EcomLabs

STEP-BY-STEP GUIDE

# 5 GitHub repos for AI agents

The five week 1 repositories with their link, what they do, how they install per their README, their license and what they run on your machine. First, two read-only commands to vet any repo without cloning it.

| | |
|---|---|
| FOR | Claude Code and other coding agents |
| TESTED WITH | GitHub CLI (gh api) · PowerShell 5.1 · Windows 11 |
| UPDATED | October 30, 2026 |
| DOCS | code.claude.com/docs/en/plugins/security |
| ARTICLE | ecomlabs.dev/g/repos-en |
| AUTHOR | Luis Morales · @luismoralesdev |

# Contents


| | |
|---|---|
| Before you start | How to use this list |
| Step 01 | Check license and activity |
| Step 02 | Read what runs on its own |
| Repo 01 | obra/superpowers |
| Repo 02 | thedotmack/claude-mem |
| Repo 03 | addyosmani/agent-skills |
| Repo 04 | pbakaus/impeccable |
| Repo 05 | manaflow-ai/cmux |
| Final | Checklist |

BEFORE YOU START

## How to use this list

These are the five repositories for week 1. The data (license, last push, version and stars) is from October 30, 2026 and comes from the GitHub API. The install commands are the ones each README publishes: we did not install or run any of these projects to write this guide.

The only things we ran are the two read-only commands in steps 01 and 02, with GitHub CLI 2.96 in PowerShell and Git Bash. They work for vetting any repo before you install it.

STEP 01

## Check license and activity

```
gh api repos/obra/superpowers --jq "{license: .license.spdx_id, archived, pushed_at, stars:
.stargazers_count}"
```

Real response:

```
{"archived":false,"license":"MIT","pushed_at":"2026-10-29T23:35:34Z","stars":309369}
```

- `license`: the SPDX identifier. If it says `NOASSERTION`, GitHub could not classify it: read the `LICENSE` file.
- `archived`: if `true`, the owner froze it and no fixes will come.
- `pushed_at`: date of the last push. Months without changes in an agent project is a bad sign.
- `stars`: popularity, not quality or safety.

Swap `obra/superpowers` for the repo you want to check. You need `gh`, signed in.

STEP 02

## Read what runs on its own

A Claude Code plugin can execute code with your permissions, outside the sandbox, through hooks and MCP servers. Read its files without cloning anything:

```
gh api -H "Accept: application/vnd.github.raw" repos/obra/superpowers/contents/hooks/hooks.json
```

The `Accept` header returns the file as is. Look at which event triggers each hook and which command it runs, then open that script. Do the same with `.mcp.json` and the `bin/` folder. Paths in these five:

| REPO | HOOKS | MCP SERVER |
|---|---|---|
| obra/superpowers | `hooks/hooks.json` | None |
| thedotmack/claude-mem | `plugin/hooks/hooks.json` | `plugin/.mcp.json` |
| addyosmani/agent-skills | None registered (opt-in scripts in `hooks/`) | None |
| pbakaus/impeccable | `plugin/hooks/hooks.json` | None |
| manaflow-ai/cmux | Not a plugin: it is an app | Not applicable |

REPO 01

## obra/superpowers

`github.com/obra/superpowers` · MIT license · v7.0.1 · about 309k stars.

A development method made of skills: before coding it asks what you want, shows you the design in chunks, writes a plan of short tasks and executes it with TDD and reviews between tasks. The skills trigger on their own.

```
/plugin install superpowers@claude-plugins-official
```

**What it runs:** a `SessionStart` hook (on start, clear or compact) that runs a script with bash. Its optional visual companion loads a logo from the project's website with the version; `SUPERPOWERS_DISABLE_TELEMETRY` turns that off.

REPO 02

## thedotmack/claude-mem

`github.com/thedotmack/claude-mem` · Apache-2.0 license · v15.7.0 · about 106k stars.

Memory across sessions: it captures what the agent does, compresses it with an AI model and injects it into future sessions. It uses a local worker service, SQLite, Chroma and a search MCP server. It needs Node 20 or newer.

```
npx claude-mem install
```

Inside Claude Code: `/plugin marketplace add thedotmack/claude-mem` and `/plugin install claude-mem`. `npm install -g claude-mem` installs only the library.

**What it runs:** hooks on 9 events (including `UserPromptSubmit`, `PreToolUse` and `PostToolUse`), a local MCP server and a service with a web viewer. It installs Bun and uv if they are missing.

> **CAUTION**
>
> The installer offers to sign you in to claude-mem's hosted service. What it compresses goes to the provider you choose (their service, your OpenRouter or Gemini key, or your Anthropic plan). To skip the sign-in: `--provider` or `CLAUDE_MEM_ONLINE_OPTIN=false`. Keep sensitive content out with `<private>` tags.

REPO 03

## addyosmani/agent-skills

`github.com/addyosmani/agent-skills` · MIT license · 0.6.14 · about 112k stars.

25 engineering skills and 9 commands for the whole lifecycle: `/spec`, `/plan`, `/build`, `/test`, `/constraints`, `/review`, `/webperf`, `/code-simplify` and `/ship`. Every skill demands evidence before calling something done.

```
/plugin marketplace add addyosmani/agent-skills
/plugin install agent-skills@addy-agent-skills
```

**What it runs:** nothing on its own; it registers no hooks. Use `/agent-skills:review` so it does not clash with Claude Code's own `/review`. If `marketplace add` fails over SSH, its README suggests the repository's HTTPS URL.

REPO 04

## pbakaus/impeccable

`github.com/pbakaus/impeccable` · Apache-2.0 license · cli-v4.8.0 · about 97k stars.

Design vocabulary for your agent: 24 commands (`critique`, `audit`, `polish`, `bolder`…) and a detector with 62 deterministic rules, no model and no API key.

```
npx impeccable install
```

Then run `/impeccable init` once per project. In Claude Code there is also `/plugin marketplace add pbakaus/impeccable`, then `/plugin`.

**What it runs:** hooks on `SessionStart`, `PostToolUse` (after `Edit` or `Write`) and `Stop`, which run its own binary; that binary ships with the skill or downloads once into `~/.impeccable/bin/`.

REPO 05

## manaflow-ai/cmux

`github.com/manaflow-ai/cmux` · GPL-3.0-or-later (server under BUSL-1.1) · v0.65.2.2 · about 28.6k stars.

A native macOS terminal for running several agents: vertical tabs with branch and PR, notifications when an agent is waiting, a built-in browser and SSH. macOS only.

```
brew tap manaflow-ai/cmux
brew install --cask cmux
```

**What it runs:** it is your terminal, with full access to your shell. Its CLI and socket can send keystrokes, read the screen and drive the browser. It updates itself through Sparkle.

FINAL

## Checklist

- [ ] The license is clear; if it says `NOASSERTION`, you read the `LICENSE`.
- [ ] The repo is not archived and has recent pushes.
- [ ] You read its hooks and know what runs on each event.
- [ ] You know whether it starts an MCP server or a local service.
- [ ] You know where it sends data and how to turn that off.
- [ ] You installed one at a time, on a project that is not production.