**Title: Installing DevOps Tools (Git, Docker, Python) on Ubuntu 22.04 LTS in WSL2**

This recipe uses the **What – How – Why** format to guide you through installing essential DevOps tools on Ubuntu 22.04 inside WSL2 — preparing your Linux terminal for automation, scripting, and cloud-native workflows.

**🔸 What (Introduction & Theory)**

This recipe teaches you how to install the **core DevOps toolset** inside your Ubuntu 22.04 WSL2 environment, including:

1. **Git** – For version control and source code management.
2. **Python 3 + pip** – Used for automation, scripting, and DevOps tooling.
3. **curl & wget** – To interact with APIs and download files from the terminal.
4. **Docker** – For containerized development and deployments.
5. **htop & net-tools** – For system monitoring and basic networking.

These tools are the **minimum working kit** for any DevOps practitioner. They help you build pipelines, manage code, interact with APIs, run containers, and automate environments — all from your local Ubuntu shell.

**🔸 How (Step-by-Step Process)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

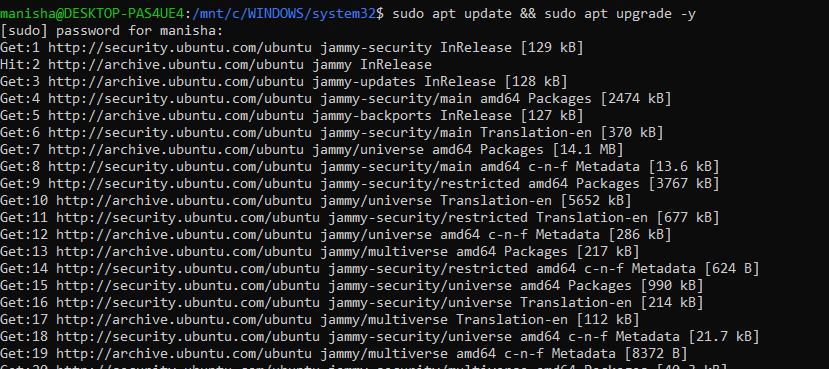
**Part 1: Update System & Install Core CLI Tools**

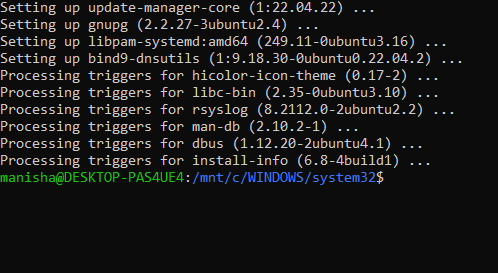
**Step 1: Open Ubuntu 22.04 (WSL Terminal)**

Launch Ubuntu 22.04 from the Start Menu or by typing wsl (if it's your default distro).

**Step 2: Update package index and upgrade system**

**sudo apt update && sudo apt upgrade –y**

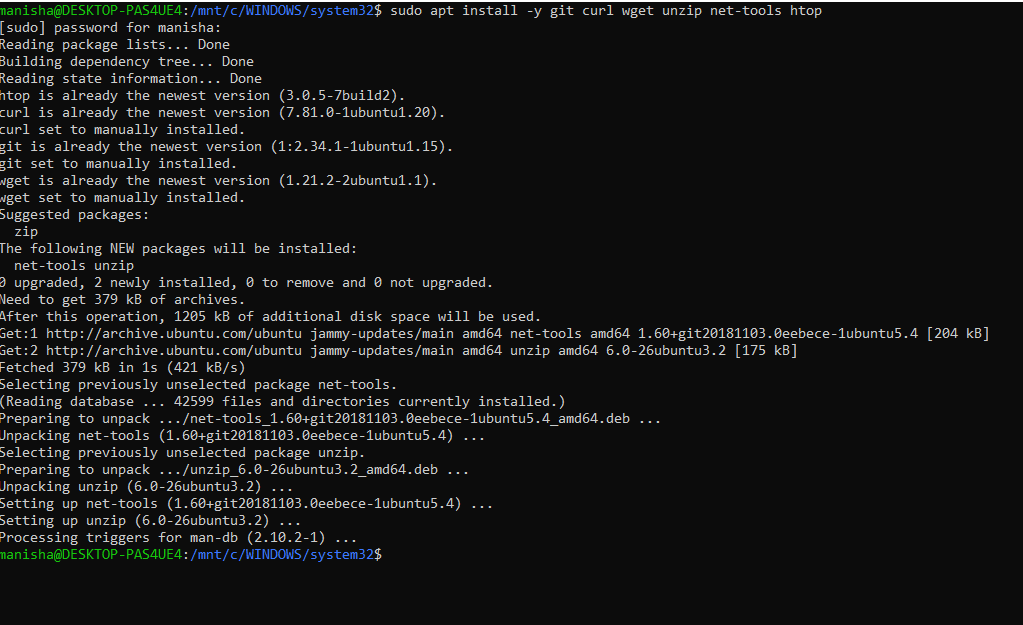
****

****

This ensures you're installing from the latest repositories.

