WSL Hardware Resource Usage



explain wsl completely like how it uses hardware resources and how much resources it will use

Sure, here's a **complete but simple explanation of WSL (Windows Subsystem for Linux)** — including how it works, how it uses hardware resources, and how much it consumes.



WSL (**Windows Subsystem for Linux**) is a compatibility layer that lets you run a **full Linux environment** directly inside **Windows**, without needing a virtual machine or dual-boot setup.

There are **two versions**:

Version	Description	
WSL 1	Translates Linux calls into Windows calls (lightweight, uses less resources).	
WSL 2	Runs a real Linux kernel in a lightweight virtual machine (VM) (better compatibility, slightly heavier).	

Today, **WSL 2** is the standard and preferred version.

How WSL Works Internally (WSL 2)

- **Virtual Machine**: WSL 2 uses a lightweight VM with a real Linux kernel running inside a special Microsoft-built VM.
- VHD File: Your Linux filesystem is stored in a .vhdx file (virtual hard disk).
- **Integration**: You can access Windows files from Linux (/mnt/c/), and Linux files from Windows (\\wsl\$).

How WSL Uses Hardware Resources

WSL shares your system's hardware resources with Windows.

Resource	How It Works	
СРИ	Shared. WSL can use all available CPU cores (no limit unless you set one).	
RAM	Dynamic. WSL uses only what it needs, but can grow up to 80% of your total RAM unless limited.	
Disk	VHDX file grows as needed (can be several GB depending on what you install).	
GPU	Supported via WSLg + GPU compute (for GUI apps and AI workloads).	
Network	Bridged through Windows — behaves like part of your system network.	

You can configure limits on CPU/RAM/IO using .wslconfig.



How Much Resources WSL Typically Uses

Depends on what you're running, but here's a general idea:

Use Case	RAM	Disk Space	CPU
Idle (just open terminal)	~100–300 MB	~1-2 GB	~0%
Coding/Dev (Node, Python, etc.)	~500 MB-2 GB	~2-5 GB	Medium
Running Docker or LLMs	4-8+ GB	10+ GB	High
GUI apps (VS Code, Firefox, etc.)	1-4 GB	Depends	Moderate

Can I Control or Limit Resources?

Yes! Create a file:

%UserProfile%\.wslconfig

ini [wsl2] memory=4GB processors=2 swap=2GB

Restart WSL after changes using:

sh wsl --shutdown

WSL vs VM vs Docker (for resource usage)

Feature	WSL 2	VM (VirtualBox, VMware)	Docker
Lightweight	✓	X (heavier)	✓
Performance	High	Medium	High
Manual resource control	(via config)	✓	✓
GUI support	(WSLg)	✓	(CLI only unless configured)

Summary

- WSL 2 runs Linux in a lightweight VM, sharing system resources dynamically.
- It's efficient for development, supports full Linux apps, and is easy to configure.
- You can cap CPU, RAM, swap usage via .wslconfig if needed.
- WSL is ideal for devs, students, or researchers who want Linux tools without leaving Windows.

Would you like a guide on how to set .wslconfig or monitor usage in real time?



how wsl manages multiple linux os images

Great question! Here's how **WSL manages multiple Linux OS images** (like Ubuntu, Kali, Debian, etc.) — explained simply and clearly:

Multiple Linux Distros in WSL

WSL supports **multiple Linux distributions side-by-side** — each acts like an **independent environment (container/VM)**.

For example, you can install:

- Ubuntu
- Kali Linux
- Debian
- openSUSE
- Arch Linux (via custom import)

Each of these is treated as a **separate instance**, with its own:

- Filesystem
- Users
- Packages
- Processes

How WSL Stores and Manages Distros

1. Filesystem Location:

• Each distro is stored in its own folder under:

```
shell
%LOCALAPPDATA%\Packages\
```

For example:

```
%LOCALAPPDATA%\Packages\CanonicalGroupLimited.UbuntuonWindows_<...>\
```

• Inside, there's a .vhdx file — this is a virtual disk image that contains your Linux root filesystem.

2. Internal Names and Management:

• WSL registers each distro with a **name**.

• Use this command to see them:

```
bash
wsl --list --verbose
```

Example output:

```
NAME STATE VERSION
Ubuntu Running 2
Kali-Linux Stopped 2
Debian Stopped 2
```

3. Switching Between Distros:

You can start any distro by name:

```
bash
wsl -d <DistroName>
```

Or set a default:

```
bash
wsl --set-default <DistroName>
```

How It Handles Resources for Multiple Distros

- Shared VM Kernel (in WSL 2): All WSL 2 distros use the same Linux kernel managed by WSL.
- **Separate Filesystems and Environments**: Each distro has **its own root filesystem** and runs isolated, like its own container.
- Only Active Distros Use Resources: If a distro is not running, it uses zero CPU/RAM.

