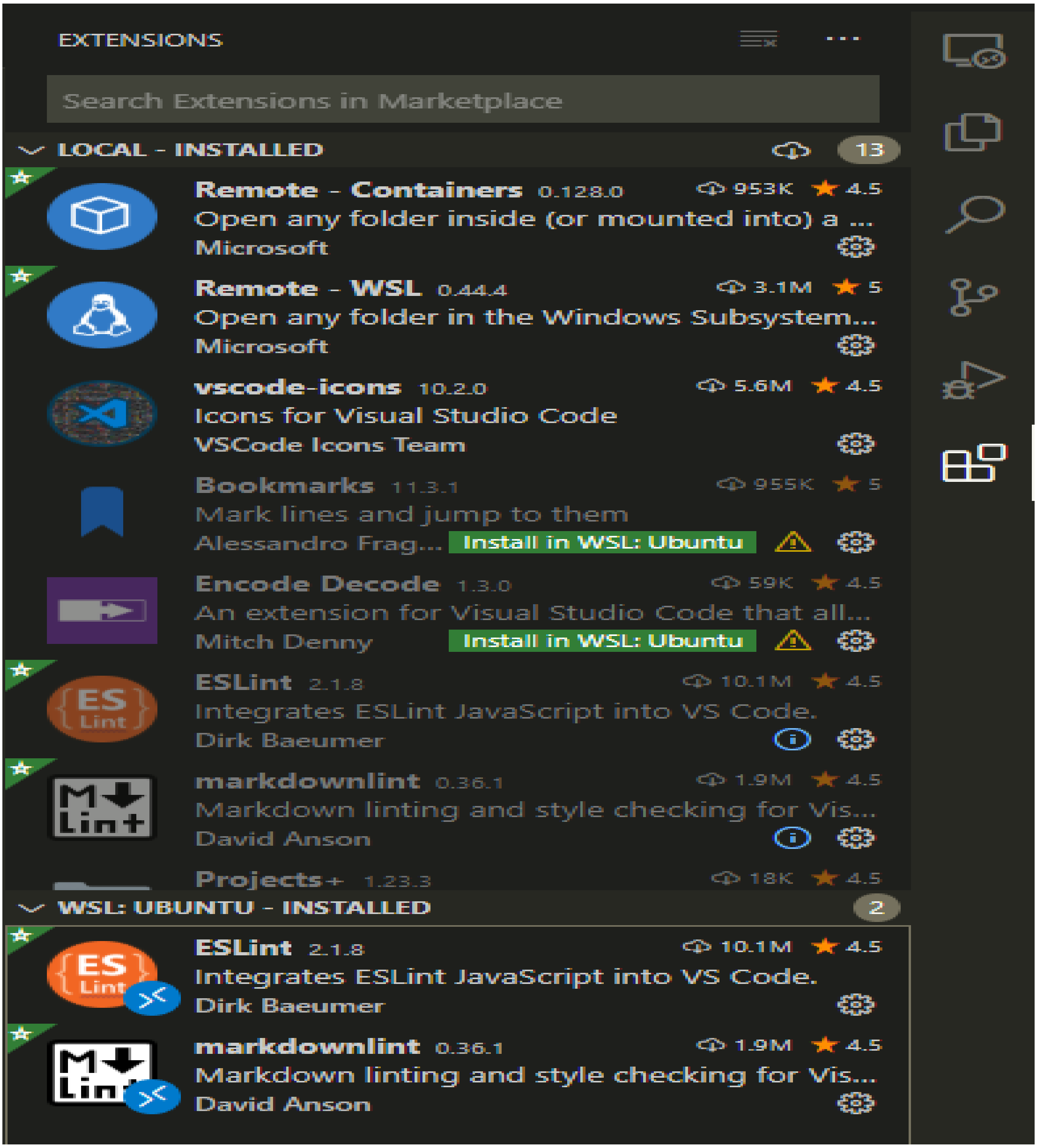
```
cd ~/myproject
code .
```

Alternatively, click the **remote window** icon in the bottom left of the application, then choose a connection option.

Or, you can use the new **Remote Explorer** icon in the Activity Bar. This lets you choose a target type from the drop-down menu and connect accordingly.



The remote instance of VS Code has separate extensions and settings for Linux use, but some can be globally enabled. The **Extensions** panel allows you to enable and configure extensions in the remote instance.



