AI Plugins Marketplace

User Guide · main · 2026-10-17

Contents

1. Overview
2. Installation
3. Quick start
4. The plugins
5. Configuration reference
6. Recipes
7. Troubleshooting
8. Appendix: quick reference

Overview

The **AI Plugins Marketplace** is a monorepo that hosts, manages, and distributes plugins for autonomous AI coding agents. It ships the same set of capabilities across four agents, so you can keep your tooling no matter which host you run:

- **Claude Code** — installs via the Claude plugin marketplace.
- **OpenCode** — installs as plain file copies (no registry, no build).
- **Antigravity** — installs via `agy plugin install`.
- **Qwen** — installs via `qwen extensions`.

Four plugin families are shared across the agents. The agent-specific half is a thin front end; the engines live once in `shared/`:

Family	What it does
`aiplugins-devtools`	Dev-tool skills (code review, simplification, etc.) as a skills container.
`rtk-mcp`	Rewrites shell commands through `rtk` for 60–90% fewer output tokens, plus analytics MCP tools.
`re-ghidra-mcp`	A persistent headless Ghidra JVM exposed as 19 reverse-engineering MCP tools.
`repo-user-guide`	Generates a PDF user guide for a repository, with run-and-verified examples.
`todov2` OpenCode only	Restores the V2-removed `todowrite` tool plus a live todo sidebar.

> **Context cost.** Every plugin reports a *resident* cost (what the host holds all session) and an *invocation* cost (loaded only when it runs), measured in bytes. The marketplace README carries the current table; it is regenerated by `just footprint-regen`.

Installation

Each agent has its own install mechanism. Pick the section for your host.

Claude Code ran

The repository is itself a Claude Code marketplace. Add it once, then install plugins by name:

```
$ claude plugin marketplace add ckir/aiplugins
$ claude plugin install aiplugins-devtools@aiplugins
```

Output

```
Adding marketplace…✔ Marketplace 'aiplugins' already on disk — declared in user
settings
Installing plugin "aiplugins-devtools@aiplugins"...✔ Plugin "aiplugins-
devtools@aiplugins" is already installed (scope: user)
```

Both commands are idempotent — re-running them is safe. Installs resolve to plugin zips on the latest GitHub release that already contain binaries for Windows x64 and Linux/macOS on x86_64 and aarch64, so nothing is cloned, built, or compiled at install time. Working inside a clone, the marketplace is already declared in `.claude/settings.json`, so you can skip the first line.

OpenCode ran

OpenCode plugins install as **file copies** — no registry, no build of the TypeScript. Each plugin's `INSTALL.md` is the executable procedure. For example, `repo-user-guide-opencode` (a pure skills plugin):

```
$ VERSION=0.3.0
$ curl -fsSL -o /tmp/repo-user-guide-opencode.zip \
    "https://github.com/ckir/aiplugins/releases/download/repo-user-guide-opencode-
v$VERSION/repo-user-guide-opencode-opencode.zip"
$ unzip -q -o /tmp/repo-user-guide-opencode.zip -d /tmp/repo-user-guide-opencode-
stage
$ cp -r /tmp/repo-user-guide-opencode-stage/skills/repo-user-guide
~/.config/opencode/skills/repo-user-guide
```

Output

```
downloaded: 17505 bytes
staged contents:
CHANGELOG.md  INSTALL.md  LICENSE  README.md  package.json
skills/repo-user-guide/SKILL.md
skills/repo-user-guide/assets/guide-template.html
skills/repo-user-guide/scripts/check_pre.py
skills/repo-user-guide/scripts/escape_pre.py
skills/repo-user-guide/scripts/html_to_pdf.sh
copied to /home/user/.config/opencode/skills/repo-user-guide
```

Plugins that ship a binary (`rtk-mcp-opencode`, `re-ghidra-mcp-opencode`) additionally stage the built binary and merge an `opencode.jsonc` entry; their `INSTALL.md` covers the extra steps. OpenCode auto-discovers skills by folder name — nothing is registered in a config file for skills-only plugins.

Antigravity illustrative

Download the agent's zip from the release, extract it into a folder named after the plugin, and install that folder:

```
$ curl -fsSL -o /tmp/rtk-mcp-agy.zip \
    "https://github.com/ckir/aiplugins/releases/download/rtk-mcp-agy-vX.Y.Z/rtk-mcp-agy-agy-plugin.zip"
$ unzip -q -o /tmp/rtk-mcp-agy.zip -d /tmp/rtk-mcp-agy
$ agy plugin install /tmp/rtk-mcp-agy
```

Put the installed `bin/` on your `PATH` and verify with `agy plugin validate /tmp/rtk-mcp-agy`. A repo-scoped alternative points `.agents/plugins.json` at the directory.

