NOW AVAILABLE The 512GB Mac Studio is on sale now, shipping February to March. See every configuration.

THE AI TOOLCHAIN

NO. 087

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 8, 2026 · EVERY DAY

EDITIONS **tail** | grep | head | ls | diff | uniq | perf

The daily firehose — everything the toolchain shipped today, already filtered.

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 82 of 84 tools

OFFENSIVE SECURITY

vim ? help

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.183 NOTES ↗

The xalgorix platform runs AI pentesting agents for reconnaissance, vulnerability detection, and exploitation workflows.

xalgorix adds converge-finish and converge-reconcile defaults that automatically close scans once coverage thresholds are met

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.183 https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.183
```

└──▷ TRY IT

Restore the previous open-ended scan behaviour on a CI system where scans must not auto-close before a manual review step.

```
$ XALGORIX_CONVERGE_FINISH=false XALGORIX_CONVERGE_RECONCILE=false
xalgorix scan <target>
```

Tune the idle-iteration window so a long-running scan waits longer before closing, while keeping the lead-verification window tight.

```
$ XALGORIX_CONVERGE_FINISH=true XALGORIX_CONVERGE_FINISH_IDLE=1000
XALGORIX_CONVERGE_FINISH_LEAD_WINDOW=50 xalgorix scan <target>
```

Scans now close automatically after `XALGORIX_CONVERGE_FINISH_IDLE` idle iterations (default 500) once a scan clears its coverage threshold, with high-confidence open leads granted `XALGORIX_CONVERGE_FINISH_LEAD_WINDOW` iterations (default 100) to be verified before close — enabled by setting `XALGORIX_CONVERGE_FINISH=true`, now the default.

Exploratory plan tasks without a vulnerability class now close once every measurable obligation is done and the minimum iteration count has passed, controlled by `XALGORIX_CONVERGE_RECONCILE`, now defaulting to `true`.

Content-discovery artifacts (VCS metadata, config and secret files, logs, OS cruft) now owe only information-exposure checks rather than every injection class when `XALGORIX_CONVERGE_FINISH` is enabled.

└──▷ **BREAKING ON UPGRADE**

! `XALGORIX_CONVERGE_FINISH` now defaults to `true`; scans that previously relied on running indefinitely past the coverage threshold will now close automatically — set `XALGORIX_CONVERGE_FINISH=false` to restore prior behaviour.

! `XALGORIX_CONVERGE_RECONCILE` now defaults to `true`; exploratory plan tasks without a vulnerability class that previously kept a fully covered scan open indefinitely will now close once obligations are met — set `XALGORIX_CONVERGE_RECONCILE=false` to restore prior behaviour.

WAS THIS USEFUL? 👍 👎

APPLICATION SECURITY

◆ Code Security

jscpd

Feedback ⓘ v5.4.3 NOTES ↗

Jscpd is a copy/paste detector for source code. 220+ languages, Rust engine, SARIF/HTML/badge reporters, GitHub Action, MCP server for AI agents.

jscpd adds a runJscpd(args, options) Node.js API and now shows both clone fragments with syntax highlighting in HTML reports

└──▷ GET THIS VERSION ▾

Homebrew

```
$ brew install jscpd
# already installed? upgrade to this version:
$ brew upgrade jscpd
```

Docker

```
$ docker pull ghcr.io/kucherenko/jscpd:5.4.3
```

pipx

```
$ pipx install jscpd==5.4.3
# already installed? upgrade to this version:
$ pipx install --force jscpd==5.4.3
```

npm

```
$ npm install -g cpd@5.4.3
```

Cargo

```
$ cargo install jscpd --version 5.4.3
# already installed? upgrade to this version:
$ cargo install jscpd --version 5.4.3 --force
```

git

```
$ git clone --branch v5.4.3 https://github.com/kucherenko/jscpd.git
# already have the repo? check out this version:
$ git checkout v5.4.3
```

Homebrew lists this version · published under the project's GitHub account · PyPI attestation · signed build provenance · named in the project's README · checked 8 Nov

Wrong command?

Call runJscpd(args, options) from Node.js via `import { runJscpd } from 'jscpd'` to invoke the jscpd binary without spawning a second Node.js process; types are in `index.d.ts` and errors from the binary throw as `Error` objects.

The HTML report now shows both fragments of a clone side by side (or stacked on narrow screens), each with its filename, line-number gutter, and horizontal scroll per pane, and highlights code with Prism 1.30.0, inlining only the languages present in the report.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

zot

Feedback ⓘ v0.4.67 NOTES ↗

zot manages coding agent workflows from the command line.

zot adds provider disabling, model visibility settings, custom compaction summary instructions, and an optional Bash startup file.

└──▷ GET THIS VERSION ▾

Go

```
$ go install github.com/patriceckhart/zot/cmd/zot@v0.4.67
```

git

```
$ git clone --branch v0.4.67 https://github.com/patriceckhart/zot.git
# already have the repo? check out this version:
$ git checkout v0.4.67
```

Go module path matches the repo · checked 8 Nov Wrong command?

Disable providers and control model visibility with the new provider disabling and model visibility settings.

Customize how sessions are summarized by supplying custom summary instructions to the compaction feature.

Load an optional zot Bash startup file at launch for environment setup.

NEW INTEGRATION GPT-6.1 Sol support for GitHub Copilot

NEW INTEGRATION GPT-6.1 Terra support for GitHub Copilot

NEW INTEGRATION Copilot Gimini 4 Argon and MAI-Code model support

WAS THIS USEFUL? 👍 👎

Red Hat ripwire

Feedback ⓘ v0.6.10 NOTES ↗

With ripwire, coding agents get a ranked repository map showing call graphs, blast radius and test coverage.

ripwire adds `--impact=<path>`, call-site printing for `--path` and `--connect`, alias resolution for `--defs` and `--uses`, and reST/Sphinx document support

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.6.10 https://github.com/redhat-et/ripwire.git
# already have the repo? check out this version:
$ git checkout v0.6.10
```

└──▷ **TRY IT**

Find every file that imports a specific module to understand blast radius before refactoring it.

```
$ ripwire --impact=src/auth/session.ts
```

Let an AI agent edit files through Claude Code's native ranged Read and Edit instead of raw file writes.

```
$ ripwire wrap claude
```

List a file's importers directly with `--impact=<path>`, and get a summary count of importers that a cut listing hides when using `--impact`.

Get call sites, the call expression, and the callee's signature printed for each hop when using `--path` and `--connect`.

Show an alias's target definition with `--defs` and list every write site through an alias with `--uses`.

Route Claude Code edits through a ranged Read and Claude Code's native Edit with `ripwire wrap claude`.

Support reST and Sphinx documents as first-class indexed units, with `--recall` now serving the best-matching part of an HTML page.

+2 more

└──▷ **BREAKING ON UPGRADE**

! Caches rebuild once on upgrade (`kParserVer` 301, `kCacheVersion` 49); quality snapshots of the older scheme (scheme 25) are not used.

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.3314 CODE ↗ LATEST STABLE

Magpie is a macOS menu-bar app, TUI, and CLI that lists every AI coding agent on the machine and switches each one's model, such as Codex on DeepSeek or Claude Code on Kimi, by rewriting only that key in the agent's own config.

magpie's Routing stage now labels each request by the gateway key name it arrived with, not just the User-Agent string

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3314
```

git

```
$ git clone --branch v0.1.3314 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3314
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Requests in the Routing stage now display the named gateway key (`callerKey` / `callerKeyName`) they arrived with — so a script sending `Python-urllib` as its User-Agent is identified by its key's human-readable name instead.

▾ 9 more releases in this issue · 2026-11-07 → 2026-11-08

v0.1.3313 CODE ↗ STABLE

magpie's gateway `GET /v1/models` endpoint now answers with an `ETag` and returns `304 Not Modified` when the client's `If-None-Match` header matches the current list

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3313
```

```
git

$ git clone --branch v0.1.3313 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3313
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ TRY IT

Poll the model list efficiently from a script — skip processing when the list has not changed since the last fetch.

```
$ curl -s -o /dev/null -w "%{http_code}" -H 'If-None-Match: "<etag-from-previous-response>"' http://localhost:8080/v1/models
```

v0.1.3310 CODE ↗ STABLE

magpie now tracks multiple expiring credit grants per API key, spending the soonest-expiring grant first

└──▷ GET THIS VERSION ▾

```
Docker

$ docker pull ghcr.io/yetone/magpie:0.1.3310
```

```
git

$ git clone --branch v0.1.3310 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3310
```

published under the project's GitHub account · checked 8 Nov Wrong command?

A key's balance now supports multiple independent expiring credit grants — each with its own amount and expiry date — spent in soonest-first order, so promotional credits added on separate days are each tracked and depleted correctly.

The balance card UI now shows each expiring grant as a separate pill with its own remaining amount and days countdown, turning amber within a week of expiry, and lets you add or remove individual grants from the provider editor.

v0.1.3309 CODE ↗ STABLE

magpie adds sign-in expiry notifications that alert when an account needs re-authentication or when a Claude session is about to expire

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3309
```

git

```
$ git clone --branch v0.1.3309 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3309
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Adds desktop notifications for accounts that have been signed out or need re-authentication, naming the exact provider row affected — previously, sign-in failures were visible only inside the magpie Providers page.

Adds advance warning notifications for Claude accounts whose sign-in refresh token is nearing expiry, mirroring the same countdown Claude Code itself shows (`Your login expires in N days`), starting three days before the token ends.

v0.1.3305 CODE ↗ STABLE

magpie now tracks expiring credit on provider keys, spending it first and stopping it at local midnight on its last day

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3305
```

git

```
$ git clone --branch v0.1.3305 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3305
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Track expiring credit on a provider key with the new `Expiring` field (`expiring` in JSON), which records the amount, the last day it counts (`until`, as `2006-01-02`),

and when it was recorded (`at`); the credit is spent before the rest of the balance and counted through midnight local time on its last day.

Map the model into memory instead of reading it when running `pf-serve` on the CPU, reducing resident memory use for local inference.

v0.1.3303 CODE ↗ STABLE

magpie adds optional SOCKS5 proxy username and password fields in Settings > Network and sharing

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3303
```

git

```
$ git clone --branch v0.1.3303 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3303
```

published under the project's GitHub account · checked 8 Nov Wrong command?

v0.1.3302 CODE ↗ STABLE

magpie's redact engine now passes hex literals (up to 64-bit) through unmasked and masks values on both the masked-words and never-mask lists

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3302
```

git

```
$ git clone --branch v0.1.3302 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3302
```

published under the project's GitHub account · checked 8 Nov Wrong command?

A value present in both the Words (masked words) list and the Allow (never-mask) list is now masked — the safer outcome — while Allow still wins over user-defined Rules for the specific value it names.

v0.1.3300 CODE ↗ STABLE

magpie adds per-model request headers that override provider-level headers across all API routes

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3300
```

git

```
$ git clone --branch v0.1.3300 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3300
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Set per-model HTTP request headers in Model settings via `ModelHeaders`, keyed by `<provider id>/<model id>`, so a model can carry its own routing headers alongside — and overriding — the provider's shared headers on every request route (chat, `count_tokens`, embeddings, images, videos, and model test).

Use `WithModel` in the gateway context to attach the active model's identity so that `Sign` applies per-model headers only to that provider's requests, never to a fallback provider serving the same model ID.

Allow a key's provider to have its model list turned off independently of other providers.

v0.1.3295 CODE ↗ STABLE

magpie adds yuan-denominated key balances and automatic monthly or weekly refill schedules to `magpie provider key-balance`

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3295
```

git

```
$ git clone --branch v0.1.3295 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3295
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ TRY IT

Track a CNY-priced API key so magpie converts spend to yuan automatically instead of requiring a USD amount.

```
$ magpie provider key-balance myprovider Main ¥100
```

Track a key that tops up ¥100 on the 1st of every month, so magpie resets the balance counter automatically each cycle.

```
$ magpie provider key-balance myprovider Main ¥100 --refill
monthly:1
```

Set a provider key balance in Chinese yuan with `magpie provider key-balance <id> <key> ¥<cny>` or `<amount>cny`; magpie counts it down at the USD/CNY rate it last fetched, leaving the balance unknown until a rate is available.

Schedule automatic balance refills with `--refill monthly[:<day>]`, `--refill weekly[:<mon…sun>]`, or `--refill off` on `magpie provider key-balance`; at each refill midnight the balance resets to the originally typed amount and spend counting restarts.

v0.1.3291 CODE ↗ STABLE

magpie now releases browser tabs 20 seconds after the last CDP call ends, preventing sites from detecting long-running focus emulation

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.3291
```

git

```
$ git clone --branch v0.1.3291 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.3291
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Browser extension now releases a tab's CDP attachment via `release` and `releaseAll` 20 seconds after the last call in that tab ends — matching Claude in

Chrome's behaviour — so a site is never held under a debugger with focus emulated for minutes at a stretch.

└──▷ **BREAKING ON UPGRADE**

! The browser extension's CDP idle timeout drops from 5 minutes (`IDLE_MS = 5 * 60 * 1000`) to 20 seconds (`IDLE_MS = 20 * 1000`); tabs that were kept attached across a long pause between agent calls will now be detached and must re-attach on the next call.

WAS THIS USEFUL? 👍 👎

mjolnir

Feedback ⓘ v2.69.0 NOTES ↗ LATEST STABLE

Mjolnir is a manage Codex, Claude Code, Muse Code, Kimi Code, Grok Build, and DeepSeek Harness with durable sessions, isolated environments, quotas, and remote control. A Rust terminal and web control plane for Agent Client Protocol (ACP) coding agents.

Mjolnir now re-discovers models when a profile's harness home is edited, and install.sh refuses releases with no checksum sidecar

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.69.0
```

git

```
$ git clone --branch v2.69.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.69.0
```

signed build provenance · checked 8 Nov Wrong command?

Mjolnir's `install.sh` now refuses to install a release that has no checksum sidecar, preventing installation of unverified binaries.

▾ 7 more releases in this issue · 2026-10-15 → 2026-11-08

v2.66.0 NOTES ↗ STABLE

Mjolnir now offers to configure the host Git identity instead of failing when it is missing at launch

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.66.0
```

git

```
$ git clone --branch v2.66.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.66.0
```

signed build provenance · checked 8 Nov Wrong command?

v2.58.0 NOTES ↗ STABLE

mjolnir now removes a machine's runtimes automatically when that machine is deleted

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.58.0
```

git

```
$ git clone --branch v2.58.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.58.0
```

signed build provenance · checked 8 Nov Wrong command?

v2.57.0 NOTES ↗ STABLE

Mjolnir adds Podman support on hosts without CPU/memory cgroup delegation and review sessions in Git-less folders

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.57.0
```

git

```
$ git clone --branch v2.57.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.57.0
```

signed build provenance · checked 8 Nov Wrong command?

Mjolnir now launches Podman sessions on hosts that do not delegate CPU/memory cgroups, removing a common Linux container setup barrier.

Mjolnir now creates review-enabled sessions in folders that have no Git commits, so adversarial review is available from the very start of a project.

Mjolnir now uses the key saved in DSH before falling back to environment variables and Mjolnir secrets, giving DSH-stored credentials priority in the lookup order.

Session pickers now show only the options that detection determines are relevant, reducing noise when creating or moving a session.

v2.56.0 NOTES ↗ STABLE

mjolnir adds ACP sign-in from Settings and the CLI, Apple container sessions over SSH, and Git-free local sessions

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.56.0
```

git

```
$ git clone --branch v2.56.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.56.0
```

signed build provenance · checked 8 Nov Wrong command?

Run Apple container sessions on a Mac reached over SSH, extending container support to remote Apple hosts.

Run plain local sessions without requiring Git, allowing use in directories that are not version-controlled.

v2.55.0 NOTES ↗ STABLE

Mjolnir adds automatic steering for all compatible harnesses, SELinux-safe mbx cache for Podman, and zero-install DSH launch via DeepSeek's CLI

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.55.0
```

git

```
$ git clone --branch v2.55.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.55.0
```

signed build provenance · checked 8 Nov Wrong command?

Mjolnir now auto-steers every harness that advertises steering capability, removing the need to configure it manually per harness.

Mjolnir now lets Podman sessions read the shared mbx cache on SELinux hosts, enabling shared project memory in SELinux-enforcing environments.

Mjolnir now detects DSH through DeepSeek's CLI and launches it without requiring a separate install.

v2.52.0 NOTES ↗ STABLE

Mjolnir adds OpenCode 2 session support using a person's own OpenCode logins, with live session reload when profile logins change

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.52.0
```

git

```
$ git clone --branch v2.52.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.52.0
```

signed build provenance · checked 8 Nov Wrong command?

Mjolnir now runs OpenCode 2 sessions authenticated with the person's own OpenCode logins rather than a shared credential.

Mjolnir now reloads a running OpenCode session automatically when the profile's logins change, so credential rotations take effect without restarting the session.

v2.46.0 NOTES ↗ STABLE

Mjolnir now reports update checks in the dashboard and falls back to GitHub login after a rate limit

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @brokkai/mjolnir@2.46.0
```

```git
$ git clone --branch v2.46.0 https://github.com/BrokkAi/mjolnir.git
# already have the repo? check out this version:
$ git checkout v2.46.0
```

signed build provenance · checked 8 Nov Wrong command?

WAS THIS USEFUL? 👍 👎

octomind

Feedback ⓘ 0.70.0 NOTES ↗

Octomind is an open-source AI coding agent and agent runtime: one binary, any model, MCP-native. Runs in terminal, CI, or as a daemon.

Octomind adds an interactive provider setup wizard with back navigation and now exposes shell grants in MCP session metadata

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/muvon/octomind:0.70.0
```

git

```
$ git clone --branch 0.70.0 https://github.com/Muvon/octomind.git
# already have the repo? check out this version:
$ git checkout 0.70.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Octomind's `configure` command now has an interactive provider setup wizard with back navigation, Escape-to-skip, and sign-in status display during provider setup.

Octomind's `configure` command now shows API key sources for each provider during setup.

Octomind's learning system now preserves verified user-rule authority across learning cycles.

Octomind's compression now compacts after prompt-cache expiry and preserves exact artifact evidence during folds.

└──▷ BREAKING ON UPGRADE

! The supervisor now evaluates unsettled conditions against evidence — existing supervisor condition logic that relied on the previous evaluation behaviour may produce different outcomes.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ desktop-v0.27.0 NOTES ↗

LATEST STABLE

Qwen Code is an open-source AI coding agent that lives in your terminal.

Qwen Code adds GitHub repository event filtering, FileChanged hook notifications, OSC 7501 status reporting, and Skill archive install support

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/qwenlm/qwen-code:0.27.0
```

npm

```
$ npm install -g @qwen-code/qwen-code@0.27.0
```

git

```
$ git clone --branch desktop-v0.27.0 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout desktop-v0.27.0
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

Filter which GitHub repository events reach the agent using `allowlist` and `blocklist` controls in the channels configuration.

Receive `FileChanged` hook notifications for local sessions so external tooling can react to agent-driven file edits.

Report interactive program status to the terminal host using `OSC 7501`, enabling terminal multiplexers and wrappers to track agent activity.

Install, activate, and query Skill archives through a new file-backed archive installation kernel with native archive routes, receipts, and SDK lookup support.

Surface task Q&A activity above the composer in the web shell so users see pending agent questions without scrolling.

+3 more

▾ 1 more release in this issue · 2026-11-07 → 2026-11-08

v0.27.0 NOTES ↗ STABLE

Qwen Code adds parallel child-question handling in managed-agent sessions, a GitHub channel Repository Filter, and a PR-state watch endpoint in daemon mode

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/qwenlm/qwen-code:0.27.0
```

npm

```
$ npm install -g @qwen-code/qwen-code@0.27.0
```

git

```
$ git clone --branch v0.27.0 https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout v0.27.0
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

Enable the unified agent loop in Ink TUI with `experimental.unifiedAgentLoop` to send continuation rounds after tool batches and handle queued input and retries.

Managed-agent sessions now keep multiple child questions pending simultaneously, answer them in any order, and track drafts and errors separately per child.

Adds a Repository Filter to the GitHub channel editor with allowlist and blocklist modes, URL-friendly entries, and skipped-notification timeline rows.

Adds an Ask for details action to pending Ask User Question panels in Web Shell, opening a side task that explains options without answering.

Adds a single-pull-request view by URL for daemon PR-state watch, returning state, mergeability, and head SHA.

+3 more

WAS THIS USEFUL? 👍 👎

pairlet

Feedback ⓘ v2.6.3 NOTES ↗

Pairlet (formerly CC Pocket / cc-pocket) — Continue your local AI coding tasks from phone, tablet, or desktop.

Pairlet adds custom listen-port configuration for background service installs, with port-range and conflict checking

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.6.3 https://github.com/heypandax/pairlet.git
# already have the repo? check out this version:
$ git checkout v2.6.3
```

Background service install now accepts a custom listen port, with automatic port-range validation and conflict detection.

Codex long-session observation now processes new records incrementally, reducing repeated scan overhead.

Code blocks now wrap automatically to match message width, while copy preserves the original unformatted content.

After Codex context compression completes, a completion notice now appears in both the live session and history.

└──▷ BREAKING ON UPGRADE

! Server infrastructure has been upgraded; both the mobile and desktop Pairlet clients must be updated simultaneously to maintain connections.

WAS THIS USEFUL? 👍 👎

agentty

Feedback ⓘ v0.9.50 NOTES ↗

Agentty is an AI pair programming in your terminal — one static binary, sub-ms startup, any model

agentty adds a `tool_gallery` command to render and snapshot every tool-card surface from real recorded calls

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.9.50 https://github.com/agentty-
org/agentty.git
# already have the repo? check out this version:
$ git checkout v0.9.50
```

└──▷ **TRY IT**

Inspect how running tool cards look before shipping a UI change, to catch visual regressions in review.

```
$ tool_gallery --running --reasoning --panels
```

WAS THIS USEFUL? 👍 👎

gentle-ai

Feedback ⓘ v4.4.0 NOTES ↗

Gentle-AI configures the AI coding agents you already use: Claude Code, Cursor, OpenCode, Codex, Pi, and more. Choose persistent memory, Organic-Driven Development, curated skills, MCP servers, personas, and optional bounded review.

gentle-ai adds Antigravity 1.4 agent parity, a Manage Custom Agents TUI entry, `--phase-timings`, `--force-community-tools`, and `gs.cmd` on Windows

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v4.4.0 https://github.com/Gentleman-Programming/gentle-ai.git
# already have the repo? check out this version:
$ git checkout v4.4.0
```

└──▷ **TRY IT**

Profile how long each STATUS review phase takes when diagnosing slow pre-work in CI or locally.

```
$ gentle-ai review --phase-timings
```

Re-install community tools at the latest version without first uninstalling, useful after a CodeGraph contract change.

```
$ gentle-ai install --force-community-tools
```

Direct Codex to write all config, guidance, and skill files to a non-default location instead of `~/.codex`.

```
$ export CODEX_HOME=/workspaces/my-project/.codex
gentle-ai sync
```

Print per-phase durations of the native review STATUS pre-work to stderr with `--phase-timings` or the `GENTLE_AI_STATUS_PHASE_TIMINGS` environment variable, with phases that were still in flight or unmeasured marked explicitly.

Force a fresh community-tool install at `@latest` with `gentle-ai install --force-community-tools`, even when the installed CLI is already current.

Gentle Shell now provisions Stable from a reviewed gentle-pi supplier list, with each release pinned by its npm tarball SRI and installer hash; gentle-pi 4.5.1 is the first reviewed entry.

Stable Gentle Shell launches drop from 2.81 seconds to 0.33 seconds and Windows warm launches reach about 52 ms, using ctime-bound stamp caches instead of re-reading unchanged bytes on every start.

The Windows Separate install now publishes both `gentle-shell.cmd` and `gs.cmd`, completing the cross-platform launcher contract that shipped for Unix in v4.3.1.

+3 more

└─▷ **BREAKING ON UPGRADE**

- ! After `gentle-ai sync`, Antigravity loads Gentle AI guidance from agy rule files instead of `~/.gemini/GEMINI.md`; Gemini CLI receives guidance through `GEMINI.md` imports of owned modules.
- ! Codex config, guidance, skills, and Engram instruction files are now written under `CODEX_HOME` when it is set, instead of `~/.codex`.
- ! Workspace-scoped Context7 for Claude Code is now written to `<workspace>/.mcp.json` (which Claude Code requires you to approve once) instead of the inert `.claude/settings.json` entry.
- ! Claude Code secret deny rules are now anchored to the home directory (`~/.ssh/*` and similar) instead of resolving relative to the project.
- ! The published `review-integration/v2` intended-untracked selection schema now describes one repo-relative path string per entry; the object form it previously published was always refused at runtime.
- ! The `gentleman` theme overlay is no longer written into settings of adapters without theme assets, such as Pi and Gemini.

WAS THIS USEFUL? 👍 👎

Yuval Sarel cones

Feedback ⓘ v0.4.12 NOTES ↗

Cones is a terminal dashboard for managing coding-agent sessions across projects.

cones adds `cones resume` to reopen background sessions and `cones wait` to block until a just-launched session is ready

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.4.12 https://github.com/YuvalSarel1/cones.git
# already have the repo? check out this version:
$ git checkout v0.4.12
```

└──▷ **TRY IT**

Reopen a Claude background session that was previously stopped with `cones stop`, to continue the conversation.

```
$ cones resume <session-id>
```

Launch a session and block in CI or a script until it is ready before sending follow-up work.

```
$ cones launch --dir /path/to/project 'implement the new feature' --
wait
```

`cones wait` now blocks until a just-launched session is ready instead of returning immediately.