**Step 3: Install core CLI utilities**

**sudo apt install -y git curl wget unzip net-tools htop**

****

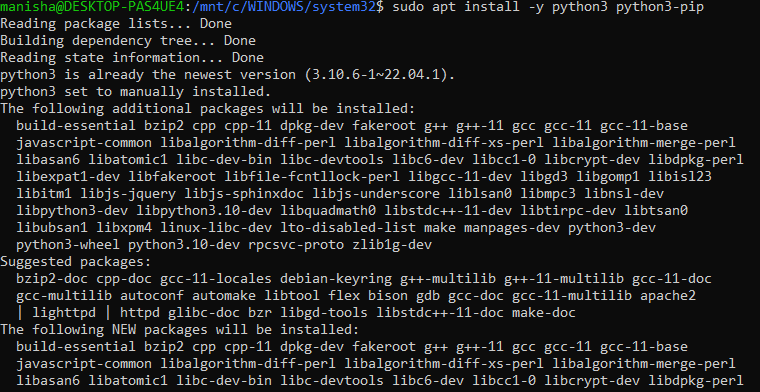
These tools allow you to manage files, monitor system resources, and interact with URLs or APIs.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Part 2: Install Python 3 and pip**

**Step 4: Install Python 3 and pip3**

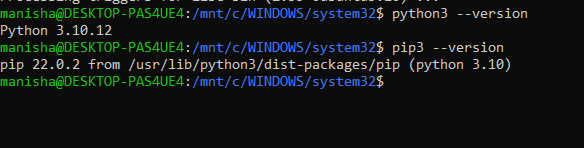
**sudo apt install -y python3 python3-pip**

****

**Step 5: Verify Python and pip**

python3 --version

pip3 –version



Expected output:

**Python 3.10.x**

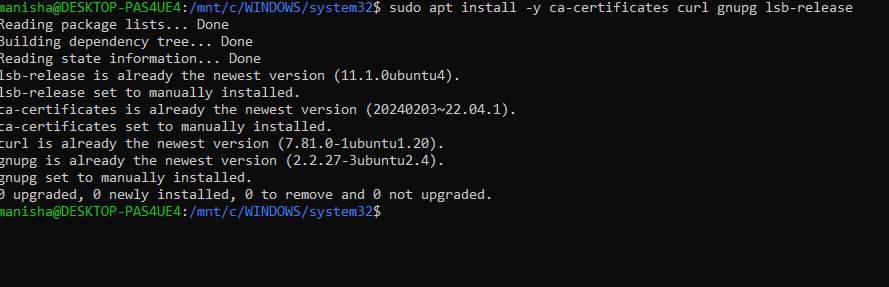
**pip 22.x.x**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Part 3: Install Docker on Ubuntu 22.04 (WSL2)**

**Step 6: Install Docker dependencies**

**sudo apt install -y ca-certificates curl gnupg lsb-release**

****

**Step 7: Add Docker’s official GPG key**

**sudo mkdir -p /etc/apt/keyrings**

**curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \**

**sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg**

**Step 8: Set up Docker's APT repository**

**echo \**

**"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \**

**https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | \**

**sudo tee /etc/apt/sources.list.d/docker.list > /dev/null**

**Step 9: Update repo list and install Docker Engine**

bash

CopyEdit

sudo apt update

sudo apt install -y docker-ce docker-ce-cli containerd.io

**Step 10: Enable Docker inside WSL2**

Allow Docker to run without sudo (optional):

bash

CopyEdit

sudo usermod -aG docker $USER

newgrp docker

Verify Docker:

bash

CopyEdit

docker --version

Expected:

nginx

CopyEdit

Docker version 24.x.x, build xxxxx

🎉 Output:

* Installed **Git**, **Python**, **pip**, **Docker**, **curl**, and more.
* Your Ubuntu WSL2 shell is now ready for DevOps workflows.

**🔸 Why (Practical Thinking & Reasoning)**

**Why update apt before installing?**

✔️ To get the latest and most secure versions of software.  
Outdated packages could contain bugs or missing features.

**Why use apt install -y?**

✔️ The -y flag auto-confirms prompts so the install runs without interruptions — ideal for scripting and automation.

**Why install Git?**

✔️ Git is used everywhere — CI/CD, version control, DevOps pipelines, GitHub Actions, GitLab, Bitbucket, etc.

**Why Python 3 and pip?**

✔️ Python is the go-to language for DevOps scripting, automation, infrastructure as code (IaC), and API interactions.

**Why curl and wget?**

✔️ You’ll often download files or call APIs in scripts. These tools are essential for CLI-based interactions with services.

**Why install Docker manually if Docker Desktop exists?**

✔️ Docker Desktop is optional — but on **WSL2**, you can run Docker natively using the Docker Engine.  
It's lighter, faster, and more realistic for server-side workflows.

**Why add yourself to the docker group?**

✔️ So you don’t have to type sudo before every docker command. Makes working smoother and avoids permission issues.

**Why install htop and net-tools?**

✔️ htop gives a nice system monitor in the terminal.  
✔️ net-tools provides ifconfig, netstat, etc., which are useful in networking and cloud debugging.

**Can I use this same setup in cloud VMs (AWS, Azure)?**

✔️ Yes — these exact steps apply to **Ubuntu cloud instances**, meaning your WSL2 environment mirrors what’s used in production.