Installing and Managing Distros

You can:

- Install via Microsoft Store (e.g., search "Ubuntu" or "Kali Linux").
- Or import/export distros using .tar files:

```
bash
wsl --import MyCustomDistro C:\WSL\CustomDistro customdistro.tar
```

You can **backup/export** a distro too:



bash

wsl --export Ubuntu ubuntu-backup.tar

Example Use Case

Let's say you have:

• Ubuntu: for development

• Kali: for cybersecurity tools

• Debian: for testing software compatibility

You can run each separately:

bash

wsl -d Kali-Linux

They behave like completely separate machines under the same system.

🖈 Summary

- WSL manages multiple Linux OSes like lightweight VMs with shared kernel but isolated environments.
- Each has its own storage, packages, and users.
- You can easily install, remove, or switch between them.
- WSL 2 makes sure they all run efficiently, and only use system resources when running.

Would you like a demo command list to install and manage multiple distros via terminal?



how much heavy to the windows system like if we install wslq

Installing **WSLg** (**Windows Subsystem for Linux GUI**) adds some overhead to your system, but it's still much lighter than traditional virtual machines. Let me break down exactly **how heavy WSLg is** on a Windows system:

What is WSLg?

WSLg lets you run **Linux GUI apps** (like Firefox, Gedit, or even graphical IDEs) directly on Windows via WSL 2. It includes:

- A Wayland server
- A PulseAudio server (for sound)



- A XWayland server (for X11 apps)
- FreeRDP (to render the GUI in a Windows window)

These run behind the scenes when needed.

lack (Disk Usage)

Component	Approx Size
Base WSL 2 (kernel, tools)	~500 MB
WSLg	~1.5 – 2 GB
One Linux distro (e.g., Ubuntu)	~1.5 – 3 GB after install
Total (typical setup)	~4 - 6 GB

So WSL + WSLg + a distro ≈ 4-6 GB of disk space.

This grows as you install more packages or distros.

Memory & CPU Usage (RAM & CPU)

Activity	RAM Usage	CPU Usage
WSL (idle, CLI only)	100-300 MB	~0%
WSLg GUI app (e.g., Gedit)	500 MB – 1 GB	Low (~1-5%)
Heavy GUI app (e.g., VS Code, Firefox, GIMP)	1–2+ GB	Moderate to High

- Idle WSLg services use minimal RAM (less than 100 MB when no GUI app is open).
- Apps only consume resources when running.

GPU-accelerated apps (e.g., AI tools, rendering software) can use **your real GPU**, which may increase load — but this is a benefit, not a burden.

CPU/GPU Impact

- WSLg offloads rendering to **Windows GPU** via **DirectX and WDDM** (Windows Display Driver Model).
- Minimal CPU usage unless you're running **GPU-heavy** apps (e.g., ML models or games).
- No noticeable lag for typical apps like terminal editors, browsers, or file managers.

Performance Impact on Windows

Metric	Impact
Boot time	None
Background resource usage	Low
Foreground multitasking (with apps open)	Moderate (depends on app)
Overall system slowdown	Negligible for normal use unless running many GUI apps at once



Factor	With WSLg	
Disk Space	+2 GB approx	
RAM Usage	+500 MB to 2 GB per active GUI app	
CPU Usage	Low to Moderate, only when GUI apps are open	
GPU	Used for acceleration; can improve performance for GUI apps	
Windows performance	Unaffected when WSL is idle; minimal impact when in use	

Recommendation

If you're running modern hardware (8GB+ RAM, SSD, decent CPU) — WSLg is very lightweight and won't slow down Windows noticeably. It's ideal for devs, cybersecurity students, or AI practitioners who want Linux tools with GUI support.

Would you like help checking your current system suitability or optimizing WSLg performance?



how to only terminal, only gui then

Great! Here's how to **run only the terminal** or **only GUI apps** in WSL (with or without WSLg), depending on your use case:

Option 1: Only Terminal (No GUI / No WSLg)

If you want just the Linux terminal (text-based), without WSLg:

✓ How:

• Use WSL 2 without installing WSLq:

bash
wsl

or

bash

wsl -d <DistroName>

Avoid GUI features:

- Don't run any GUI apps like gedit, firefox, etc.
- Don't install GUI libraries or desktop environments (like GNOME, KDE).





If WSLg is installed but you still want terminal-only behavior, **just don't open GUI apps** — they won't load background GUI servers unless needed.

☐ Option 2: Only GUI Apps (No Terminal)

If you want to run **Linux GUI apps** directly, without opening a terminal first:

✓ How:

- Open Start Menu → Search for your distro (e.g., "Ubuntu").
- Right-click → Open File Location → Create a shortcut.
- You can also create shortcuts to GUI apps directly:

```
echo "[Desktop Entry]
Name=Gedit
Exec=gedit
Type=Application" > ~/.local/share/applications/gedit.desktop
```

This app will appear in the Windows Start Menu under "Linux Apps".

Optional: Disable or Enable WSLg

If you want to **prevent WSLg from starting**, you can try these:

Disable GUI in WSL (advanced):

Add this to your .bashrc or .zshrc:

```
bash
export DISPLAY=""
```

This disables GUI output — even if you accidentally run a GUI app.

