DevOps Tools Installation

Git: Git is a distributed version control system, meaning it allows developers to track changes in their code over time and collaborate effectively on projects

Installation Steps:

✅ **Step-by-Step: Install Git on Windows**

1. **Download Git for Windows**
    - Go to: https://git-scm.com/download/win
    - The download will start automatically.
2. **Run the Installer**
    - Double-click the downloaded .exe file to launch the installer.
3. **Follow the Setup Wizard**
    - Click **Next** on the welcome screen.
    - Choose the installation path (default is fine for most users).
    - Keep clicking **Next**, accepting default options, unless you need something specific.
4. **Important Installer Options**
    - **Choose the default editor used by Git**: Pick an editor like Notepad++ or VS Code, or leave it as Vim if unsure.
    - **Adjusting your PATH environment**: Choose **"Git from the command line and also from 3rd-party software"**.
    - **Choosing HTTPS transport backend**: Pick **"Use the OpenSSL library"**.
    - Other options can be left as default unless you know what you're changing.
5. **Install**
    - Click **Install** and wait for it to complete.
6. **Finish Setup**
    - After installation, click **Finish**.
    - You can choose to launch **Git Bash** right away.

✅ **Optional: Verify Git Installation**

Open **Git Bash** or **Command Prompt** and type:

git --version

You should see the installed version of Git.

Docker Desktop: Docker is a platform that enables developers to package and run applications in lightweight, isolated containers. These containers include everything needed for the application to run, including code, runtime, system tools, and libraries, allowing for consistent execution across different environments

---

🚀 Install Docker Desktop on Windows

1. Check System Requirements

- **Windows 10 (Pro, Enterprise, or Education) or Windows 11** (with WSL2 enabled)
- **Windows Home users:** Requires WSL 2 backend
- At least **4GB RAM**

2. Download Docker Desktop

- Go to: https://www.docker.com/products/docker-desktop/
- Click **Download for Windows**

3. Install Docker

1. Run the downloaded .exe file.
2. Select **"Use WSL 2 instead of Hyper-V"** (if you have WSL2 installed).
3. Click **Next** → **Install**.

4. Complete Installation

- Once installed, **restart your computer**.
- Open **Docker Desktop** and follow the setup instructions.

5. Verify Installation

Open **PowerShell** or **Command Prompt** and run:

```
docker --version
```

It should display the installed Docker version.

🍏 Install Docker Desktop on macOS

1. Check System Requirements

- macOS **Monterey (12), Ventura (13), or Sonoma (14)**
- **Apple Silicon (M1/M2) or Intel Mac**
- At least **4GB RAM**

2. Download Docker Desktop

- Go to: https://www.docker.com/products/docker-desktop/
- Click **Download for Mac** (choose **Apple Chip** for M1/M2 or **Intel Chip** if applicable).

3. Install Docker

1. Open the downloaded .dmg file.
2. Drag **Docker.app** to the **Applications** folder.

4. Run Docker Desktop

- Open **Docker** from Applications.
- Follow the onboarding steps.
- Grant permissions if prompted.

5. Verify Installation

Open **Terminal** and run:

docker --version

If Docker is running, you should see the installed version.

Kubectl: kubectl is the primary command-line tool used to interact with and manage Kubernetes clusters. It allows you to deploy applications, manage resources, and perform various operations on your Kubernetes cluster from the command line.

Install kubectl on Windows

Method 1: Using PowerShell

1. **Open PowerShell as Administrator**
2. Run this command to download the latest version:

curl -LO "https://dl.k8s.io/release/$(curl -s https://dl.k8s.io/release/stable.txt)/bin/windows/amd64/kubectl.exe"

3. **Move the file to a folder in your PATH**
   For example, move it to C:\Program Files\kubectl\ and then:

- Open **System Properties** → **Environment Variables**
- Add that folder to the **Path** variable

4. **Verify the installation**

kubectl version --client

Install kubectl on macOS

✅ **Method 1: Using Homebrew (Recommended)**

1. **Open Terminal**
2. Run:

brew install kubectl

3. **Verify the installation**

kubectl version --client

✅ **Alternative (Manual Download for macOS)**

1. **Download binary**

curl -LO "https://dl.k8s.io/release/$(curl -s https://dl.k8s.io/release/stable.txt)/bin/darwin/amd64/kubectl"

2. **Make it executable and move to /usr/local/bin**

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

3. **Verify**

kubectl version --client

Terraform: The Terraform CLI is a command-line interface tool used to manage Infrastructure as Code (IaC). It allows you to define, provision, and manage infrastructure resources using the HashiCorp Configuration Language (HCL). The CLI provides commands to interact with your infrastructure configuration, including initializing the environment, planning changes, applying them, and destroying resources.

