🧠How to Set Up and Run MCP on Windows

🖕 Step 1: Enable and Set Up WSL (Windows Subsystem for Linux)

1. Open **PowerShell**.
2. Run the following command to install WSL and Ubuntu:

```
wsl --install
```

3. Approve the admin-access prompt when asked. Wait until installation completes and Ubuntu is downloaded.
4. Once finished, start Linux for the first time:

```
wsl
```

   Create your **Linux username** and **password** when prompted.
5. Use these basic commands to explore:
6. `pwd` → shows your current working directory
7. `ls` → lists files and folders
8. `cd` → changes directories (for example `cd ~` returns to your home folder)

💡 **Tip:** You now have two separate home directories — one in Windows and another inside Linux. Remember which environment you're working in!

😾Step 2: Install `uv` and Clone Your Repository

1. In PowerShell, start your Ubuntu environment (use `ubuntu`, not `wsl`, so you land in your Linux home directory):

```
ubuntu
```

2. Follow the Linux install guide for **uv** at:
   🔔https://docs.astral.sh/uv/getting-started/installation/
   or run directly:

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

3. After installation completes, exit and reopen Ubuntu so your updated `PATH` is loaded:

```
exit
ubuntu
```

4. Confirm you're in your Linux home folder:

```
pwd
cd ~
ls
```

5. Create a workspace for projects:

```
mkdir projects
cd projects
```

6. Clone the repository you want to work on:

```
git clone <repo-url>
```

7. Move into your new project directory (for example):

```
cd agents
```

8. Sync all project dependencies:

```
uv sync
```

🤔Step 3: Re-opening Later or After Reboot

If you close PowerShell and want to return later:

- Check Ubuntu's WSL status:

```
wsl -d Ubuntu
```

- Launch Ubuntu again:

```
wsl -d Ubuntu
```

🎉Step 4: Open Your Project in Cursor Editor

From PowerShell, navigate to your project directory, then run:

```
cursor .
```

This opens your current Linux project folder in the Cursor editor.

🐱Step 5 – Set Up Cursor for WSL Development

🤩Open Cursor Normally

Launch **Cursor** from your Windows Start menu (not inside Ubuntu).

🤪Install the WSL Extension

- Open the **Extensions Panel** → `View → Extensions` or press `Ctrl + Shift + X`.
- Search for **"WSL by Anysphere"** (the developers of Cursor).
- Click **Install** to enable remote Linux integration.

🤫Start a WSL-Connected Session

- Press `Ctrl + Shift + P` to open the Command Palette.
- Type and select **"Remote-WSL: New Window"**.
- Cursor will launch a new window that runs directly inside your Ubuntu environment.

👹Open Your Project

- Choose **File → Open Folder** (or *Open Project*).
- Navigate to:

  ```
  /home/<your-username>/projects/agents
  ```

- Click **Select Folder** or **Open**.
  (It may take a minute the first time—Cursor is setting up the remote bridge.)

😺Step 6 – Install Python and Jupyter Support in WSL

Once your Cursor window is connected to Ubuntu:

1. Re-open the **Extensions Panel** (`Ctrl + Shift + X`).
2. Search for and install these extensions **inside WSL-Ubuntu** if they're not already present:
3. 🐍**Python** (by *Microsoft*)
4. 📓**Jupyter** (by *Microsoft*)
5. When prompted, click **"Install in WSL-Ubuntu"** so the tools run natively within Linux.

😺Step 7 – Verify Your Environment

Inside Cursor's integrated terminal (`Ctrl + ~`):

```
which python
uv --version
```

🔗You should see Linux-style paths (`/home/...`) and valid versions displayed.

Then activate your project's environment and run a simple test:

```
uv venv
source .venv/bin/activate
python --version
```

😥Step 8 – Run and Develop

Your project is now fully ready inside WSL with Cursor. You can execute, debug, and install new Python packages seamlessly within this Linux setup while enjoying Windows-native UI performance.

😺Step 9 – Troubleshooting Common Issues

Even with WSL and Cursor correctly configured, a few common hiccups can occur. Here's how to resolve them quickly 🔑

👉Issue 1: “Ubuntu is not recognized” in PowerShell

Cause: WSL wasn’t added to PATH or Ubuntu didn’t register after reboot.
Fix:

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

If Ubuntu appears in the list, start it manually:

```
wsl -d Ubuntu
```

If you get an error that `wsl` itself isn’t recognized, re-enable it:

```
wsl --install
```

Then restart your computer.

👉Issue 2: Cursor fails to connect to WSL

Cause: Cursor’s WSL extension can’t find its launcher binary.
Fix:
1. Ensure the **WSL by Anysphere** extension is installed.
2. If you see an ENOENT error referencing `cursor/resources/app/bin/code`, create a shim: - Navigate to:
`C:\Program Files\cursor\resources\app\bin\` - Create a file named `code.cmd` containing:

```
@echo off
"C:\Program Files\cursor\cursor.exe" %*
```

- Reopen Cursor → `Ctrl + Shift + P` → **Remote-WSL: New Window** → select Ubuntu.

👉Issue 3: Playwright install hangs asking for password

Cause: Cursor’s integrated terminal doesn’t support `sudo` prompts.
Fix:
Run this in your **Ubuntu terminal (outside Cursor):**

```
uv run playwright install --with-deps chromium
```

Enter your Linux password when prompted. After that, Cursor can run Playwright commands normally.

👉Issue 4: OpenAI Tracing works in Windows but not in WSL

Cause: Environment variables aren't loaded or `.env` path differs.
Fix:
1. Export your variables inside WSL:

```
export OPENAI_API_KEY="sk-xxxxx"
export OPENAI_TRACE="1"
```

2. Make it permanent by adding them to `~/.bashrc`.
3. Ensure your `.env` file is stored under `/home/<user>/projects/...` (not `/mnt/c/...`).

👉Issue 5: File Permissions or Sandbox Errors in MCP

Cause: The sandbox directory doesn't exist or has restricted access.
Fix:

```
mkdir -p sandbox
chmod -R 755 sandbox
```

Then retry running the MCP server.

🖕 Quick Validation Checklist

Run the following inside Ubuntu to confirm everything is working:

```
node -v
npx -v
uv --version
python --version
git --version
```

All versions should return successfully without errors.