To re-enable:

```
bash
unset DISPLAY
```

Summary Table

Mode	How to Use	Notes
Terminal only	wslorwsl -d distro	No GUI apps started

Mode	How to Use	Notes
GUI only (apps)	Launch via Start Menu or shortcut	No terminal shown
Both	Use terminal, then run firefox , gedit , etc.	Starts WSLg if needed
Disable GUI apps	export DISPLAY=""	Prevents GUI launching

Would you like help creating shortcuts for Linux GUI apps on your Windows desktop or menu?



how to uninstall wsl current version and install wsl2 and wslg

Here's a step-by-step guide to fully uninstall your current WSL version and then cleanly install WSL 2 with WSLq on your Windows system.

Step 1: Uninstall Current WSL Components (Optional/Clean Slate)

1 Only do this if you want to reset or clean up everything.

🔄 Unregister & Remove Linux Distros

```
bash
wsl --list --all --verbose
```

For each distro listed:

bash wsl --unregister <DistroName>

Uninstall WSL Components

- Open Settings > Apps > Installed Apps
- Uninstall the following if present:
 - Windows Subsystem for Linux
 - Ubuntu / Kali / Debian (your distros)
 - Windows Subsystem for Linux Update
 - Windows Subsystem for Linux GUI App (WSLq)

You can also uninstall from PowerShell:

```
powershell
wsl --unregister <DistroName> # removes individual distro
```

Then in PowerShell (Admin):

```
powershell
wsl --uninstall
```

Step 2: Install WSL 2 with WSLg

Requirements

Requirement	Notes
OS Version	Windows 10 (21H2+) or Windows 11
Features	Virtual Machine Platform + WSL
RAM	8 GB+ recommended for WSLg
Internet	Required to download packages

Installation (1 Command)

Open PowerShell as Administrator and run:

```
powershell
wsl --install
```

This command will:

- Install the **latest WSL version** (WSL 2)
- Install WSLg (if on Windows 11 or 10 22H2+)
- Install **Ubuntu** as default distro
- Enable required Windows features (like Virtual Machine Platform)

If you're using Windows 10 **before 22H2**, WSLg may not be supported.

Nerify Installation

After reboot:

```
bash
wsl --version
```

You should see:

```
WSL version: 2.x.x
Kernel version: ...
WSLg version: x.x.x
```

Check distros:

```
bash
wsl --list --verbose
```

It should show:

```
pgsql

NAME STATE VERSION
Ubuntu Running 2
```

Enable WSLg GUI Support (Optional Step – if missing)

If WSLg didn't install (e.g., on Win 10 21H2), you can install the **preview manually**:

```
powershell
wsl --install --web-download
```

Or download WSLg manually:

• https://github.com/microsoft/wslg/releases

But on **Windows 11**, WSLg is bundled and works out-of-the-box.

🧽 Optional Cleanup

To remove leftover files:

```
bash
rd /s /q %LOCALAPPDATA%\Packages\CanonicalGroupLimited.*
```

Also clear:

```
bash
C:\Users\<You>\AppData\Local\Docker
C:\Users\<You>\AppData\Local\Packages\MicrosoftCorporationII.WindowsSubsystemForLinux*
```

📝 Summary

Task	Command
Remove old WSL	wsluninstall
Unregister distros	wslunregister <distroname></distroname>
Fresh install WSL 2 + WSLg	wslinstall
Check version	wslversion
List distros	wsllistverbose

Would you like help installing a **specific Linux distro**, or setting up .wslconfig for resource limits after installation?



how to install gedit in ubuntu

To install **Gedit** (a simple GUI text editor) in **Ubuntu under WSL (with WSLg)**, follow these steps:

☑ Step-by-Step: Install Gedit on Ubuntu (in WSL 2)

1. Open Ubuntu terminal:

bash wsl

(or from Start Menu > Ubuntu)

2. Update package list:

bash sudo apt update

3. Install Gedit:

bash sudo apt install gedit -y

4. Run Gedit (with WSLg GUI):

bash gedit

If WSLq is correctly installed, **Gedit will open in a separate window** like a native app.



If gedit gives an error like DISPLAY not set:

- Make sure WSLg is installed.
- You're on Windows 11 or Windows 10 (build 22000+).
- Run:

```
bash
echo $DISPLAY
```

If it returns something like :0, WSLg is working.

📝 Summary

Command	Purpose
sudo apt update	Refresh package list
sudo apt install gedit -y	Install Gedit
gedit	Launch GUI app

Would you like to create a desktop shortcut or Start Menu entry for Gedit as well?



uninstall gedit

To uninstall Gedit from Ubuntu (in WSL), follow this simple command:

Uninstall Gedit

Open your Ubuntu terminal and run:

```
bash
sudo apt remove --purge gedit -y
```

This removes Gedit and its configuration files.

✓ Optional: Clean Up Unused Dependencies

After removing Gedit, clean up extra packages:



bash

sudo apt autoremove -y

This clears any leftover dependencies that are no longer needed.

Summary

Command	Purpose
sudo apt removepurge gedit -y	Uninstalls Gedit completely
sudo apt autoremove -y	Cleans up unused packages

Let me know if you'd also like to remove other GUI apps or reassign a different default editor.