Agents can now start separate harnesses with `cones launch` and limit `--parent self` to their own helpers, following updated cones skill guidance.

Sessions running in their own terminal, such as an agent's `codex exec`, now show a conversation preview on hover, and `ctrl+y` takes over from that preview.

Panes now resize like a real terminal, pulling lines back from scrollback when growing and pushing lines into scrollback when shrinking, so Claude Code's prompt is no longer cut off on resize.

└──▷ **BREAKING ON UPGRADE**

! Commands that take a session id now require the full id; a prefix is refused (with the full ids it matches shown). `--parent self` continues to be accepted.

WAS THIS USEFUL? 👍 👎

swttch

Feedback ⓘ v0.36.0 NOTES ↗

Swttch is a claude Code GUI plugin for JetBrains IDEs. (ex - Claude Code with GUI)

swttch adds Claude Desktop plugin session import and a native Windows launcher command

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.36.0 https://github.com/Swttch/swttch.git
# already have the repo? check out this version:
$ git checkout v0.36.0
```

└──▷ **TRY IT**

Launch swttch on Windows directly from a terminal or Run dialog instead of navigating to the install directory.

```
$ swttch
```

WAS THIS USEFUL? 👍 👎

mira

Feedback ⓘ v0.11.0 NOTES ↗ LATEST STABLE

Mira is a self-hosted AI code reviewer with indexed PR reviews, walkthroughs, vulnerability scanning, dependency graphs, custom rules, and a learning loop.

mira adds Bitbucket Cloud as a fourth Git provider and now posts review progress via check runs on pull requests

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.11.0 https://github.com/miracodeai/mira.git
# already have the repo? check out this version:
$ git checkout v0.11.0
```

mira now supports Bitbucket Cloud as a fourth Git provider alongside GitHub, GitLab, and Forgejo, enabling auto-reviews and inline `@miracodeai` questions on Bitbucket Cloud pull requests.

mira now shows review progress and results directly on pull requests via check runs, giving teams a native CI status view of each review.

mira now caches the stable review prompt prefix on Anthropic models, reducing token usage on repeated reviews.

▾ 1 more release in this issue · 2026-10-26 → 2026-11-07

v0.10.0 NOTES ↗ STABLE

mira adds command authorization and makes indexing LLM concurrency configurable via `index.llm_concurrency`

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/miracodeai/mira:0.10.0
```

git

```
$ git clone --branch v0.10.0 https://github.com/miracodeai/mira.git
# already have the repo? check out this version:
$ git checkout v0.10.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ USE IT

Cap the number of parallel LLM calls during codebase indexing to stay within API rate limits.

mira.yaml

```
index:
  llm_concurrency: 4
```

Control indexing LLM concurrency by setting `index.llm_concurrency` in `mira.yaml`.

WAS THIS USEFUL? 👍 👎

kane-cli

Feedback ⓘ 0.10.3 NOTES ↗

Kane CLI by TestMu AI (Formerly LambdaTest)

kane-cli adds LT tunnel support for real-device runs and per-key typing delay for web type steps

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install -g @testmuai/kane-cli@0.10.3
```

git

```
$ git clone --branch 0.10.3 https://github.com/LambdaTest/kane-cli.git
# already have the repo? check out this version:
$ git checkout 0.10.3
```

named in the project's README · checked 8 Nov Wrong command?

└──▷ **TRY IT**

Run a real Android device test against a private staging backend reachable only through your LT tunnel.

```
$ kane-cli run "Sign in and open the account tab" --target android --app ./app-release.apk --tunnel-name my-tunnel
```

Connect real devices to private backends through LT tunnel by passing `--tunnel` or `--tunnel-name <name>` to `run`, `testmd run`, or `testrun run`; for `testrun run --remote`, start LT tunnel with `--env ht-prod`.

Web type steps now accept a per-key delay so fields that ignore a normal fill are typed character by character, carrying the delay through replay, upload, and code export.

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ announced · Nov 5, 2026 BLOG ↗

Kiro is an agentic development environment that uses spec-driven workflows to plan, build, and maintain software.

Kiro adds bots in Kiro Crew: named, long-running agents with memory, scheduled runs, scoped tool access, and dashboards

.
.
.
.

└──▷ **HOW TO FIND IT**

Create a build-health bot that checks your CI pipeline on a schedule and brings back only new failures that need attention.

📍 1. In Kiro Crew, open the bots workspace in the sidebar. 2. Select `Add a bot` and give it a name, e.g. `Build Health`. 3. Choose a custom agent as its base and scope its tools to CI pipeline access only (no write permissions). 4. In the bot's chat, type: 'Check the CI pipeline every 30 minutes. Sort new failures from known flaky tests and only bring back what needs my attention.' 5. The bot confirms the schedule; scheduled activity goes to its work log, findings return to chat.

Split planning and routine work across two model tiers inside a single bot to keep costs down while preserving quality on complex decisions.

📍 1. Open the bots workspace in the Kiro Crew sidebar and select your bot. 2. In the bot's settings, choose a stronger model (e.g. a frontier model) as the bot's own model for planning. 3. Set a faster, cheaper model as the worker agent model for routine checks. 4. The bot now plans on the stronger model and delegates execution to the lighter one automatically.

Kiro Crew now supports bots — named, persistent agents each with their own memory, chat, notes, and workspace, so recurring responsibilities are owned by a dedicated agent rather than tracked across isolated sessions.

Each bot can run sessions in parallel across worker agents, monitor them, and redirect work that drifts from the plan, while surfacing only findings and decisions that need your attention back to its single ongoing chat.

Set up schedules in a bot's chat so it can investigate overnight CI failures or prepare release notes on a recurring cadence without manual kicks.

Each bot gets independently scoped tool access and permissions — a research bot can read files but not write them; a deployment bot can run scripts but requires your approval before each change — and nothing changes until you approve a request.

Each bot maintains its own notes and a work log so you can inspect decisions, open items, and completed activity without opening individual sessions.

+3 more

WAS THIS USEFUL? 👍 👎

tokeneconomizer

Feedback ⓘ v0.1.0 NOTES ↗

Tokeneconomizer is an experience-driven Black-Box Cost Control for Coding Agents

TokenEconomizer adds a plugin for Claude Code, Codex, GitHub Copilot CLI, and Pi that learns from past sessions to cut agent token costs 18–31%

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.0
https://github.com/microsoft/TokenEconomizer.git
# already have the repo? check out this version:
$ git checkout v0.1.0
```

└──▷ TRY IT

Install the plugin for Claude Code so it starts learning from sessions in a repository you work in frequently.

```
$ git clone https://github.com/microsoft/TokenEconomizer.git
claude plugin marketplace add "$PWD/TokenEconomizer"
claude plugin install tokeneconomizer@tokeneconomizer
```

After Claude Code updates, rebuild the tool catalog so tool pruning has an accurate baseline of available tools.

```
$ /tokeneconomizer:toolprune refresh-catalog
```

Re-enable a tool that was pruned from a repository's sessions when you need it again.

```
$ toolprune restore
```

Check plugin status and trigger tool-catalog refresh inside Claude Code with the `/tokeneconomizer:status` and `/tokeneconomizer:toolprune refresh-catalog` slash commands.

TokenEconomizer adds a plugin for Claude Code, Codex, GitHub Copilot CLI, and Pi that observes finished sessions in a repository and applies three mechanisms — micro-skills, tool pruning, and failure-loop intervention — to reduce token spend in subsequent sessions without modifying the agent.

Tool pruning (on by default) removes tools a repository never uses from later sessions through the agent's own settings, cutting per-request token overhead.

Failure-loop intervention (on by default) uses a judge model to detect stuck turns and injects a redirect note into context, up to once per turn, without stopping the turn automatically.

All learned state is stored locally under `~/.tokeneconomizer`, keeping repository experience on the user's machine.

+1 more

WAS THIS USEFUL?  

◆ Coding Agent Add-ons

Superset

Feedback ⓘ desktop-v2.0.6 NOTES ↗

Superset is a desktop workspace for running CLI coding agents like Claude Code with terminals, diff viewer, and browser previews.

Paradev adds cloud workspace host reach, multi-org GitHub connections, and a CLI switch to control host accessibility from cloud workspaces

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v2.0.6
https://github.com/ParadevAI/Paradev.git
# already have the repo? check out this version:
$ git checkout desktop-v2.0.6
```

Control whether cloud workspaces can reach a host via a new `v2_hosts.allow_cloud_workspaces` database field.

Enable or disable cloud workspace access to a host using a new switch in the desktop app and CLI.

Connect multiple GitHub organizations to a single Paradev organization.

The Chat interface setting is now out of Experimental and in the main settings panel.

Cloud workspaces now reach hosts their owner has opened.

+2 more

WAS THIS USEFUL? 👍 👎

JuliusBrussee Caveman

Feedback ⓘ v4.1.0 NOTES ↗

LATEST STABLE

Caveman is a why use many token when few token do trick. Viral skill + proxy for coding agents that cuts 65% of tokens by talking like a caveman.

Caveman adds a recovery-store CLI, per-prompt reminder toggle, a Node-version hook flag, Pi session scanning, and memory revert to v4.1.0

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.1.0
https://github.com/JuliusBrussee/caveman.git
# already have the repo? check out this version:
$ git checkout v4.1.0
```

└──▷ TRY IT

Check the health of a damaged recovery store before an upgrade to avoid silent corruption.

```
$ caveman-proxy recovery-store check
```

Disable the per-prompt ruleset reminder repo-wide so only the session-start ruleset is sent, reducing per-prompt token overhead.

.caveman.json

```
{"reinforce": false}
```

Manage a damaged recovery store with new `caveman-proxy recovery-store status|check|repair` subcommands; a damaged store now refuses writes instead of silently corrupting, repair copies only rows whose bytes match their hash, and `caveman status` and `caveman doctor` link to it.

Turn off the per-prompt ruleset reminder by setting `CAVEMAN_REINFORCE=0` or `"reinforce": false` in `~/.config/caveman/config.json` or a repo's `.caveman.json`; the full ruleset is still sent at session start.

Make installed hooks find Node at run time via `PATH` instead of the pinned install-time path using new `--hook-node node` flag, for environments where Node versions change.

`caveman learn` now scans Pi sessions and honors the `PI_CODING_AGENT_DIR` and `PI_CODING_AGENT_SESSION_DIR` environment variables.

`caveman mem` now supports reverting a memory to an earlier value, and `supersede --stdin` accepts replacements too large for a command line.

+3 more

└──▷ **BREAKING ON UPGRADE**

! `forward_headers` under a `providers.<name>` block now stops the proxy from starting. Configure that upstream as a compat mount instead: `compat.<name>` with `base_url` and `forward_headers`, plus `wire_dialect: anthropic` if it speaks the Anthropic protocol (see `docs/technical/configuration.md`).

! `caveman-compress` now refuses a file whose text holds a credential such as a private key block or a cloud, Git host, or model API token; move the secret into an env var or secret manager before compressing.

▾ **2 more releases in this issue** · 2026-11-08

`middleware-python-v1.0.1` NOTES ↗ STABLE

Caveman middleware now supports anthropic SDK 2.x without version warnings

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch middleware-python-v1.0.1
https://github.com/JuliusBrussee/caveman.git
# already have the repo? check out this version:
$ git checkout middleware-python-v1.0.1
```

Caveman middleware now accepts anthropic SDK versions `>=1.0,<3`, ending the `unsupported_version` warning that appeared with anthropic 2.x.

`middleware-ts-v1.1.0` NOTES ↗ STABLE

Caveman's middleware now supports Anthropic SDK versions up to the 1.x major and flags outdated MCP adapter installs with a named advisory warning.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch middleware-ts-v1.1.0
https://github.com/JuliusBrussee/caveman.git
# already have the repo? check out this version:
$ git checkout middleware-ts-v1.1.0
```

Caveman now accepts `@anthropic-ai/sdk` versions 0.132.0 through 0.137.0 and the 1.x major (tested 1.0.0 and 1.1.0) within the supported range `>=0.124.0 <2`; previously the latest SDK triggered an `unsupported_version` warning.

With `@modelcontextprotocol/sdk` 1.30.x or 1.31.x installed, MCP results now pass through uncompressed and a warning names advisory GHSA-6prh-2h8m-c8cw and the version to upgrade to (`@modelcontextprotocol/sdk` 1.32.0 or later).

└─▷ **BREAKING ON UPGRADE**

! The MCP adapter now requires `@modelcontextprotocol/sdk` 1.32.0 or later; earlier versions cause MCP results to pass through uncompressed.

WAS THIS USEFUL? 👍 👎

Token Optimizer

Feedback ⓘ v5.13.68 NOTES ↗

Token Optimizer is a coding-agent plugin that trims wasted context, checkpoints work across sessions and compactions, and reports token spend on a live dashboard.

Token Optimizer adds custom command filter support for Copilot, Cursor, and Grok, and now masks secrets in doctor output

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @alexgreensh/token-optimizer@5.13.68
```

git

```
$ git clone --branch v5.13.68 https://github.com/alexgreensh/token-optimizer.git
# already have the repo? check out this version:
$ git checkout v5.13.68
```

signed build provenance · checked 8 Nov Wrong command?

Custom command filters now work on Copilot, Cursor, and Grok, and continue working on Antigravity and Devin with Python 3.9 and 3.10.

The desktop status bar now supports 12-hour and 24-hour time display formats.

The `doctor` command now hides likely secrets found in hook commands and limits flagging to hooks it can identify as its own.

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

fizgig

Feedback ⓘ v9.2.1 NOTES ↗

Fizgig — LoRA & Fine-tune Studio for Klein 9B, Krea 2, MiniMax H3, Qwen Image 2.1, Anima & Z-Image Turbo and Ace-Step XL: train, fine-tune, profile, repair and extract LoRAs & LoKR for Image, Video & Music Models

Fizgig adds PyTorch 2.15 with major training speed and memory reductions, plus telemetry opt-out and adaptive learning-rate ceiling logic

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/shootthesound/fizgig:9.2.1
```

git

```
$ git clone --branch v9.2.1
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v9.2.1
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Training now uses far less GPU memory with gradient checkpointing on PyTorch 2.15: Krea 2 drops from 22.9 GB to 14.9 GB, MOSS-Image at batch 3 from 21.7 GB to 7.1 GB.

The Muon optimizer now processes a LoRA's matrices together instead of one at a time, making each step 12× faster with identical results.

Batched steps are faster when captions vary in length: 18% faster on MOSS-Image, 12% on Krea 2, 9% on Qwen Image 2.1, and 8% on Z-Image.

LoKR training is now 4–6× faster on every model, including LoKR Context LoRA runs and LoKR files in Studio and other tools.

Fizgig now disables onnxruntime telemetry (device ID and machine details to Microsoft) and Hugging Face usage reporting on every platform.

+3 more

└──▷ BREAKING ON UPGRADE

! Python 3.10 is no longer supported: PyTorch 2.15 has no Python 3.10 build, so installs on Python 3.10 cannot update. Upgrade to Python 3.11, 3.12 or 3.13 and run

`install_fizgig.bat` (Windows) or `python3.12 install_fizgig.py` (Linux) to rebuild the environment.

! An NVIDIA driver that supports CUDA 13 (version 580 or newer) is now required. If Fizgig cannot see your GPU after updating, update the NVIDIA driver.

! RunPod pod images move to Ubuntu 24.04 (Python 3.12) to carry PyTorch 2.15; existing pods must be restarted to pick up v9.2.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

claw-orchestrator

Feedback ⓘ v7.20.1 NOTES ↗

Claw-orchestrator is a run Claude Code, Codex, Antigravity, Grok Build and OpenCode behind one runtime, and a run only counts as done when checks it didn't write pass. Cross-engine fan-out, councils, session handoff, spend caps, MCP server, ACP agent.

claw-orchestrator now surfaces rejected Planner controls in Autoloop replies and logs them as `planner_control_error` rows in `decisions.jsonl`

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @enderfga/claw-orchestrator@7.20.1
```

git

```
$ git clone --branch v7.20.1 https://github.com/Enderfga/claw-orchestrator.git
# already have the repo? check out this version:
$ git checkout v7.20.1
```

signed build provenance · checked 8 Nov Wrong command?

└──▷ TRY IT

Raise the OpenAI-compatible turn timeout when your gateway agent produces replies that take longer than 5 minutes.

```
$ OPENAI_COMPAT_TURN_TIMEOUT_MS=1800000 clawo serve
```

Point `clawo serve` at a specific Claude binary in a non-default location.

```
$ CLAUDE_BIN=/usr/local/bin/claude-custom clawo serve
```

Configure the OpenAI-compatible turn timeout with `OPENAI_COMPAT_TURN_TIMEOUT_MS`; the default is now 30 minutes (up from 5) so long-running gateway agent replies are no longer cut off.

A rejected Planner control now appears in the Autoloop reply with a named validation error and is recorded as a `planner_control_error` row in `decisions.jsonl`.

Set `CLAUDE_BIN` to control which Claude binary `clawo serve` and the MCP server use.

WAS THIS USEFUL?  

cotal

Feedback ⓘ v0.148.2 NOTES ↗ LATEST STABLE

Cotal is the open standard for agent coordination

cotal adds a command to export registered remote user-auth spaces to private connection-bundle files

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.148.2 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.148.2
```

└──▷ TRY IT

Bundle a remote user-auth space for offline handoff or backup without exposing login sessions or signing authority.

```
$ cotal space export prod-auth --out ./prod-auth-bundle.private
```

▾ 1 more release in this issue · 2026-11-07 → 2026-11-08

v0.148.0 NOTES ↗ STABLE

cotal adds Codex mesh integration via `cotal setup`, local mesh support for Claude Code and Codex conversations, and MCP Events discovery for hosted gateways

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.148.0 https://github.com/Cotal-AI/Cotal.git
# already have the repo? check out this version:
$ git checkout v0.148.0
```

└──▷ TRY IT

Add a Codex conversation to your Cotal mesh so it receives and queues messages via `cotal_inbox` after you approve the identity.

```
$ cotal setup
```

Let a Claude Code or Codex conversation join a local open or static mesh without requiring a signed-in space.

```
$ cotal up
```

Connect Codex conversations to a Cotal mesh by running `cotal setup`, which installs a Cotal plugin into Codex; the plugin binds to the conversation named in each tool call's request metadata, manages identity through the `cotal app-identity` approval flow, queues notices via Codex's `thread/queue/add`, and delivers messages through `cotal_inbox`.

Allow Claude Code and Codex conversations to join an open or static mesh started with `cotal up` without requiring a signed-in space; a conversation's `cotal_connect` opens its identity with the same grant a bare `cotal spawn` gets on that mesh.

Allow authenticated hosted gateways to implement MCP Events discovery and subscription methods through an optional adapter, with transport-bound accounts and secret-safe protocol errors.

WAS THIS USEFUL? 👍 👎

assistant-ui

Feedback ⓘ @assistant-ui/react@0.15.33 NOTES ↗

Assistant-ui provides React components and state management for chat interfaces with streaming, tool calls, and attachments.

assistant-ui adds `GeminiLiveVoiceAdapter`, a browser WebSocket adapter for the Gemini Live API with transcripts, interruptions, mute, and typed input

└──▷ GET THIS VERSION ▾

npm

```
$ npm install @assistant-ui/react@0.15.33
```

git

```
$ git clone --branch @assistant-ui/react@0.15.33
https://github.com/assistant-ui/assistant-ui.git
# already have the repo? check out this version:
$ git checkout @assistant-ui/react@0.15.33
```

signed build provenance · checked 8 Nov Wrong command?

WAS THIS USEFUL? 👍 👎

CopilotKit OpenBot

Feedback ⓘ v0.3.3 NOTES ↗

OpenBot runs self-hosted AI coworkers that each get their own browser, files, and gated tools, with every action decided and recorded.

OpenBot adds goal tools to every Bot, repeating-goal lifecycle controls, routine titles, and tightened archive/delete approval prompts

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/copilotkit/openbot:v0.3.3
```

git

```
$ git clone --branch v0.3.3 https://github.com/CopilotKit/OpenBot.git
# already have the repo? check out this version:
$ git checkout v0.3.3
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Every Bot now has access to the goal tools from Routines — `create_goal`, `list_goals`, `update_goal`, `run_goal`, `pause_goal`, `resume_goal`, and `complete_goal` — so asking a Bot in conversation to set up a goal works without manually assigning tools.

(improved) Archive actions (such as HubSpot's `HUBSPOT_ARCHIVE_CONTACT`, archiving a Slack conversation, or archiving mail in Gmail) now prompt with Allow once or Deny every time, the same as delete actions; any tool an app marks destructive — via an MCP server's `destructiveHint`, Composio's destructive tag, or Arcade's destructive flag — now prompts for approval every time regardless of its name; and an app action that publishes no behaviour flags (such as every AirtableApi and GithubApi Arcade action, or a Composio action with no label) is now treated as an ordinary change, asking only when Ask before making changes or a rule applies, rather than being treated as destructive.

(improved) A goal that a routine runs or that an event wakes now stays active after each run — only the user can end it, via conversation or its card — a goal's run now waits up to approximately 9 minutes for a busy conversation rather than failing immediately with a 409 error, and Bots now report a routine's next run in localised natural language ('Saturday, November 7, 2026 at 9:00 AM UTC, 6 minutes from now') rather than raw ISO-8601 timestamps.

A routine can now have a title up to 100 characters, set by asking a Bot to name it; the title appears as the heading on the Routines page and in a Bot's Activity list.

(improved) A question with no predefined answer choices (such as 'Which city are you in?') now reaches the user with a free-text field rather than being refused before the user is asked, and a Claude Agent SDK Bot's question now surfaces above the message box so the user can answer it, with a Bot's messages to another Bot delivered the same way.

+1 more

└─▷ **BREAKING ON UPGRADE**

- ! The tools `create_responsibility`, `list_responsibilities`, `update_responsibility`, `report_responsibility_progress`, `run_responsibility`, `pause_responsibility`, `resume_responsibility`, and `complete_responsibility` are renamed to `create_goal`, `list_goals`, `update_goal`, `report_goal_progress`, `run_goal`, `pause_goal`, `resume_goal`, and `complete_goal`; the old names are removed — update any Bot instructions, skills, or goals that reference them.
- ! `create_trigger` now takes `goalId` instead of a responsibility identifier; update any instructions or skills that call it.
- ! Audit trail event counts change: a person's denial is now recorded as one `computer.action_refused` or `mcp.call_rejected` event; an approved computer action is one `computer.action_allowed` (previously up to three); a Bot's actions in folders are `computer.action_*` events (previously two `configuration.changed` rows each); and assigning an app's tools to a Bot is one `configuration.changed` row per Bot and app (previously one per tool). Log collectors or exports that relied on the old counts will see different totals.
- ! Five database migrations run on first start: migration `0078_routine_title` adds nullable `routines.title`; `0079_tool_unclassified` adds `mcp_tools.unclassified` (default false); `0080_listing_revision` adds `mcp_servers.listing_revision` (default 0); `0081_goal_recurring` adds `responsibilities.recurring` (default false); `0082_routine_goal` adds `routines.goal_id` and `routines.goal_by_title` (default true). The database advances to `0082_routine_goal`.
- ! Any Bot at its own address now receives at most 128 tools per request; app tools least relevant to the request are omitted when that limit is reached.
- ! A goal that was completed by a routine's run before this upgrade will remain completed; going forward, goals run by routines or events no longer auto-complete.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.9.22 NOTES ↗ LATEST

STABLE

HyperFrames is an open-source HTML-to-video renderer that runs in AI-agent workflows.