EXTENSIONS

Search Extensions in Marketplace

LOCAL - INSTALLED 13

Remote - Containers 0.128.0 953K ★ 4.5
Open any folder inside (or mounted into) a ...
Microsoft

Remote - WSL 0.44.4 3.1M ★ 5
Open any folder in the Windows Subsystem...
Microsoft

vscode-icons 10.2.0 5.6M ★ 4.5
Icons for Visual Studio Code
VSCode Icons Team

Bookmarks 11.3.1 955K ★ 5
Mark lines and jump to them
Alessandro Frag... Install in WSL: Ubuntu

Encode Decode 1.3.0 59K ★ 4.5
An extension for Visual Studio Code that all...
Mitch Denny Install in WSL: Ubuntu

ESLint 2.1.8 10.1M ★ 4.5
Integrates ESLint JavaScript into VS Code.
Dirk Baeumer

markdownlint 0.36.1 1.9M ★ 4.5
Markdown linting and style checking for Vis...
David Anson

Projects+ 1.23.3 18K ★ 4.5

WSL: UBUNTU - INSTALLED 2

ESLint 2.1.8 10.1M ★ 4.5
Integrates ESLint JavaScript into VS Code.
Dirk Baeumer

markdownlint 0.36.1 1.9M ★ 4.5
Markdown linting and style checking for Vis...
David Anson

Settings are stored in your Linux ~/.vscode-server directory.

This WSL2 tutorial has shown how to utilize WSL2 Windows 11 for running Linux applications on your Windows machine. Running Linux applications on Windows used to require virtual machines, complex Samba folder shares, and lots of patience. WSL2 has revolutionized web development. It makes it easy to write code using Windows tools and then run it in a production-like Linux environment. The tight integration offers the benefits of both OSes.

Further WSL2 resources:

- Windows Subsystem for Linux Documentation
- Developing in WSL
- VS Code Remote development in WSL
- Run Linux GUI apps on the Windows Subsystem for Linux
- Windows Terminal: The Complete Guide
- How to Install Docker on Windows 10 Home
- How to Install PHP on Windows 10 (with Apache & MySQL)
- *PHP & MySQL: Novice to Ninja*
- How to Build an Image with the Dockerfile
- Understanding Docker, Containers and Safer Software Delivery
- *Docker for Web Developers*

FAQs About Windows Subsystem for Linux 2

What Is the Difference Between WSL1 and WSL2?

WSL1 and WSL2 are both Microsoft's Windows Subsystem for Linux, but they operate differently. WSL1 translates Linux system calls into Windows system calls, which can lead to compatibility issues with certain Linux applications. On the other hand, WSL2 uses a lightweight virtual machine to run a full Linux kernel, providing better compatibility and performance, especially for file system heavy operations. However, WSL2 may consume more system resources than WSL1.

How Can I Upgrade from WSL1 to WSL2?

Upgrading from WSL1 to WSL2 is a straightforward process. First, you need to ensure that your Windows 10 version supports WSL2. Then, open PowerShell as an administrator and run the command wsl –set-version <distribution name> 2. Replace <distribution name> with the name of your Linux distribution. The upgrade process may take a few minutes.

Can I Use GUI Applications with WSL2?

Yes, you can use GUI applications with WSL2. Microsoft has introduced support for GUI applications in WSL2 with the Windows 10 May 2021 update. This means you can run Linux GUI applications directly on your Windows desktop without needing a separate X server.

How Can I Install a New Linux Distribution with WSL2?

You can install new Linux distributions with WSL2 Windows 11 from the Microsoft Store. Open the Microsoft Store, search for the Linux distribution you want to install, and click the “Get” button. Once the installation is complete, start WSL2 by launching the distribution from the Start menu.

How Can I Access My Windows Files from WSL2?

You can access your Windows files from WSL2 through the /mnt/ directory. For example, your C: drive is available at /mnt/c/. You can navigate to this directory using the cd command.

Is It Possible to Run Docker with WSL2?

Yes, it is possible to run Docker with WSL2. Docker has introduced support for WSL2 with the Docker Desktop for Windows. This allows you to run Docker containers directly from your WSL2 Linux distribution.

How Can I Update My WSL2 Linux Kernel?

You can update your WSL2 Linux kernel by downloading and installing the latest WSL2 Linux kernel update package from the Microsoft website. After the installation is complete, you need to restart your computer for the changes to take effect.

Can I Use Multiple Linux Distributions with WSL2?

Yes, you can manage multiple Linux distributions with Windows WSL2. Each linux distribution runs in its own environment, and you can switch between them as needed. You can install additional linux distributions from the Microsoft Store.

How Can I Uninstall a Linux Distribution from WSL2?

You can uninstall a Linux distribution from WSL2 by right-clicking on the distribution in the Start menu and selecting “Uninstall”. Alternatively, you can use the wsl –unregister <distribution name> command in PowerShell.

Can I Use WSL2 with Windows 11?

Yes, you can use WSL2 with Windows 11. In fact, Windows 11 has introduced several improvements to WSL2, including faster startup times and better system performance.

How Can I Switch Between Linux Distros in WSL2?

By default, the wsl command will open the default linux distro, but you can specify a different one by using the “-d” flag followed by the distro’s name. For example, to switch to a distro named “Debian,” you would use the command wsl -d Debian.

How Do I Update a Linux Distro in WSL2?

Open the terminal for the specific Linux distro and use the package manager to update it. For example, on an Ubuntu-based distro, you can run “sudo apt update && sudo apt upgrade”.