EcomLabs

PRACTICAL GUIDE

Claude Code plugins: command cheat sheet and a team marketplace

The plugin commands you will use and the version each one needs, how to install from a marketplace in a single command, and how to create and validate your team's marketplace.json with a subagent that sets its own effort level.

FOR	Teams on Claude Code that want to share their own plugins
TESTED WITH	Claude Code 2.1.238 (claude plugin validate and --help) · Windows 11
UPDATED	November 11, 2026
DOCS	code.claude.com/docs/en/plugins
ARTICLE	ecomlabs.dev/g/plugins-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Before you start	What changed and how to check your version
Step 01	Command cheat sheet
Step 02	Install with --marketplace
Step 03	Your team's marketplace
Step 04	Validate before you publish
Step 05	Publish and share
When you finish	Checklist

BEFORE YOU START

What changed and how to check your version

Claude Code 2.1.292, released October 6, 2026, added `--marketplace` to `claude plugin install`: it adds the marketplace if needed, under the same policy checks as `marketplace add`, then installs. In a session, `/plugin install` has accepted the same flag since 2.1.275. Commands and fields come from the official docs (code.claude.com/docs/en/plugins), read on November 11, 2026; we validated the Step 03 files with `claude plugin validate` on 2.1.238, without installing anything.

```
claude --version
claude plugin install --help | grep -- --marketplace
claude update
```

If the `grep` prints nothing, your version lacks the flag (in PowerShell: `Select-String -SimpleMatch "--marketplace"`). `claude update` updates right away; on Homebrew run `brew upgrade claude-code`, and on WinGet `winget upgrade Anthropic.ClaudeCode`.

STEP 01

Command cheat sheet

COMMAND	WHAT IT DOES	SINCE
`claude plugin marketplace add <source>`	Registers a marketplace	—
`claude plugin marketplace list`	Lists your marketplaces and their sources	—
`claude plugin marketplace remove <name>`	Removes it and uninstalls its plugins	—
`claude plugin install <plugin>@<marketplace>`	Installs from a marketplace you already added	—
`claude plugin install <plugin> --marketplace <source>`	Adds the marketplace if needed and installs, with no confirmation	2.1.292
`/plugin install <plugin> --marketplace <source>`	The same in a session: shows the source and asks first	2.1.275
`claude plugin details <plugin>`	Component inventory and projected token cost	—
`claude plugin update <plugin>@<marketplace>`	Updates one plugin	—
`claude plugin validate <path> --strict`	Validates a marketplace, a plugin or an agents folder	—

`install` flags: `--scope` (`user` by default, `project` or `local`), `--config <key=value>`, `-y`, `--json` (2.1.268) and `--accept-command` (2.1.271). "—" means it existed before.

STEP 02

Install with --marketplace

- **The plugin name goes alone**: `claude plugin install deploy-helper --marketplace acme/tools`, with no `@marketplace`. In the old form, what follows the `@` is the `name` in `marketplace.json`, not the repository's.
- **The source** takes a GitHub `owner/repo` or `owner/repo#ref`, a git URL ending in `.git`, an `https://` URL to a `marketplace.json`, or a path starting with `./`, `../` or `/`. Without `https://`, a host such as `gitlab.example.com` is read as GitHub and fails.
- **In the shell** there is no confirmation: it prints `Successfully added marketplace: <name> (declared in user settings)`, and the marketplace stays in your user settings even with `--scope project`.
- **In a session**, the source cannot contain spaces; Claude Code shows the source, asks you to confirm and opens the details so you pick a scope.

> **CAUTION**
> A plugin can execute arbitrary code with your user privileges, and its hooks, monitors and MCP servers run outside the sandbox. Before you paste a one-step command, read `hooks/hooks.json`, `.mcp.json` and everything in `bin/`, and check the inventory with `claude --plugin-dir <directory> plugin details <name>`.