HyperFrames adds full-quality decoded preview frames when scrubbing video in Studio

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.22 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.9.22
```

▾ 2 more releases in this issue · 2026-11-07 → 2026-11-08

v0.9.21 NOTES ↗ STABLE

HyperFrames adds multipart cloud uploads with per-part retry so a dropped connection does not restart the whole transfer

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.21 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.9.21
```

HyperFrames CLI now uploads large files to the cloud in parts, so a dropped connection only resends the failed part rather than restarting the whole transfer.

Voice credits now identify the actual source — your own recording or a vendor — rather than a generic label.

Your own voice recording now counts as a voice line, with playback speed applied to it.

v0.9.20 NOTES ↗ STABLE

HyperFrames adds JSON telemetry status output, `check_report` finding names, and `--read --layer` single-layer audio reading to its CLI

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.20 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.9.20
```

└──▷ **TRY IT**

Ingest a single-layer audio track without triggering stem separation — useful when your source file is already a final mix.

```
$ npx hyperframes audio --read --layer my-soundtrack.wav
```

Get telemetry status as machine-readable output with `--json` or the JSON telemetry status flag on the CLI.

Read a one-layer audio file whole, without separation, using `audio --read --layer` on the CLI.

Check subsystem improvements: `check_report` now counts the browser-playable copies a check waited for, a `line_unread` finding now carries the frames it measured around where the line was first read, and you can name the saved report file and get one stderr line per check step.

Studio now accepts a typed exact preview zoom value and shows a pixel grid at high zoom levels.

Studio now shows the agent's storyboard in a Storyboard tab beside the preview, and shows canvas gesture position, size, and rotation.

+1 more

WAS THIS USEFUL? 👍 👎

openexecutive

Feedback ⓘ v0.7.0 NOTES ↗

Openexecutive is an ai-powered virtual executive team — a single coherent executive persona backed by 8 specialist agents (FastAPI + Next.js).

OpenExecutive adds interactive alert cards, an Approvals gate across all channels, memory editing, and email-draft editing in chat

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.7.0
https://github.com/SenteLabsAI/OpenExecutive.git
# already have the repo? check out this version:
$ git checkout v0.7.0
```

OpenExecutive expands chat card interactions: alert cards can now request an answer, a pick, an acknowledgment, or no response from the recipient; Needs you cards can be answered directly in chat by saying yes, no, or a typed answer; and email drafts can be edited directly on their chat card in place.

OpenExecutive adds Approvals as the single gate on every channel, replacing per-channel approval handling.

OpenExecutive adds a Memory page (replacing the previous Pulse view) that lets people edit or remove what is kept in their memory, shows a notification with Undo whenever memory changes, retains calculated figures together with their source and date, and lets mail chats teach the owner's own memory.

OpenExecutive adds watches that report real changes and monitor for awaited email.

OpenExecutive now updates lessons from what people actually say in conversation.

+3 more

WAS THIS USEFUL? 👍 👎

Microsoft Agent Framework

Feedback ⓘ python-1.25.0

NOTES ↗

The Microsoft Agent Framework builds, orchestrates, and deploys agents and multi-agent workflows in Python and .NET, consolidating AutoGen and Semantic Kernel.

Microsoft Agent Framework adds callable MCP discovery approval policies with tool-name, read-only, and caller-override filters, plus bounded MCP catalog discovery

└──▷ GET THIS VERSION ▾

pip

```
$ pip install agent-framework==1.25.0
# already installed? upgrade to this version:
$ pip install --upgrade agent-framework==1.25.0
```

git

```
$ git clone --branch python-1.25.0 https://github.com/microsoft/agent-framework.git
# already have the repo? check out this version:
$ git checkout python-1.25.0
```

named in the project's README · checked 8 Nov Wrong command?

└──▷ USE IT

Cap MCP catalog discovery at a custom page and item limit when connecting to a large but trusted internal MCP server.

python

```
from agent_framework.mcp import DefaultMCPToolHandler

handler = DefaultMCPToolHandler(
    server_url="https://mcp.internal.example.com",
    max_discovery_pages=200,
    max_discovery_items=5000,
    request_timeout=60,
)
```

Isolate workflow executors per caller to satisfy the new single-scope authorization requirement in multi-user deployments.

```python
from agent_framework.hosting import AgentFrameworkWorkflow

app = AgentFrameworkWorkflow(
    workflow_factory=lambda: MyWorkflow(),
)
```

Support callable MCP discovery approval policies and honor Foundry Toolbox approval metadata, including tool-name and read-only filters, explicit caller overrides, and policy refresh for retained tools in `agent-framework-core` and `agent-framework-foundry-hosting`.

Bound MCP tool and prompt discovery to 100 pages and 1,000 items per catalog via `max_discovery_pages` and `max_discovery_items` on `DefaultMCPToolHandler`, default framework-owned request timeouts to 30 seconds, and preserve the complete catalog snapshot when refresh fails.

Redact host-tool exception details before generated code resumes in `agent-framework-hyperlight` and `agent-framework-monty`; control exception detail exposure with `include_detailed_errors=True` on the host side for trusted channels.

Migrate `agent-framework-foundry-local` to Foundry Local SDK 2.3.1, adding support for SDK-registered local models and model preparation before first inference, on Apple Silicon macOS, Linux x86_64/aarch64, and Windows x64/ARM64 with Python 3.11–3.14.

Enable Redis integration (`agent-framework-redis`) and its standard-install dependency on Python 3.15, and widen the supported RedisVL range to include 0.31.

+1 more

└──▷ **BREAKING ON UPGRADE**

- ! Python 3.10 is no longer supported across all packages; Python 3.11 or later is now required.
- ! MCP tool and prompt discovery is now bounded at 100 pages and 1,000 items per catalog by default; set `max_discovery_pages` or `max_discovery_items` to a higher value or `None` (trusted servers only) to restore previous behavior. `DefaultMCPToolHandler` now requires Core 1.25.
- ! Live workflow agent executors and sessions are now bound to one authorization scope; cross-scope reuse raises `WORKFLOW_SCOPE_CONFLICT` and WorkflowAgent.as_tool() in AG-UI endpoints raises `WORKFLOW_AGENT_TOOL_UNSUPPORTED`. Migrate to AgentFrameworkWorkflow(workflow_factory=...) with a per-call factory.
- ! OpenTelemetry workflow invocation spans are renamed: replace `workflow.run` span-name filters with `gen_ai.operation.name == 'invoke_workflow'` and

replace `workflow.name` attribute lookups with `gen_ai.workflow.name`. `OtelAttr.WORKFLOW_RUN_SPAN` now contains the base operation name only.

! In `agent-framework-declarative`, `DefaultMCPToolHandler` no longer caches initialized MCP sessions; replace `async with DefaultMCPToolHandler(...) as handler:` with ordinary construction handler = DefaultMCPToolHandler(...) and remove aclose() calls. The `cache_max_size` parameter is now a deprecated no-op and emits `DeprecationWarning`.

! In `agent-framework-foundry-local`, Foundry Local SDK 2.3.1 or later is now required (`foundry-local-sdk>=2.3.1,<3`); the `do_sample` and `early_stopping` options are removed. `bootstrap=False` now requires a pre-initialized manager and running web service.

! In `agent-framework-hyperlight` and `agent-framework-monty`, host-tool exception details are redacted by default; code matching original host exception types will receive generic `RuntimeError` instead unless `include_detailed_errors=True` is set on the host side.

WAS THIS USEFUL? 👍 👎

busabase

Feedback ⓘ v1.2.0 NOTES ↗

Busabase is an open-source database & workspace for AI agents — structured data, durable knowledge, reusable skills, runnable apps, and human review on the writes that matter. Local-first and self-hostable.

busabase adds per-video poster support in templates and cleans up preview tiles

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g busabase@1.2.0
```

git

```
$ git clone --branch v1.2.0 https://github.com/busabase/busabase.git
# already have the repo? check out this version:
$ git checkout v1.2.0
```

signed build provenance · checked 8 Nov Wrong command?

Templates now support per-video posters and cleaner preview tile rendering.

WAS THIS USEFUL? 👍 👎

Arcade

Feedback ⓘ announced · Nov 5, 2026 BLOG ↗

Arcade provides an actions runtime that lets AI agents securely call external services with authorization and governance controls.

Arcade adds Enterprise MCP Hosting so self-hosted customers can deploy custom MCP servers with `arcade deploy`

.
.
.
.

└──▷ **TRY IT**

Deploy a custom MCP server from a local checkout or CI pipeline to your self-hosted Arcade environment without needing cluster access.

```
$ arcade deploy
```

Deploy custom MCP servers to self-hosted Arcade environments with the `arcade deploy` command, without filing tickets for a host, identity setup, or cluster access.

Every deployed server automatically inherits Arcade's authorization, policy enforcement, and audit logging from its first call.

Manage deployed servers through the dashboard's Servers page, with versioned releases, status, logs, restart, removal, and per-organization and per-project deployment limits.

Arcade adds Enterprise MCP Hosting support for Python servers built with the Arcade MCP framework, with FastMCP support planned next.

WAS THIS USEFUL? 👍 👎

opencrabs

Feedback ⓘ v0.5.14 NOTES ↗ LATEST STABLE

Opencrabs is the all-in-one AI agent. Build mobile apps, frontend, backend, manages files, deep research, schedule tasks and events.

opencrabs adds Chinese and Arabic language packs with CJK and Arabic script detection

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.14
https://github.com/opencrabs/opencrabs.git
# already have the repo? check out this version:
$ git checkout v0.5.14
```

opencrabs now adds a Chinese language pack with CJK script detection to the phantom engine.

opencrabs now adds an Arabic language pack with Arabic script detection to the phantom engine.

▾ 4 more releases in this issue · 2026-10-10 → 2026-10-25

v0.5.13 NOTES ↗ STABLE

opencrabs adds full-screen TUI sessions and help dialogs, a Wayland-aware clipboard, and an `--no-path` installer flag

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.13
https://github.com/opencrabs/opencrabs.git
# already have the repo? check out this version:
$ git checkout v0.5.13
```

Install opencrabs with `--no-path` to skip the shell PATH entry, or let the installer write it once and verify with `command -v` before reporting success.

The TUI help dialog now opens full screen with each column scrolling independently.

The TUI sessions dialog now opens full screen with its commands listed in the footer.

Copy in the TUI now tries every clipboard backend in order and writes to Wayland on Linux.

The editor handoff in the TUI now matches `nvim`, `helix`, and `$EDITOR`.

v0.5.12 NOTES ↗ STABLE

OpenCrabs adds file upload to browser actions, enforces mid-turn planning after a tool-call threshold, and extends full session capabilities to Telegram guest queries

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.12
https://github.com/opencrabs/opencrabs.git
# already have the repo? check out this version:
$ git checkout v0.5.12
```

Attach local files in browser automation without putting their bytes in the context using the new `upload` action on `browser_act`.

The agent now stops and writes a plan once the number of tool calls in a single turn crosses a threshold, enforced by a new plan gate.

The `bash` tool now refuses `sleep` while a detached task is running and names the specific task it is waiting on in the refusal message.

Telegram guest queries now get the same live status block, rich rendering, and tool set that a normal session receives, with one cancellable turn per guest session and rate limiting.

Sensitive and keyed prompt calls now run from a Python script instead of inside the session, isolating them from the session context.

+1 more

v0.5.10 NOTES ↗ STABLE

OpenCrabs adds a built-in provider serving DeepSeek v4.1 Flash at 200k context, a Telegram userbot-download command, and voice STT/TTS model selection from any provider

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.10
https://github.com/opencrabs/opencrabs.git
# already have the repo? check out this version:
$ git checkout v0.5.10
```

└──▷ **USE IT**

Configure voice to use specific STT and TTS models from your chosen provider instead of the defaults.

~/.opencrabs/config.toml

```
stt_model = "whisper-1"
tts_model = "tts-1-hd"
```

Install the motion-graphics opt-in skill so the `/motion-graphics` command becomes available in your session.

```
$ cp -r src/docs/reference/templates/optional-skills/motion-graphics
~/.opencrabs/skills/motion-graphics
```

OpenCrabs built-in provider improvements (new): you can now set `generation_model` via `config` to control which model the built-in endpoint uses; OpenCrabs includes a built-in provider serving DeepSeek v4.1 Flash, a 200k-context multimodal model requiring no account, API key, or sign-up on first run; and `ocp_` built-in provider install tokens are now redacted from logs and kept out of Debug output.

Select speech-to-text and text-to-speech models from any provider using the `stt_model` and `tts_model` voice config fields.

Run a new `userbot-download` CLI command under the `telegram` subcommand to download files via the Telegram userbot.

The TUI gains a full-screen responsive `/usage` screen and a `/mission-control` screen that adapts to terminal width.

The `/models` TUI command now opens on the session's current provider and model, and accepts a full `<provider/model>` argument to switch directly without opening the picker.

+5 more

v0.5.5 NOTES ↗ STABLE

OpenCrabs adds native Discord polls, voice messages, slash commands, per-channel modes, governance policy linting, and ACP v1 session lifecycle across Discord, Slack, and Tel

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.5.5
https://github.com/opencrabs/opencrabs.git
# already have the repo? check out this version:
$ git checkout v0.5.5
```

└──▷ **USE IT**

Restrict a Discord channel so the agent only responds to direct mentions, while keeping a second channel fully open — set per-channel modes without touching global config.

~/.opencrabs/config.toml

```
[agent]
session_notify = false

[channels.discord.channels."#ops-alerts"]
respond_to = "mention"
open = false

[channels.discord.channels."#general"]
respond_to = "all"
open = true
```

Suppress agent session notifications globally — useful when running OpenCrabs as an always-on daemon where notification noise is unwanted.

~/.opencrabs/config.toml

```
[agent]
session_notify = false
```

Adds per-channel Discord respond-to mode control via `channels.discord.channels` in `config.toml` (enforced in the message filter, ACL, and command gate) and the `/respond_to` slash command to set a channel or thread's respond-to mode.

Disable session notifications with the `session_notify` kill switch under `[agent]` in `config.toml`, which defaults off

Adds new Discord capabilities: sending native Discord polls via `send_poll`, automatic projection of `commands.toml` entries onto native Discord slash commands, the `/cowork` command to open a channel or thread to its members, and TTS replies sent as native Discord voice messages.

Telegram now renders mermaid fences inside trailer bubbles via the hooked rich send, delivers markdown links to local files as documents, and gains a new flag to keep Telegram markdown documents out of the rich media array.

> ACP now implements the five v1 session lifecycle methods, publishes the plan as a v1 plan update, and decodes image prompt blocks into IMG file attachments.
>
> +5 more

WAS THIS USEFUL? 👍 👎

runx

Feedback ⓘ cli-v0.11.0 NOTES ↗ LATEST STABLE

Runx is the governed runtime for agent skill workflows, off the leash but on the record

runx adds a secret management command, browser-session connectors, Stripe and PayPal payment connectors, and automatic newest-version resolution for skills and connectors

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/runxhq/runx:0.11.0
```

npm

```
$ npm install -g @runxhq/cli@0.11.0
```

git

```
$ git clone --branch cli-v0.11.0 https://github.com/runxhq/runx.git
# already have the repo? check out this version:
$ git checkout cli-v0.11.0
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

└──▷ TRY IT

Store an API key as a named secret and wire it to a connector so every run picks it up without manual credential setup.

```
$ runx secret set openai-key
runx connector use openai --secret openai-key
```

Query the registry for all published versions of a connector when you need to pin an exact version for a reproducible run.

```
$ curl https://registry.runx.ai/v1/connectors/github/versions
```

NEW INTEGRATION Adds secrets management via `runx secret set <name>`, `runx secret list`, `runx secret rm`, and `runx secret read`, storing values sealed in

the runx home or delegating to 1Password, AWS Secrets Manager, AWS Systems Manager Parameter Store, Vault, the system keychain, or a file via `--from <reference>`; binds a secret to a connector with `runx connector use <connector> --secret <name>`, sets the managed agent's secret with `runx config set agent.secret <name>`, imports an MCP connector secret with `runx connector import-mcp --secret`, and reads a secret programmatically via `secrets.read` in the client library.

Look up a connector's available versions from the registry at `GET /v1/connectors/<name>/versions`, returning results page by page.

Read files by line range with `fs.read` using `start_line` and `line_count`, which also returns the whole file's digest; `fs.edit`, `fs.write`, and `fs.apply_bundle` now return each file's diff.

Packages runx adds now go to `.runx/added-skills` and `.runx/added-connectors`, leaving `skills/` and `connectors/` for user-written code; a local copy answers its name before any registry version.

Browser-session connectors now support running as a user's signed-in browser account, with setup pinning the account and each operation verifying it.

+5 more

NEW INTEGRATION Connector implementations: HTTP, MCP, process tools, and data stores

└──▷ **BREAKING ON UPGRADE**

- ! Each project gets a new kernel store at `.runx/kernel-v2.sqlite`; 0.11.0 reads nothing 0.10 recorded in `.runx/kernel.sqlite`, so connector instances, standing approvals, paused runs, and effect custody start empty. If 0.10 holds a provider effect in flight, 0.11.0 refuses to start the new store and names those acts.
- ! The home config upgrades on first use: each credential profile becomes a secret of its name, the managed agent's key becomes the secret `agent`, and sealed value ids become `material_id`, `api_token_material_id`, and `publish_api_token_material_id`.
- ! `runx connector use --from`, `runx config set api-key --from`, and `--bearer-stdin` are removed; use `runx secret` commands instead.
- ! The `reddit` connector's bind command now takes `--profile` for the browser profile field named `profile`; each account must bind its browser again.
- ! The `sql` connector's connection config and requests now name the configured source `source` (rename `source_ref` to `source` in each `sql` connection config).
- ! Packages runx adds now live under `.runx/`; files previously added under `skills/<owner>/<name>/<version>/` and `connectors/<name>/<version>/`

are treated as user-written code and never updated — delete them and let 0.11.0 re-add them under `.runx/`.

- ! runx no longer reads `MEMORY.md` or `RUNX_MEMORY_PROFILE_PATH` into an act.
- ! Packages published for 0.11.0 may use manifest fields (such as a runner's `description` and the new connector session and payment shapes) that 0.10.2 refuses.
- ! All contract fields follow a single naming convention: `_ref` for a runx reference, `_uri` for a URI, `_id` for a bare id, `_path` for a file path, `_locator` for a provider's locator, every digest as `sha256:<hex>`; `state` for lifecycle position, `status` for check outcome, `outcome` for run result.
- ! Packet ids now follow the form `runx.<name>.v<N>` for core packets and `<publisher>/<package>.<name>.v<N>` for package packets (e.g. `runx/lead-qualify.qualification.v1`), with schema at `packets/<name>.v<N>.schema.json`; runx refuses any other form.
- ! The registry digest of a published bundle's files is now `bundle_digest` (written `sha256:<hex>`); `package_digest` is only ever the execution package digest a run records. The `root package.json` `runx.packets` field is gone, and workspace `dist/packets` no longer resolves package packets.

▾ **3 more releases in this issue** · 2026-10-13 → 2026-10-25

cli-v0.10.2 NOTES ↗ STABLE

runx adds agent sessions, process supervision, secrets manager credential bindings, and paid skill purchasing to its governed skill runtime

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/runxhq/runx:0.10.2
```

npm

```
$ npm install -g @runxhq/cli@0.10.2
```

git

```
$ git clone --branch cli-v0.10.2 https://github.com/runxhq/runx.git
# already have the repo? check out this version:
$ git checkout cli-v0.10.2
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

└──▷ **TRY IT**

Pull a connector credential from 1Password (or another secrets manager) so runx never stores the secret value itself.

```
$ runx connector use github --from op://Personal/GitHub/token
```

Cap what a run and its child skills can spend on paid registry calls, refusing any call that would exceed the limit.

```
$ runx skill publisher/my-skill --max-cost 500 --json
```

Read a full run record — its summary, every request digest, and stream events — for audit or replay.

```
$ runx history my-run-id --json
```

Adds `runx connector use <connector> --from <reference>` to bind a connector credential from 1Password, AWS Secrets Manager, AWS Systems Manager Parameter Store, Vault, the system keychain, or a file, read via that manager's own CLI before every call and never stored, and `runx connector use --hosted --from <reference>` to seed a Connect grant from a secrets manager.

Inspect a run as `runx.run_read.v1` — its summary, every request with its digest, and events after a per-stream cursor — with `runx history <run-id> --json`.

Adds run authority controls: `runx skill --max-cost` or a run authority's `max_cost_micro` field caps per-run spending, refusing any call that would exceed the limit with `cost_limit_exceeded`; and `agent_bounds` requires approval before any workspace edit or process execution, showing the unified diff or argv of each change as a standing or per-step approval.

Validate up to 128 values against a bounded JSON Schema 2020-12 schema using the new `data.validate` tool, returning schema and value digests plus error paths without copying private values.

Include the complete-package content digest in `runx registry package --json` via the new `registry.content_digest` field.

+4 more

└──▷ **BREAKING ON UPGRADE**

- ! Resume documents must be `runx.resume.v1`. Each entry under `answers` must be `{request_digest, value}` and each entry under `approvals` must be `{request_digest, approved, reason}`, carrying the digest of the request it answers. A claimed call's retry is now `{request_digest, value: {retry: true}}` in the answers lane; the answers lane can no longer answer an approval.
- ! A run id may not begin with `sha256:`.
- ! A managed agent now requires `agent.model`; a managed agent whose provider has no `agent.model` is refused.
- ! Connectors are named by name, not by publisher-qualified refs: `coinbase-cdp` becomes `coinbase/cdp`; `google-analytics`, `google-calendar`, `google-search-console`, and `google-sheets` become `google/analytics`, `google/calendar`, `google/search-console`, and `google/sheets`; `x402` becomes `wallet/evm`. Connector installs now live at `connectors/<name>/<version>/`.
- ! Registry skill names have changed: `agency` to `team-delegate`, `content-pipeline` to `content-package`, `contract-drafter` to `contract-draft`, `crm-cleanup` to `crm-update-plan`, `data-subject-request` to `privacy-request-review`, `deep-research` to `research-brief`, `github-sync` to `github`, `helpdesk` to `support-triage`, `issue-intake` to `code-intake`, `issue-to-pr` to `github-issue-to-pr`, `issue-triage` to `github-issue-triage`, `list-hygiene-judge` to `contact-consent-review`, `nws-weather-forecast` to `nws-weather`, `open-meteo-weather-forecast` to `open-meteo-weather`, `ops-desk` to `operations-review`, `postmortem-maker` to `postmortem-draft`, `purchase-approval` to `purchase-policy-check`, `sbom-maker` to `sbom-generate`, `schema-guard` to `schema-check`, `settle-invoice` to `invoice-pay`, `sql-analyst` to `sql-analysis`, and `work-plan` to `code-plan`. The old names are not aliases.
- ! The skills `marketplace-invoke`, `vault-unseal`, `vuln-disclosure`, and `weather-forecast` are removed, as are the connectors `mpp`, `stripe`, `stripe-spt`, and `vault`.
- ! The bundled skills `spend`, `x402-pay`, `stripe-pay`, `send-as`, `conversation-reply`, `slack-notify`, `github-pr-comment`, `ecosystem-brief`, `n8n-handoff`, `zapier-handoff`, and `data-store` are removed; use `pay`, `message-send`, `github`, `research-brief`, `webhook-handoff`, and direct `store` operations instead.
- ! `plugin.json` namespace files are no longer supported.
- ! Current catalog aliases for renamed packages are removed; existing receipt bindings keep their original bytes.

`cli-v0.10.1` NOTES ↗ STABLE

runx adds automatic registry skill fetching, hosted catalog access, and connector credential binding by environment variable

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/runxhq/runx:0.10.1
```

npm

```
$ npm install -g @runxhq/cli@0.10.1
```

git

```
$ git clone --branch cli-v0.10.1 https://github.com/runxhq/runx.git
# already have the repo? check out this version:
$ git checkout cli-v0.10.1
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

└──▷ **TRY IT**

Search the hosted catalog for a skill when no local registry is configured.

```
$ runx registry search weather --json
```

Bind a connector credential to its declared environment variable (such as `EXA_API_KEY`) with `runx connector use <connector>` — the variable's value is read on every call, never stored, so a rotated key needs no rebinding; with the variable unset, the CLI prompts without echo.

Open a hosted grant for any publisher's connector with `runx connector use <connector> --hosted`, not only runx's own connectors.

Reach the hosted catalog at `https://api.runx.ai` from `runx registry search`, `runx registry install`, and `runx registry publish` when no registry is named; override with `--registry <dir>` or the `RUNX_REGISTRY_DIR` environment variable.

