AI TOOLS WIKI · MÉMO

Ruflo *mémo*

Les commandes à connaître pour **Ruflo** (anciennement Claude Flow), la couche d'orchestration multi-agents pour Claude Code et Codex : essaims d'agents (swarms), mémoire vectorielle, workers en arrière-plan, plugins.

À lire d'abord : Ruflo a été très critiqué au printemps 2026 (audit indépendant de la v3.5.51 en avril), puis l'équipe a publié de nombreux correctifs et son propre audit chiffré (v3.10.7, fin mai). La version actuelle (v3.45, septembre 2026) n'a pas été réauditée de façon indépendante à notre connaissance. Voir **les avis sur Ruflo** pour la chronologie. À essayer sur un projet non critique, en mesurant le coût, avant de l'adopter.

Famille Orchestration et catalogues d'agents · **Éditeur** ruvnet · **Licence** MIT
Note du wiki 8/15 · **Commandes vérifiées le** 25 septembre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/orchestration-et-catalogues/ruflo

Installation : deux chemins très différents

Chemin	Ce que tu obtiens	Pour qui
A. Plugins Claude Code	Quelques slash commands et agents, aucun fichier ajouté au projet, pas de hooks.	Tester sans s'engager.
B. CLI (`ruflo init`)	Le système complet : ~98 agents, 60+ commandes, 30 skills, serveur MCP, hooks, daemon. Ajoute **`.claude/`**, **`.claude-flow/`** et **`CLAUDE.md`** au projet.	Usage réel, tel que documenté.

`/plugin marketplace add ruvnet/ruflo`
`/plugin install ruflo-core@ruflo`
`/plugin install ruflo-swarm@ruflo`
	Chemin A – dans Claude Code

Chemin B – CLI	
`npx ruflo@latest init wizard`	assistant interactif (toutes plateformes)
`npm install -g ruflo@latest`	ou installation globale
`claude mcp add claude-flow -- npx ruflo@latest mcp start`	Ajouter Ruflo comme serveur MCP dans Claude Code

Commandes CLI principales

`npx ruflo init --wizard`	Initialise un projet avec l'assistant.
`npx ruflo doctor --fix`	Vérifie l'installation et corrige les problèmes courants. **Premier réflexe en cas de souci.**
`npx ruflo swarm init --topology hierarchical --max-agents 8`	Crée un essaim d'agents (topologies : hiérarchique, maillée, adaptative).
`npx ruflo agent spawn -t coder --name mon-coder`	Lance un agent d'un type donné.
`npx ruflo agent list`	Liste les agents actifs.
`npx ruflo hive-mind init` / **`hive-mind`** spawn `--agents 8`	Mode « hive-mind » : plusieurs agents avec consensus.
`npx ruflo memory store` / **`search`** / **`list`** / **`retrieve`**	Mémoire vectorielle persistante : stocker, chercher sémantiquement, lister, relire.
`npx ruflo task …`	Crée et suit des tâches.
`npx ruflo workflow …`	Workflows réutilisables en plusieurs étapes.
`npx ruflo daemon start`	Démarre les workers d'apprentissage en arrière-plan.
`npx ruflo hooks …`	Gère les hooks d'auto-apprentissage.
`npx ruflo security …`	Analyse de sécurité.
`npx ruflo testgen …`	Analyse de couverture et génération de tests.
`npx ruflo plugins …`	Gère les plugins npm.

La liste complète est dans la référence CLI du wiki.

Plugins Claude Code utiles

Plugin	Rôle
`ruflo-core`	Socle : serveur MCP, diagnostics, découverte des plugins.
`ruflo-swarm`	Coordination de plusieurs agents en équipe.
`ruflo-rag-memory`	Recherche hybride dans la mémoire.
`ruflo-testgen`	Trouve et génère les tests manquants.

Plugins Claude Code utiles (suite)

Plugin	Rôle
`ruflo-security-audit`	Recherche de vulnérabilités et CVE.
`ruflo-cost-tracker`	Suivi de la consommation de tokens, budgets, alertes. **Recommandé si tu utilises Ruflo sérieusement.**
`ruflo-sparc`	Méthodologie guidée en 5 phases avec contrôles qualité.

Conseils d'utilisation

- **Mesurer avant d'adopter.** Un test comparatif publié le 11 mai 2026 (version non précisée) mesurait environ 220 000 tokens et 35 minutes pour une fonctionnalité de complexité moyenne. Ce coût n'a pas été remesuré publiquement sur les versions récentes : le vérifier sur ton propre cas, avec `ruflo-cost-tracker`.
- **Replacer les critiques dans leur contexte.** L'audit indépendant d'avril 2026 (v3.5.51) jugeait que la plupart des 300+ outils MCP n'exécutaient rien. Le mainteneur a reconnu les problèmes et publié des correctifs dès mai, puis un audit interne mesuré (v3.10.7) qui a corrigé des bugs, retiré une métrique générée au hasard et remplacé les chiffres de performance gonflés par des valeurs mesurées.
- **Préférer les chiffres mesurés** : le README actuel indique par exemple ~1,9× à 4,7× pour la recherche vectorielle HNSW, loin des « 150× à 12 500× » annoncés auparavant.
- **Garder Claude Code seul pour les tâches simples** : une tâche sur un fichier, un agent, terminée dans la journée, n'a pas besoin d'un essaim.
- **Commencer par le chemin A** (plugins) pour tester sans modifier le projet.
- **Installer `ruflo-cost-tracker`** et fixer un budget avant de lancer un hive-mind.
- **Commiter avant `ruflo init`** : la commande ajoute plusieurs dossiers et modifie **`CLAUDE.md`**.

Sources (documentation officielle)

- README Ruflo
- Référence CLI (wiki)
- Comparatif des plugins d'orchestration multi-agents – Gerald Chen
- Audit Ruflo / Claude-Flow sur la v3.5.51 – gist roman-rr (avril 2026)
- Audit interne mesuré de Ruflo (mai 2026)
- Guide complet Claude Flow (Ruflo) – Pasquale Pillitteri