**Installing VS Code Extension Windows**

**Connecting VS Code to WSL for Windows Development**

This lesson guides Windows users through installing and using the official Microsoft WSL extension for Visual Studio Code (VS Code). Integrating VS Code with your Windows Subsystem for Linux (WSL) environment, typically Ubuntu for this context, provides a seamless development experience. This setup allows you to leverage Linux command-line tools and environments directly within VS Code, which is the recommended configuration for subsequent development tasks.

**Installing the VS Code WSL Extension**

There are two primary methods to install the necessary extension.

**Method 1: Automatic Installation Prompt**

Often, when you open VS Code after installing WSL, the editor will detect its presence.

1. Look for a notification prompt in the bottom-right corner of the VS Code window.
2. The prompt may read: "You have Windows Subsystem for Linux (WSL) installed on your system. Do you want to install the recommended 'WSL' extension from Microsoft for it?"
3. Click the "Install" button on this prompt.

**Method 2: Manual Installation via Extensions View**

If the automatic prompt doesn't appear, or if you dismissed it, follow these steps:

1. Open the Extensions view by clicking the square icon on the left-hand sidebar or pressing Ctrl+Shift+X.
2. In the search bar at the top of the Extensions view, type wsl.
3. Locate the extension named "WSL".
4. **Crucially, verify the publisher.** Ensure the extension is published by "Microsoft" and displays a blue checkmark next to the publisher name (microsoft.com). This confirms you are installing the official, trusted extension.
5. Click the "Install" button for the verified Microsoft WSL extension.

**Understanding the Core Concepts**

Before proceeding, let's clarify the key components involved:

* **WSL (Windows Subsystem for Linux):** A Windows feature enabling you to run native Linux distributions (like Ubuntu) directly on Windows, without requiring traditional virtual machines.
* **VS Code:** A versatile source code editor developed by Microsoft.
* **VS Code WSL Extension:** An add-on that bridges VS Code with your WSL environment, enabling deep integration.
* **Integrated Terminal:** A terminal panel built into VS Code. On Windows, it typically defaults to PowerShell or Command Prompt.
* **Remote Development:** The core feature provided by the WSL extension. It allows VS Code's backend processes (file handling, terminal execution, debugging) to run *inside* your WSL Linux environment, while the user interface runs on Windows. This ensures commands and tools operate natively within Linux.

**Using the Integrated Terminal with WSL**

Once the extension is installed, you can access your WSL terminal within VS Code.

1. Open a new terminal using the menu (Terminal -> New Terminal) or the shortcut `Ctrl+Shift+``` (backtick).
2. By default, this might open a Windows shell like PowerShell (indicated by a prompt like PS C:\Users\YourUser>).
3. To switch to your WSL distribution's terminal *within this specific panel*, click the dropdown arrow next to the + icon in the terminal panel's top-right corner.
4. Select your WSL distribution, typically named "Ubuntu (WSL)" or similar.
5. The terminal panel will reload, presenting your Linux shell prompt (e.g., username@hostname:/mnt/c/Users/YourUser$). Note that the starting directory might be a mounted Windows path (/mnt/c/...).
6. You can navigate to your Linux home directory by typing cd and pressing Enter. The prompt will change to reflect your Linux home (e.g., username@hostname:~$).

While this method works for running quick Linux commands, it doesn't connect your entire VS Code workspace to WSL. Files opened in the editor might still be treated as Windows files.

**Connecting VS Code Directly to WSL (Recommended)**

For a fully integrated experience where VS Code operates entirely within the context of your WSL environment, use the remote connection feature. This is the strongly recommended approach.

1. Locate the green button with a "><" symbol in the bottom-left corner of the VS Code window (Status Bar). Hovering over it might show "Open a Remote Window".
2. Click this green button. A command palette dropdown will appear at the top.
3. Select "Connect to WSL" from the list. (You might see other options like "Connect to WSL using Distro..." - the general "Connect to WSL" is usually sufficient).
4. VS Code will reload. This might take a few moments as it establishes the connection and installs necessary components within WSL.
5. **Verification:** Once connected, the green button in the bottom-left corner will now clearly display "WSL: Ubuntu" (or the name of your specific WSL distribution). This is your primary indicator that VS Code is running in the WSL context.

**Working Effectively in the WSL-Connected Environment**

When VS Code is connected to WSL (indicated by "WSL: Ubuntu" in the status bar):

* **Default Terminal:** Opening a new integrated terminal (Ctrl+Shift+```) will now \*automatically\* launch your default WSL shell (e.g., Bash in Ubuntu), typically starting in your Linux home directory (~). The prompt will look like username@hostname:~$`.
* **File System:** Opening files and folders (File > Open Folder...) will browse the Linux file system within your WSL distribution, not your Windows drives directly (though Windows drives are usually accessible via /mnt/c, /mnt/d, etc.).
* **Persistence:** If you close VS Code while connected to WSL, it will attempt to automatically reconnect the next time you open it.
* **Disconnecting:** To return VS Code to operating locally on Windows, click the green "WSL: Ubuntu" status bar indicator and select "Close Remote Connection". VS Code will reload in the local context.
* **Linux Commands:** All commands run in the integrated terminal will execute within the Linux environment. Tools like git, package managers (apt, npm, pip), compilers, and interpreters will be the versions installed within your WSL distribution.

**Course Recommendation: Always Use the WSL Connection**

For all subsequent development work in this course context, **ensure your VS Code is connected to WSL**.

1. Always check for the "WSL: Ubuntu" (or similar) indicator in the bottom-left status bar before starting work.
2. If VS Code opens in the local Windows context (no WSL indicator), use the green "><" button to "Connect to WSL".
3. When presented with command-line instructions that differ for Linux and Windows, **always use the Linux versions** within the VS Code integrated terminal while connected to WSL.

This setup provides a consistent, powerful, and Linux-native development environment directly on your Windows machine, aligning with common deployment targets and toolchains used in web3 and broader software development.