Module 02

Preparing Your Laptop

🦞 WSL2, Node.js, and Everything You Need

OpenClaw Course

🧭 Navigation Chart

By the end of this module, you will be able to:

1. **Verify** your laptop meets the minimum requirements
2. **Explain** why WSL2 is required and what it does
3. **Install** and configure WSL2 with Ubuntu on Windows 10
4. **Enable** systemd inside WSL2 (required for the 🦞 OpenClaw daemon)
5. **Install** Node.js 22+ inside your WSL2 environment
6. **Verify** your entire environment is ready for 🦞 OpenClaw
7. **Configure** your laptop for 24/7 operation
8. **Optionally** set up Tailscale for secure remote access

📜 Ship's Logbook (Part 1)

Term	Definition
WSL2	Windows Subsystem for Linux v2 -- run a full Linux OS inside Windows
Ubuntu	A beginner-friendly Linux distribution we install inside WSL2
Terminal	Text-based interface for typing commands
systemd	Linux service manager -- OpenClaw's daemon needs it to start automatically
Daemon	A program that runs continuously in the background
Node.js	JavaScript runtime that 🦞 OpenClaw is built on (version 22+ required)

📜 Ship's Logbook (Part 2)

Term	Definition
npm	Node Package Manager -- installs JavaScript packages
nvm	Node Version Manager -- install and switch between Node.js versions
PATH	List of directories your computer checks when you type a command
Tailscale	Secure networking tool for private remote access between your devices
Loopback	Network address (127.0.0.1) that only your own computer can reach

Check Your Hardware: RAM

Open **Task Manager** (`Ctrl + Shift + Esc`) > **Performance** tab > **Memory**.

RAM	Verdict
4 GB or less	Not enough -- will struggle
8 GB	Minimum -- works but slow when multitasking
16 GB	Excellent -- plenty for 🦞 OpenClaw
32 GB+	Overkill, but great for local models later

What uses the RAM: WSL2 uses 1-4 GB, Node.js a few hundred MB. The rest is yours.

Check Your Hardware: Disk Space

Open **File Explorer** > **This PC** > Check your C: drive.

Free Space	Verdict
Less than 10 GB	Not enough
10-20 GB	Tight but workable
20-50 GB	Comfortable
50 GB+	Plenty of room

Total footprint: WSL2 + Ubuntu (~5 GB) + Node.js (~200 MB) + 🦞 OpenClaw (~500 MB) + agent data over time (1-10 GB). About **20 GB** covers everything comfortably.

Check Your Hardware: Windows Version

Press `Win + R`, type `winver`, press Enter.

Version	WSL2 Support
Windows 10 v1903+ (Build 18362)	Supported
Windows 10 v2004+	Simplified install (`wsl --install`)
Windows 10 22H2	Current -- fully supported
Windows 11	Fully supported

> ***If your version is older than 1903:*** *Update Windows first. Go to* ***Settings > Update & Security > Windows Update*** *and install all available updates.*

Why WSL2 Is Required

🦞 **OpenClaw is built for Linux/macOS.** The ⛵ gateway, daemon, 🐚 shell commands, and file permissions all assume a Unix-like OS.

> *"OpenClaw on Windows is recommended via WSL2 (Ubuntu recommended)." -- Official docs*

What WSL2 Actually Is

- Runs a **real Linux kernel** inside Windows
- Not an emulator, not a traditional VM
- Runs **alongside** Windows, not instead of it
- Access Windows files from Linux, and vice versa

Apartment analogy: Your laptop is a building with two tenants -- Windows and Linux. They share the hardware but have their own separate units.

Step 1: Open PowerShell as Administrator

1. Click the **Start menu**
2. Type `PowerShell`
3. **Right-click** on Windows PowerShell
4. Click **Run as administrator**
5. Click **Yes** when prompted

You should see:

```
PS C:\WINDOWS\system32>
```

The blinking cursor is waiting for your input.

Step 1: Install WSL with Ubuntu

Type this command exactly and press Enter:

```
wsl --install
```

What this does:

- Enables the WSL feature in Windows
- Downloads the WSL2 Linux kernel
- Downloads and installs Ubuntu
- Sets WSL2 as the default version

You should see messages about installing Virtual Machine Platform, WSL, and Ubuntu.

> ***You MUST restart your computer after this step.*** *Use Restart, not Shut Down.*

Step 1: Complete Ubuntu Setup

After restarting, open the **Ubuntu** app from the Start menu.

- Type a **lowercase** username (e.g., `openclaw`)
- Set a password you will remember
- **Nothing appears on screen when typing the password** -- this is normal!

When you see `openclaw@YOURPC:~$` — that is your 🐚 **Linux prompt**. You are inside Ubuntu.

Step 2: Enable systemd (Edit Config)

🦞 OpenClaw's daemon needs systemd. It is **not enabled by default** in WSL2.

Edit the WSL config file

```
sudo nano /etc/wsl.conf
```

Add these two lines

```
[boot]
systemd=true
```

Save and exit

- `Ctrl + O` to save, then `Enter`
- `Ctrl + X` to exit

Step 2: Enable systemd (Restart and Verify)

Restart WSL from PowerShell

```
wsl --shutdown
```

Wait 5 seconds, then reopen Ubuntu and verify:

```
systemctl is-system-running
```

Expected: `running` (or `degraded` is OK)

Step 3a: Install nvm

We use nvm (Node Version Manager) -- the recommended way.