🪟 **Install Terraform CLI on Windows**

✅ **Method 1: Using Windows Package Manager (winget)**

1. **Open PowerShell as Administrator**
2. Run:

winget install --id HashiCorp.Terraform -e

📝 *If winget is not available, use the manual method below.*

✅ **Method 2: Manual Installation**

1. Go to: https://developer.hashicorp.com/terraform/downloads
2. Select **Windows** → Click to download the ZIP.

3. Extract the ZIP file (it contains terraform.exe).
4. Move terraform.exe to a folder (e.g., C:\Terraform\).
5. Add that folder to the system **Path**:
    - Control Panel → System → Advanced → Environment Variables → Edit "Path"
6. **Verify installation**:

terraform -version

🍏 **Install Terraform CLI on macOS**

✅ **Method 1: Using Homebrew (Recommended)**

1. **Open Terminal**
2. Run:

brew tap hashicorp/tap

brew install hashicorp/tap/terraform

3. **Verify installation**:

terraform -version

✅ **Method 2: Manual Installation**

1. Download Terraform for macOS from:
   https://developer.hashicorp.com/terraform/downloads
2. Unzip the downloaded file.
3. Move the terraform binary to /usr/local/bin:

sudo mv terraform /usr/local/bin/

4. **Verify**:

terraform -version

AWS CLI: The AWS CLI (Command Line Interface) is a free, open-source tool that lets you interact with Amazon Web Services (AWS) services using commands in your terminal

---

🪟 **Install AWS CLI on Windows**

✅ **Step-by-Step (Installer Method)**

1. **Download the AWS CLI Installer**
   Go to:

https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html
Or direct download:
https://awscli.amazonaws.com/AWSCLIV2.msi

2. **Run the Installer**
    - Double-click the AWSCLIV2.msi file
    - Follow the installation wizard
3. **Verify Installation**
   Open **Command Prompt** or **PowerShell**, then run:

```
aws --version
```

You should see something like: aws-cli/2.x.x

🍏 **Install AWS CLI on macOS**

✅ **Method 1: Using Homebrew (Recommended)**

1. **Open Terminal**
2. Run:

```
brew install awscli
```

3. **Verify Installation**

```
aws --version
```

✅ **Method 2: Manual Install (No Homebrew)**

1. **Download the Installer**

```
curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
```

2. **Install the Package**

```
sudo installer -pkg AWSCLIV2.pkg -target /
```

3. **Verify Installation**

```
aws --version
```

Virtualization: Setup the Virtual Box and Inside the Box setup the Ubuntu os : Virtualization is the creation of a virtual, simulated environment from a single physical machine. This allows multiple operating systems, applications, and other resources to share the same hardware, effectively creating virtual machines (VMs) that act as independent computers.

✅ **What You'll Need:**

- VirtualBox (Free VM software)
- Ubuntu ISO file (from https://ubuntu.com/download/desktop)

Steps to Install Ubuntu on VirtualBox – Windows

1. Install VirtualBox

- Download VirtualBox for Windows: https://www.virtualbox.org/wiki/Downloads
- Run the installer → Click **Next** → Keep defaults → Install

2. Download Ubuntu ISO

- Go to: https://ubuntu.com/download/desktop
- Choose **Ubuntu 22.04 LTS** (or latest) and download the .iso file

3. Create a New Virtual Machine

1. Open **VirtualBox**
2. Click **New**
    - Name: Ubuntu
    - Type: Linux
    - Version: Ubuntu (64-bit)
3. Click **Next**

4. Allocate Memory (RAM)

- Set at least **2GB (2048MB)**; **4GB+** recommended

5. Create a Virtual Hard Disk

- Select **Create a virtual hard disk now**
- Type: **VDI**
- Storage: **Dynamically allocated**
- Size: **25GB+**

6. Load the Ubuntu ISO

1. Select your VM → Click **Settings**
2. Go to **Storage**
    - Under “Controller: IDE” click **Empty**
    - On the right, click the disc icon → **Choose a disk file**
    - Select the Ubuntu .iso file

7. Start the VM

- Click **Start** and it will boot into the Ubuntu installer

8. Install Ubuntu

- Choose language → Click **Install Ubuntu**
- Follow setup (Normal Installation, use entire virtual disk, etc.)
- Set username/password → Continue

🍏 **Steps to Install Ubuntu on VirtualBox – macOS**

Almost identical steps, just with macOS as the host OS

1. Install VirtualBox

- Download VirtualBox for macOS: https://www.virtualbox.org/wiki/Downloads
- Open .dmg → Install
- Grant permission in **System Preferences > Security & Privacy** if blocked

2. Download Ubuntu ISO

- Same as Windows (see above)

3. Create the Virtual Machine

- Open VirtualBox
- Click **New** → Name: Ubuntu, Type: Linux, Version: Ubuntu (64-bit)
- Allocate memory
- Create virtual hard disk (VDI, 25GB+, dynamically allocated)

4. Mount Ubuntu ISO

- Go to **Settings > Storage**
- Under "Empty" → Choose a disk → Select the Ubuntu .iso

5. Start the VM and Install Ubuntu

- Click **Start**
- Ubuntu installer will boot → Follow prompts to install
- After install, restart the VM

✅ **After Installation**

1. **Eject the ISO**
    - Go to **Devices > Optical Drives > Remove disk from virtual drive**
2. **Install Guest Additions** (optional but improves performance)

- With VM running, go to:
  Devices > Insert Guest Additions CD image
- Follow on-screen instructions inside Ubuntu

---

choco install awscli

choco install terraform

choco install kubectl

choco install vscode

choco install docker-desktop

choco install git