

Ruflo

Give Claude Code and Codex a nervous system

100+ AGENTS · SWARM COORDINATION · SELF-LEARNING MEMORY

An agent is a model plus a harness. The model writes; the harness gives it tools, memory, loops, and controls so it can actually work. Ruflo is that harness — the execution layer around Claude Code and Codex that adds specialized agents, coordinated swarms, memory that persists across sessions, and security guardrails.

You don't need to learn every tool or command. After setup, keep using Claude Code exactly as you do now — Ruflo routes tasks, learns from what works, and coordinates agents in the background.

01 Pick your install path

There are two routes with very different surface areas. Choose based on how deep you want to go:

	PLUGIN PATH	CLI PATH — FULL
What you get	Slash commands, a few skills, agent definitions	98 agents, 60+ commands, 30 skills, MCP server, hooks, daemon
Files added	None	.claude/, .claude-flow/, CLAUDE.md, settings
MCP server	Not registered	Registered
Hooks	No	Yes
Best for	Trying one plugin without committing	Production use — everything works as documented

If you're unsure, take the CLI path. It's what the documentation assumes.

02 Install via CLI (recommended)

Works on every platform — Windows included

```
npx ruflo@latest init wizard
```

This interactive wizard runs identically in PowerShell, cmd, macOS Terminal, and Linux. It's the safest default.

macOS / Linux / WSL / Git-Bash only

```
curl -fsSL https://cdn.jsdelivr.net/gh/ruvnet/ruflo@main/scripts/install.sh | bash
```

Or install globally

```
npm install -g ruflo@latest
```

WINDOWS USERS, READ THIS

The curl ... | bash one-liner needs a POSIX shell. If you see a “'bash' is not recognized” error, use the npx ruflo@latest init wizard line instead — it runs the exact same setup flow natively.

03 Register the MCP server

This is what makes Ruflo's tools callable from inside Claude Code. Run it once:

```
claude mcp add ruflo -- npx ruflo@latest mcp start
```

Without this step, tools like memory_store, swarm_init, and agent_spawn won't be available to Claude.

04 Or take the lite plugin path

If you'd rather just sample a plugin's slash commands, paste these into Claude Code one at a time:

```
/plugin marketplace add ruvnet/ruflo
/plugin install ruflo-core@ruflo
/plugin install ruflo-swarm@ruflo
/plugin install ruflo-rag-memory@ruflo
```

Remember: this adds commands and agent definitions only. No MCP server, no hooks, no files in your workspace.

05 Just use Claude Code normally

This is the part people overthink. Once init has run, there's nothing new to memorize — write your prompts the way you always have. The hooks system routes tasks to the right

agents, learns from successful patterns, and coordinates swarms in the background while you work.

06 Try the web UIs (optional)

Multi-model chat with parallel MCP tool calling

<https://flo.ruv.io/>

Plain-English goals to executable agent plans

<https://goal.ruv.io/>

Both are free to try in the browser with no account and no API key. Handy for seeing what the agents actually do before you wire them into your own project.

THAT'S IT — HAPPY BUILDING

Follow [@ai_aryaman](#) on Instagram