NemoClaw

Installation Guide for Windows

A complete step-by-step guide to installing NVIDIA NemoClaw on Windows using WSL2, Docker Desktop, and OpenShell

Platform:	Windows 10/11 with WSL2
Tools:	Docker Desktop, Node.js 22, OpenShell, NemoClaw
Model:	Nemotron 3 Super 120B (NVIDIA Cloud API)
Time:	45-90 minutes (including downloads)
Date:	March 2026

What is NemoClaw?

NemoClaw is NVIDIA's open-source security plugin that wraps OpenClaw AI agents inside a sandboxed environment (OpenShell). It enforces strict network policies, blocks unauthorized connections, and prevents privilege escalation — making it the safe way to run powerful AI agents on your machine.

What You'll Set Up

- WSL2 (Windows Subsystem for Linux) with Ubuntu
- Docker Desktop with WSL2 integration
- Node.js 22 LTS inside WSL
- NVIDIA OpenShell CLI (sandbox runtime)
- NemoClaw (security + inference layer)

• OpenClaw AI agent (interactive TUI)

Table of Contents

Section 1: Quick Start

For experienced developers who are comfortable with WSL, Docker, and CLI tools. If you're new to any of these, skip to Section 2 for the detailed walkthrough.

1. Enable WSL2 + Ubuntu

```
wsl --install # then restart PC
```

2. Install Docker Desktop

```
Download from docker.com/products/docker-desktop
Enable WSL2 integration for Ubuntu in Settings > Resources
```

3. Install Node.js 22 in WSL

```
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -
sudo apt install -y nodejs
```

4. Install OpenShell

```
curl -LsSf https://raw.githubusercontent.com/NVIDIA/
OpenShell/main/install.sh | OPENSHELL_VERSION=v0.0.10 sh
export PATH="$HOME/.local/bin:$PATH"
echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.bashrc
```

5. Clone NemoClaw (in WSL)

```
cd ~
git clone --config core.autocrlf=false \
https://github.com/NVIDIA/NemoClaw.git
cd NemoClaw
sudo npm install -g .
```

6. Run the CRLF fix script + onboard

```
# Download fix-and-onboard.sh from GitHub repo
chmod +x fix-and-onboard.sh
./fix-and-onboard.sh
```

7. Connect

```
nemoclaw mynemo connect
openclaw tui
```

Section 2: Prerequisites — Install WSL2

Windows Subsystem for Linux (WSL2) lets you run a full Linux environment directly on Windows without a virtual machine. NemoClaw and OpenShell are Linux-native tools, so WSL2 is essential.

Step 2.1: Open PowerShell as Administrator

Press the **Windows key**, type **PowerShell**, right-click it, and select **Run as administrator**.

Step 2.2: Install WSL

```
wsl --install
```

This installs WSL2 with Ubuntu as the default Linux distribution. You will need to **restart your PC** after this completes.

Step 2.3: Set Up Ubuntu

> **Note:** All WSL/Linux commands from this point forward should be run in this Ubuntu terminal, **not** in Windows CMD or PowerShell (unless explicitly stated).

ou'll
your

Step 2.4: Enable PowerShell Script Execution (Windows side)

If you encounter errors about scripts being disabled, run this in your Administrator PowerShell:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Type **Y** when prompted to confirm.

Section 3: Install Docker Desktop

NemoClaw uses Docker containers to create isolated sandboxes for AI agents. Docker Desktop provides the easiest way to run Docker on Windows with WSL2 integration.

Step 3.1: Download and Install

- Go to **https://docker.com/products/docker-desktop**
- Download Docker Desktop for Windows
- Run the installer and follow the prompts
- Restart your PC if prompted

Step 3.2: Enable WSL2 Integration

This is a critical step that many people miss:

- Open **Docker Desktop**
- Click the **gear icon** (Settings) in the top right
- Go to **Resources** → **WSL Integration**
- Toggle **ON** your Ubuntu distribution
- Click **Apply & Restart**

Step 3.3: Verify Docker in WSL

Open your Ubuntu terminal and run:

```
docker --version
docker ps
```

Both commands should succeed. If **docker ps** gives a permission error, run:

> **Important:** Docker Desktop must be running on Windows whenever you use NemoClaw. The whale icon in your system tray should be steady (not animating).