Reference a run's own ID in an operation step's `idempotency_key` field using `$run.id`, so a write that must not repeat within a single run executes exactly once.

Allow a connector's declared request to send the paid invocation token `{"$runx": "invocation"}` to let a vendor confirm a sale through runx Connect.

+2 more

└──▷ **BREAKING ON UPGRADE**

! With no registry named, `runx registry search`, `runx registry install`, and `runx registry publish` now reach the hosted catalog at `https://api.runx.ai` instead of the local registry cache. Use `--registry <dir>` or `RUNX_REGISTRY_DIR` to target a local registry.

! Opening a hosted grant requires runx 0.10.1; runx Connect refuses the request sent by 0.10.0 with 'upgrade runx to 0.10.1'. Existing grants continue to work.

`cli-v0.10.0` NOTES ↗ STABLE

runx 0.10.0 rebuilds around a connector system with declared authority, a unified `run.yaml` skill graph, MCP agent turns, idempotent run submission, and verifiable signed receipts.

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/runxhq/runx:0.10.0
```

npm

```
$ npm install -g @runxhq/cli@0.10.0
```

git

```
$ git clone --branch cli-v0.10.0 https://github.com/runxhq/runx.git
# already have the repo? check out this version:
$ git checkout cli-v0.10.0
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

Bind, list, revoke and import connector authority with `runx connector list|inspect|use|revoke|import-mcp`; `use` accepts `--session`, `--credential <name>`, or `--hosted` (which prints an authorization link and waits) to set up an instance, profile, or new Connect grant.

Install a skill or connector by name from the registry with `runx registry install <publisher>/<name>[@version]`; a skill automatically brings the connectors it binds.

Serve agent turns of a paused run as MCP tools over stdio with `runx mcp --run <run-id>`, scoped to that run alone.

Pass caller context into a run with `runx skill --context <json>`, which sets the `current_context` and `historical_context` entries an agent context envelope carries without granting authority.

Pin a run's submission identity with `runx skill --run-id <id>` so that resubmitting continues the same run and a sealed run returns its sealed result, preventing duplicate step execution.

+7 more

└─▷ **BREAKING ON UPGRADE**

- ! The 0.9 skill, credential and receipt contracts are broken; migration is required before upgrading.
- ! `X.yaml` execution profiles are removed; skills using them must be rewritten.
- ! The runner source kinds `cli-tool`, `javascript`, `mcp`, `external-adapter`, `thread-outbox` and `approval` are removed.
- ! The graph-language `fanout:` groups, step `retry:`, `charter_from` and `mint` are removed; fan-out and retry now belong to the driving agent and host.
- ! `runx data migrate`, `runx skill -p` and `--credential-profile`, and `runx login --from-gh` are removed.
- ! A credential's `--auth-mode` and `--audience` flags are removed; a connector's manifest now declares where its credential may be sent.
- ! `runx dev`, its fixture lanes and the `runx.dev.v1` report are removed; proofs now go through `runx harness`.
- ! `runx kernel eval` and `runx parser eval` are removed from the CLI.
- ! `runx verify --allow-local-development-signatures` is removed; use `--trust home|configured` instead.
- ! The `--receipts` spelling of `--receipt-dir` is removed (`-R` still works).
- ! `runx add`, `runx registry read`, `runx registry resolve` and the `runx registry connector` verbs are removed.
- ! `.runx/continuations` is removed; `runx resume` now reads answers from one document on stdin or a named file.
- ! Graph references now read values, not artifact envelopes.
- ! Approvals are now addressed by the step that asks (`<child>#<step>` below the root), not by prior addressing schemes.
- ! The `lead-router` skill is replaced by `lead-qualify`.

WAS THIS USEFUL?  

narrafork

Feedback ⓘ v0.9.8 NOTES ↗ LATEST STABLE

Narrafork is an efficient durable harness for collaboration across people, models, and devices

NarraFork adds Bot-mode narrators, text-selection actions, grouped global search, and opaque server-side session tokens replacing JWTs

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.8 https://github.com/NarraFork/NarraFork.git
# already have the repo? check out this version:
$ git checkout v0.9.8
```

NarraFork now adds Bot-mode narrators, each with a dedicated chat room and narrator composer, where Bots coordinate delegated work rather than executing tasks directly.

Import provider configurations from an existing local installation during first-time setup or when adding a new provider.

Add quote and ask-in-passing actions for selected text, plus reasoning-effort and permission-mode controls in the dashboard composer.

Global search results now group by session, with excerpts, filters, pagination, and navigation to the matching row; independent-term and two-character searches are now improved.

Store and render conversation content as native blocks, use a shared provider-history projection, and move search reindexing and legacy cleanup into paced background jobs.

+2 more

└──▷ BREAKING ON UPGRADE

! Existing session JWTs are no longer accepted: all users and integrations that use a login-session token must sign in again after upgrading.

! Narrative lines, chapters, the story graph, ruler, chapter review/fork/merge/split, and `project.db` synchronization are retired; old chapter sessions become ordinary sessions and chapter metadata, review conclusions, edges, and commit records are deleted.

! The former Chapters & Containers settings page becomes Workspace & Containers, removing chapter-specific settings.

! Automatically created projects are now public: sessions with project visibility are readable by any logged-in user who can access the project. Review session visibility

and working directories after upgrading.

! Database migration is included: stop active runs in chapter sessions and back up `~/.narrafork/narrafork.db` before upgrading. Legacy messages are converted to block storage by background jobs.

▾ **2 more releases in this issue** · 2026-10-12 → 2026-10-24

v0.9.1 NOTES ↗ STABLE

NarraFork adds bounded request/response rolling recordings and immediate `/save_dump` snapshots with separate storage accounting

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.9.1
https://github.com/NarraFork/NarraFork.git
# already have the repo? check out this version:
$ git checkout v0.9.1
```

Preserve custom, aggregated, plugin and temporarily unavailable model references across settings reloads instead of clearing them during startup.

Improve failed-request diagnostics so error cards can inspect the matching recorded request and response evidence, including partial or truncated recordings.

v0.9.0 NOTES ↗ STABLE

NarraFork adds GitHub as the default update source, TokenDance multi-protocol model access, and sortable paginated worktree lists

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.9.0
https://github.com/NarraFork/NarraFork.git
# already have the repo? check out this version:
$ git checkout v0.9.0
```

Make GitHub the default update source, with fork-aware repository identity, atomic update indexes, verified incremental updates, and explicit support for self-hosted update servers; existing explicit source settings are preserved.

Add dedicated network proxy settings for software updates and auxiliary tools; Ripgrep, zstd, and the remote executor follow the selected distribution source, with integrity-checked caches and system PATH priority.

> Add controlled GitHub Release workflows: build-only by default, eight-platform builds and native smoke checks, immutable artifacts, approval-gated publication, and recovery from the original verified bundle.
>
> Add TokenDance authorization and multi-protocol model access, with safer balance and top-up handling and improved provider setup with popular and regional presets.
>
> Add sortable, paginated worktree lists with smoother loading, plus undo for individually closed recent tabs.
>
> +1 more

WAS THIS USEFUL?  

◆ MCP Servers & Tooling

premiere-pro-mcp

Feedback ⓘ v1.30.0 NOTES ↗

Adobe Premiere Pro MCP — independent, local-first server for supported workflows. Connect Claude, Codex, or Cursor through a local CEP bridge.

premiere-pro-mcp adds compact CEP recovery inspection mode and stateless HTTP idempotency key sharing across requests

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g premiere-pro-mcp@1.30.0
```

git

```
$ git clone --branch v1.30.0
https://github.com/leancoderkavy/premiere-pro-mcp.git
# already have the repo? check out this version:
$ git checkout v1.30.0
```

signed build provenance · checked 8 Nov Wrong command?

└──▷ USE IT

When inspecting CEP recovery state without needing the full snapshot, pass `include_snapshot: false` to cut down response size.

json

```
{ "include_snapshot": false }
```

Reduce CEP recovery inspection payload size by setting `include_snapshot: false` on the recovery inspection tool.

Timeline health and snapshot comparison tools now accept raw sequence inspection results as direct input.

WAS THIS USEFUL? 👍 👎

claudexor

Feedback ⓘ v5.1.0 NOTES ↗

Claudexor is a multi-harness control plane for Claude Code, Codex, Cursor, and OpenCode with quota-aware rotation across multiple subscriptions.

Claudexor adds a typed `turn_limit_reached` outcome, managed run-data lifetimes, exact thread export, and conditional thread retirement in v5.1.0

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v5.1.0 https://github.com/razzant/claudexor.git
# already have the repo? check out this version:
$ git checkout v5.1.0
```

└──▷ **TRY IT**

Prepare the publication scanner for npm-backed `pr` delivery in an environment where the npm registry is available.

```
$ claudexor scanner prepare
```

Prepare the publication scanner offline from an official archive when the npm registry is not reachable.

```
$ claudexor scanner prepare --from-archive <file>
```

Claudexor now manages lifetimes for newly uploaded run data.

Thread export is now exact, and threads can be conditionally retired.

Scanner preparation for npm installs is now explicit, callable with `claudexor scanner prepare` or `claudexor scanner prepare --from-archive <file>`.

The MCP Accounts tool now projects newer engines' responses through its published schema.

The macOS Accounts row now shows a model-scoped quota window when no account-wide quota is reported.

+2 more

└──▷ **BREAKING ON UPGRADE**

! Opening an existing engine store migrates it from schema 6 to 7; the migration is forward-only, and returning to an older binary does not undo it.

WAS THIS USEFUL?  

gflow-cli

Feedback ⓘ v0.96.0 NOTES ↗ LATEST STABLE

Gflow-cli is a drive Google Flow from the command line: Veo video and Imagen images, scripted, batched and pipeline-ready. Ships an MCP server so coding agents can drive it too, giving you and your agents full programmatic control over generation.

gflow-cli adds Omni 1.2 Flash video generation with 3–15s durations and r2v support at 4s, 6s, and 10s

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install gflow-cli==0.96.0
# already installed? upgrade to this version:
$ pipx install --force gflow-cli==0.96.0
```

git

```
$ git clone --branch v0.96.0 https://github.com/ffroliva/gflow-cli.git
# already have the repo? check out this version:
$ git checkout v0.96.0
```

PyPI attestation · checked 8 Nov Wrong command?

▾ 1 more release in this issue · 2026-11-07

v0.95.0 NOTES ↗ STABLE

gflow-cli adds a telemetry system with a first-run consent notice and an on/off toggle

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install gflow-cli==0.95.0
# already installed? upgrade to this version:
$ pipx install --force gflow-cli==0.95.0
```

git

```
$ git clone --branch v0.95.0 https://github.com/ffroliva/gflow-cli.git
# already have the repo? check out this version:
$ git checkout v0.95.0
```

PyPI attestation · checked 8 Nov Wrong command?

WAS THIS USEFUL? 👍 👎

magic

Feedback ⓘ v24.4.17 NOTES ↗ LATEST STABLE

Magic is an instant SECURE Full Stack Apps and AI Agents

magic adds a single dialog to configure both backend and frontend URLs together

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v24.4.17 https://github.com/polterguy/magic.git
# already have the repo? check out this version:
$ git checkout v24.4.17
```

▾ 2 more releases in this issue · 2026-11-07

v24.4.16 NOTES ↗ STABLE

magic adds image lightbox zoom and JSON array item counts to the Chat Ops interface

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v24.4.16 https://github.com/polterguy/magic.git
# already have the repo? check out this version:
$ git checkout v24.4.16
```

Chat Ops now displays a lightbox when you click an attached image, allowing you to zoom in without leaving the conversation.

Chat Ops now shows the item count when an AI response is a JSON array, giving a quick sense of result size at a glance.

v24.4.15 NOTES ↗ STABLE

magic adds a [whitelist] slot restriction mechanism to prevent privilege escalation via signal grants

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v24.4.15 https://github.com/polterguy/magic.git
# already have the repo? check out this version:
$ git checkout v24.4.15
```

> Declare slots that a `[whitelist]` can never grant, blocking privilege escalation paths in signal-based execution.

WAS THIS USEFUL? 👍 👎

3dsmax-mcp

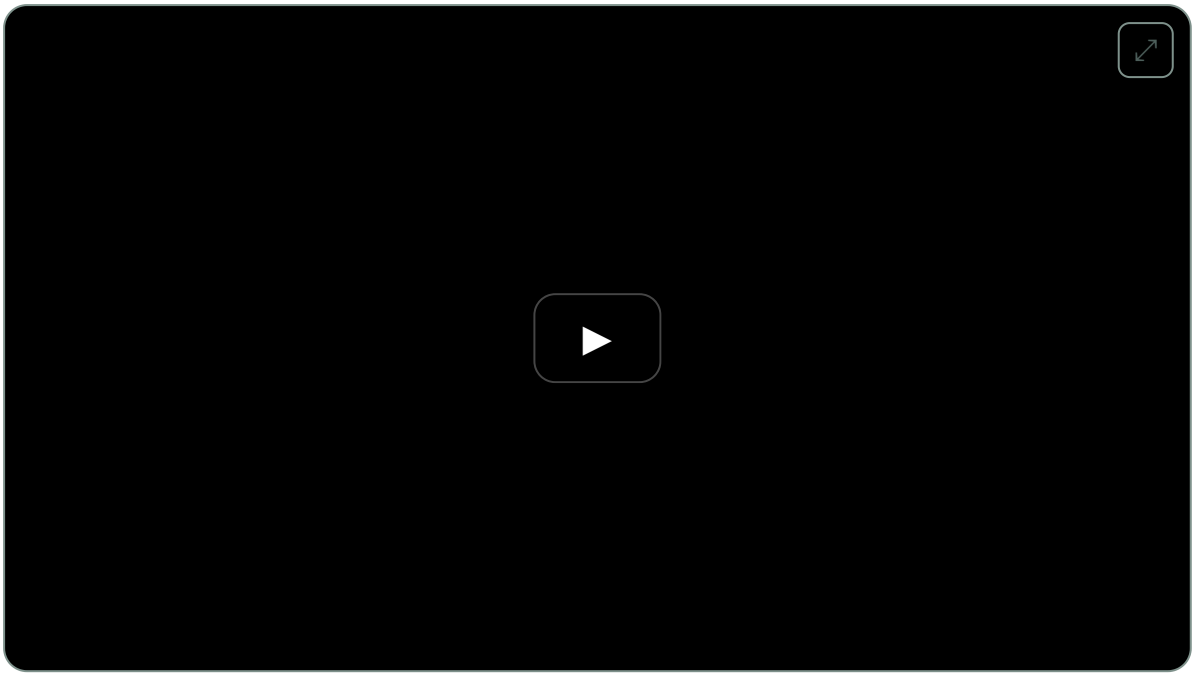
Feedback ⓘ v1.7.8 NOTES ↗

3dsmax-mcp is a 3ds Max MCP server

3dsmax-mcp adds tyFlow C# script compilation, flow graph reading, and image-based Cosmos asset search

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.7.8 https://github.com/cl0nazepamm/3dsmax-mcp.git
# already have the repo? check out this version:
$ git checkout v1.7.8
```



└──▷ TRY IT

Find Cosmos assets that visually match a local reference image, with real-world sizes included, to quickly source geometry that fits a scene.

```
$ cosmos_search(image_path='/renders/reference_chair.jpg', sizes=true)
```

Compile and apply C# against the tyFlow script API with `tyflow_script`, which reports line and column errors, Print() output, and exceptions — none of which tyFlow surfaces on its own.

Read tyFlow wiring and operator types directly from the flow with `get_tyflow_graph`, including hand-drawn edges; Shape, Birth Burst, Actor, and Spline Paths can now be

renamed, enabled, and removed like any other operator.

Search Cosmos assets by visual similarity to a local image with cosmos_search(image_path=...), returning best matches first; add `sizes=true` to include each model's real-world dimensions (requires Cosmos Browser 2026.11.04).

Return a single labeled contact sheet instead of full-size previews with cosmos_search(thumbnails=true).

WAS THIS USEFUL? 👍 👎

tableau-mcp

Feedback ⓘ v5.5.0 NOTES ↗ LATEST STABLE

Tableau's official MCP Server. Helping agents see and understand data.

tableau-mcp adds environment variables to toggle Knowledge tools, a flag for claude.ai MCP Apps access, and workbook extension listing for Desktop

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/tableau/tableau-mcp:5.5.0
```

git

```
$ git clone --branch v5.5.0 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v5.5.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Control whether Knowledge tools are active by setting the new environment variables added for enabling or disabling them.

Let claude.ai access MCP Apps tools by enabling the new flag added for that purpose.

List workbook extensions in Tableau Desktop with the new `list workbook extensions` capability.

Scaffold and export workbook data apps from Tableau Desktop.

tableau-mcp now syncs with External Client API 0.2.31, using its routes instead of raw commands.

▾ 7 more releases in this issue · 2026-10-10 → 2026-11-04

v5.1.1 NOTES ↗ STABLE

tableau-mcp adds Desktop authoring integration, a skills/get method, and new External Client API tools

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/tableau/tableau-mcp:5.1.1
```

```git
$ git clone --branch v5.1.1 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v5.1.1
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Adds tools for External Client API 0.2.25 endpoints, updating Desktop support to the latest client version.

Adds a `skills/get` method to the MCP server for retrieving skills.

Integrates Tableau Desktop authoring into the main MCP server.

Returns server and site info for datasources in the scaffold data app tool.

Adds `Data/` and `Resources/.agents/` directories to the data app template.

v4.29.1 NOTES ↗ STABLE

tableau-mcp adds Flow tools on by default, .tds/.tdsx publish tools, external OAuth token support, and row-limit transparency across query tools

└──▷ GET THIS VERSION ▾

```Docker
$ docker pull ghcr.io/tableau/tableau-mcp:4.29.1
```

```git
$ git clone --branch v4.29.1 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v4.29.1
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ USE IT

Restrict an MCP server instance to read-only Flow access by enabling read tools but disabling write tools via config.

```yaml
flow-read-tools: true
flow-write-tools: false
```

Enable or disable Tableau Flow tools independently using the `flow-read-tools` and `flow-write-tools` flags, now on by default and overridable via config.

Publish `.tds` and `.tdsx` data source files to Tableau Server with new publish tools.

Accept OAuth tokens from an external authorization server for authentication.

Register Pulse insight bundles automatically when Pulse is enabled.

Query tools now report when a row limit cut the results short, covering `query-datasource` and `query-admin-insights`.

+1 more

v4.27.2 NOTES ↗ STABLE

tableau-mcp now scopes embedded datasource queries by workbook with `workbookLuid`

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.27.2 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v4.27.2
```

Query embedded datasources scoped to a specific workbook by passing `workbookLuid` as a filter parameter.

v4.26.3 NOTES ↗ STABLE

tableau-mcp now serves admin prompt workflows as model-readable MCP resources and lowers the scaffold-data-app version floor to 2026.3.0

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.26.3 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v4.26.3
```

Lower the minimum supported version for `scaffold-data-app` to `2026.3.0`, enabling use on earlier Tableau deployments.

Admin prompt workflows are now served as model-readable MCP resources, making them accessible to AI models via the MCP protocol.

Flow tools are now grouped by operation, improving discoverability in MCP tool listings.

v4.25.4 NOTES ↗ STABLE

tableau-mcp now gates Knowledge write operations independently from reads

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.25.4 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v4.25.4
```

Gate Knowledge write operations independently from read operations, allowing separate access control over Knowledge tool reads and writes.

Improve Knowledge tool descriptions and schemas, and add delete/create guards to prevent unintended mutations.

v4.25.3 NOTES ↗ STABLE

tableau-mcp adds skills listing, activity logging, admin-tools flag, Personal Space workbook moves, and data-app scaffolding

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.25.3 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v4.25.3
```

Scaffold data apps from a JSON `manifest.json` using the `scaffold-data-app` action.

Move workbooks into Personal Space using the `move-workbook` action.

Manage pending-deletion tags using the `delete-content` action.

List available agent skills with the new `skills/list` endpoint.

Record MCP tool call events to Tableau's Activity Log for audit visibility.

v4.22.3 NOTES ↗ STABLE

tableau-mcp adds a skills registry over MCP, personal-space publishing, permission-rule disclosure, and a project-capability filter to list-projects

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/tableau/tableau-mcp:4.22.3
```

git

```
$ git clone --branch v4.22.3 https://github.com/tableau/tableau-mcp.git
# already have the repo? check out this version:
$ git checkout v4.22.3
```

published under the project's GitHub account · checked 8 Nov **Wrong command?**

└──▷ **USE IT**

Let Slack users publish and download workbooks by enabling the authoring-with-slack flag alongside authoring-tools in your server config.

yaml

```
authoring-tools: true
authoring-with-slack: true
```

Enable access to authoring tools from Slack clients by turning on the new `authoring-with-slack` flag alongside `authoring-tools`; without it, `download-workbook`, `publish-workbook`, and `request-workbook-upload` remain blocked for Slack.

Filter projects by publish permission with `capability: "Write"` on `list-projects`, which now requires REST API 3.30+; on older servers the argument is hidden.

Publish a workbook to the caller's Personal Space via the new `personalSpace` option on `publish-workbook` (defaults to `true` when `data-apps` is on); an explicit `projectId` always takes precedence.

Read workbook permission rules after a successful project publish: the result now includes a `permissions` field (user and group rules) using the optional `tableau:permissions:read` scope, or a `permissionsNote` if consent is missing.

Query a datasource without parent-project permissions: `query-datasource` now falls back through the Metadata API and `GET /datasources?filter=name:eq:<name>` on a 403, honoring `INCLUDE_TAGS` and `INCLUDE_PROJECT_IDS` allowlists.

+4 more

└──▷ **BREAKING ON UPGRADE**

- ! `scaffold-data-app` now requires the `tableau:packages:read` OAuth scope; users with existing tokens must sign in again, and tokens without this scope receive HTTP 403 `insufficient_scope` if scope enforcement is on.
- ! `publish-workbook` no longer requests `tableau:content:read`; Connected Apps must now grant `tableau:projects:read`, and the full required set is `tableau:workbooks:create`, `tableau:file_uploads:create`, and `tableau:projects:read`.
- ! `publish-workbook` rejects calls without a `projectId` when the `data-apps` flag is off, including calls from clients holding an older cached tool schema.
- ! Changes to file-based feature flags only take effect after a server restart; clients must reconnect to see updated tool schemas.

WAS THIS USEFUL? 👍 👎

open-computer-use

Feedback ⓘ v1.1.3 NOTES ↗ LATEST

STABLE

Open Computer Use – Open-Source Alternative to Codex Computer Use

open-computer-use adds macOS text selection and opt-in virtual displays, with wider cross-platform input and observation improvements

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g open-computer-use@1.1.3
```

git

```
$ git clone --branch v1.1.3 https://github.com/iFurySt/open-computer-use.git
# already have the repo? check out this version:
$ git checkout v1.1.3
```

signed build provenance · checked 8 Nov Wrong command?

Virtual displays now require explicit opt-in rather than being enabled by default.

open-computer-use now supports optional accessibility-tree or screenshot collection across platforms.

open-computer-use now improves Windows scrolling and scaled-display input handling.

└──▷ BREAKING ON UPGRADE

! Virtual displays now require explicit opt-in; setups that relied on virtual displays being active by default will need to update their configuration.

▾ 2 more releases in this issue · 2026-10-10 → 2026-11-01

v1.1.0 NOTES ↗ STABLE

open-computer-use adds macOS locked-screen GUI automation, targeted accessibility queries, and persistent screenshot format configuration

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.1.0 https://github.com/iFurySt/open-computer-use.git
# already have the repo? check out this version:
$ git checkout v1.1.0
```

└──▷ **TRY IT**

Set up GUI automation to keep running on a Mac that may lock during long agent tasks.

```
$ ocu locked-use enable
```

Audit required permissions in CI or a script without triggering the interactive onboarding wizard.