STEP 03

Your team's marketplace

`team-plugins/.claude-plugin/marketplace.json`. Required: `name`, `owner` (with a `name`) and `plugins`; every entry needs a `name` and a `source`, a path from the marketplace root that starts with `./`:

```
{
  "name": "acme-team-plugins",
  "description": "Internal plugins for the team",
  "owner": {
    "name": "Platform team",
    "email": "platform@example.com"
  },
  "plugins": [
    {
      "name": "reviewer",
      "source": "./plugins/reviewer",
      "description": "Reviewer subagent and PR checklist",
      "category": "review"
    }
  ]
}
```

`plugins/reviewer/.claude-plugin/plugin.json`. The `version` goes here, not in the entry: when both declare one, `plugin.json` wins.

```
{
  "name": "reviewer",
  "version": "1.0.0",
  "description": "Reviewer subagent and PR checklist",
  "author": { "name": "Platform team" }
}
```

`plugins/reviewer/agents/reviewer.md`, with its own effort level:

```
---
name: reviewer
description: Reviews the diff before a pull request. Use it after changes to business logic.
tools: Read, Grep, Glob
model: sonnet
effort: high
---

Read the changed files and report logic errors, uncovered edge cases and missing tests. Do not edit files.
```

`effort` takes `low`, `medium`, `high`, `xhigh` or `max`, depending on the model. `CLAUDE_CODE_EFFORT_LEVEL` wins, then the `effort` Claude passes on the call (2.1.292), then this field. Inside the plugin, the agent is named `reviewer:reviewer`.

STEP 04

Validate before you publish

```
claude plugin validate ./team-plugins --strict
claude plugin validate ./team-plugins/plugins/reviewer --strict
```

Both ended in `✓ Validation passed` on 2.1.238. With files we broke on purpose, it answered:

CHANGE	VALIDATOR OUTPUT
No `version` in `plugin.json`, with `--strict`	`version: No version specified`, exit 1
No `owner`	`owner: Invalid input: expected object, received undefined`
`"source": "./../plugins/reviewer"`	`Path contains ".."`
`"name": "acme team plugins"`	`Marketplace name cannot contain spaces`

`validate` does not see a mistyped remote source or a missing folder: those fail at install time. On 2.1.238 it also let `effort: extreme`, a level that does not exist, pass; check that field by hand.

STEP 05

Publish and share

- **Push the folder** to a git repository, with an uncommon `name`: a user cannot register two marketplaces with the same name, and the official names are reserved.
- **Put this in the README**: `claude plugin install reviewer --marketplace your-org/your-repo`. Since 2.1.295, `validate` suggests it when it is missing.
- **For a project**, install with `--scope project` and commit `.claude/settings.json`; each person runs the install once in their own clone.
- **Private repository**: Claude Code clones with your git credentials and never prompts (over SSH, the host must be in `known_hosts`).

If you administer the organization, this managed settings policy, adapted from the docs, allows only the official marketplace and your organization's, and registers yours. It applies before anything downloads and at every session start, `--marketplace` included:

```
{
  "strictKnownMarketplaces": [
    { "source": "github", "repo": "anthropics/claude-plugins-official" },
    { "source": "github", "repo": "your-org/*" },
    { "source": "skills-dir" }
  ],
  "extraKnownMarketplaces": {
    "acme-team-plugins": {
      "source": { "source": "github", "repo": "your-org/team-plugins" }
    }
  }
}
```

WHEN YOU FINISH

Checklist

- [ ] `claude --version` shows 2.1.292 or later on every machine on the team.
- [ ] `marketplace.json` has `name`, `owner.name` and `plugins`, with sources that start with `./`.
- [ ] The `version` lives in `plugin.json` and `validate --strict` passes.
- [ ] Each agent has an `effort` chosen for its task, checked by hand.
- [ ] The README carries the `--marketplace` line.
- [ ] Someone read every plugin's hooks, `.mcp.json` and `bin/` before sharing it.