Section 4: Install Node.js 22 in WSL

NemoClaw requires Node.js version 20 or higher. Ubuntu's default repository ships an older version (v18), so we'll install Node.js 22 LTS from NodeSource.

Step 4.1: Add NodeSource Repository and Install

```
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -
sudo apt install -y nodejs
```

Step 4.2: Verify Installation

> ! **Why not v18?** NemoClaw and its dependencies (OpenClaw, cmake-js, octokit, etc.) all require Node.js >= 20. Installing with the default **apt install nodejs** gives you v18.19 which will produce dozens of EBADENGINE warnings and may cause runtime failures.

Step 4.3: Install Additional Tools

We'll need **dos2unix** and **git** for later steps:

```
sudo apt update && sudo apt install -y dos2unix git
```

Section 5: Install OpenShell CLI

OpenShell is NVIDIA's container runtime that NemoClaw uses to create secure, isolated sandboxes. It provides Landlock, seccomp, and network namespace isolation.

Step 5.1: Install OpenShell

In your Ubuntu terminal:

```
curl -LsSf https://raw.githubusercontent.com/NVIDIA/OpenShell/main/install.sh \
| OPENSHELL_VERSION=v0.0.10 sh
```

Step 5.2: Add to PATH

The installer places the binary in **~/.local/bin** which may not be in your PATH. Fix that:

```
export PATH="$HOME/.local/bin:$PATH"
echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.bashrc
```

Step 5.3: Verify

> **Tip:** The first command makes OpenShell available in your current session. The second command (with echo) makes it permanent for all future sessions.

> **Critical:** You must clone NemoClaw inside the WSL filesystem (e.g., ~/NemoClaw), **not** on the Windows filesystem (/mnt/c/...). Files on /mnt/c/ have permission restrictions that prevent dos2unix from working and cause CRLF issues.

Step 6.1: Clone the Repository

```
cd ~
git clone --config core.autocrlf=false \
https://github.com/NVIDIA/NemoClaw.git
```

The **--config core.autocrlf=false** flag ensures Git does not convert line endings to Windows-style CRLF.

Step 6.2: Install Globally

```
cd ~/NemoClaw
sudo npm install -g .
```

This makes the **nemoclaw** command available system-wide in WSL.

Step 6.3: Verify

```
nemoclaw --help
```

Section 7: The CRLF Fix — Automation Script

This is the most important section of this guide. Even when you clone NemoClaw with **core.autocrlf=false**, some files that NemoClaw generates at runtime in temporary directories (**/tmp/nemoclaw-build-*/**) can still contain Windows-style CRLF line endings. This causes the dreaded error:

```
/usr/bin/env: 'bash\r': No such file or directory
```

The **\r** (carriage return) character is invisible but breaks Linux shell scripts. Our solution is a wrapper script that runs a background file watcher to strip CRLF from temp files as NemoClaw creates them.

Step 7.1: Create the Script

Create a file called **fix-and-onboard.sh** in your NemoClaw directory:

```
#!/bin/bash
# fix-and-onboard.sh
# Fixes CRLF line endings in NemoClaw temp build files
# GitHub: github.com/your-repo/nemoclaw-windows-fix

# Run line-ending fixer in background
(
while true; do
for dir in /tmp/nemoclaw-build-*/; do
if [ -d "$dir" ]; then
find "$dir" -type f -exec sed -i 's/\r$//' {} + 2>/dev/null
fi
done
sleep 0.2
done
) &
FIXER_PID=$!

# Run nemoclaw onboard in foreground (accepts interactive input)
nemoclaw onboard

# Clean up background fixer when done
```

How it works: The script spawns a background loop that checks /tmp/nemoclaw-build-*/ every 200ms and strips carriage returns from any files it finds. Meanwhile, NemoClaw runs in the foreground so you can interact with the setup wizard normally. When onboarding completes, the background fixer is automatically killed.