```
$ ocu doctor --json
```

Enable macOS locked-screen GUI automation with `ocu locked-use enable`, which installs signed components, grants permissions, and runs GUI automation behind a full-screen display shield while the Mac is locked — tasks relock the Mac and remove shields on completion.

Check permissions in machine-readable form with `ocu doctor --json`, which returns structured JSON without opening the onboarding UI.

Persist screenshot output format — PNG, JPG, or WebP — using `ocu config`, stored in `~/.config/ocu/config.json`.

v1.0.0 NOTES ↗ STABLE

open-computer-use adds a persistent JavaScript REPL, native MCP compatibility, and global configuration via `~/.config/ocu/config.json` in its v1.0.0 release

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.0.0 https://github.com/iFurySt/open-computer-use.git
# already have the repo? check out this version:
$ git checkout v1.0.0
```

└──▷ **USE IT**

Connect any MCP-compatible AI agent (e.g. Gemini CLI) to open-computer-use by wiring it in as an MCP server.

```json
{
  "mcpServers": {
    "open-computer-use": {
      "command": "open-computer-use",
      "args": ["mcp"]
    }
  }
}
```

Run a multi-step computer-use sequence (get app state, then press Return) in one process so element index state is reused across calls.

```
$ ocu call --calls '[{"tool":"get_app_state","args":
{"app":"TextEdit"}},{"tool":"press_key","args":
{"app":"TextEdit","key":"Return"}}]' --sleep 0.5
```

Register open-computer-use with Codex in one step so it is available immediately as a Computer Use MCP tool.

```
$ ocu install-codex-mcp
```

Install open-computer-use into Codex, Claude Code, Gemini CLI, or opencode with dedicated subcommands: `install-codex-mcp`, `install-codex-plugin`, `install-claude-mcp`, `install-gemini-mcp` (with optional `--scope user`), and `install-opencode-mcp`, each writing to the respective agent config file.

Call a single Computer Use tool and print MCP-style JSON output with `ocu call <tool>` and pass arguments with `--args`, or run a multi-step sequence with `--calls` (inline JSON) or `--calls-file` (a JSON file path), with `--sleep` controlling the delay between steps.

Check and diagnose missing macOS permissions with `ocu doctor`, and query permission status as machine-readable JSON with `ocu doctor --json`.

Manage global configuration stored at `~/.config/ocu/config.json` with the new `ocu config` subcommands, available from v1.0.0.

Adds a persistent JavaScript REPL for scripting computer-use sessions.

+3 more

WAS THIS USEFUL? 👍 👎

proxmoxmcp-plus

Feedback ⓘ v0.6.8 NOTES ↗ LATEST STABLE

Proxmoxmcp-plus is an use MCP and OpenAPI to safely control Proxmox VE VMs, LXCs, backups, and snapshots from LLMs and AI agents.

ProxmoxMCP-Plus adds MCP tool annotations — readOnlyHint, destructiveHint, idempotentHint and openWorldHint — across all 65 built-in tools so clients can distinguish read-onl

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.8
```

git

```
$ git clone --branch v0.6.8 https://github.com/RekklesNA/ProxmoxMCP-Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.8
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Advertise `readOnlyHint`, `destructiveHint`, `idempotentHint` and `openWorldHint` MCP annotations on all 65 built-in tools so MCP clients that support these hints can distinguish read-only queries from mutations.

Annotate Code Mode's three public tools: local search and schema lookup are marked read-only and closed-world; execution is marked potentially destructive, non-idempotent and open-world because it can invoke mutating domain tools.

▾ 7 more releases in this issue · 2026-10-09 → 2026-10-29

v0.6.7 NOTES ↗ STABLE

ProxmoxMCP-Plus adds optional `purge` parameter to `delete_vm` and `delete_container` to remove HA resources and backup/replication references on deletion

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.7
```

git

```
$ git clone --branch v0.6.7 https://github.com/RekklesNA/ProxmoxMCP-Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.7
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Delete VMs and containers with optional HA and reference cleanup by passing `purge=true` (MCP) or `purge=True` (Python) to `delete_vm` and `delete_container`; defaults to `purge=false`, preserving previous behavior.

The `purge` argument is keyword-only in Python callers, so existing positional approval tokens cannot accidentally trigger additional cleanup.

v0.6.6 NOTES ↗ STABLE

ProxmoxMCP-Plus adds opt-in Compose profiles for OpenAPI and native MCP HTTP modes with explicit bind-address overrides

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.6
```

git

```
$ git clone --branch v0.6.6 https://github.com/RekklesNA/ProxmoxMCP-Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.6
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ TRY IT

Start only the native MCP HTTP service, binding it to a specific LAN address instead of the default IPv4 loopback.

```
$ MCP_HTTP_BIND_ADDRESS=192.168.1.10 docker compose --profile mcp-http up -d --build --force-recreate proxmox-mcp-http
```

Migrate from the old default OpenAPI service to MCP-only mode by stopping the old container and starting the new profile.

```
$ docker compose --profile openapi stop proxmox-mcp-api
docker compose --profile mcp-http up -d --build --force-recreate
proxmox-mcp-http
```

Put the OpenAPI service behind the `openapi` Compose profile and the native MCP HTTP service behind the `mcp-http` profile, making both opt-in so plain `docker compose up -d` selects no services by default.

Harden both example services with `no-new-privileges` and all Linux capabilities dropped.

└──▷ **BREAKING ON UPGRADE**

- ! Plain `docker compose up -d` now selects no services; existing containers keep their old configuration until explicitly recreated with `--profile openapi` or `--profile mcp-http`.
- ! MCP-only deployments no longer require `PROXMOX_API_KEY`; the cross-mode required-variable interpolation has been removed, so each mode now only enforces its own credential.

v0.6.5 NOTES ↗ STABLE

ProxmoxMCP-Plus adds a `download_template` tool to fetch LXC container templates from HTTPS sources via the Proxmox `download-url` API

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.5
```

git

```
$ git clone --branch v0.6.5 https://github.com/RekklesNA/ProxmoxMCP-
Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.5
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ **USE IT**

Restrict `download_template` to require explicit approval before it runs, so template downloads are gated like other high-risk operations.

command_policy.yaml

```
high_risk_operations:
  - download_template
```

Download LXC container templates over HTTPS with the new `download_template` tool, which calls the native Proxmox `download-url` API with `content=vztmpl`, validates identifiers and archive names, verifies optional SHA-256/SHA-512 digests, requires the target storage to be enabled and active, and rejects an already-existing destination volume.

Apply target-specific high-risk, read-only, and client-grant policy to `download_template` by adding it to `command_policy.high_risk_operations` in your policy configuration; approval metadata is accepted but not forwarded to Proxmox.

Task tracking for `download_template` now reports submission separately from completion, preserves UPIDs when local tracking fails, classifies uncertain responses as unsuccessful and non-retryable, and cancels via the native task `DELETE` endpoint.

v0.6.4 NOTES ↗ STABLE

ProxmoxMCP-Plus adds offline QEMU VM migration, cloud image import, and idempotent serial console configuration tools

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.4
```

git

```
$ git clone --branch v0.6.4 https://github.com/RekklesNA/ProxmoxMCP-Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.4
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ USE IT

Extend your command policy to enforce approval for all three new high-risk tools before allowing workers to process migration or import requests.

```yaml
high_risk_operations:
  - migrate_vm
  - import_cloud_image
  - configure_vm_serial_console
```

Migrate stopped QEMU VMs offline with new `preflight_vm_migration` and `migrate_vm` tools, reading native Proxmox migration preconditions alongside VM ownership, HA, storage, and network evidence; rejects local ISO media and unsupported host/device dependencies.

Import already-uploaded raw, qcow2, or vmdk volumes on import-enabled storage with new `import_cloud_image`, submitting an asynchronous Proxmox `import-from` update with the current config digest into an unused SCSI slot without URL downloads, disk overwrites, or boot-order changes.

Configure socket-backed ttyS0 and serial VGA output idempotently with new `configure_vm_serial_console`, preserving other serial devices.

Add `migrate_vm`, `import_cloud_image`, and `configure_vm_serial_console` to `command_policy.high_risk_operations` to apply approval policy to these new tools.

└──▷ **BREAKING ON UPGRADE**

! Custom `command_policy.high_risk_operations` lists must be updated to include `migrate_vm`, `import_cloud_image`, and `configure_vm_serial_console` or these new tools will not be subject to your existing approval policy.

v0.6.3 NOTES ↗ STABLE

ProxmoxMCP-Plus adds node diagnostics, cluster join info, and guarded container migration tools in v0.6.3

└──▷ **GET THIS VERSION** ▾

```Docker
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.3
```

```git
$ git clone --branch v0.6.3 https://github.com/RekklesNA/ProxmoxMCP-Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.3
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ **TRY IT**

Before moving a container, run a dry-run readiness check to surface storage, HA, and mount issues without committing anything.

```
$ preflight_container_migration
```

Submit a container migration with explicit migration and downtime confirmations so the tool applies the target node's high-risk approval policy.

```
$ migrate_container confirm_migration=true confirm_downtime=true
```

Add `migrate_container` to a custom high-risk operations list so your approval policy gates every migration submission.

```yaml
high_risk_operations:
  - migrate_container
```

Inspect node clock and timezone with the new `get_node_time` tool, storage capacity with `get_node_storage`, and CPU, memory, storage and RRD I/O metrics with `get_node_diagnostics`; missing optional metrics return partial results rather than a false completeness claim.

Retrieve public cluster metadata and certificate fingerprints — without credentials or unrecognized extensions — with the new `get_cluster_join_info` tool.

Assess container migration readiness before committing with the new read-only `preflight_container_migration` tool, which checks node availability, guest state, rootfs storage and capacity, mounts/devices, HA membership and backup inventory.

Submit container migrations with the new `migrate_container` tool, which requires `confirm_migration=true` and, when downtime is involved, `confirm_downtime=true`; returns the submitted task UPID and an optional local job

ID, and responds with `SUBMISSION_UNCERTAIN` on ambiguous submission outcomes.

`migrate_container` now applies the target node's configured `command_policy.high_risk_operations` approval policy and never automatically retries a migration.

+1 more

└──▷ **BREAKING ON UPGRADE**

! The default `command_policy.high_risk_operations` list now includes `migrate_container`; if you supply a custom list, add `migrate_container` explicitly or your configured approval policy will not apply to it.

v0.6.1 NOTES ↗ STABLE

ProxmoxMCP-Plus adds poll-result caching, queue limits, latency histograms, and an opt-in independent Proxmox API session pool

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.1
```

git

```
$ git clone --branch v0.6.1 https://github.com/RekklesNA/ProxmoxMCP-Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.1
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ **USE IT**

Raise the session pool to 4 to allow parallel Proxmox API calls under heavy concurrent tool use, while bounding how long a caller waits for a free session.

yaml

```yaml
proxmox:
  session_pool_size: 4
  session_pool_timeout: 60
mcp:
  queue_limit: 128
  queue_timeout: 45
```

Disable the poll-result cache for a specific job to get an immediate backend refresh when you need the latest task state.

```
$ poll_job(force=true)
```

Limit the number of queued MCP tool calls with `mcp.queue_limit` (default 64 per target) and cap how long they wait with `mcp.queue_timeout` (default 30 seconds); overflow and expired waits return retry-later errors before execution.

Enable independent Proxmox API session pooling by setting `proxmox.session_pool_size` or a named-target `session_pool_size` (range 1–32, default 1); `session_pool_timeout` bounds pooled waits (default 30 seconds).

Cache same-job poll results for `jobs.poll_cache_ttl` seconds (server default 1; set to 0 to disable) to reduce repeated backend calls; bypass the cache with poll_job(force=true) on MCP or OpenAPI.

Adds cumulative, fixed-bucket latency histograms in seconds, with JSON snapshots exposing approximate p95/p99 (null when empty or overflowing), covering dispatch wait, API-session wait, API requests, and tool execution separately.

v0.6.0 NOTES ↗ STABLE

ProxmoxMCP-Plus adds opt-in SSH node command execution with per-target access controls and client-scoped tool discovery

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/rekklesna/proxmoxmcp-plus:v0.6.0
```

git

```
$ git clone --branch v0.6.0 https://github.com/RekklesNA/ProxmoxMCP-Plus.git
# already have the repo? check out this version:
$ git checkout v0.6.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ **USE IT**

Enable SSH node command execution for a specific target after reviewing privilege implications of any sudo wrapper.

node-command-execution.md

```
ssh:
  allow_node_commands: true
```

Run SSH commands on Proxmox nodes via the new execute_node_command(node, command, approval_token?, target?) function, enabled per target with `ssh.allow_node_commands=true`; checks node inventory, command policy, operation approval, client grants, and read-only restrictions before execution.

Filter Code Mode search, schemas, and callable tools by the current client's grants and configured target access, evaluated per request so one client's discovery results cannot appear in another client's catalog.

Start up with cached job summaries without loading historical audit events, reducing median startup time from 0.960 s to 0.0043 s and peak allocation from ~38 MB to ~68 KB on a 50-job / 50,050-event workload.

WAS THIS USEFUL? 👍 👎

qveris-agent-toolkit

Feedback ⓘ cli-v0.13.0 NOTES ↗

LATEST STABLE

Qveris-agent-toolkit is a bring professional data and tools to the AI you already use: open-source MCP, CLI, SDKs, skills, and docs to find services, call them, and audit usage.

qveris-agent-toolkit adds `qveris auth trial --bounded` for secure pre-registration trial possession with read-first recovery

└─▷ **GET THIS VERSION** ▾

npm

```
$ npm install @qverisai/cli@0.13.0
```

git

```
$ git clone --branch cli-v0.13.0 https://github.com/QVerisAI/qveris-agent-toolkit.git
# already have the repo? check out this version:
$ git checkout cli-v0.13.0
```

signed build provenance · checked 8 Nov Wrong command?

└─▷ **TRY IT**

Start a trial before creating an account, keeping credentials in local possession storage so the original trial can be recovered if the response is uncertain.

```
$ qveris auth trial --bounded
```

Start a bounded anonymous trial before registration using `qveris auth trial --bounded`, which stores credentials securely via possession storage and reads back the original trial during recovery after uncertain responses.

Hands off a human claim privately without putting credentials in links, enforcing quote admission under the bounded policy.

▾ **5 more releases in this issue** · 2026-10-21 → 2026-10-29

> python-sdk-v0.9.0 NOTES ↗ STABLE
>
> qveris-agent-toolkit's Python SDK adds credential-free public discovery, caller-supplied idempotency keys, and typed Probe alternatives

└──▷ **GET THIS VERSION** ▾

pip

```
$ pip install qveris==0.9.0
# already installed? upgrade to this version:
$ pip install --upgrade qveris==0.9.0
```

npm

```
$ npm install @qverisai/cli@0.9.0
```

git

```
$ git clone --branch python-sdk-v0.9.0
https://github.com/QVerisAI/qveris-agent-toolkit.git
# already have the repo? check out this version:
$ git checkout python-sdk-v0.9.0
```

PyPI attestation · signed build provenance · checked 8 Nov **Wrong command?**

└──▷ **USE IT**

Browse available tools in a CI pipeline or unauthenticated agent context where no API key is present.

python

```
from qveris import PublicDiscoveryClient

client = PublicDiscoveryClient()
results = client.discover('weather forecast API')
for tool in results:
    print(tool)
```

Re-submit a call safely in a retry loop by supplying an idempotency key, then retrieve the original result if the call already settled.

```python
from qveris import QVerisClient

client = QVerisClient(api_key='<QVERIS_API_KEY>')

# Submit with a stable key
result = client.call(tool_id='<tool_id>', params={...},
idempotency_key='order-42-step-3')

# Later: look up the settled state without re-executing
record = client.reconcile_call('order-42-step-3')
print(record)
```

Discover and inspect tools without credentials using the new `PublicDiscoveryClient` class or by passing `public_read_only=True` at construction; mixed credentials are rejected and a fresh authenticated context is required before calling a previously selected public tool.

Look up the original execution and settlement state of any keyed call using the new typed reconcile_call(key) method, which accepts a caller-supplied `idempotency_key` and operates in strict single-submit compatibility mode with no automatic preflight or retry.

Access typed zero-quote settlement review details via the exposed zero-quote Call guard, which requires strict single-submit compatibility with no automatic preflight or retry.

Read typed Probe one-of alternatives via `required_any_of` and unsatisfied parameter groups via `missing_field_groups` while remaining compatible with older responses.

`js-sdk-v0.10.0` NOTES ↗ STABLE

qveris-agent-toolkit: Create a credential-free client for catalog-only discovery using Qveris.publicReadOnly() or `publicReadOnly: true` construction; mixed credentials are

└──▷ **GET THIS VERSION** ▾

```npm
$ npm install @qverisai/cli@0.10.0
```

```git
$ git clone --branch js-sdk-v0.10.0
https://github.com/QVerisAI/qveris-agent-toolkit.git
# already have the repo? check out this version:
$ git checkout js-sdk-v0.10.0
```

signed build provenance · checked 8 Nov Wrong command?

└──▷ **USE IT**

Browse the service catalog without credentials — useful in CI or agent bootstrap steps where no API key is available yet.

```javascript
import { Qveris } from '@qverisai/sdk';

const client = Qveris.publicReadOnly();
const results = await client.discover('weather forecast API');
```

Submit a Call with an idempotency key so you can safely retry submission and later reconcile the settled result without double-executing.

```javascript
import { Qveris } from '@qverisai/sdk';

const client = new Qveris({ apiKey: process.env.QVERIS_API_KEY });

// Submit once — safe to retry the same key
const result = await client.call(serviceId, params, {
idempotencyKey: 'order-42-weather' });

// Later: look up the original execution and settlement state
const settled = await client.reconcileCall('order-42-weather');
```

Look up the original execution and settlement state of a past Call by supplying a caller-chosen `idempotencyKey` at submission time and retrieving it later with reconcileCall(key); both require strict single-submit compatibility mode.

cli-v0.12.0 NOTES ↗ STABLE

qveris-agent-toolkit CLI adds credential-free public discovery, call idempotency keys, and a tolerant context consumer command

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install @qverisai/cli@0.12.0
```

git

```
$ git clone --branch cli-v0.12.0 https://github.com/QVerisAI/qveris-agent-toolkit.git
# already have the repo? check out this version:
$ git checkout cli-v0.12.0
```

signed build provenance · checked 8 Nov Wrong command?

└──▷ **TRY IT**

Browse available services in a credential-free environment, such as a sandboxed agent or CI pipeline, without creating a trial or triggering paid execution.

```
$ qveris discover "weather forecast API" --public-read-only --limit 1 --json
```

Replay a failed or uncertain call safely by looking up its result via idempotency key rather than re-submitting — avoiding double execution.

```
$ qveris execution lookup <key>
```

Run a call from a copied v1 context piped in from another tool, letting the CLI validate and refresh the snapshot before execution.

```
$ cat saved-context.json | qveris call --context -
```

Discover and inspect services without credentials using the `--public-read-only` flag; the CLI rejects mixed identities and keeps public selections separate from authenticated execution, refreshing inspection, schema, and quote after login before an upgraded selection is called.

Look up a previous execution or recover a call without replaying it using `qveris execution lookup <key>` and `qveris reconcile-call --idempotency-key <key>`; use `--require-zero-quote` to gate execution after an exact zero-credit Probe.

Pass a copied v1 service or task context directly to `qveris call --context <json|@file|->` to run it with strict execution, snapshot refresh for expired

contexts, and optional Inspect, Probe, or quote gates.

mcp-v0.15.0 NOTES ↗ STABLE

qveris-agent-toolkit adds host-controlled Call idempotency keys, `lookup_execution`, reconcileToolExecution(key), `require_zero_quote`, and machine-readable `next_action` gui

└──▷ GET THIS VERSION ▾

```
$ git clone --branch mcp-v0.15.0 https://github.com/QVerisAI/qveris-agent-toolkit.git
# already have the repo? check out this version:
$ git checkout mcp-v0.15.0
```

└──▷ USE IT

Recover a Call that may have been lost mid-flight without resubmitting it, using an idempotency key your host supplies.

```javascript
const result = await reconcileToolExecution(key);
```

MCP success and error payloads now include a machine-readable `next_action` field with guidance for authentication, required parameters, billing, retries, and unknown Call settlement.

mcp-v0.14.6 NOTES ↗ STABLE

qveris-agent-toolkit adds a QVeris Professional Data plugin for Claude

└──▷ GET THIS VERSION ▾

```
$ git clone --branch mcp-v0.14.6 https://github.com/QVerisAI/qveris-agent-toolkit.git
# already have the repo? check out this version:
$ git checkout mcp-v0.14.6
```

WAS THIS USEFUL? 👍 👎

linux-mcp-server

Feedback ⓘ 1.8.0 NOTES ↗

Linux-mcp-server tools to allow LLM clients to interact with Linux systems remotely

linux-mcp-server adds a pattern-based redaction engine for tool output, with built-in credential rules and custom YAML rule support

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install linux-mcp-server==1.8.0
# already installed? upgrade to this version:
$ pipx install --force linux-mcp-server==1.8.0
```

git

```
$ git clone --branch 1.8.0 https://github.com/rhel-lightspeed/linux-mcp-server.git
# already have the repo? check out this version:
$ git checkout 1.8.0
```

PyPI attestation · checked 8 Nov Wrong command?

Add custom redaction rules for tool output by supplying a YAML file, extending the built-in credential patterns with your own.

linux-mcp-server adds a pattern-based redaction engine that automatically scrubs common credentials from tool output, with built-in rules active by default.

Configure the logging destination for linux-mcp-server output.

Suppress INFO-level log messages from Python dependencies by default, reducing noise in server logs.

└──▷ BREAKING ON UPGRADE

! Minimum Python version raised to 3.11; installations running Python 3.10 will no longer be supported.

WAS THIS USEFUL? 👍 👎

slashdocs

Feedback ⓘ v0.4.1 NOTES ↗ LATEST STABLE

Slashdocs is a give your agents the power to publish reviewable HTML documents. Self-hosted server + CLI + remote MCP; agents push, humans review with comments and versions.

slashdocs now allows editing of comments and replies

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.4.1 https://github.com/razorpay/slashdocs.git
# already have the repo? check out this version:
$ git checkout v0.4.1
```

Users can now edit comments and replies after posting.

▾ 4 more releases in this issue · 2026-10-12 → 2026-10-28

v0.4.0 NOTES ↗ STABLE

slashdocs adds rotating refresh tokens for the MCP OAuth flow and public document viewing for signed-out readers

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.4.0
https://github.com/razorpay/slashdocs.git
# already have the repo? check out this version:
$ git checkout v0.4.0
```

Issue rotating refresh tokens during the MCP OAuth flow to improve session security for authenticated users.

Allow signed-out readers to view documents shared with `anyone` on public deployments, without requiring login.

v0.3.0 NOTES ↗ STABLE

slashdocs now shows commenter names in Slack notifications

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3.0
https://github.com/razorpay/slashdocs.git
# already have the repo? check out this version:
$ git checkout v0.3.0
```

Slack notifications now include the name of the commenter alongside the comment content.

v0.2.1 NOTES ↗ STABLE

slashdocs adds a local mode CLI flag and live cursor playback, plus restricts people search on public servers

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.2.1
https://github.com/razorpay/slashdocs.git
# already have the repo? check out this version:
$ git checkout v0.2.1
```

└──▷ TRY IT

Run a slashdocs command against a local server instance during development or air-gapped testing.

```
$ slashdocs local
```

Run the CLI in local mode with `slashdocs local` or add `--local` to any command to work against a local server instance instead of a remote host.

Live cursors now follow collaborators in real time, play back smoothly, and can be hidden.

v0.1.0 NOTES ↗ STABLE

slashdocs adds its first public release with Docker, Helm, and CLI install paths for self-hosted deployment

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.0
https://github.com/razorpay/slashdocs.git
# already have the repo? check out this version:
$ git checkout v0.1.0
```

└──▷ **TRY IT**

Stand up slashdocs on Kubernetes in its own namespace with custom values on first install.

```
$ helm upgrade --install slashdocs ./slashdocs-0.1.0.tgz -n
slashdocs --create-namespace -f my-values.yaml
```

Bootstrap the slashdocs CLI on any machine that can reach a running server instance.

```
$ curl -fsSL https://<your-slashdocs-host>/install.sh | bash
```

Deploy slashdocs via Docker using `docker pull razorpay/slashdocs:0.1.0`.

Deploy slashdocs to Kubernetes via Helm chart `slashdocs-0.1.0.tgz` with `helm upgrade --install slashdocs ./slashdocs-0.1.0.tgz -n slashdocs --create-namespace -f my-values.yaml`.

Install the slashdocs CLI from a running server via `curl -fsSL https://<your-slashdocs-host>/install.sh | bash`.

Configure self-hosted deployments using `docs/self-hosting.md`.

WAS THIS USEFUL? 👍 👎

browser-tools-mcp

Feedback ⓘ v2.0.4 NOTES ↗ LATEST

STABLE

Browser-tools-mcp is a monitor browser logs directly from Cursor and other MCP compatible IDEs.

browser-tools-mcp adds bearer-token authentication, per-call action gating, and `--extension-id` pairing for the local connector API

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.0.4 https://github.com/AgentDeskAI/browser-tools-mcp.git
# already have the repo? check out this version:
$ git checkout v2.0.4
```

└──▷ **TRY IT**

Start the connector locked to a specific extension and allow state-changing API calls, so only your known extension can pair and destructive actions (reload, screenshot, wipe) are permitted.