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
```

Then activate it:

```
source ~/.bashrc
nvm --version          # Should show: 0.40.1
```

> *If you see `nvm: command not found` -- close and reopen your Ubuntu terminal, then try again.*

Step 3b: Install Node.js 22

```
nvm install 22
nvm alias default 22
```

Verify both

```
node --version         # Should show: v22.x.x
npm --version          # Should show: 10.x.x
```

As long as `node` starts with **v22**, you're fueled up.

Steps 4 and 5: Update Packages and Install Tools

Update Ubuntu packages (2-5 minutes)

```
sudo apt update && sudo apt upgrade -y
```

Install essential tools

```
sudo apt install -y git curl wget build-essential
```

Tool	Purpose
`git`	Version control for your 🪸 workspace
`curl` / `wget`	Download tools used by the installer
`build-essential`	Compilation tools for npm packages

Step 6: Configure Power Settings

For 24/7 operation, your laptop must stay awake.

Disable sleep mode

1. Settings > System > Power & sleep
2. Set "When plugged in, PC goes to sleep after" → Never

Disable hibernate (optional)

```
powercfg /hibernate off
```

Display can turn off -- that's fine

- "Turn off screen after" → 5 minutes (display off, computer still runs)

Verify

Lock screen (`Win + L`), wait a minute, unlock, run `uptime` -- should show continuous uptime

Step 7: Understand the File System Boundary

There are **two separate file systems**. Knowing which you are in matters.

Location	Path Style	Example
Windows	`C:\Users\...`	`C:\Users\GC\Documents\`
Linux (WSL2)	`/home/username/...`	`/home/openclaw/`

The Golden Rule

> 🦞 *OpenClaw's data should live in the Linux file system, NOT the Windows file system.*

Files in `/home/openclaw/` are fast. Files through `/mnt/c/` are **significantly slower** (every operation crosses the WSL2 boundary). Run `pwd` to check which side you're on.

Step 8: Tailscale (Optional)

Tailscale lets you access 🦞 OpenClaw from your phone or another computer. **Skip if** you'll only use this laptop.

Setup:

1. Create account at **tailscale.com** (free plan works)
2. Install on **Windows** (download from website)
3. Install inside **WSL2** and authorize:

```
curl -fsSL https://tailscale.com/install.sh | sh
sudo tailscale up
```

What access looks like

- From this laptop: `http://127.0.0.1:18789/`
- From your phone (Tailscale): `http://100.x.x.x:18789/`

Step 8.5: Disk Encryption (Recommended)

If your device stores API keys and personal data, encrypt the disk.

Platform	How
Windows Pro/Enterprise	BitLocker
Windows Home	Device Encryption (requires Microsoft account)
Linux (native, not WSL)	LUKS -- enable during OS installation
Mac	FileVault

> *If that laptop walks off, encrypted data is unreadable without your password.*

Step 9: Dedicated User Account (Optional)

A security measure from Module 01's **Principle 2: Least Privilege**.

```
sudo adduser myagent
su - myagent
```

- Set a password, skip optional fields
- 🦞 OpenClaw installs under this account only
- If something goes wrong, damage is limited to this account
- Type `exit` to return to your regular account

Skip this if you want simplicity or plan to use Docker sandboxing (Module 10).

🚩 Shoals and Sandbars

Problem	Cause	Fix
Error `0x80370102`	Virtualization disabled in BIOS	Enable "Intel VT" or "AMD-V" in BIOS settings
Error `0x80004002`	Windows features not enabled	Run `dism.exe` commands to enable WSL and VM Platform
`node: command not found`	nvm not loaded	Run `source ~/.bashrc` or reopen terminal
Password not accepted	Using Windows password	Use the **Linux** password from Ubuntu setup
WSL2 uses too much RAM	Default is half your RAM	Create `.wslconfig` with `memory=4GB`
Ubuntu terminal is slow	Windows Defender scanning	Exclude WSL2 directory from Defender

The Verification Checklist

Run each command in your 🐚 Ubuntu terminal:

Check	Command	Expected
WSL2 version	`wsl.exe -l -v`	Ubuntu, VERSION 2
systemd	`systemctl is-system-running`	`running` or `degraded`
Node.js	`node --version`	`v22.x.x`
npm	`npm --version`	`10.x.x`
Git	`git --version`	`git version 2.x.x`
curl	`curl --version`	Any version output
File system	`pwd`	`/home/openclaw` (not `/mnt/c/`)

All seven must pass before moving to Module 03.

⚙ Hands on Deck

Complete this checklist:

- [ ] **RAM:** At least 8 GB (how much? _______ GB)
- [ ] **Disk space:** At least 20 GB free (how much? _______ GB)
- [ ] **Windows version:** 1903+ or Windows 11 (your version: _______)
- [ ] **WSL2:** Running, VERSION 2
- [ ] **Ubuntu username:** Created (username: _______)
- [ ] **systemd:** Enabled and running
- [ ] **Node.js:** v22.x.x installed
- [ ] **npm:** 10.x.x installed
- [ ] **Git:** Installed
- [ ] **Power settings:** Sleep disabled (if running 24/7)

💎 Treasure Chest

1. 🦞 **OpenClaw requires WSL2** -- it runs inside Linux, not directly on Windows
2. **systemd must be enabled** -- add `[boot] systemd=true` to `/etc/wsl.conf`
3. **Node.js 22+ is required** -- install with nvm for easy version management
4. **Keep 🦞 OpenClaw files in the Linux file system** -- `/home/username/`, not `/mnt/c/`
5. **Disable sleep** for 24/7 operation -- Windows pauses WSL2 when it sleeps
6. **Tailscale is optional** but useful for remote access
7. **A dedicated user account** adds security isolation
8. **Most errors** come from virtualization disabled in BIOS or nvm not sourced

🌊 Next Port of Call

Module 03: Installing 🦞 OpenClaw

> *Your laptop is ready. Time to run the installer, walk through the onboarding wizard, and bring your agent to life.*