Section 8: Run NemoClaw Onboarding

Step 8.1: Start Onboarding

```
cd ~/NemoClaw
./fix-and-onboard.sh
```

The onboarding wizard has 7 steps. Here's what to expect:

- **[1/7] Preflight checks** — Verifies Docker, OpenShell, and GPU detection. All should show green checkmarks.
- **[2/7] Starting OpenShell gateway** — Downloads and starts the gateway container. The endpoint will be https://127.0.0.1:8080 (don't open this in a browser — it uses mutual TLS).
- **[3/7] Creating sandbox** — Enter a name for your sandbox (e.g., **mynemo**) or press Enter for the default.
- **[4/7] Configuring inference** — You'll need your NVIDIA API key from **build.nvidia.com**. The model will be set to **nvidia/nemotron-3-super-120b-a12b**.
- **[5/7] Building sandbox image** — This builds the Docker image for your sandbox. Thanks to our CRLF fix script, this should now succeed.
- **[6/7] Security policies** — Choose which network policies to apply. Recommended: accept the suggested presets (**pypi** + **npm**) by typing **Y**.
- **[7/7] Complete** — Shows your sandbox summary with connection commands.

Step 8.2: Get Your NVIDIA API Key

Before running onboarding, get your API key:

- Visit **https://build.nvidia.com**
- Sign in or create a free NVIDIA developer account
- Navigate to your API keys section
- Generate a new key and copy it

The onboarding wizard will prompt you to paste this key during Step 4.

Section 9: Connect and Use Your AI Agent

Step 9.1: Connect to Your Sandbox

```
nemoclaw mynemo connect
```

This drops you into a secure Linux shell **inside** the sandbox. Note: this is a bash shell, not a chatbot — you need to launch the AI agent from here.

Step 9.2: Launch the AI Agent

```
openclaw tui
```

This opens the interactive Terminal UI where you can chat with the Nemotron 3 Super model. You can now give it tasks, ask questions, have it write and execute code — all within the secure sandbox.

Step 9.3: Other Useful Commands (inside sandbox)

Command	Description
`openclaw tui`	Interactive AI chat interface
`openclaw status`	Check agent and channel health
`openclaw doctor`	Run health checks and diagnostics
`openclaw dashboard`	Open the web control panel
`openclaw configure`	Setup wizard for channels and keys
`openclaw --help`	Full list of all commands

Step 9.4: Exit the Sandbox

Press **Ctrl+D** or type **exit** to leave the sandbox shell.

Section 10: Starting and Stopping NemoClaw

To Stop NemoClaw

```
# Exit sandbox first (if inside)
exit

# Stop the sandbox
nemoclaw mynemo stop

# To fully shut down the gateway
openshell gateway destroy --name nemoclaw
```

To Start Again Later

```
# Open Ubuntu terminal, then:
export PATH="$HOME/.local/bin:$PATH"
openshell gateway create --name nemoclaw
nemoclaw mynemo connect

# Inside the sandbox:
openclaw tui
```

Quick Reference Card

Action	Command	Where to Run
Enter sandbox	`nemoclaw mynemo connect`	Ubuntu terminal
Exit sandbox	`exit` or `Ctrl+D`	Inside sandbox
Start AI chat	`openclaw tui`	Inside sandbox
Check status	`nemoclaw mynemo status`	Ubuntu terminal
View logs	`nemoclaw mynemo logs --follow`	Ubuntu terminal
Stop sandbox	`nemoclaw mynemo stop`	Ubuntu terminal
Destroy gateway	`openshell gateway destroy --name nemoclaw`	Ubuntu terminal

Section 11: Troubleshooting Common Issues

'sudo' is not recognized (Windows)

You're running a Linux command in Windows CMD/PowerShell. Open an **Administrator PowerShell** and use commands without sudo, or switch to the Ubuntu terminal for Linux commands.

npm scripts disabled (PowerShell ExecutionPolicy)

Run in Administrator PowerShell:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

/usr/bin/env: 'bash\r': No such file or directory

Windows CRLF line endings are breaking Linux scripts. Use the **fix-and-onboard.sh** wrapper script from Section 7. Also ensure you cloned NemoClaw inside WSL (~/NemoClaw), not on /mnt/c/.

Docker is not running (inside WSL)

1. Make sure Docker Desktop is running on Windows.
2. Go to Docker Desktop → Settings → Resources → WSL Integration → Toggle ON Ubuntu → Apply & Restart.
3. If docker ps gives a permission error: `sudo usermod -aG docker $USER` then reopen the terminal.

openshell CLI not found

OpenShell is not in your PATH. Run:

```
export PATH="$HOME/.local/bin:$PATH"
```

Or reinstall: `curl -LsSf https://raw.githubusercontent.com/NVIDIA/OpenShell/main/install.sh | OPENSHELL_VERSION=v0.0.10 sh`

EBADENGINE warnings (Node.js too old)

NemoClaw needs Node.js >= 20. Install v22:

```
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -
sudo apt install -y nodejs
```

Permission denied: mkdir '/usr/local/lib/node_modules'

Use **sudo** for global npm installs in WSL:

```
sudo npm install -g .
```

dos2unix: Operation not permitted

You're trying to convert files on the Windows filesystem (/mnt/c/). Copy files to the Linux filesystem first:
`cp -r /mnt/c/path/to/NemoClaw ~/NemoClaw`

Access to 127.0.0.1 was denied (browser)

This is normal. The OpenShell gateway at https://127.0.0.1:8080 uses mutual TLS (client certificates) and is not meant to be accessed from a web browser. NemoClaw communicates with it internally. Just close the browser tab.

Section 12: GitHub Script — fix-and-onboard.sh

Below is the complete automation script ready to share on GitHub. It handles the CRLF line ending issue that affects NemoClaw when installed via WSL on Windows.

fix-and-onboard.sh

```
#!/bin/bash
# ============================================
# fix-and-onboard.sh
# NemoClaw CRLF Fix + Onboarding Wrapper
# ============================================
#
# Fixes the 'bash\r' error caused by Windows
# CRLF line endings in NemoClaw temp build
# files when running on WSL.
#
# Usage:
# chmod +x fix-and-onboard.sh
# cd ~/NemoClaw
# ./fix-and-onboard.sh
#
# Requirements: sed, nemoclaw (npm install -g .)
# ============================================

set -euo pipefail

echo "[*] Starting CRLF fixer in background..."

# Background process: watch for temp build dirs
# and strip carriage returns every 200ms
(
while true; do
for dir in /tmp/nemoclaw-build-*/; do
if [ -d "$dir" ]; then
find "$dir" -type f \
-exec sed -i 's/\r$//' {} + 2>/dev/null
fi
done
sleep 0.2
done
) &
FIXER_PID=$!

# Ensure cleanup on exit
trap 'kill $FIXER_PID 2>/dev/null' EXIT

echo "[*] CRLF fixer running (PID: $FIXER_PID)"
echo "[*] Starting NemoClaw onboarding..."
```

```
echo ""

# Run onboard in foreground for interactive input
nemoclaw onboard

echo ""
echo "[*] Onboarding complete!"
echo "[*] Run: nemoclaw <sandbox-name> connect"
echo "[*] Then: openclaw tui"
```

GitHub Repository Structure (suggested)

```
nemoclaw-windows-guide/
    README.md
    fix-and-onboard.sh
    LICENSE
```

NemoClaw Installation Guide for Windows | March 2026
References: docs.nvidia.com/nemoclaw | github.com/NVIDIA/NemoClaw | github.com/NVIDIA/OpenShell