```
$ browser-tools-mcp --extension-id abcdefghijklmnopabcdefghijklmnop --allow-actions
```

Pair a browser extension with the connector to obtain a bearer token; subsequent HTTP API calls must supply that token via the Authorization header.

```
$ curl -X POST http://localhost:<port>/.pair \
  -H 'Origin: chrome-extension://abcdefghijklmnopabcdefghijklmnop'
```

Restrict state-changing local API calls (reload, screenshot, wipe, settings changes, tab selection) behind the `--allow-actions` flag or the Allow state-changing calls checkbox in the BrowserTools panel; calls are refused by default.

Limit connector pairing and the extension WebSocket to a single extension with the `--extension-id` flag.

The HTTP API and extension WebSocket now accept authentication via `Authorization: Bearer` header or, for the extension, a WebSocket subprotocol; tokens in the URL are refused, and tokens are issued only from `POST /.pair` to a

verified browser-extension origin (`chrome-extension://`, `moz-extension://`, or `safari-web-extension://`).

Apply `--only` and `--exclude` filters to the local API — not only to the MCP tool list — so a shared connector enforces the filters it was started with.

└──▷ **BREAKING ON UPGRADE**

- **!** 1.x is unsupported; vulnerabilities fixed in 2.0 are not patched on 1.x — users on 1.x must upgrade.
- **!** Tokens supplied in the URL are refused; callers must switch to `Authorization: Bearer` header or the WebSocket subprotocol.
- **!** State-changing calls (reload, screenshot, wipe, settings changes, tab selection) are refused by default and require `--allow-actions` to re-enable.

▾ **1 more release in this issue** · 2026-10-26 → 2026-10-27

v2.0.3 NOTES ↗ STABLE

browser-tools-mcp adds `--audit-allow-origin` and `--audit-allow-private-network` flags and an in-process proxy to block SSRF in Lighthouse audit tools

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.0.3 https://github.com/AgentDeskAI/browser-tools-mcp.git
# already have the repo? check out this version:
$ git checkout v2.0.3
```

└──▷ **TRY IT**

Audit a local dev server that is not open in a browser tab, by allowlisting its origin at connector startup.

```
$ browser-tools-connector --audit-allow-origin http://localhost:3000,http://localhost:8080
```

Allow audits of private-network addresses when running behind a fake-IP proxy, accepting that local-reach checks are skipped.

```
$ browser-tools-connector --audit-allow-private-network
```

Disable all four Lighthouse audit tools entirely on a host with an IMDSv1 metadata service when upgrading is not immediately possible.

```
$ browser-tools-connector --exclude
runAccessibilityAudit,runPerformanceAudit,runSEOAudit,runBestPractic
esAudit
```

URLs with embedded credentials are now rejected by the audit tools.

Fake-IP proxy answers (Clash, mihomo, sing-box, Surge) are now refused for audited pages and everything they load, with the error message naming the workaround.

└─▷ **BREAKING ON UPGRADE**

- ! Audits no longer connect through a system, PAC, or `HTTP(S)_PROXY` proxy; on networks where direct connections are blocked, audits of public sites will fail.
- ! Audits of `localhost` or private-network pages that are not open in a connected browser tab now fail unless `--audit-allow-origin` or `--audit-allow-private-network` is set.
- ! Fake-IP proxy setups (Clash, mihomo, sing-box, Surge in fake-ip mode) will cause audits to fail unless fake-ip mode is disabled or `--audit-allow-private-network` is passed.

WAS THIS USEFUL?  

canvas-mcp

Feedback ⓘ v1.16.0 NOTES ↗ LATEST STABLE

Canvas LMS MCP server — up to 102 tools and 8 agent skills for students & educators. Works with Claude, Cursor, Codex, and 40+ agents.

canvas-mcp adds confirmed group creation with conflict detection and opt-in bounded submission body retrieval for assignments

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install canvas-mcp==1.16.0
# already installed? upgrade to this version:
$ pipx install --force canvas-mcp==1.16.0
```

git

```
$ git clone --branch v1.16.0 https://github.com/vishalsachdev/canvas-mcp.git
# already have the repo? check out this version:
$ git checkout v1.16.0
```

PyPI attestation · checked 8 Nov Wrong command?

Retrieve assignment submission bodies on an opt-in basis with bounded content via the new `include_body` path in `list_submissions`.

Create groups with confirmed creation and conflict-refusing resume, so duplicate group names are rejected rather than silently overwritten.

▾ 1 more release in this issue · 2026-10-23 → 2026-10-25

v1.15.0 NOTES ↗ STABLE

canvas-mcp adds a read-student-submission tool and now surfaces when write tools are withheld in server instructions

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install canvas-mcp==1.15.0
# already installed? upgrade to this version:
$ pipx install --force canvas-mcp==1.15.0
```

git

```
$ git clone --branch v1.15.0
https://github.com/vishalsachdev/canvas-mcp.git
# already have the repo? check out this version:
$ git checkout v1.15.0
```

PyPI attestation · checked 8 Nov Wrong command?

Read one student submission with permission and privacy guards using the new student submission tool (#487).

canvas-mcp now states in its server instructions when write tools are withheld, so agents know what actions are unavailable.

WAS THIS USEFUL? 👍 👎

search2ai

Feedback ⓘ v0.3.2 NOTES ↗ LATEST STABLE

Search2ai is a self-hosted search gateway for AI agents. Unified search, news and page extraction with provider fallback.

search2ai adds best-effort parameter translation across providers, surfacing ignored options and warnings in new response headers

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g search2ai@0.3.2
```

git

```
$ git clone --branch v0.3.2 https://github.com/fatwang2/search2ai.git
# already have the repo? check out this version:
$ git checkout v0.3.2
```

signed build provenance · checked 8 Nov Wrong command?

└──▷ TRY IT

Inspect which parameters a backend silently dropped or clamped, without changing your existing SDK call.

```
$ curl -i https://<your-search2ai-host>/search \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"query": "open source LLMs", "language": "fr", "recency":
"week", "count": 50}' \
| grep -E 'X-Search2AI-Ignored|X-Search2AI-Warnings'
```

Report ignored parameter names in the `X-Search2AI-Ignored` response header and adjustments in `X-Search2AI-Warnings`, both exposed through CORS, so callers can see exactly what each backend silently dropped or clamped.

Map available domain, language, date and recency options across providers and ignore unsupported options without returning 422 errors or skipping usable backends.

Clamp translated result counts to per-backend limits and retain search results when optional page-content enrichment is unavailable or partially fails.

▾ 1 more release in this issue · 2026-10-22

v0.3.1 NOTES ↗ STABLE

search2ai adds a compatibility gateway layer so existing Exa, Tavily, Search1API, SearXNG, and Firecrawl SDKs can route through the gateway without code changes

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g search2ai@0.3.1
```

git

```
$ git clone --branch v0.3.1
https://github.com/fatwang2/search2ai.git
# already have the repo? check out this version:
$ git checkout v0.3.1
```

signed build provenance · checked 8 Nov Wrong command?

└──▷ TRY IT

Redirect an existing Tavily SDK client through the search2ai gateway without touching response-parsing code.

```
$ curl https://YOUR_GATEWAY/compat/tavily/search \
  -H 'Authorization: Bearer YOUR_GATEWAY_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"query": "latest CVEs in OpenSSL", "api_key":
"YOUR_GATEWAY_KEY"}'
```

Redirect an existing Exa SDK client to retrieve page contents through the gateway instead of hitting Exa directly.

```
$ curl https://YOUR_GATEWAY/compat/exa/contents \
  -H 'Authorization: Bearer YOUR_GATEWAY_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"ids": ["https://example.com/report"]}'
```

Point any supported client SDK at `https://YOUR_GATEWAY/compat/PROTOCOL` with your gateway key to route Exa, Tavily, Search1API, SearXNG, or Firecrawl calls through the gateway without changing SDK methods or response-parsing code.

SearXNG JSON search now supports root and `/search` endpoints via GET and form POST through the compatibility layer.

> Exa compatibility covers `/search` and `/contents` endpoints.
>
> Tavily compatibility covers `/search` and `/extract` endpoints.
>
> Search1API compatibility covers `/search`, `/news`, and single-page `/crawl` endpoints.
>
> +1 more

WAS THIS USEFUL? 👍 👎

mcpvault

Feedback ⓘ v0.17.0 NOTES ↗

Mcpvault is a lightweight Model Context Protocol (MCP) server for safe Obsidian vault access

mcpvault adds repeatable `--exclude` CLI options and per-call `excludePaths` to filter vault subtrees from search and directory listing

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g @bitbonsai/mcpvault@0.17.0
```

git

```
$ git clone --branch v0.17.0 https://github.com/bitbonsai/mcpvault.git
# already have the repo? check out this version:
$ git checkout v0.17.0
```

signed build provenance · checked 8 Nov Wrong command?

└──▷ TRY IT

Exclude sensitive or noisy vault subtrees from all search and listing results when starting the server.

```
$ mcpvault --exclude private/journal --exclude work/archive
```

Narrow a single search call beyond the server-level exclusions by supplying additional paths in the per-call field.

```
$ { "tool": "search_notes", "query": "meeting notes", "excludePaths":
["team/drafts", "inbox"] }
```

Filter vault subtrees from `search_notes` and `list_directory` persistently by passing repeatable `--exclude <subtree>` or `--exclude=<subtree>` CLI options at startup.

Extend configured exclusion defaults per individual call using the `search_notes.excludePaths` field, without affecting direct read and write tool access to those paths.

WAS THIS USEFUL? 👍 👎

obsei

Feedback ⓘ v1.1.0 NOTES ↗ LATEST STABLE

Obsei is an open source, privacy-first Voice of Customer analytics. A self-hosted alternative to Enterpret and Unwrap that strips PII from feedback before any LLM sees it.

obsei adds GPU acceleration for the local classifier and labels the demo with the default model in use

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/obsei/obsei:1.1.0
```

pipx

```
$ pipx install obsei==1.1.0
# already installed? upgrade to this version:
$ pipx install --force obsei==1.1.0
```

git

```
$ git clone --branch v1.1.0 https://github.com/obsei/obsei.git
# already have the repo? check out this version:
$ git checkout v1.1.0
```

published under the project's GitHub account · PyPI attestation · checked 8 Nov Wrong command?

▾ 1 more release in this issue · 2026-10-15 → 2026-10-16

v1.0.0 NOTES ↗ STABLE

obsei adds guided setup commands `obsei add` and `obsei check` with an integration catalog

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/obsei/obsei:1.0.0
```

pipx

```
$ pipx install obsei==1.0.0
# already installed? upgrade to this version:
$ pipx install --force obsei==1.0.0
```

git

```
$ git clone --branch v1.0.0 https://github.com/obsei/obsei.git
# already have the repo? check out this version:
$ git checkout v1.0.0
```

published under the project's GitHub account · PyPI attestation · checked 8 Nov

Wrong command?

└──▷ **TRY IT**

Bootstrap a new obsei integration without hand-editing config files.

```
$ obsei add
```

Verify that a configured integration is reachable and correctly set up before running a pipeline.

```
$ obsei check
```

WAS THIS USEFUL? 👍 👎

openrouter-mcp-multimodal

Feedback ⓘ v5.1.0 NOTES ↗

Openrouter-mcp-multimodal is a MCP server for OpenRouter — chat with 300+ LLMs (Claude, Gemini, GPT), analyze images / audio / video, generate images / speech / music / video (Veo 3.1, Seedance, Kling, Wan) from Claude Desktop, Cursor, Kiro, VS Code.

openrouter-mcp-multimodal adds vector embeddings and calibrated decision tools, plus 13 new chat completion parameters

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/stabgan/openrouter-mcp-multimodal:5.1.0
```

pipx

```
$ pipx install mcp-server-openrouter-multimodal==5.1.0
# already installed? upgrade to this version:
$ pipx install --force mcp-server-openrouter-multimodal==5.1.0
```

git

```
$ git clone --branch v5.1.0 https://github.com/stabgan/openrouter-mcp-multimodal.git
# already have the repo? check out this version:
$ git checkout v5.1.0
```

published under the project's GitHub account · PyPI attestation · checked 8 Nov Wrong command?

└──▷ TRY IT

Generate embeddings for a batch of documents to feed a RAG pipeline.

```
$ curl -X POST /api/v1/embeddings \
  -H 'Content-Type: application/json' \
  -d '{"input": ["doc1 text", "doc2 text"], "model": "<embedding-model>"}'
```

Run a calibrated classification decision with probability scores using the decide tool.

```
$ curl -X POST /api/alpha/decisions \
  -H 'Content-Type: application/json' \
  -d '{"prompt": "Is this email phishing?", "type": "classification",
"labels": ["phishing", "legitimate"]}'
```

Get vector embeddings for RAG, semantic search, and clustering via the new `embed_text` tool, backed by `POST /api/v1/embeddings` with 37+ models and batch sizes up to 2048.

Make yes/no, classification, and scoring decisions with probabilities via the new `decide` tool, backed by `POST /api/alpha/decisions`.

Control chat completions with 13 new parameters on `chat_completion`: `fusion`, `subagent`, `response_healing`, `web_blocked_domains`, `stop`, `top_p`, `frequency_penalty`, `presence_penalty`, `logprobs`, `top_logprobs`, `seed`, `reasoning_effort`, and `response_format`.

WAS THIS USEFUL? 👍 👎

◆ **Applied AI Agents**

nanobot

Feedback ⓘ CVE-2026-121865 CVE ↗ STABLE

Nanobot is a self-hosted personal AI assistant framework with tools, memory, scheduled tasks, and integrations with chat applications.

`CVE-2026-121865`: Standard WebSocket clients could select a workspace outside configured authority (CVSS 7.6, affects `>= 0.3.5, <= 0.4.0`)

▾ **1 more release in this issue** · 2026-11-08

> CVE-2026-121864 CVE ↗ STABLE
>
> `CVE-2026-121864`: Sandbox escape through caller-selected shell executable (CVSS 8.5, affects `>= 0.2.1, < 0.4.1`)

WAS THIS USEFUL? 👍 👎

◆ Agent Memory

claude-mem

Feedback ⓘ v15.12.0 NOTES ↗

claude-mem gives coding agents persistent memory by capturing session activity, compressing it with a model, and injecting relevant context into later sessions.

claude-mem adds image capture and gallery in My Work, a 25 GiB cloud image quota, Delphi/Pascal smart-read support, and two new AI provider presets

└──▷ GET THIS VERSION ▾

npm

```
$ npm install -g claude-mem@15.12.0
```

git

```
$ git clone --branch v15.12.0 https://github.com/thedotmack/claude-mem.git
# already have the repo? check out this version:
$ git checkout v15.12.0
```

signed build provenance · checked 8 Nov Wrong command?

Screenshots and images an agent reads or produces are now kept by default and linked to the observations from that turn, without sending them to an AI model; control this with the Keep images toggle in the memory viewer's Settings > Images section, which respects a saved `"false"` value.

My Work now shows a gallery: request cards display an image count and the first four thumbnails, the request drawer shows every linked image, and any thumbnail opens a full-size gallery with keyboard navigation.

Each owner can store up to 25 GiB of images on cmem.ai, matching the limit Pro enforces.

Screenshots now convert locally on your own computer as lossy WebP, queued so each conversion gets 20 seconds and the stored copies stay small.

Smart-read now supports Delphi/Pascal source files.

+5 more

└──▷ BREAKING ON UPGRADE

! The Claude app plugin is renamed to `cmem-pro`; any reference to the old plugin name will need to be updated.

WAS THIS USEFUL? 👍 👎

◆ Agent Sandboxes

Sandbox0

Feedback ⓘ announced · Oct 17, 2026 BLOG ↗

Sandbox0 Cloud is a hosted execution environment for long-running AI agents, with a writable root filesystem that keeps installed dependencies, repositories and outputs across checkpoints, plus pause and resume.

Sandbox0 adds out-of-band sandbox recovery, granular health reporting, and experimental memory pause/resume that work even when the internal command service is down

Sandbox0 now reports runtime, supervisor, command-execution, and storage health as separate signals rather than a single `running` flag, so applications can distinguish a failed command service from a broader runtime or storage problem.

Sandbox0 adds an explicit runtime restart path that operates outside the internal `procd` command service, so a damaged `procd` no longer blocks lifecycle management of the sandbox.

Sandbox0 adds experimental memory pause and resume that can retain supported processes and in-memory state on compatible restores, operable without a healthy `procd`; memory preservation must be selected on both pause and resume — default pause/resume remains filesystem-only.

Sandbox0 now records restart operations with a request key and expected runtime generation so that a retry after a lost connection follows the original request instead of issuing a duplicate restart.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ ML Platforms & SDKs

Heretic

Feedback ⓘ v2.1.0 NOTES ↗

Heretic removes safety alignment from language models by tuning directional-ablation parameters.

Heretic adds a custom config file path via `HERETIC_CONFIG_FILE` and a new response length scorer plugin.

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install heretic-llm==2.1.0
# already installed? upgrade to this version:
$ pipx install --force heretic-llm==2.1.0
```

git

```
$ git clone --branch v2.1.0 https://github.com/p-e-w/heretic.git
# already have the repo? check out this version:
$ git checkout v2.1.0
```

named in the project's README · checked 8 Nov Wrong command?

└──▷ TRY IT

Run Heretic with a project-specific config file without renaming it or moving it to your working directory.

```
$ HERETIC_CONFIG_FILE=/path/to/my-custom-config.toml heretic
Qwen/Qwen3-4B-Instruct-2507
```

Set a custom configuration file location with the `HERETIC_CONFIG_FILE` environment variable instead of relying on the default `config.toml` in the working directory.

Heretic adds a response length scorer plugin available in the `scorers` directory, which can be activated via the `scorers` setting in the configuration file.

WAS THIS USEFUL? 👍 👎

emisar

Feedback ⓘ runner-v0.32.1 NOTES ↗ LATEST STABLE

Emisar lets AI agents securely run infrastructure actions through MCP, with approval controls that protect production systems.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch runner-v0.32.1
https://github.com/AndrewDryga/emisar.git
# already have the repo? check out this version:
$ git checkout runner-v0.32.1
```

UPDATED INTEGRATION Dataproc: add governed cluster and job operations

NEW INTEGRATION NATS: add governed JetStream operations

UPDATED INTEGRATION Better Stack: add governed uptime operations

UPDATED INTEGRATION Secret Manager: inspect version metadata without payloads

UPDATED INTEGRATION Dataproc: read retained bounded driver logs

UPDATED INTEGRATION HCP Terraform: show one resource's changed plan paths

NEW INTEGRATION gcp-dataproc pack

NEW INTEGRATION nats pack

NEW INTEGRATION betterstack pack

UPDATED INTEGRATION hcp-terraform pack

UPDATED INTEGRATION traefik pack

UPDATED INTEGRATION gcp-secret-manager pack

▾ 1 more release in this issue · 2026-11-08

> v0.58.0 NOTES ↗ STABLE
>
> Runner and portal now accept up to 4 MB of advertised packs, up from 2 MB; Runner 0.32.1 ships with this release.
>
> └──▷ GET THIS VERSION ▾
>
> ```
> $ git clone --branch v0.58.0
> https://github.com/AndrewDryga/emisar.git
> # already have the repo? check out this version:
> $ git checkout v0.58.0
> ```

WAS THIS USEFUL?  

MotleyAI SLayer

Feedback ⓘ v1.8.1 NOTES ↗ LATEST STABLE

SLayer is an embeddable semantic layer that lets AI agents and people query warehouse data through shared metric definitions.

SLayer adds `remedy_kind` to type-conflict findings from `validate_models`, letting applications branch on fix type without parsing text

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.8.1 https://github.com/MotleyAI/slayer.git
# already have the repo? check out this version:
$ git checkout v1.8.1
```

└──▷ BREAKING ON UPGRADE

! RetypeFinding.of(...) now takes `remedy_kind=` in place of `reingest: bool` — callers that pass the old `reingest` keyword argument must update.

▾ 1 more release in this issue · 2026-11-07

v1.8.0 NOTES ↗ STABLE

SLayer adds OSI export via `slayer export-osi` and export_osi(), and now enforces model filters on every read, including through joins

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.8.0 https://github.com/MotleyAI/slayer.git
# already have the repo? check out this version:
$ git checkout v1.8.0
```

└──▷ TRY IT

Snapshot your current semantic layer as an Ossie document for sharing or migration, then verify it round-trips cleanly.

```
$ slayer export-osi --datasource shop --out shop.osi.yaml
```

Export plain Ossie without SLayer-specific extensions for use with other tools that consume standard OSI.

```
$ slayer export-osi --datasource shop --out shop.osi.yaml --no-extensions
```

Export models to an Ossie document with `slayer export-osi --datasource <name> --out <file>` (CLI) or export_osi(...) (Python); use `--no-extensions` to write plain Ossie without SLayer-specific extensions.

MCP tools now report database failures as tool errors with `isError: true` so agents can distinguish infrastructure errors from empty results.

Model filters now apply wherever a model is read, including when it is reached through a join, not only when a query starts from that model directly.

└─▷ **BREAKING ON UPGRADE**

! Model filters that name a measure or a windowed column, or that form a cycle across models, are now refused at save time — previously such filters could be saved and would silently produce wrong results.

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

cc-router

Feedback ⓘ v6.7.0 NOTES ↗

Cc-router is a 本地运行的大模型聚合网关，GUI桌面端app，零代码部署，把Coding Plan、大模型 API 额度聚合成一个虚拟 Plan，一键接入 Claude Code、Claude Desktop App、OpenClaw、OpenCode 等工具。Bundle your scattered Token Plan, Coding Plan, and LLM API quotas into a single virtual Plan, and plug it into Claude Code, Claude Desktop App, OpenClaw, OpenCode, and more.

cc-router adds header forwarding controls, speed and first-content timing, token counting endpoint, and automatic upstream decompression

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v6.7.0 https://github.com/finch-xu/cc-router.git
# already have the repo? check out this version:
$ git checkout v6.7.0
```

└──▷ HOW TO FIND IT

Suppress Claude Code auto mode negotiation headers when your upstream does not support them.

📍 In your shell environment or .env file, set: `CLAUDE_CODE_AUTO_MODE_SERVER=0` (Copy this value from the Claude Code auto mode compatibility notice in the header forwarding settings.)

Inspect exactly which headers were sent upstream and returned to the client for a specific request, with sensitive values masked.

📍 In the cc-router web interface, open Request Logs → select a log entry → switch between the Client request headers, Upstream request headers, and Upstream response headers tabs to see sources and client-return status for each header.

Support the `/v1/messages/count_tokens` token counting endpoint and `HEAD /api/hello` warm-up probe through Anthropic passthrough subscriptions; counting requests do not count toward logs or quotas.

Add header forwarding control for Anthropic passthrough subscriptions with five modes — Don't forward, Claude Code preset, Allowlist, Denylist, and Forward all — plus a preview mode that shows send results for unsaved settings without contacting the upstream; copy the `CLAUDE_CODE_AUTO_MODE_SERVER=0` environment variable for Claude Code auto mode compatibility.

Add speed and first-content time metrics to the request log list, details, copied content, and CSV exports; speed is the average output token rate after the initial wait, and first content is timed to the first actual text, thinking, or tool-call content.

Automatically negotiate and decompress gzip, deflate, Brotli, and zstd upstream responses with no extra configuration, forwarding streaming responses incrementally.

Expand request log details with per-category tabs for overview, client request headers, upstream request headers, and upstream response headers; header tables show each header's source and whether it was returned to the client, with sensitive values masked.

+2 more

└─▷ **BREAKING ON UPGRADE**

! Request header recording is enabled by default after upgrading; turn it off under Settings → Advanced → Record request headers.

WAS THIS USEFUL? 👍 👎

llamactl

Feedback ⓘ v0.23.0 NOTES ↗

Llamactl is an unified management and routing for llama.cpp, MLX and vLLM models with web dashboard.

llamactl adds request body size limits and switches inference API key storage to HMAC fingerprints

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.23.0
https://github.com/lordmathis/llamactl.git
# already have the repo? check out this version:
$ git checkout v0.23.0
```

llamactl now enforces request body size limits to cap the maximum payload the service will accept.

llamactl now stores and verifies inference API keys as HMAC fingerprints instead of Argon2id hashes; existing keys migrate automatically on first use.