Qwen illustrative

Qwen extensions install from a zip URL or a manual extract into `~/.qwen/extensions/`:

```
$ qwen extensions install
https://github.com/ckir/aiplugins/releases/download/.../aiplugins-devtools-qwen-extension.zip
# or, manually:
$ unzip -q -o aiplugins-devtools-qwen-extension.zip -d
~/.qwen/extensions/aiplugins-devtools/
$ qwen extensions list
```

> **Known issue.** As of Qwen Code v0.22.3, `qwen extensions install <url>` can fail silently. Prefer the manual extract, or `qwen extensions link ./qwen/plugins/<name>` to develop from a clone.

Quick start

The shortest path to a working result: add the marketplace and install the dev-tool skills container, then invoke a skill.

```
$ claude plugin marketplace add ckir/aiplugins
$ claude plugin install aiplugins-devtools@aiplugins
```

Output

```
Adding marketplace…✔ Marketplace 'aiplugins' already on disk — declared in user settings
Installing plugin "aiplugins-devtools@aiplugins"...✔ Plugin "aiplugins-devtools@aiplugins" is already installed (scope: user)
```

Then ask the agent for a code review — the `code-review-and-quality` skill loads automatically when relevant. For a visible, always-on win, install `rtk-mcp` for your agent: it rewrites every shell command the agent runs, cutting output tokens 60–90% with no action on your part.

The plugins

rtk-mcp — token-saving shell rewriting ran

`rtk` (Rust Token Killer) proxies shell commands and returns token-optimized output. The plugin wires it into the agent's shell tool via a hook, so it runs automatically. The effect is easiest to see by comparing a command run natively versus through `rtk`:

```
$ rtk git status
```

Output

```
* main...origin/main
clean — nothing to commit
```

```
$ git status
```

Output

```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

The `rtk` form drops the boilerplate ("On branch", "Your branch is up to date") and compresses the rest — the same information in a fraction of the tokens. The plugin ships skills `rtk-policy` (auto) and `gain`, and MCP tools `rtk_gain`, `rtk_discover`, `rtk_check`, `rtk_proxy`.

> **Prerequisite.** `rtk` must be on your `PATH` (or point `RTK_BIN` at it). Without it the plugin installs and runs fine but never rewrites — you just get normal output.

aiplugins-devtools — dev-tool skills illustrative

A skills-only container. The first skill, `code-review-and-quality`, runs a multi-axis review (correctness, readability, architecture, security, performance). Invoke it by asking for a code review; it loads automatically when relevant. No configuration, no binary, no MCP server.

re-ghidra-mcp — Ghidra as MCP tools illustrative

Attaches a persistent headless Ghidra JVM to the session and exposes it as 19 reverse-engineering MCP tools (14 read/navigate, 5 durable writes): `list_project_programs`, `attach_program`, `inspect_function`, `find_functions`, `list_symbols`, `list_strings`, `list_data_items`, `list_segments`, `resolve_symbol`, `describe_address`, `get_xrefs`, `get_disassembly`, `read_bytes`, `get_datatype`, and the writes `rename`, `comment`, `set_datatype`, `set_prototype`, `set_local`.

```
$ # invoke a tool through your agent, e.g.:
"list the functions in the current program and disassemble the first one"
```

Output

```
(illustrative — the MCP tools run inside the agent session, not from a shell)
```

It also ships the `ghidra-re-driver` skill (auto) and a `doctor` skill, plus a `re-analyst` agent for autonomous naming/annotation sweeps. The OpenCode variant adds two session hooks (RE state across compaction, driver-skill pointer).

> **Prerequisites.** Ghidra 12.1.2 with `GHIDRA_INSTALL_DIR` set, JDK 21+ on `PATH`, and a Ghidra project that is created, imported, fully analyzed, and closed in the GUI. Without these the preflight hook reports the problem and the tools stay quiet.

repo-user-guide — PDF user guides ran

Generates a PDF that teaches a *user* (not a contributor) how to use a repository, with examples that are run and verified before inclusion. Invoke it by asking for "a user guide for this repo". The skill writes the HTML and PDF next to the repository. The OpenCode install is the file-copy procedure shown in the Installation section above (verified end-to-end). Prerequisites: Chrome/Chromium (renders the PDF), Python 3, and poppler (recommended, for checking).

todov2 — the V2-removed todo tool OpenCode only

Restores the `todowrite` session tool that OpenCode V2 removed, plus an optional live todo list in the session sidebar (a TUI). Per-session persistence, fresh per-round injection. Invoke by asking the agent to manage a todo list; the tool appears as a native `todowrite` tool. No MCP binary and no `opencode.jsonc` merge — it registers two entries in the `plugins` config list.

Configuration reference

Most plugins read an optional per-plugin settings file and a set of environment variables. Precedence is generally CLI flag > env var > settings file > default.

Plugin	Settings file	Key env vars
`re-ghidra-mcp`	`.<agent>/re-ghidra-mcp-*.local.md`	`GHIDRA_INSTALL_DIR`, `GHIDRA_MCP_PROJECT_DIR`, `GHIDRA_MCP_PROJECT_NAME`, `GHIDRA_MCP_BOOTSTRAP_PROGRAM`, `GHIDRA_MCP_MAX_HEAP`, `RE_GHIDRA_HOST`
`rtk-mcp`	`.<agent>/rtk-mcp-*.local.md`	`RTK_BIN`, `RTK_*_DISABLE`, `RTK_*_ULTRA_COMPACT`, `RTK_*_SKIP_ENV`
`todov2`	—	plugin options `enabled`, `injectEveryRound`

> **Host selection (`re-ghidra-mcp`).** `RE_GHIDRA_HOST` picks which agent's settings path the plugin reads. It defaults to `claude`; set it to `opencode` (exact) to use the OpenCode host. Unset or any other value falls back to `claude`.

Recipes

Set up the marketplace and install your first plugin ran

```
$ claude plugin marketplace add ckir/aiplugins
$ claude plugin install aiplugins-devtools@aiplugins
$ claude plugin install rtk-mcp-cc@aiplugins
```

Output

```
Adding marketplace…✔ Marketplace 'aiplugins' already on disk — declared in user
settings
Installing plugin "aiplugins-devtools@aiplugins"...✔ Plugin "aiplugins-
devtools@aiplugins" is already installed (scope: user)
Installing plugin "rtk-mcp-cc@aiplugins"...✔ Plugin "rtk-mcp-cc@aiplugins" is already
installed (scope: user)
```

See rtk save tokens on a real command ran

```
$ rtk git status
```

Output

```
* main...origin/main
clean — nothing to commit
```

Generate a user guide for a repository illustrative

```
$ # with the repo-user-guide skill installed, ask the agent:
"create a user guide for this repo"
```

Output

```
(the skill writes docs/USER_GUIDE.html and docs/USER_GUIDE.pdf next to the repo)
```

Troubleshooting

rtk does not seem to rewrite anything

Check `rtk` is on your `PATH` (or set `RTK_BIN`). The plugin installs and runs without it but silently never rewrites. Verify with `rtk git status` — if it returns compact output, the binary is found.

re-ghidra tools stay quiet

The preflight hook reports the problem. Confirm `GHIDRA_INSTALL_DIR` points at Ghidra 12.1.2, JDK 21+ is on `PATH`, and the target project is analyzed and closed in the GUI. If you are on OpenCode, set `RE_GHIDRA_HOST=opencode` so the plugin reads the right settings path.

Qwen extension install does nothing

Known silent failure as of Qwen Code v0.22.3. Use the manual extract into `~/.qwen/extensions/` or `qwen extensions link` from a clone, then confirm with `qwen extensions list`.

A plugin is installed but a skill is not found

OpenCode keys skills by folder name, so the folder name and the frontmatter `name:` must match. Re-run the plugin's `INSTALL.md` verification step, which checks both.

Appendix: quick reference

Plugin	Agent	Install	Invoke
`aiplugins-devtools`	all four	marketplace / zip / extensions	skill `code-review-and-quality`
`rtk-mcp`	all four	marketplace / zip / extensions	auto hook + MCP `rtk_*` + skills
`re-ghidra-mcp`	all four	marketplace / zip / extensions	19 MCP tools + skills + agent
`repo-user-guide`	cc, opencode	marketplace / file copy	skill `repo-user-guide`
`todov2`	opencode	file copy	tool `todowrite` + sidebar

Full per-plugin detail lives in each plugin's own `README.md` and `INSTALL.md` under `claude-code/plugins/`, `opencode/plugins/`, `antigravity/plugins/`, and `qwen/plugins/`.