└──▷ BREAKING ON UPGRADE

! `GET /api/v1/config` returns a new response format containing only local node, instance defaults, and backend command/docker summary — all other fields are removed. Any consumer of this endpoint must be updated.

! Inference API keys are now stored as HMAC fingerprints. Keys not used before the legacy Argon2id verification path is removed in a future release must be recreated.

WAS THIS USEFUL? 👍 👎

dario

Feedback ⓘ v6.23.1 NOTES ↗ LATEST STABLE

Dario is an one local endpoint that routes Cursor, Cline, Aider, Claude Code and the Agent SDK through your Claude and ChatGPT subscriptions, with failover and pooling and zero runtime dependencies.

dario adds automatic expiry for compare records via `DARIO_COMPARE_KEEP_DAYS`

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/askalf/dario:v6.23.1
```

npm

```
$ npm install -g @askalf/dario@6.23.1
```

git

```
$ git clone --branch v6.23.1 https://github.com/askalf/dario.git
# already have the repo? check out this version:
$ git checkout v6.23.1
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

└──▷ **USE IT**

Prevent unbounded disk growth from accumulated compare records by enabling automatic deletion of records older than 30 days.

.env

```
DARIO_COMPARE_KEEP_DAYS=30
```

Set `DARIO_COMPARE_KEEP_DAYS` to automatically delete compare records older than that many days from `~/.dario/compare`, checked on start and hourly.

Compare records now expire — previously, records holding full prompts and both answers accumulated in `~/.dario/compare` with no cleanup path.

▾ **1 more release in this issue** · 2026-11-07

> v6.23.0 NOTES ↗ STABLE

dario adds per-key transcript capture with a dedicated token-gated API endpoint and a 48-hour rolling store

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/askalf/dario:v6.23.0
```

npm

```
$ npm install -g @askalf/dario@6.23.0
```

git

```
$ git clone --branch v6.23.0 https://github.com/askalf/dario.git
# already have the repo? check out this version:
$ git checkout v6.23.0
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

└──▷ **TRY IT**

Turn on transcript recording for a specific named key so you can review its exact requests and responses later.

```
$ dario keys transcript my-dev-key on
```

Enable per-key request/response transcript capture with `dario keys transcript <name> on`; recordings are stored as private files under `~/.dario/transcripts` and auto-deleted after 48 hours whether read or not.

Retrieve transcripts via `GET /analytics/transcripts`, accessible only to `--transcript-token` or the `DARIO_TRANSCRIPT_TOKEN` environment variable — not the root key, named keys, or the analytics token; the route is disabled entirely when no token is set.

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

distributed-llama

Feedback ⓘ v0.17.0 NOTES ↗

Distributed-llama is a distributed LLM inference. Connect home devices into a powerful cluster to accelerate LLM inference.

distributed-llama adds support for running Qwen3.5 models across distributed clusters.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.17.0 https://github.com/b4rtaz/distributed-llama.git
# already have the repo? check out this version:
$ git checkout v0.17.0
```

NEW SUPPORTED PLATFORM Run Qwen3.5 models on a distributed cluster, with `python launch.py qwen3_5_9b_q40` available as a one-command setup for the 9B Q40 variant (8.0 GB).

WAS THIS USEFUL? 👍 👎

Niko1221 Strata

Feedback ⓘ v0.2.12 NOTES ↗

Strata is a local inference engine with one-click install for Windows and Linux, providing an OpenAI/Anthropic-compatible API on localhost for models like Qwen and DeepSeek.

Strata adds a tuned DFlash2 drafter, Strix Halo and Mac M5 Max support, and new opt-in GPU kernel switches in v0.2.12

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.2.12 https://github.com/Niko1221/Strata.git
# already have the repo? check out this version:
$ git checkout v0.2.12
```

└──▷ TRY IT

Force MTP drafting on a Radeon R9700 to avoid DFlash2 stalls that affect 11 of 50 engines on this card.

```
$ strata launch --drafting mtp
```

Return calls to undeclared tools as actual tool calls rather than text, useful when a model names tools not listed in the system prompt.

```
$ STRATA_UNDECLARED_TOOL_CALLS=call strata launch
```

On a Radeon R9700, enable the fused `Q5_0` prompt kernel to get up to 44% faster short prompts.

```
$ STRATA_PF_FUSED_Q50=1 strata launch
```

Control undeclared tool-call handling with the new `undeclared_tool_calls` setting (`text` by default; set to `call` or use `STRATA_UNDECLARED_TOOL_CALLS=call` to return them as tool calls).

On Radeon gfx12 (R9700/RX 9070), adds `STRATA_PF_FUSED_Q50=1` to enable fused `Q5_0` kernel prompts (up to 44% faster 2K and 40% faster 8K prompts), and `STRATA_GDN_CHUNKED=0` to disable the chunked GDN scan for prompts of 30,000+ tokens.

On NVIDIA, adds `STRATA_IQ4_W4A8=1` or `STRATA_IQ4_W4A8=2` to enable an int8 expert kernel for `IQ4_XS` and `IQ4_NL`, and `STRATA_SELECT_TWO_PASS=1` to enable exact two-pass expert selection for long prompts (128K +1.65%).

On CUDA, adds `STRATA_ADAPT_FETCH=1` or `STRATA_ADAPT_FETCH=2` to send fetched cache misses into the card's cache (decode +10.2% with `=2` in one setup), and, with `--peer-device`, `STRATA_ADAPT_OVERLAP=1` to overlap peer copies with the next window.

Enable 512-bit expert rows on CPUs without AVX-512-VBMI with `STRATA_VNNI_TIER=1`.

+5 more

WAS THIS USEFUL? 👍 👎

emmy

Feedback ⓘ v0.3.23 NOTES ↗ LATEST STABLE

Emmy is an optimized GPU compiler for LLM inference. Choose from a list of optimized recipes or optimize your own model via kernel fusion, autotuning, and advanced scheduling.

Emmy adds CPU compilation via LLVM, MoE expert host-memory paging, speculative decoding on GDN models, and f16/bf16 conv1d computed at f32

└──▷ GET THIS VERSION ▾

pip

```
$ pip install emmy-ml==0.3.23
# already installed? upgrade to this version:
$ pip install --upgrade emmy-ml==0.3.23
```

git

```
$ git clone --branch v0.3.23 https://github.com/cloudrift-ai/emmy.git
# already have the repo? check out this version:
$ git checkout v0.3.23
```

PyPI attestation · checked 8 Nov Wrong command?

└──▷ TRY IT

Point tune row freezes at a custom directory, or set empty to disable freeze writes entirely.

```
$ export EMMY_FREEZE_DIR=/mnt/shared/freezes
emmy run --bench --tune 20 -c "nn.Linear(4096,4096,bias=False)
(torch.randn(1,4096))"
```

Adds a new CPU backend: compiles and runs programs via LLVM and the Rust runtime with `emmy compile`, and runs MoE experts on the CPU near memory speed using a CPU schedule, CPU benching, and NVFP4 on the CPU.

Run Affine statements on the CPU backend, extending the CPU compilation path to cover the Affine IR layer.

Adds speculative decoding on GDN models using an Emmy-compiled MTP head, and captures CUDA graphs of decode steps on GDN models.

Generate natively from Qwen3.5-MoE models with resident routed experts.

Compute f16/bf16 conv1d at f32 and convert once, matching PyTorch behavior.

+2 more

▾ **1 more release in this issue** · 2026-10-11 → 2026-10-30

v0.3.21 NOTES ↗ STABLE

emmy adds a torch.compile() backend and an opt-in cold-cache bench mode that evicts L2 before every timed launch

└──▷ **GET THIS VERSION** ▾

pip

```
$ pip install emmy-ml==0.3.21
# already installed? upgrade to this version:
$ pip install --upgrade emmy-ml==0.3.21
```

git

```
$ git clone --branch v0.3.21 https://github.com/cloudrift-
ai/emmy.git
# already have the repo? check out this version:
$ git checkout v0.3.21
```

PyPI attestation · checked 8 Nov Wrong command?

└──▷ **USE IT**

Compile a PyTorch model through emmy without writing a custom compile step — drop-in replacement for other torch.compile backends.

python

```
import torch
import torch.nn as nn
import emmy  # registers the 'emmy' backend

model = nn.Linear(4096, 4096, bias=False).cuda()
compiled = torch.compile(model, backend='emmy', dynamic=False)
with torch.no_grad():
    out = compiled(torch.randn(1, 4096, device='cuda'))
```

Measure cold-cache (L2-evicted) latency when tuning a kernel, so hot-cache wins don't inflate benchmark results.

```
$ emmy run --bench --cold-cache --tune 20 -c
"nn.Linear(4096,4096,bias=False)(torch.randn(1,4096))"
```

> Use emmy as a torch.compile() backend via torch.compile(model, backend='emmy', dynamic=False) to compile PyTorch models with functionalized input writes.
>
> Add an opt-in cold-cache bench mode with `--cold-cache` to evict L2 before every timed launch and keep cold rows in their own measurement regime.
>
> NVFP4 linear layers of any token width now take the tensor core path, extending the earlier one-token-only NVFP4 tile to arbitrary widths.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

vMLX

Feedback ⓘ v1.6.84 NOTES ↗

The vMLX server runs compressed MLX models on Apple Silicon with disk caching, paged memory, continuous batching, and hybrid SSM scheduling.

vMLX adds Ripple-65B runtime support and DFlash2 text lane, plus per-request footer rows in the panel

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install vmlx==1.6.84
# already installed? upgrade to this version:
$ pipx install --force vmlx==1.6.84
```

git

```
$ git clone --branch v1.6.84 https://github.com/jjang-ai/vmlx.git
# already have the repo? check out this version:
$ git checkout v1.6.84
```

PyPI attestation · checked 8 Nov Wrong command?

Adds Ripple-65B runtime and DFlash2 text lane support for running the Ripple-65B model family.

Adds per-request footer rows in the panel for each model request within a tool exchange.

The `find_files` tool now matches path-style globs in addition to filename patterns.

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

RunPod

Feedback ⓘ announced · Nov 6, 2026 BLOG ↗

RunPod is an AI developer cloud platform for experimenting, training, fine-tuning, deploying, and scaling AI workloads across 31 global regions.

NEW SUPPORTED PLATFORM AP-JP-1 data center region (Fukushima, Japan)

WAS THIS USEFUL? 👍 👎

DATA

◆ Document Parsing

Docling

Feedback ⓘ v2.149.0 NOTES ↗

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

Docling adds opt-in embedded PDF image extraction, mlx-audio ASR support, and a MiniCPM-OCR VLM preset alongside a refactored CLI

└──▷ GET THIS VERSION ▾

pip

```
$ pip install docling==2.149.0
# already installed? upgrade to this version:
$ pip install --upgrade docling==2.149.0
```

git

```
$ git clone --branch v2.149.0 https://github.com/docling-
project/docling.git
# already have the repo? check out this version:
$ git checkout v2.149.0
```

PyPI attestation · checked 8 Nov Wrong command?

Extract images embedded in PDFs as a new opt-in capability for PDF pipelines.

Use the MiniCPM-OCR model as a new VLM preset for document parsing.

Docling's ASR pipeline now supports mlx-audio as an additional backend.

Docling's CLI has been refactored to be more intuitive.

WAS THIS USEFUL? 👍 👎

anymd

Feedback ⓘ v8.5.10 NOTES ↗

anymd converts any file to clean Markdown for AI agents: PDF, Word, PowerPoint, Excel, EPUB, HTML and web pages.

anymd adds size limits for non-PDF and ZIP conversions and tightens source-access rules for media sidecars

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/sylphxai/anymd:8.5.10
```

npm

```
$ npm install -g @sylphx/anymd@8.5.10
```

git

```
$ git clone --branch v8.5.10 https://github.com/SylphxAI/anymd.git
# already have the repo? check out this version:
$ git checkout v8.5.10
```

published under the project's GitHub account · signed build provenance · checked 8 Nov

Wrong command?

Source-access rules now apply before media sidecars are discovered or opened, so restricted callers cannot read subtitles or sibling media outside their allowed directories.

WAS THIS USEFUL? 👍 👎

kordoc

Feedback ⓘ v4.24.52 NOTES ↗

Kordoc is a 모두 파싱해버리겠다 — HWP·HWPX·PDF·Office 문서를 Markdown으로. 양식 자동 채우기와 신구대조를 갖춘 CLI·MCP 서버 | Convert Korean documents (HWP, HWPX, PDF, Office) to Markdown — CLI and MCP server with form filling and diff

kordoc adds `imagePlacement: "flow"` and `dropBlankImages` options to control how PDF images are placed and filtered during conversion

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install -g kordoc@4.24.52
```

git

```
$ git clone --branch v4.24.52 https://github.com/chrisryugj/kordoc.git
# already have the repo? check out this version:
$ git checkout v4.24.52
```

named in the project's README · checked 8 Nov Wrong command?

└──▷ **TRY IT**

Convert a PDF and keep each image anchored next to its surrounding text rather than pushed to the end of the page — useful when preserving reading order for downstream RAG pipelines.

```
$ npx kordoc 문서.pdf --image-placement flow -o 문서.md
```

Batch-convert scanned government PDFs and discard decorative blank or solid-color images to keep output clean.

```
$ npx kordoc *.pdf --drop-blank-images --jobs 4 -d ./결과
```

Place PDF images inline within the text flow instead of at the page end by setting `imagePlacement: "flow"` (CLI: `--image-placement flow`); each image block gains a `bbox` coordinate; the default remains `"page-end"`.

Strip blank, solid-color, or faint-gradient images from all output formats by enabling `dropBlankImages` (CLI: `--drop-blank-images`); off by default.

WAS THIS USEFUL?  

◆ Vector DBs & RAG

synaplan

Feedback ⓘ v5.7.1 NOTES ↗ LATEST STABLE

Synaplan is an own the flow: one open-source AI platform for your knowledge, models and channels – web, WhatsApp, email, Outlook, Nextcloud & ownCloud. Local and cloud models side by side, with the cost on every answer.

synaplan adds a phone settings hub and AI Act labeling for AI conversations and generated media

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.7.1
```

git

```
$ git clone --branch v5.7.1 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.7.1
```

published under the project's GitHub account · checked 8 Nov Wrong command?

synaplan adds a phone settings hub for managing phone-related configuration from one place, while keeping chat accessible with the keyboard.

synaplan now marks AI conversations and generated media with AI Act compliance labels.

▾ 12 more releases in this issue · 2026-10-12 → 2026-11-05

v5.7.0 NOTES ↗ STABLE

Synaplan adds approval-gate enforcement and audit logging for every tool call, plus account-menu links to both Synaplan websites

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.7.0
```

git

```
$ git clone --branch v5.7.0 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.7.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Every tool call now passes the approval gate and is audited, closing any path that previously bypassed access control or logging.

Synaplan now links both Synaplan websites from the account menu for quicker navigation.

v5.6.5 NOTES ↗ STABLE

Synaplan adds a one-file OpenAI-compatible provider and now shows Demo refusals on the statistics page

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.6.5
```

git

```
$ git clone --branch v5.6.5 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.6.5
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Synaplan now adds an OpenAI-compatible provider by defining it in a single file, reducing the work needed to connect a custom or self-hosted model server.

Synaplan now shows Synaplan Demo refusals on the statistics page, making it possible to track how often the demo model declines requests.

Synaplan now keeps AI-created files out of the default library, preventing generated content from polluting the shared file store.

v5.6.3 NOTES ↗ STABLE

synaplan adds a unified outbound HTTP client with proxy settings, a destination allowlist, and an optional CONNECT gateway for controlled egress

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.6.3
```

git

```
$ git clone --branch v5.6.3 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.6.3
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Control all outbound HTTP traffic through one egress client with proxy settings and a destination allowlist.

Newly linked cloud keys can now watch folders immediately after registration.

Research sessions now choose effort level adaptively based on the chat.

Tasks now return shortcuts and knowledge folders for a watch.

synaplan adds support for the Gemini 4 Argon model at its current introductory price.

+1 more

v5.6.2 NOTES ↗ STABLE

synaplan adds MCP server sign-in, Notion page editing, research mode in web chat, and cloud file/folder watches

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.6.2
```

git

```
$ git clone --branch v5.6.2 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.6.2
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Publish an MCP tool list without an API key, lowering the barrier to exposing internal tools.

Improves cloud connection and organisation login management: revoke one cloud connection without deleting an account that is still in use by other connections, keep a disconnected cloud's record while turning organisation login off independently, and mint a background key for organisation login via platform links.

Adds watches for cloud folders and files so tasks react to remote changes, and reports whether a cloud file is searchable, surfacing file-level search eligibility in the UI.

Read and edit Notion pages from chat.

Expands research mode by making it available in the web chat, allowing it to run as a waiting Saved Task step so long-running research fits into automated task pipelines, and adding a research tab on the AI infrastructure settings page.

+2 more

v5.6.0 NOTES ↗ STABLE

synaplan adds an agentic research loop, sign-up admission controls, server watchdogs, and invitation-code billing

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.6.0
```

git

```
$ git clone --branch v5.6.0 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.6.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Restricts new registrations to paid accounts or invitation codes, with Cloudflare Turnstile bot protection, a registration mode toggle, and automatic deletion of unpaid sign-ups after 7 days; limits API keys to paid plans when billing is enabled; starts a Stripe Checkout session with only an email address before an account exists; and lets admins manage invitation codes from the admin panel — create, edit, revoke, view usage, and redisplay any code for copying at any time.

Show server capacity on the system status page, with a watchdog process running on every server role reporting its capacity.

Adds model slots dedicated to reading sources and fact-checking, and a full agentic research loop — stages and prompts, in-process research spikes scored at a Phase C gate, citation-precision judging, evidence and budget storage, step-by-step

execution with stuck-run recovery, reflection after each round with re-search, and plan/budget-aware loop termination.

Mark models that the upstream provider no longer offers.

Enable plugin frontends to use locale files and the t() translation function.

+3 more

└──▷ **BREAKING ON UPGRADE**

! Free sign-ups on `web.synaplan.com` are disabled; new accounts require a paid plan or an invitation code.

v5.5.4 NOTES ↗ STABLE

Synaplan adds folder-to-task file access, PDF and Office file reading in web search, and maxOutput catalog advertising for desktop server tools

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.5.4
```

git

```
$ git clone --branch v5.5.4 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.5.4
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Task runs now read files directly from a connected folder.

Web search now reads PDFs and Office files it finds, not just web pages.

Desktop catalog now advertises `maxOutput` and names hidden server tools.

Removes the technical owner account requirement from the synascriber component.

v5.5.3 NOTES ↗ STABLE

Synaplan adds meeting-notes transcription, cloud file copy, and web-page reading across all models

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.5.3
```

git

```
$ git clone --branch v5.5.3 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.5.3
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Synaplan now bundles meeting notes (synascriber) with a transcriber image that includes the Prosody module.

Synaplan now reads web pages on every model, not only on Anthropic models.

Synaplan now searches the pinned knowledge folder as a tool during chat.

Files can now be copied into a linked cloud that accepts the operation, queued asynchronously.

Synaplan now prompts for confirmation before replacing a default model that keeps failing.

+1 more

v5.5.1 NOTES ↗ STABLE

Synaplan adds a Demo gateway with `/v1` /embeddings and `/v1` /images/generations endpoints, image/video preview in chat, and a document translation mode

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.5.1
```

git

```
$ git clone --branch v5.5.1 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.5.1
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ TRY IT

Check whether all backend services are ready before routing traffic to a fresh deployment.

```
$ curl https://<your-synaplan-host>/setup/status
```

Check service readiness status via the new `/setup/status` endpoint, surfaced also on the system status page and admin overview.

Preview attached images and videos directly in the chat composer before sending.

Translate documents with a new translation mode in the summary feature.

Access the Synaplan Demo from the welcome screen, setup flow, and chat — including a Demo banner, pill, card, mix row, and auto-leave toast with Undo.

v5.5.0 NOTES ↗ STABLE

synaplan adds all built-in live voices to the model menu, a what was sent reply view, and saved-context selection in incognito chat

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.5.0
```

git

```
$ git clone --branch v5.5.0 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.5.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

synaplan now offers every built-in live voice from the model menu, replacing the previous limited selection.

Chat now shows what was sent for each reply so users can inspect the exact context delivered to the model.

Users can now choose saved context in incognito chat and use live speech on mobile.

AgentSerializer now validates and completes agent definitions at serialization time.

v5.4.2 NOTES ↗ STABLE

Synaplan adds live voice model selection from the composer and improves search with thread context

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.4.2
```

git

```
$ git clone --branch v5.4.2 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.4.2
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Pick a live voice model and start a session directly from an empty composer, without needing a pre-configured model.

Search query generation now uses thread context to produce more relevant queries.

v5.4.1 NOTES ↗ STABLE

Synaplan adds Muse Wonder, Image, and Voice Transcribe AI capabilities alongside live voice conversations

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.4.1
```

git

```
$ git clone --branch v5.4.1 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.4.1
```

published under the project's GitHub account · checked 8 Nov Wrong command?

v5.4.0 NOTES ↗ STABLE

synaplan adds live captions during speech and overhauls the menu, apps directory, and guided tours

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/metadist/synaplan:5.4.0
```

git

```
$ git clone --branch v5.4.0 https://github.com/metadist/synaplan.git
# already have the repo? check out this version:
$ git checkout v5.4.0
```

published under the project's GitHub account · checked 8 Nov Wrong command?

synaplan adds live captions that display in real time while someone speaks via the transcriber.

synaplan overhauls the menu, apps directory, and guided tours with a redesigned navigation experience.

synaplan now defaults image generation and editing to the Flare model.

WAS THIS USEFUL? 👍 👎

serviette

Feedback ⓘ v0.1.8 NOTES ↗

Serviette is a no-code RAG server that keeps answers in sync with your documents. Works with 8 vector databases.

serviette adds a no-code RAG server with live document sync, 8 vector database integrations, and a REST API backed by a single YAML config

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.8
https://github.com/pathwaycom/serviette.git
# already have the repo? check out this version:
$ git checkout v0.1.8
```

└──▷ TRY IT

Spin up a fully working RAG demo against your OpenAI key to test document Q&A before configuring your own sources.

```
$ pip install serviette
export PATHWAY_LICENSE_KEY=<your_key>
export OPENAI_API_KEY=sk-...
serviette demo --embedder openai
```

Let serviette generate a YAML config for your own documents and sources, then bring up the chat UI and REST API.

```
$ serviette wizard
serviette up
```

Run the indexer and query the retrieval API separately to integrate document search into an existing service without coupling it to the chat UI.

```
$ curl http://localhost:8989/api/v1/retrieve
```

Adds new CLI entry points: `serviette demo --embedder openai` starts a demo RAG server with a chat UI at `http://localhost:8989`; `serviette wizard` interactively generates a config file and `serviette up` runs the full stack, exposing a chat UI and the `/api/v1` REST API; and `serviette indexer` runs the indexer separately,

exposing document retrieval at `/api/v1/retrieve` for stateless, horizontally scalable deployments.

Enable hybrid BM25 + vector search, reranking, adaptive RAG (`rag.adaptive`), query decomposition, and MMR each as individual options in the YAML config file.

Connect live document sources — local folders, Google Drive, S3/MinIO, SharePoint, FTP/SFTP/WebDAV — so edits, new files, and deletions appear in answers within seconds with no re-indexing step.

Choose from 8 vector databases in config: DuckDB (embedded default), pgvector, Qdrant, Milvus, ChromaDB, Weaviate, Pinecone, and MongoDB Atlas, each tested against a real instance.

Enable video parsing by setting the `TWELVELABS_API_KEY` environment variable; audio and video indexing activate automatically when the corresponding API keys are present.

+2 more

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

Braintrust

Feedback ⓘ announced · Nov 8, 2026 BLOG ↗

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

Braintrust adds application log ingestion alongside agent traces, available via Python SDK and OpenTelemetry

└──▷ USE IT

Forward existing Python application logs to Braintrust without rewriting log calls — useful when you want to correlate service errors with agent traces.

```python
import logging
from braintrust import BraintrustLogHandler, init_logger

braintrust_logger = init_logger(project="application-logs")
handler = BraintrustLogHandler(braintrust_logger)
app_logger = logging.getLogger("checkout")
app_logger.setLevel(logging.INFO)
app_logger.addHandler(handler)

app_logger.info("Payment %s started", "pay_123", extra={"attempt": 1})

handler.flush()
app_logger.removeHandler(handler)
handler.close()
```

Preserve message templates when logging errors directly so recurring failures can be grouped by pattern rather than individual parameter values.

```python
from braintrust import init_logger

logger = init_logger(project="application-logs")
logger.error("Payment {payment_id} failed", payment_id="pay_123")
logger.flush()
```

Send application logs to Braintrust using `BraintrustLogHandler` and `init_logger` from the Python SDK (v0.42.0 or later), set via `BRAINTRUST_API_KEY`, to forward logs from Python's standard `logging` module without changing existing log calls.

Send logs directly with the Python SDK using logger.error('Payment {payment_id} failed', payment_id='pay_123') to preserve message templates and group recurring

events across different values.

Send application logs from any language over OpenTelemetry (OTLP) via the `https://api.braintrust.dev/otel/v1/logs` endpoint, using `Authorization` and `x-bt-parent` headers, without requiring an active trace.

Braintrust now surfaces application logs and agent traces together in one view, letting teams investigate tool failures, timeouts, and retries with full log context alongside agent steps.

Use Patterns and Loop automations on application logs to identify recurring errors from services such as databases and background workers, independent of agent traces.

+2 more

NEW INTEGRATION Firebase Genkit flow tracing (Go SDK)

NEW INTEGRATION MCP server tracing for fastmcp and mcp-handler (TypeScript SDK)

UPDATED INTEGRATION TypeScript SDK tracing for newer framework and model provider versions

NEW INTEGRATION Anthropic Java SDK 3.x tracing support with async client wrapping

NEW INTEGRATION Temporal workflow tracing (Python SDK)

NEW INTEGRATION OpenAI Responses API tracing (Python SDK)

▾ 1 more release in this issue · seen 2026-11-01 → 2026-11-08

updated · Nov 8, 2026 NOTES ↗ LATEST STABLE

Braintrust adds thread translation, Terraform provider GA, ephemeral service tokens, MCP telemetry opt-out, trace continuation via `--parent`, and new UI features for human review and Loop.

Read a non-English thread in English in the Thread layout using the browser's built-in translation API in desktop Chrome or Edge, without consuming model credits.

Continue a coding-agent trace from an existing span by passing `--parent` to `bt trace run` with the value returned by an SDK span's export(), attaching the session below that span instead of starting a new trace.

Call a Braintrust MCP server tool with only its own arguments — `bt mcp run` no longer requires a telemetry argument.

Braintrust's Terraform provider is now generally available at v1.0.0.

Create service tokens without storing the key in Terraform state or plan files using the new ephemeral `braintrust_service_token` resource.

+8 more

└──▷ BREAKING ON UPGRADE

! The Terraform provider now requires Terraform or OpenTofu 1.11.1 or later.

! `braintrust_ai_secret` now requires `secret_version` alongside `secret`, and `braintrust_env_var` now requires `value_version` alongside `value`; changing only the value no longer updates it, and references to the old `secret` or `value` fields return null.

! The `braintrust_org_automation` Terraform resource is removed; delete its blocks from your configuration and run `terraform state rm` on each address before running `terraform plan`.

WAS THIS USEFUL? 👍 👎

agentic-os

Feedback ⓘ v1.8.37 NOTES ↗

Agentic-os is a governance framework for AI coding agents. It runs them through a five-step workflow (plan, build, review, test, ship) where no step counts as done without evidence.

agentic-os adds per-finding red-team reporting and a new `.acx-incoming` skill delivery mechanism for customized files on upgrade

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.8.37 https://github.com/KbWen/agentic-os.git
# already have the repo? check out this version:
$ git checkout v1.8.37
```

The Lite red-team review now reports every distinct actionable finding instead of capping at `1-2 findings max` and `≤300 tokens`; repeated instances of one root cause are grouped rather than suppressed.

Customized skills (`database-design`, `test-driven-development`, `red-team-adversarial`) now arrive as `.acx-incoming` files on upgrade so local edits are preserved.

New eval data added under `.agentcortex/eval/`.

WAS THIS USEFUL? 👍 👎

versus-incident

Feedback ⓘ v1.4.34 NOTES ↗ LATEST

STABLE

Versus-incident is the self-hosted AI SRE agent. You don't have to monitor your systems alone.

versus-incident adds linked log views for Kubernetes break investigations and fast history access across upgrades

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/versuscontrol/versus-incident:v1.4.34
```

git

```
$ git clone --branch v1.4.34 https://github.com/VersusControl/versus-incident.git
# already have the repo? check out this version:
$ git checkout v1.4.34
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Retire the `kubernetes.evidence.read_models` setting; any existing value is now ignored, and the first start after upgrading may take longer while history is prepared.

versus-incident now adds an optional ingestion-delay setting for linked-log data sources in Kubernetes agent settings, reusing existing data-source connections.

NEW INTEGRATION Open a Kubernetes break before all logs have arrived, with pending coverage shown and automatic retries as the source catches up.

Very large histories now prepare in the background after an upgrade, showing a retryable history upgrading status instead of timing out.

Authorized workload inventory and known counts remain visible during temporary refresh failures, including the first usable partial inventory.

+1 more

└──▷ BREAKING ON UPGRADE

! The `kubernetes.evidence.read_models` setting is retired; any existing value is silently ignored after upgrading.

! The first start after upgrading to v1.4.34 may take longer than usual while investigation history is prepared; no saved history is deleted.

▾ 3 more releases in this issue · 2026-10-17 → 2026-11-04

v1.4.33 NOTES ↗ STABLE

versus-incident adds persistent Kubernetes investigation history and durable long-running sync recovery in v1.4.33

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/versuscontrol/versus-incident:v1.4.33
```

git

```
$ git clone --branch v1.4.33
https://github.com/VersusControl/versus-incident.git
# already have the repo? check out this version:
$ git checkout v1.4.33
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Persist Kubernetes investigation history across restarts and resource deletion, keeping historical observations separate from current active-issue counts.

Investigate retained evidence across 7-day, 30-day, 90-day, or all time windows using the existing time selector, with resource filters and pagination.

File storage now enforces explicit 24-hour, 5,000-record, and 4-MiB aggregate history budgets, with capacity pressure reporting capture failure rather than silently discarding active evidence.

PostgreSQL storage now retains Kubernetes history with no application-imposed age, record-count, or total-byte limit — capacity and lifecycle remain operator-managed.

Memory storage is now documented as ephemeral, with no restart durability provided.

+3 more

v1.4.32 NOTES ↗ STABLE

versus-incident adds logs-at-break capture (up to 24 hours), per-cluster issue alerts (off by default), Agent preview settings, and expanded Kubernetes reader permissions.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/versuscontrol/versus-incident:v1.4.32
```

git

```
$ git clone --branch v1.4.32
https://github.com/VersusControl/versus-incident.git
# already have the repo? check out this version:
$ git checkout v1.4.32
```

published under the project's GitHub account · checked 8 Nov Wrong command?

Read Kubernetes container logs alongside changes and crashes in a workload Timeline, with controls to drag or keyboard-navigate the log-window bracket, filter by Pod, container mode, severity, and search, and load more matching logs without changing the window.

Automatically capture bounded Kubernetes container logs at detected breaks; captures survive Pod deletion and Versus restarts for up to 24 hours, subject to capture, storage, and retention limits.

Enable per-cluster Kubernetes issue alerts from the Issues tab (off by default), choosing Critical or Warning severity and routing notifications through configured alert channels using the existing incident flow.

Open the issue's Timeline or investigate with the DevOps Agent directly from a notification; Slack provides action buttons and other channels provide links.

Save logs, metrics, and traces learning toggles and namespace scope in the cluster's Agent tab as preview settings.

+4 more

v1.4.31 NOTES ↗ STABLE

versus-incident adds opt-in multi-cluster Kubernetes support with fleet health cards, cluster-targeted investigations, live pod log streaming, and expanded crash evidence

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/versuscontrol/versus-incident:v1.4.31
```

```git
$ git clone --branch v1.4.31
https://github.com/VersusControl/versus-incident.git
# already have the repo? check out this version:
$ git checkout v1.4.31
```

published under the project's GitHub account · checked 8 Nov Wrong command?

└──▷ **USE IT**

Opt in to multi-cluster mode and register two clusters so the fleet card view covers both environments.

```yaml
connectors:
  kubernetes:
    multiple: true
    clusters:
      - cluster_id: prod-us-east
        display_name: Production US-East
        # cluster-specific connection settings here
      - cluster_id: staging-eu-west
        display_name: Staging EU-West
        # cluster-specific connection settings here
```

Enable multi-cluster Kubernetes by setting `connectors.kubernetes.multiple: true` and placing each cluster's connection settings under `connectors.kubernetes.clusters`, with a required `cluster_id` and optional `display_name` per entry.

versus-incident now supports connecting multiple Kubernetes clusters without an Enterprise license, with fleet cards showing health, Kubernetes version, node and pod readiness, warnings, issues, and snapshot status.

Select clusters in web chat and the CLI to run cluster-targeted investigations and compare environments using read-only Kubernetes tools.

Stream live pod logs tied to a selected cluster, pod, and container, with pause/resume, timestamps, filtering, and copy/download output.

Enterprise installs can now restrict member and team infrastructure access to selected clusters, with checks applied across fleet cards, dashboards, logs, attachments, action proposals, approvals, and rejections.

WAS THIS USEFUL? 👍 👎

MLflow

Feedback ⓘ CVE-2023-1176 CVE ↗

MLflow tracks, evaluates, and monitors machine learning and LLM applications, covering experiment tracking, tracing, prompt management, and a model registry.

`CVE-2023-1176`: Remote file existence check vulnerability in `mlflow server` and `mlflow ui` CLIs (affects `<= 2.2.0`)

WAS THIS USEFUL? 👍 👎

GOVERN

◆ Model & Agent Protection

stride-gpt

Feedback ⓘ v0.21.0 NOTES ↗

Stride-GPT generates threat models and attack trees with large language models and the STRIDE framework.

stride-gpt adds per-subsystem `--max-llm-calls` and `--max-tool-calls` caps, service-scoped agent planning, DeepSeek JSON support, and a DFD editor in the web UI

└──▷ GET THIS VERSION ▾

pipx

```
$ pipx install stride-gpt==0.21.0
# already installed? upgrade to this version:
$ pipx install --force stride-gpt==0.21.0
```

git

```
$ git clone --branch v0.21.0 https://github.com/mrwadams/stride-gpt.git
# already have the repo? check out this version:
$ git checkout v0.21.0
```

PyPI attestation · checked 8 Nov Wrong command?

└──▷ TRY IT

Save a report to a nested output path — the CLI now creates missing parent directories automatically.

```
$ stride-gpt quick -o ./results/2024-Q3/my-system/report.md
```

▶ View command output v0.21.0

Adds per-subsystem call caps via `--max-llm-calls` and `--max-tool-calls`, so the limit bounds work on each subsystem instead of dropping subsystems when the cap is reached.

stride-gpt now sends `json_object` instead of `json_schema` for DeepSeek models, enabling structured output on DeepSeek providers.

The agent planner now organises subsystems by service and scopes each worker to its own files, replacing the flat planning model, and discovers every key file instead of

stopping at the first 50 per pattern, removing a silent 200-file cap on large repos.

The web UI now shows a generated or parsed data-flow diagram (DFD) in an inline editor.

The web UI now uses one provider-agnostic call for all threat-model tabs, replacing per-provider call paths.

+5 more

└─▷ **BREAKING ON UPGRADE**

- ! The deprecated `gemini-3.5-flash` model is removed; runs referencing it will fail.
- ! Claude Sonnet 5 and Haiku 4.5 are removed; configurations referencing those model names will fail.
- ! The `console` and `auto_approve` parameters are removed from `run_analysis`; callers that pass either will break.
- ! The `anthropic` and `groq` SDKs are dropped from the web extras; installations that relied on them via that extra must install them separately.

WAS THIS USEFUL? 👍 👎

Minimal Software Research minimal

Feedback ⓘ

release-6a01d5ae NOTES ↗ LATEST STABLE

Minimal is a secure, declarative runtimes for Claude Code, Gemini, Codex, Pi and more, so you can run multiple agents safely on your stack from a verified software supply chain.

minimal adds a Box Egress Proxy, secret management via `min secret`, audit trail commands, OpenTelemetry tracing, and a vulnerability reporting flag to `diag`

└──▷ GET THIS VERSION ▾

```
$ git clone --branch release-6a01d5ae
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-6a01d5ae
```

└──▷ TRY IT

Store a secret token for a box to consume, then list all secrets to confirm it was saved.

```
$ min secret set MY_API_TOKEN
min secret
```

Review the full egress audit trail for a box, merging in its parent chain, to verify no unauthorised proxy decisions occurred.

```
$ min box audit --parent my-box
```

File a vulnerability report about a suspected security issue in minimal directly from the CLI.

```
$ min diag --security
```

Read a box's audit trail with `min box audit`, merge parent chains with `min box audit --parent`, and report an edited audit chain in `min doctor`.

minimal adds a Box Egress Proxy (bep) that issues leaf certificates under a name-constrained root and signing CA, holds proxy keys in the host keychain (Secure Enclave on macOS), appends hash-chained audit records per proxy decision, rotates the signing CA through the control socket, bounds the client-key registry with `--max-client-keys` in `bep serve`, and ships as a signed, notarized `bep.app` on macOS.

Send a suspected vulnerability report with `min diag --security`.

Enable opt-in OpenTelemetry export with a local spool via `mlog`, propagating one trace per request across `min` and `minimald`.

Adds GitHub access across the stack: sign in with `min login` under the published GitHub App with credentials kept in the host keychain, minimald ships a git credential helper inside grant boxes so agents can reach GitHub without a manual PAT, and minvmd starts the Box Egress Proxy alongside gvproxy so `git` and `gh` reach GitHub from inside a VM guest.

+4 more

▾ **7 more releases in this issue** · 2026-10-15 → 2026-11-07

release-1e236070 NOTES ↗ STABLE

minimal: Make `min login` the GitHub sign-in entry point, completing authentication on a real page with repository choice.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch release-1e236070
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-1e236070
```

└──▷ **TRY IT**

Store a GitHub token as a named secret in a box so the agent can access it without it ever being typed in plain text in the session.

```
$ min secret set GH_TOKEN
```

Review the tamper-evident audit trail for a box, merging its parent chain, to verify no proxy decisions were altered.

```
$ min box audit --parent
```

Inspect every store reference a box holds alongside the authorities that govern each one.

```
$ min box spec
```

Minimal now ships `minzoned` in a signed macOS app bundle with an app icon.

Minimal now keeps box names answering across reboots and a second VM, and gives a re-created box its own redeemable identity.

`release-47eba5d3` NOTES ↗ STABLE

minimal: Wires a client-key bound into `bep serve` with `--max-client-keys`, admits widened store rules without restarting `bep serve`, and keeps store rules under min

└──▷ GET THIS VERSION ▾

```
$ git clone --branch release-47eba5d3
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-47eba5d3
```

└──▷ TRY IT

Store a secret in a box so an agent can access it at runtime without it ever being pasted into the session.

```
$ min secret set MY_API_KEY
```

Inspect the full audit trail for a box, merging its parent chain, to verify no proxy decisions were tampered with.

```
$ min box audit --parent mybox
```

List all store references and their declared authorities for a box before activating a session.

```
$ min box spec mybox
```

Block UDP port 443 for a `host_ip` cohort with a blocked QUIC stance, and reject UDP to port 443 where a box's QUIC stance is blocked.

`release-6bddc077` NOTES ↗ STABLE

minimal: Adds hash-chained audit trails: read a box's trail with `min box audit`, merging parent chains with `min box audit --parent`, and appends one hash-chained audit

└──▷ GET THIS VERSION ▾

```
$ git clone --branch release-6bddc077
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-6bddc077
```

└──▷ TRY IT

Store a GitHub token as a secret so an agent box can access it without it being pasted into the session environment.

```
$ min secret set GH_TOKEN
```

Inspect the tamper-evident audit trail for a specific box, merging its parent chain, to verify no decisions were edited.

```
$ min box audit --parent mybox
```

Sign in to GitHub under the published App so grant boxes receive sealed credentials via the Box Egress Proxy.

```
$ min login
```

Sign in to GitHub with `min login`, which mints sealed sign-in references and registers the box with the proxy; signing out deletes the reference and refuses all further redemptions.

release-0be7561f NOTES ↗ STABLE

minimal: Store, list, and remove secrets with the new `min secret` command (subcommands `set` and the implied list/remove).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch release-0be7561f
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-0be7561f
```

└──▷ TRY IT

Store a GitHub token as a named secret and make it available to a box without pasting it into the session.

```
$ min secret set GH_TOKEN
```

Inspect the tamper-evident audit trail for a specific box to verify no proxy decisions were edited.

```
$ min box audit <box-name>
```

Sign in to GitHub under the published App so grant boxes can reach GitHub through the Box Egress Proxy.

```
$ min login
```

Admit a widened store rule without restarting `bep serve`.

release-03466f74 NOTES ↗ STABLE

minimal adds opt-in OpenTelemetry tracing across the CLI, daemon, and VM guests, plus a new BEP crate with certificate issuance, proxy-key management, and hash-chained audit

└──▷ GET THIS VERSION ▾

```
$ git clone --branch release-03466f74
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-03466f74
```

Add the `bep` crate with a GitHub module and sealed-envelope primitive for bound-execution-policy enforcement.

(new) Stores proxy keys and the GitHub App credential in the host keychain, publishing proxy key identity so relying parties can verify the source of a proxied request, and signs in to GitHub under the published App credential rather than a user-supplied PAT.

Append one hash-chained audit record per proxy decision to give a tamper-evident log of every BEP redemption.

Evaluate redemption as a pure function over host key, attribution, and bounds so BEP decisions are deterministic and auditable.

release-6c6c39ae NOTES ↗ STABLE

minimal adds opt-in OpenTelemetry export, `min finalize-install`, `min diag upload`, and staged-version install across CLI and daemon

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch release-6c6c39ae
https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout release-6c6c39ae
```

└──▷ **TRY IT**

Pin the installation to a specific staged release instead of pulling the latest stable.

```
$ curl --proto '=https' --tlsv1.2 -fsSL
'https://go.minimal.dev/stable' | sh -s -- --version <version>
```

Complete host DNS setup interactively after installation, review the generated script, and undo it if needed.

```
$ min finalize-install --show --script
min finalize-install --undo
```

Upload a collected diagnostic bundle to the portal for analysis without providing a token.

```
$ min diag collect && min diag upload
```

Install a specific staged version of minimal with `--version` on the installer command.

Upload diagnostic bundles to the portal for analysis with `min diag upload` (no token required).

Run `min finalize-install` (replacing the removed `min net setup`) to complete host DNS setup, with `--show` and `--script` flags replacing `--print`, and `--undo` to reverse it.

Include the telemetry spool in `min bug` reports, with automatic redaction of telemetry settings.

Warn within a minute when the switch datapath is lost.

`+1 more`

└──▷ **BREAKING ON UPGRADE**

! The `[harness]` table in `minimal.toml` no longer loads a stack and now fails to load; rename it to `[stack]`.

! `--token` is removed from `min bug`, `min diag collect`, and `min diag upload`; passing it is now a parse error. The `Status:` line is no longer printed after an upload.

! `min net setup` is removed with no alias; run `min finalize-install` instead (`--show --script` replaces `--print`).

v0.7.0 NOTES ↗ STABLE

minimal adds per-session network isolation with egress rules, named addresses, port publishing, and a new CLI for diagnostics and multi-VM support

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.7.0 https://github.com/gominimal/minimal.git
# already have the repo? check out this version:
$ git checkout v0.7.0
```

└──▷ **TRY IT**

Sandbox an agent so it can only reach one external host — useful when you want Claude to pull from GitHub but nothing else.

```
$ min session activate --network own_ip --allow-dns-hosts github.com
--attach .
```

Forward a port running inside a session to localhost so you can hit it from a browser or curl, without installing anything in the session.

```
$ min net forward my-session 8080:3000
```

Upload a diagnostic bundle for a suspected daemon issue without needing an account, then share the printed link with the team.

```
$ min bug --upload
```

Adds comprehensive session networking controls: pick a session's network mode with `min session activate --network <none|host_ip|own_ip>` (`own_ip` gives the session its own IP and is required for egress rules and port publishing; `none` removes all network access; `host_ip`, the default, shares the host's network); restrict what a session can reach with `--allow-dns-hosts <host>`, `--allow-subnets <cidr>`, `--deny-subnets <cidr>`, `--allow-protocols <tcp|udp|icmp>`, and `--deny-all-egress` (an `own_ip` session with no rules reaches nothing external); inspect the effective policy and runtime-published ports with `min session policy <session>`; publish ports at startup with `--ingress <external>:<internal>` or control dynamic port requests with `--dynamic-ingress <allow|ask|deny>` and `--dynamic-range <lo-hi>`; request a port from inside a session with `min net expose <port>` or forward a session port to localhost with `min net forward <session> <local>:<port>` (works in every network mode); and every session now gets a stable DNS name `<name>.min.internal`, reachable from a browser via a local HTTP proxy printed at activation, or system-wide after `min finalize-install`.

Collect and upload a diagnostic bundle in one step with `min bug --upload`, which prints a shareable report link and a delete command; upload an existing bundle with `min diag upload <path>` and remove it with `min diag delete <id> <token>`.

Run a named VM alongside the default with `--vm <name>` on `min session activate`, giving each VM its own independent state.

Refer to a session by a unique prefix of its session ID in `min session attach`, `exec`, and `policy`.

BREAKING ON UPGRADE

- `[harness]` in `minimal.toml` no longer loads — rename the table to `[stack]`.
- `min ssh-forward` is removed — use `min net forward <session> <local>:<port>` instead.
- The daemon's HTTPS reverse proxy and client certificates are removed, and `min login` no longer issues a certificate — reach sessions by their `<session>.min.internal` name or with `min net forward`.
- Session names must now be valid hostnames (letters, digits, hyphens only; case-insensitive; `host`, `local`, `localhost`, `minimald`, and session-ID-shaped names are reserved) — rename non-conforming sessions with `min session rename`.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic Claude Opus 4.5 Feedback ⓘ

new snapshot · Nov 8, 2026 MODEL ↗

Claude Opus 4.5 is a multimodal model from Anthropic.

New snapshot 20251101 listed on Amazon Bedrock (amazon-bedrock/anthropic.claude-opus-4-5-20251101-v1:0), first seen 8 Nov 2026.

WAS THIS USEFUL? 👍 👎

xAI Grok 4.20 Multi-Agent Feedback ⓘ

new snapshot · Nov 8, 2026 MODEL ↗

Grok 4.20 Multi-Agent is a reasoning model from xAI.

New snapshot 0309 listed on models.dev (xai/grok-4.20-multi-agent-0309), first seen 8 Nov 2026.

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

StepFun Step 3.5 Flash Feedback ⓘ

new snapshot · Nov 8, 2026 MODEL ↗

Step 3.5 Flash is a reasoning model from StepFun.

New snapshot 2603 listed on models.dev (stepfun/step-3.5-flash-2603), first seen 8 Nov 2026.

WAS THIS USEFUL? 👍 👎

Google Gemma 4 26B A4B Feedback ⓘ

price cut · Nov 7, 2026 MODEL ↗

Gemma 4 26B A4B is a multimodal model from Google.

Price cut on OpenRouter (google/gemma-4-26b-a4b-it): $0.10 → $0.0675 in, $0.30 → $0.225 out per million tokens (-32%), first seen 7 Nov 2026.

WAS THIS USEFUL? 👍 👎

Liquid AI LFM2.5-2.6B (Liquid AI) Feedback ⓘ

retirement scheduled · Nov 7, 2026 MODEL ↗

LFM2.5-2.6B (Liquid AI) is a reasoning model from Liquid AI.

OpenRouter lists a retirement date of 13 Nov 2026 for liquid/lfm-2.5-2.6b, first seen 7 Nov 2026.

WAS THIS USEFUL? 👍 👎

inclusionAI Ling 3.1 Flash

Feedback ⓘ

price cut · Nov 7, 2026 MODEL ↗

Ling 3.1 Flash is a chat model from inclusionAI.

Price cut on OpenRouter (inclusionai/ling-3.1-flash): $0.075 → $0.045 in, $0.225 → $0.135 out per million tokens (-40%), first seen 7 Nov 2026.

WAS THIS USEFUL? 👍 👎

Meta Llama 3.2 1B Instruct

Feedback ⓘ

retirement scheduled · Nov 7, 2026 MODEL ↗

Llama 3.2 1B Instruct is a chat model from Meta.

OpenRouter lists a retirement date of 20 Nov 2026 for meta-llama/llama-3.2-1b-instruct, first seen 7 Nov 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · November 8, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue