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

THE AI TOOLCHAIN

NO. 092

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED NOVEMBER 14, 2026 · EVERY DAY

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.

all 413 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 85 of 87 tools

▾ // THE BRIEF

HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: GPT-5.5 | Opus 5

What stands out across today's releases, grouped by what it lets you do. Every tool named links to its entry below.

vim ? help

Three unrelated things stand out today: TRL can keep RL rollouts flowing through a vLLM server instead of blocking training; Mastra can resume interrupted agent streams instead of losing live work; llama.cpp can shrink KV-cache memory and harden local serving defaults.

BUILD

Train RL agents without letting generation stall the trainer

AsyncGRPOTrainer is now a stable top-level trainer that separates sample generation from weight updates through a vLLM server, with new controls for timeout, concurrency, LoRA sync, rollout customization, and agent environments. Teams tuning reasoning or tool-use policies can keep the expensive training loop fed instead of hand-managing rollout workers and synchronization.

TRL

OPERATE

Resume a live agent run after the client drops

Mastra added resumable agent streams with reconnect and cancel semantics. That turns a dropped browser tab or mobile connection into a recoverable stream state rather than a lost run that must be restarted or replayed manually.

Mastra

DEPLOY

Serve local models with less context memory and safer defaults

llama.cpp shipped FP8 KV cache quantization and a host-memory MoE expert cache, reducing the memory pressure that makes long-context and mixture-of-experts serving hard on constrained hardware. Its server also now defaults to SSRF, CORS, and DNS-rebinding hardening, so fewer local deployments start life exposed by configuration mistakes.

llama.cpp

EVALUATE

Run broader agent and safety evals in remote sandboxes

evalscope added nine safety and agent benchmarks plus remote sandbox execution via E2B and Aliyun Agent Sandbox, while e2e added E2B and Cloudflare Browser Run providers with a self-contained HTML reporter. That makes it cheaper to run risky or browser-dependent test cases off the developer machine and send reviewers a portable failure artifact.

evalscope · e2e

Does **GPT-5.5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.189 NOTES ↗ RANK

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

> Xalgorix hardened its local-machine SSRF detection to catch parser-divergent URL forms and locked down file-tool access to credential stores, while also changing ripgrep-based search to include hidden and ignored files.

└──▷ WHAT SHIPPED · 3 FEATURES most completely described first ? what's the number?

01 Credential file protection in file tools

BREAKING 89

`str_replace_editor`, `search_files` and `list_files` now refuse access to `xalgorix.env` files, the provider key store and auth profile store in `Data_Dir` and their temporary copies, and `/proc/*/environ`. The file-tool deny-list now applies to writes as well as reads, so a deny-listed path stays refused even when it falls inside the write allow-list, and entries added through `XALGORIX_READ_DENY_LIST` now also block writes (previously reads only). The `read_deny` list in `/api/status` now shows these newly protected entries.

Check the current deny-list after upgrading to confirm that credential-file paths are present before running scans.

```
$ curl -s http://localhost:<port>/api/status | jq '.read_deny'
```

— Names exact files, config env var and a verifiable status endpoint v4.6.189

02 Expanded SSRF-bypass detection for local-machine checks

65

BREAKING

Local-machine host checks now catch URL forms that Go's URL parser reads differently from curl, Python, glibc and browsers — including missing or extra slashes, backslashes, percent-encoded hosts, and tabs or newlines inside the host. A host that looks numeric but that no IPv4 parser accepts is now refused outright instead of being looked up in DNS and allowed when the lookup fails, so targets that previously relied on that DNS fallback will be rejected.

— Mechanism and parser edge-cases named but no config or command shown v4.6.189

THINNER COVERAGE BELOW

03 ripgrep search now includes hidden and ignored files

`search_files` with ripgrep now searches the path it was given and includes hidden and ignored files, matching `grep -r` behavior; scans that previously excluded such files will now surface them.

— Clear behavior change but no command or config to act on v4.6.189

└──▷ **BREAKING ON UPGRADE**

! Entries added through `XALGORIX_READ_DENY_LIST` now also block file-tool writes — previously they only blocked reads.

! `search_files` with ripgrep now includes hidden and ignored files when searching the given path — scans that previously excluded such files will now surface them.

! A host that looks numeric but that no IPv4 parser accepts is now refused outright instead of being looked up in DNS and allowed when the lookup fails — targets relying on that DNS fallback will be rejected.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI App Builders

Replit Agent

Feedback ⓘ changelog · Nov 14, 2026 CHANGELOG ↗

RANK

Replit Agent is an AI coding agent that builds, tests, and deploys applications in Replit.

> Replit Agent this window gained agentic Google Ads campaign creation and Stripe Link-based checkout approval, a Performance Agent scanner for live apps, per-user Viewer sharing, and full CRUD support for Microsoft Fabric apps.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Google Ads campaign creation via Agent NEW

77

Agent can create Google Ads campaigns — including ad copy, images, and conversion tracking — either by connecting an existing ad account with separate billing or by funding a Replit-managed account with Replit credits. Managed accounts require Replit Pro, a personal Workspace, and auto-reload enabled.

Run an end-to-end Google Ads campaign from inside Replit, funding it with Replit credits under a managed account.

> 📍 In your Replit project (Replit Pro, personal Workspace, auto-reload enabled), open Agent and ask it to create a Google Ads campaign. Review the generated ad copy, images, and conversion tracking setup, then approve spending before ads run. See Agentic Ads for spending controls.

— Names requirements and funding mechanism, includes runnable steps

changelog-20261114-1a76e8df

02 Performance Agent scanner for deployed apps NEW

68

A scan can be run from Monitoring > Run scan with Agent to identify and fix performance bottlenecks — flagged slowdowns and log issues — in a deployed Replit app, with suggested fixes applied and tested before publishing.

Identify and fix performance bottlenecks in a deployed Replit app before pushing changes live.

> 📍 In your Replit project, go to Monitoring › Run scan with Agent, review the flagged slowdowns and log issues, then apply and test the suggested fixes before publishing.

— Exact UI navigation path makes it directly runnable changelog-20261114-1a76e8df

THINNER COVERAGE BELOW

03 **Stripe Link checkout approval in Agent** NEW 54

Ask Agent to compare products and prepare checkout, then connect Link, Stripe's digital wallet, to approve the merchant, total, and any recurring terms before spending is authorized.

– Describes flow but no exact UI path or command changelog-20261114-1a76e8df

04 **Microsoft Fabric CRUD app support** IMPROVED 52

The Microsoft Fabric integration now supports database-backed apps that can create, read, update, and delete records, alongside server-side functions and analytics.

– Names capability scope but no concrete steps given changelog-20261114-1a76e8df

05 **Per-user Viewer sharing** IMPROVED 42

Admins and members can now share projects with specific individual Viewers instead of the entire Viewer group.

– States the change but no UI path or mechanism detail changelog-20261114-1a76e8df

WAS THIS USEFUL? 👍 👎

◆ AI Coding Agents

SST OpenCode

Feedback ⓘ updated · Nov 14, 2026 NOTES ↗ RANK

OpenCode is an open-source AI coding agent for terminal-based development tasks.

> OpenCode ships concurrent background subagents, a configurable transcript verbosity setting, and a Jump to message navigation command.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Configurable transcript verbosity NEW

70

Set `session.verbosity` to `medium` or `high` in your TUI config to see every tool call and thought step instead of the default collapsed summary row.

Show every tool call and thought step in the transcript instead of the collapsed default view.

yaml

```
session.verbosity: high
```

— Names exact config key and values snapshot-20261114

02 Jump to message navigation NEW

70

Jump to any earlier prompt in the current session with the Jump to message command palette entry or the keyboard shortcut Cmd+Alt+G (or Ctrl+Alt+G on Windows and Linux) in the desktop app.

— Exact keyboard shortcut and command named snapshot-20261114

THINNER COVERAGE BELOW

03 Background subagents run concurrently IMPROVED

50

Subagents now run in the background by default, returning immediately so you can keep working in the parent session while multiple subagents run concurrently.

— Describes behavior change but no config or flag named snapshot-20261114

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.4156 CODE ↗ RANK

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 biggest change this window is a breaking rearchitecture of its memory store from a single JSON file to per-memory Markdown files with metadata tracking and automatic retention pruning, alongside a new `magpie provider keys` command for copying API keys between instances and expanded provider/model support (Qwen Token Plan quotas, Gemini native tools, fx reasoning-effort metadata, per-model Codex context windows).

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 **Copy provider API keys between instances** NEW 90

A new `magpie provider keys` command copies all API keys from one provider to another, including disabled ones: `magpie provider keys <id>` prints keys and `magpie provider keys <id> add` imports them, optionally piping via stdin (`-`). All provider keys can also be retrieved at once via the `/api/provider/key` endpoint with the `allKeys: true` field, returning each key one per line plus a `Count` of how many were returned.

Pipe all keys from a local provider to a remote magpie instance over SSH, including any disabled keys.

```
$ magpie provider keys relay | ssh cloud magpie provider keys relay add
```

Add specific keys directly on the command line instead of via stdin.

```
$ magpie provider keys cloud add k4 k1
```

— runnable commands and endpoint given v0.1.4148

02 **Memory store rearchitected to per-file Markdown** BREAKING 80

magpie now stores each memory as its own `<id>.md` file with YAML front matter (fields like `kind`, `scope`, `tags`, `origin`) and the memory text as the body, replacing the single `memory.json` file; hand-edits to any file are detected and treated as user

changes the agent won't overwrite. A new `meta.json` tracks project metadata, published digests, dismissed texts, removed memories, declined suggestions, and per-memory last-used timestamps; an `index.json` cache records each file's size, timestamp, hash, and content so reads parse only changed files; a `journal.json` records every file in a multi-file write before it begins so an interrupted change completes on next read. Existing `memory.json` stores are migrated automatically on first read and renamed to `memory.json.migrated`.

— names every file and its role but no reader command v0.1.4138

03 **Toggle for billion-context compression markers** NEW

75

The `compress.visibilityMarkers` config key controls billion-context's compression markers: the `Markers` field in `BiliSettings` / `BiliChange` is on by default and writing `false` suppresses the lines added after `compress`, `decompress`, or `search_context` calls, while turning it back on removes the key rather than leaving a magpie-chosen value. A `Show compression markers` toggle now appears in the GUI under the Advanced section of billion-context library settings.

— config key and GUI location both given v0.1.4144

04 **Reasoning effort levels surfaced to fx** NEW

70

magpie now populates `reasoning_efforts` in fx's `providers.magpie.model_metadata.<model>` settings, so fx 0.0.17 and later can surface per-model effort levels (none, low, high) in its `/effort` view and Ctrl+T and send them as `reasoning_effort` to the provider.

— config path and UI entry points named, no direct command v0.1.4156

05 **Hidden usage cards and rows synced across tray and TUI**

IMPROVED

70

The tray panel's Allowances tab and the TUI's Usage page now omit cards and individual quota rows hidden in the app's Usage view, using `settings.UsageCardsHidden` and `settings.UsageWindowsHidden`, matching by `cardHideKey` (tray card id plus `|balance` or `|plan`) and by provider plus pool, family, or row name — no restart required. A card is never left with zero rows visible: if hiding individual rows would remove all rows from a card, that card's full row set is preserved.

— settings keys and matching rule named v0.1.4136

06 **Automatic memory retention and pruning** NEW

65

Automatic memory retention now archives candidate memories left unreviewed for 30 days and removes archived or superseded memories after a further 30 days via the new Prune() function, with the window controlled by the `Keep` constant; user-written and pinned (`Fixed`) memories are never removed. Undoing a tidy (`Untidy`) now preserves a

merged memory when one of its source members was already removed by `Prune`, so no data is silently lost on rollback.

— mechanism and constant named but no user-facing command `v0.1.4134`

07 **Per-agent masking coverage counts** IMPROVED 65

The Masking page now counts each agent's turns by day: turns masking looked over (`Looked`), turns where content was changed (`Masked`), and turns sent as written to an unmasked model (`Whole`). This distinguishes an agent with no masked records from one that isn't monitored at all.

— fields named, UI page given as starting point `v0.1.4153`

08 **Qwen AI platform Token Plan quota support** NEW 65

magpie now reads Qwen AI platform Token Plan usage (monthly/weekly window) for providers with `sk-sp-` keys at `token-plan.maas.qianwenaiapi.com` by signing in via the platform's device-code OAuth flow, mirroring the `@qianwenai/qianwen-cli` `qianwen auth login` process.

— endpoint, key prefix, and auth flow named `v0.1.4147`

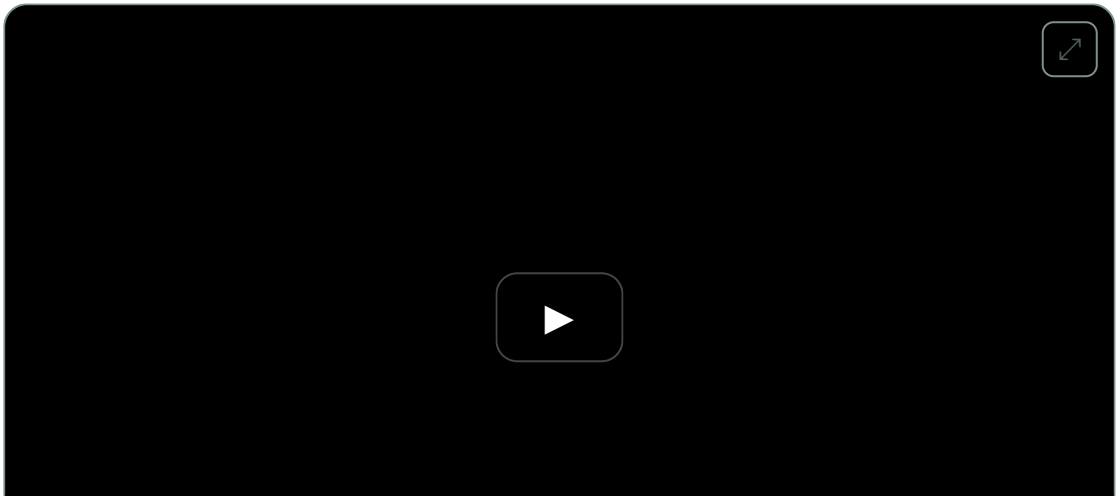
09 **Per-model context windows from Codex catalog** IMPROVED 60

magpie now reads each OpenAI model's context window from Codex's bundled catalog via `codexcat.OwnWindow`, so GPT-6.1 Terra gets its full 500K window rather than being capped at 272K alongside most other models.

— mechanism and numbers given, nothing for reader to run `v0.1.4140`

10 **Bilibili and YouTube ID allowlisting in redaction** IMPROVED 60

magpie's redaction engine now recognises Bilibili BV IDs (e.g. `BV1VMnk68E9Q`) and av numbers by their own shape, and YouTube video, playlist, and channel IDs by their surrounding link, passing them through unmasked so the local model can use them as written — a named `api_key=…` before an ID of either shape is still redacted.





— exact ID formats named, no reader action beyond behaviour v0.1.4132

11 **Native Gemini tool pass-through** NEW 60

Gemini's native tools `googleSearch`, `urlContext` (sent as `url_context`), and `codeExecution` now pass through to Gemini-family models and return `groundingMetadata` (including `webSearchQueries` and `groundingChunks`) in the response, instead of being dropped.

— tool names and response fields given, no command v0.1.4131

THINNER COVERAGE BELOW

12 **OpenMausBot conversation transcripts** NEW INTEGRATION NEW 55

magpie now shows OpenMausBot conversations — including the person's messages, the bot's replies, options-card text, and each tool call with its input and output — via the `openmausbot` agent identifier in the transcript subsystem.

— identifier named but no navigation or command v0.1.4151

└──▷ BREAKING ON UPGRADE

! The `memory.json` file is replaced by a folder of per-memory `<id>.md` files; existing stores are migrated automatically on first read and the original is renamed to `memory.json.migrated`.

v0.1.4146 CODE ↗ STABLE

magpie adds OpenMausBot conversation reading, surfacing chat history, token usage, and tool calls under agent id `openmausbot`.

└──▷ **GET THIS VERSION** ▾

Docker

```
$ docker pull ghcr.io/yetone/magpie:0.1.4146
```

git

```
$ git clone --branch v0.1.4146 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.4146
```

published under the project's GitHub account · checked 14 Nov **Wrong command?**

Reads OpenMausBot conversations from `$OMB_DATA_DIR` or `~/.openmausbot`, pulling prompts, replies, tool calls, titles, working-folder pins, and per-turn token counts into magpie's session list under agent id `openmausbot`.

WAS THIS USEFUL?  

mvschwarz OpenRig

Feedback ⓘ v0.7.15 NOTES ↗ RANK

OpenRig is a network manager for AI coding agents: build persistent teams with roles, shared context and owned work.

> OpenRig's two releases center on operational safety and connectivity: Slack link previews now default to off with explicit toggle control, runner activity grants are bound to the launch lifecycle via a non-reversible migration, and assistants gain MCP-based seat lookup, exact queue filters, and connected-assistant visibility across `rig ps` and the TUI.

└──▷ **WHAT SHIPPED · 14 FEATURES** most completely described first ? what's the number?

01 Slack link preview control per connector

BREAKING 85

`rig slack setup --unfurl on|off` toggles link and media previews per connector, and `rig slack status` reports the active setting. The default has changed to off, so Slack posts no longer show previews after the next connector rewire or daemon restart unless re-enabled.

Turn Slack link and media previews back on for a team that relied on them, after upgrading to 0.7.15 where the default flipped to off.

```
$ rig slack setup --unfurl on
rig slack status
```

— Exact flags, status command, and default-change migration path given. v0.7.15

02 Exact filters on rig queue list

NEW 80

`rig queue list` now supports exact `--tag`, `--mission`, and `--slice` filters, with multi-tag AND matching when several `--tag` values are given.

Narrow a queue listing to items that carry all of several tags at once, in a specific mission and slice.

```
$ rig queue list --tag backend --tag auth --mission onboarding-v2
--slice 3
```

— Exact flags and a runnable example command. v0.7.14

03 Seat restart failure names recovery command

IMPROVED 80

When `rig seat restart --keep-conversation` refuses a recoverable seat whose tmux session is gone, it now names the recovery path directly in the error: `rig launch '<rig>' '<node>'` (add `--plan` to preview before running).

Recover a seat whose tmux session has disappeared after a failed `rig seat restart --keep-conversation`, using the path the error message now names.

```
$ rig launch '<rig>' '<node>' --plan
rig launch '<rig>' '<node>'
```

— Exact recovery command given verbatim. v0.7.15

04 **Runner activity grants bound to launch lifecycle** BREAKING 75

Runner activity grants are now bound to the actual launch, and activating a replacement revokes its predecessor's grant. Migration 165 adds four columns and two indexes to `activity_grants` to support this and cannot be rolled back — downgrading to 0.7.14 afterward is unsupported, so keep a pre-upgrade database backup.

— Mechanism and migration specifics given with backup caveat. v0.7.15

05 **Experimental rig network command** NEW 75

A new experimental `rig network` command starts a connector, sweeps leftover credential folders, and schedules local rounds. It is backed by migration 163, which adds the `network_installations` table and four columns to `network_bindings`, plus migrations 164 and 166 which build supporting indexes; rolling back to 0.7.13 after these run is untested, so keep a pre-upgrade backup.

— Command behavior and migration details both named. v0.7.14

06 **Node reads no longer single snapshot** BREAKING 70

Node reads (`rig ps --nodes`, the TUI) are no longer a single snapshot across steps: in a burst each response arrives sequentially, raising median latency from ~21 ms to ~43 ms in a 24-request burst.

— Exact before/after latency numbers but no mitigation. v0.7.14

07 **MCP seat lookup by plain name** NEW 70

An assistant connected over MCP can now reach a seat by its plain name; ambiguous or unknown names return `destination_ambiguous`, `destination_not_found`, or `destination_unchecked` errors.

— Named error codes with moderate mechanism detail. v0.7.14

08 **Conversation-based restore for seats missing startup context**

IMPROVED 65

Restore of an eligible seat with a resume token and no startup context now attempts the saved conversation and may report `partially_restored`, where it previously stopped outright with `Harness not started: no startup context`.

— States exact before/after status and error string. v0.7.15

09 **Setup prompts for missing logins and unregistered persons**

NEW 65

`rig doctor` and setup now warn when a seat's runtime isn't covered by a configured login, naming the missing sign-in and prompting for it. Separately, when no persons are registered, `rig daemon start` and `rig gateway participants` now prompt the setup agent to offer registration.

— Named commands for both prompts, no deeper mechanism given. v0.7.14

10 **Daemon startup readiness budget extended** IMPROVED 60

`rig daemon start` now gives the daemon child a 180-second readiness budget instead of 20 seconds, printing progress about every 20 seconds while it waits.

— Exact before/after numbers, no further mechanism. v0.7.15

11 **Connected assistant visibility in ps and TUI** NEW 60

Connected assistants, along with their check-in time, now appear in `rig ps`, `rig_team`, and the TUI's tables.

— Named surfaces but no deeper mechanism. v0.7.14

THINNER COVERAGE BELOW

12 **Automatic Codex hook approval renewal** IMPROVED 50

OpenRig now renews its own changed Codex hook approvals automatically at daemon start and at each matching seat's launch, instead of stopping at Codex's hook review.

— Explains behavior change but no config surface to act on. v0.7.15

13 **Higher tmux open-file limit** IMPROVED 50

New tmux servers now raise their soft open-file limit to 8,192 (or the hard limit), reducing descriptor exhaustion as panes and viewers accumulate.

— Specific number given but no user action needed. v0.7.14

14 **Slow-operation log covers more queue and inventory steps**

IMPROVED 50

The slow-operation log now covers three additional queue sites and three nodes-read steps: `nodes.inventory.build`, `nodes.inventory.activity_sync`, and `nodes.inventory.terminal_work`.

— Named steps but no detail on log format or access. v0.7.15

└──▷ BREAKING ON UPGRADE

! Slack posts to a person no longer show link and media previews by default after the next connector rewire or daemon restart. Run `rig slack setup --unfurl on` to restore previous behaviour.

! Migration 165 adds four columns and two indexes to `activity_grants` and cannot be rolled back; downgrading to 0.7.14 after the migration runs is unsupported — keep the pre-upgrade database backup.

! Restore of an eligible seat with a resume token and no startup context now attempts the saved conversation and may read `partially_restored`, where it previously stopped with `Harness not started: no startup context`.

! Node reads (`rig ps --nodes`, the TUI) are no longer a single snapshot across steps: in a burst, each response arrives sequentially, raising median latency from ~21 ms to ~43 ms in a 24-request burst.

! Migration 163 adds the `network_installations` table and four columns to `network_bindings`; migrations 164 and 166 build indexes. Rolling back to 0.7.13 after these migrations is untested — keep the database backup made before upgrading.

WAS THIS USEFUL? 👍 👎

ogcode

Feedback ⓘ v0.59.2 NOTES ↗ RANK

ogcode executes code tasks via a local-first AI agent that manages tools, memory, permissions, and human oversight to turn intent into completed work.

> ogcode's v0.59.2 release overhauls its logging and telemetry pipeline, adding per-process log files, configurable console verbosity and retention, automatic credential redaction, and clearer telemetry drop reporting.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Per-process log files with console level and retention controls

NEW 90

Each command now writes its own rotated log file under `~/.ogcode/logs/<project>/`: `ogcode.log` for `ogcode` and `serve`, plus `run.log`, `index.log`, `worker.log`, and `nativehost.log`, with the control plane writing to `~/.ogcode/logs/controlplane.log`. Terminal log verbosity is overridable with `OGCODE_LOG_CONSOLE` (set to `off` to silence it; new default is warnings instead of errors-only), and idle project log folder retention is configurable with `OGCODE_LOG_PROJECT_MAX_AGE_DAYS` (default 90 days, `0` keeps them forever).

Silence all terminal log output and disable telemetry shipping in a deployment where stdout must stay clean.

```
$ OGCODE_LOG_LEVEL=off ogcode serve
```

Keep project log folders for 30 days and show warnings in the terminal during a development session.

```
$ OGCODE_LOG_PROJECT_MAX_AGE_DAYS=30 OGCODE_LOG_CONSOLE=warn
ogcode serve
```

— Names every log file path and both env vars with defaults v0.59.2

02 Credential redaction for logs and telemetry NEW 75

A new redaction layer scrubs log records and shipped telemetry of credential-shaped content, including `token`, `secret`, `password`, `api_key`, `authorization`, `cookie`, and `credential` attributes, `Bearer` / `Basic` header values, `sk-…`, `ghp_…`, `AKIA…`, and JWT formats, as well as content-bearing query parameters (`q`, `prompt`, `text`, `path`, `directory`, `url`).

— Lists exact fields and patterns but no user-facing control v0.59.2

THINNER COVERAGE BELOW

03 **PostHog dropped-record summary at shutdown** IMPROVED

40

PostHog telemetry shipping now counts dropped records across six categories (queued, encode, request, transport, status, unflushed) and logs one summary when the process shuts down.

— Describes the counting mechanism but offers no reader action v0.59.2

WAS THIS USEFUL? 👍 👎

coffee-cli

Feedback ⓘ v4.0.2 NOTES ↗ RANK

coffee-cli is a clean AI terminal workspace with quick input, folder switching, copy & paste, and image pasting.

> Coffee CLI's v4.0.2 release introduces QR-code browser remote control for mirroring sessions from any device on the local network or over a virtual LAN, while retiring the older Coffee Agent and its chat-app integrations.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Browser remote control via QR code** NEW 62

Coffee CLI adds browser-based remote control: scanning a QR code from any phone, tablet, or other computer mirrors and operates sessions, terminals, settings, and themes live over the local network with no app install required. It also supports remote access over virtual LANs such as Tailscale or ZeroTier, surfacing an extra away-from-home pairing address when both ends run a virtual LAN tool.

– Describes mechanism and named VLAN tools but no exact UI steps or commands v4.0.2

THINNER COVERAGE BELOW

02 **Coffee Agent and chat channels retired** BREAKING 30

Coffee Agent and its WeChat and Telegram channels are retired; the new browser remote control feature replaces them.

– States the removal and replacement but no migration steps v4.0.2

03 **Worktrees grouped in usage dashboard** IMPROVED 25

Coffee CLI now groups worktrees by project in the usage dashboard, which also appears immediately on open.

– Brief description with no further mechanism detail v4.0.2

└──▷ **BREAKING ON UPGRADE**

! Coffee Agent and its WeChat and Telegram channels are retired; the browser remote control feature replaces them.

WAS THIS USEFUL? 👍 👎

keploy

Feedback ⓘ v3.7.5 NOTES ↗ RANK

keploy is an open-source platform for creating safe, isolated production sandboxes for API, integration, and E2E testing.

> Keploy's latest releases add external observability exports from its agent and a naming mechanism for worker-mock rules to ensure consistent mock matching across test runs.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Worker-mock rule naming** NEW 37

Mocks can now be marked with a user-defined worker-mock rule name so they are identified and consistently served to every test that matches that rule.

— Names the mechanism but lacks exact syntax or config key v3.7.4

02 **Agent caller and liveness read exports** NEW 33

Keploy's agent now exports caller and liveness reads, making agent observability data accessible externally.

— Describes what changed but no mechanism or access method v3.7.5

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.112 NOTES ↗ RANK

GitHub Copilot CLI is a terminal-based coding agent that can plan, write, and test code in local repositories.

> GitHub Copilot CLI's two releases focus on MCP server reliability and OAuth proxy support, extend the AI credit cap to subagents, add Microsoft 365 sign-in for workflows and regex support in enterprise permission rules, plus a batch of smaller fixes to plugins, `/diff`, sessions, and exports.

└──▷ **WHAT SHIPPED · 15 FEATURES** most completely described first ? what's the number?

01 MCP command reliability and reporting improvements IMPROVED

90

`copilot mcp remove` and `/mcp delete` now explain that a server passed with `--additional-mcp-config` comes from that option and cannot be removed, instead of reporting it as not found or built-in; `copilot mcp get` correctly reports unconfigured server names such as `toString` as not found in both text and JSON output; `/mcp delete` now removes a server without restarting other connected MCP servers; `/mcp search` installs registry servers with URL variables using the values you enter; OAuth sign-in for remote MCP servers now works through proxies that require Kerberos (Negotiate) authentication; and MCP token counts now exclude tools deferred by client-side tool search.

— Describes exact before/after behavior with named commands Four named mechanisms with clear before/after behavior

v1.0.112 v1.0.111

02 Stricter /model plan keyword handling IMPROVED

75

`/model plan` now reads `PLAN` and `--plan plan` as the plan keyword, and refuses extra arguments instead of silently saving or clearing the plan model.

— Names exact command and keyword variants v1.0.112

03 Remote session creation and control NEW

75

The `--host-allow-session-creation` flag allows one-off remote session creation without permanently changing saved settings, and sessions started with `/new` or from the Sessions tab can now be controlled from GitHub when remote sessions are enabled.

Let a remote SDK client create a new session on a CLI host for a single launch, without permanently changing saved settings.

```
$ copilot --host-allow-session-creation
```

— Named flag with a runnable example command Names UI surfaces but no exact steps

v1.0.111

04 **Silent compaction output flag** NEW 75

Running `copilot -p "/compact" --silent` suppresses compaction result output in non-interactive pipelines, so only the agent's response goes to stdout.

Run a compaction prompt silently in a pipeline so only the agent's response goes to stdout, not the compaction result line.

```
$ copilot -p "/compact" --silent
```

— Named flag with a runnable example command v1.0.111

05 **MCP tool snapshot cache control** IMPROVED 73

Setting the `COPILOT_MCP_TOOL_CACHE` environment variable controls the MCP tool snapshot cache for every server; it is now documented in `copilot help environment`.

Disable the MCP tool snapshot cache when troubleshooting stale tool definitions from an MCP server.

```
$ export COPILOT_MCP_TOOL_CACHE=0
copilot
```

— Names exact env var and shows a runnable example v1.0.112

06 **/diff rendering fixes** IMPROVED 70

`/diff` now shows staged renames under their new names including edits made while renaming, displays a clear notice for binary files instead of garbled text or an empty pane, lists copied files correctly when Git copy detection is enabled, and preserves files whose names end in spaces.

— Names command and several concrete fixed cases v1.0.111

07 **Trusted folder hook load warning** IMPROVED 65

Running `copilot -p` with `--trust-folder` from a subdirectory now warns when trusted repository hooks are not loaded, with instructions on how to load them.

— Names flag and trigger condition clearly v1.0.111

08 **Skip /tmp permission prompts on macOS** IMPROVED 65

Copilot CLI now skips `/tmp` path permission prompts on macOS for scratch files unless `--disallow-temp-dir` is set.

— Names the opt-out flag and exact condition v1.0.111

09 **Plugin hook variable expansion and config error naming** IMPROVED 65

`${COPILOT_PLUGIN_ROOT}` now expands in plugin hook working directories and environment values, and `copilot plugin enable`, `disable`, `update`, and `uninstall` now fail and name `config.json` when it cannot be parsed.

— Names variable, commands and file, but no runnable example v1.0.111

10 **Microsoft 365 sign-in for workflows** NEW 60

`copilot workflow run --help` now lists a Microsoft 365 account among the sign-ins that can run workflows, alongside usage-based billing and a custom model provider (BYOK) while signed out of GitHub.

— Names surfaces but no command to try it directly v1.0.112

THINNER COVERAGE BELOW

11 **Expanded /chronicle cost-tips recommendations** IMPROVED 50

`/chronicle cost-tips` now includes Copilot App sessions and recommends cheaper models and reasoning settings based on your usage.

— Names the command but mechanism is only described briefly v1.0.112

12 **HTML export preserves generic type syntax** IMPROVED 50

HTML exports and rich-text copies now preserve generic types and placeholders written in angle brackets, such as `Vec<String>` and `<file>`.

— Concrete examples given but passive output behavior v1.0.111

13 **Team plugin install from branch, tag, or subfolder** NEW 40

Team plugins now install from marketplaces on a specific branch or tag, or in a repository subfolder.

— One-line capability with no command or flag given v1.0.112

14 **Regex support in enterprise permission rules** NEW 40

Enterprise-managed Shell, Bash, and PowerShell permission rules now support regular expressions.

— States capability without example syntax v1.0.111

15 **Model-recommended default context window** IMPROVED 35

GitHub Copilot CLI now uses each model's recommended context window by default, while preserving explicit context choices.

— Describes a default behavior change with no named control v1.0.112

WAS THIS USEFUL? 👍 👎

codealta

Feedback ⓘ 4.4.0 NOTES ↗ RANK

Codealta is an agentic Development Environment

> CodeAlta 4.4.0 adds inline shell commands in the prompt, Vega-Lite charting in answers and plugins, a global network proxy setting, custom profile folder locations, and nested monorepo project maps.

└──▷ **WHAT SHIPPED · 8 FEATURES** most completely described first ? what's the number?

01 Custom profile folder via --profile / CODEALTA_HOME NEW

90

Keep CodeAlta's whole profile in a folder of your choice with `alta --profile <folder>` or the `CODEALTA_HOME` environment variable, so an organization-trusted directory can be used instead of `~/.alta`.

Run CodeAlta Desktop with a company-approved profile directory so credentials and settings stay in an audited location instead of the default home folder.

```
$ alta --profile /path/to/company-profile
```

— Named flag, env var and default path with runnable example. 4.4.0

02 Inline shell commands in the prompt NEW

85

Type `!` before a command in the prompt to run it directly — the output appears live in a panel and is attached to your next message for the agent to read, and `Alt+↑` recalls previous commands.

Paste a shell command into the agent prompt to capture its output as context — useful when an agent asks you to run something and you want it to see the result immediately.

```
$ ! git log --oneline -5
```

— Exact trigger character, recall shortcut, and runnable example given. 4.4.0

03 Vega-Lite charts in answers and plugins NEW

85

Ask an agent for a chart and it writes a Vega-Lite block that CodeAlta Desktop renders inline in answers, Code Mode cards, and Markdown files — data can come from inline values or a CSV, TSV, or JSON file in the project, and charts can be copied as PNG or SVG. Plugins can also draw charts via Observable Plot and `VegaLite` from the `codealta`

module, demonstrated by the new `canvas-plot` sample in the `codealta-plugin-runtime` skill.

– Names rendering surfaces, formats and a sample, but no exact command. 4.4.0

04 **Interactive vault unlock in headless CLI runs** NEW

80

`altacli --interactive-vault` unlocks interactive vault access in headless runs, letting a CLI session prompt for a locked vault's credentials.

Let a CI script using altacli unlock a vault-protected provider key interactively instead of failing silently when the vault is locked.

```
$ altacli --interactive-vault run "Run the nightly integration tests and open a PR for any failures"
```

– Named flag with a runnable CLI example. 4.4.0

05 **Global network proxy setting** NEW

75

Set the network proxy for CodeAlta and every program it starts — Claude Code, Codex, git, npm, MCP servers, shells and terminals — via Settings > Advanced > Network, with options for system proxy, no proxy, or a manual proxy with bypass list and credentials.

– Exact UI path and affected tools named, but no config key or CLI. 4.4.0

06 **Model tier selector for plans and tasks** NEW

65

Start a plan or task on a model tier (Complex, Standard, Quick) or on Adaptive using the new Runs with selector on work-item cards and in the Work items tab.

– Names tiers and UI location but no exact steps or config. 4.4.0

07 **In-window confirmation for sensitive secrets** NEW

60

Secrets that require confirmation before use now prompt inside the window with a named dialog at the bottom right, confirmed via Windows Hello or Touch ID, with sub-agents inheriting their parent's answer.

– Describes mechanism and inheritance but no named flag or setting. 4.4.0

THINNER COVERAGE BELOW

08 **OpenCode Zen and OpenCode Go providers** NEW

55

Use OpenCode Zen (pay-as-you-go) and OpenCode Go (subscription) as built-in providers by adding a key from the OpenCode console.

– Names both providers and setup source, lacks exact config key. 4.4.0

WAS THIS USEFUL? 👍 👎

gentle-ai

Feedback ⓘ v4.5.0 NOTES ↗ RANK

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's v4.5.0 release surfaces previously hidden review findings, brings Codex up to parity with other supported agent hosts, and lowers the Windows installer's minimum supported build.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Non-blocking findings for stopped reviews** NEW 81

A new `--non-blocking-findings` flag lets callers retrieve informational and refuted findings from a review that stopped at `correction_required`, which previously went unreported. The provider contract (now 1.5.0) also adds a `correction_request_unavailable` stop reason.

Retrieve informational and refuted findings from a review that stopped at `correction_required`, which previously went unreported to the caller.

```
$ gentle-ai review status --non-blocking-findings --contract gentle-ai.review-integration/v2
```

— Names flag, stop reason, contract version, and includes runnable command v4.5.0

THINNER COVERAGE BELOW

02 **Lower Windows installer minimum build** IMPROVED 40

The Separate installer now accepts Windows 10 22H2 (build 19045 and newer) on amd64, where build 22000 was previously the minimum.

— Exact build numbers given but no action beyond installing v4.5.0

03 **Codex module split and multi-agent configuration** IMPROVED 32

Codex now gets the core-plus-on-demand module split and current multi-agent capability configuration, bringing it in line with the other supported agent hosts.

— States what changed but no concrete config or command v4.5.0

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.325 NOTES ↗

CODE ↗ RANK

Claude Code is Anthropic's terminal coding agent that plans, edits, and tests code in local repositories.

> Claude Code's single release in this window focused on small UX clarifications: clearer recovery guidance when org policy blocks a request, a notebook re-run tip in integrated terminals, and a warning when sandboxed git operations can't touch protected files.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Recovery guidance after policy-blocked requests NEW

65

When an org policy blocks a request, Claude Code now shows how to continue: the error includes instructions to run `/rewind` and change the message, or run `/clear`.

— Names exact commands but limited mechanism detail v2.1.325

02 Sandbox warning for blocked git operations on protected files

IMPROVED

60

Claude Code now adds a note for Claude when a sandboxed `git checkout`, `git stash`, or autostash pull cannot update protected files such as those in `.claude`.

— Names specific git commands and protected directory v2.1.325

THINNER COVERAGE BELOW

03 Notebook re-run tip in integrated terminals IMPROVED

40

Claude Code now shows a spinner tip in VS Code and similar integrated terminals suggesting the user re-run a notebook cell after Claude has read or edited a Jupyter notebook.

— Describes UI nudge but no config or command v2.1.325

WAS THIS USEFUL? 👍 👎

The Open Engine Zeroshot

Feedback ⓘ v16.0.0 NOTES ↗

RANK

Zeroshot orchestrates executor-verifier agent loops that implement and validate software changes while logging workflow state to SQLite.

> Zeroshot v16 makes checkpoint rollback during resume explicit and consent-gated, while adding richer error codes, Claude Code permission-denial logging, GitLab pipeline failure diagnostics, and bounded reconnects.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 Explicit consent for checkpoint rollback on resume

BREAKING

93

Restoring a selected checkpoint with `zeroshot resume RUN_ID --from-checkpoint ID` now requires the additional flag `--discard-workspace-changes`, so rollbacks are always explicit and the workspace is never silently overwritten; omitting it fails the command before credentials are resolved or the workspace changes. Plain `zeroshot resume RUN_ID` now restarts on the retained checkout as-is, preserving commits, staged, unstaged, untracked and ignored files, the attached branch, `HEAD` and index, instead of restoring the latest checkpoint snapshot over work done after a failure.

Resume a failed local run from a specific checkpoint while explicitly consenting to discard workspace changes — required as of v16.

```
$ zeroshot resume RUN_ID --from-checkpoint ID --discard-workspace-changes
```

Resume a failed local run on the retained checkout without touching staged or unstaged work — the new default behavior in v16.

```
$ zeroshot resume RUN_ID
```

— Exact flags and commands given, full before/after behavior v16.0.0

02 Claude Code permission denial logging NEW

70

`zeroshot logs` output now reports Claude Code `permission_denials` — including denied tool names and counts — as error-level lines for unattended nodes whose tool calls were refused.

— Names the exact log field and command surface v16.0.0

03 **Distinct error codes for failed and invalid runs** IMPROVED

65

Failed runs now return `run.failed` with `details.runId`, and `request.invalid` is returned for unknown targets and registry resume refusals, distinguishing those cases from `target.unavailable` transport errors; SDK mappings and documentation were updated to match.

— Names the specific error codes but no command to trigger them v16.0.0

THINNER COVERAGE BELOW

04 **GitLab pipeline failure diagnostics** IMPROVED

55

Failed required GitLab pipelines now report failed job names, stages, reasons, configuration errors, and bounded redacted log excerpts.

— Describes what's reported but no command or config shown v16.0.0

05 **Bounded reconnect attempts for streaming commands** IMPROVED

48

Reconnect attempts are now bounded when `run`, `watch`, `logs` or `attach` follows a direct or hosted target, with the first connection announced.

— Names affected commands but no limit or config given v16.0.0

06 **Publish now delivers from checked-out branch** BREAKING

40

Publishing now delivers from the checked-out branch instead of the previously documented branch behavior; worktrees are now opt-in rather than automatic.

— States the behavior change but no migration mechanism given v16.0.0

└──▷ **BREAKING ON UPGRADE**

! Plain `zeroshot resume RUN_ID` no longer restores the latest checkpoint snapshot over the working directory; it restarts on the retained checkout as-is. Runs that relied on the previous snapshot-restore behavior on plain resume will now keep post-failure workspace changes instead of reverting them.

! Restoring a selected checkpoint with `--from-checkpoint ID` now requires `--discard-workspace-changes`; omitting it causes the command to fail before credentials are resolved, a successor is claimed, or the workspace changes.

! Publishing now delivers from the checked-out branch instead of the previously documented branch behavior; worktrees are opt-in.

WAS THIS USEFUL? 👍 👎

agentty

Feedback ⓘ v0.9.55 NOTES ↗ RANK

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

> Agentty added a single environment variable to work around an empty model list issue with the ChatGPT backend.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Override Codex client version reported to ChatGPT** NEW

60

Set `AGENTTY_CODEX_CLIENT_VERSION` to report a different client version to the ChatGPT backend when the model list comes back empty.

— Names exact env var and condition, but lacks broader mechanism detail v0.9.55

WAS THIS USEFUL? 👍 👎

forgedock

Feedback ⓘ v1.60.0 NOTES ↗ RANK

ForgeDock — autonomous AI dev pipeline for Claude Code. GitHub as agent memory: issue in, PR out.

> ForgeDock's v1.60.0 release overhauls its release pipeline with a planner, rollback and incident handling, adds per-repo lane container images with sandbox tool installs, and introduces a passive triage mode for issue-watch alongside new investigation and telemetry data.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 Per-repo lane tool images and sandbox installs NEW

83

ForgeDock can build a per-repo lane container image from a tool recipe registry by setting `execution.lane_tools` in `forge.yaml`, and can install missing tools in a remote sandbox from a recipe, with opt-in host fallback (off by default when `host.role` is `orchestrator`).

Build a custom lane image for a repo that needs non-standard tools, without modifying the host environment.

```yaml
# in forge.yaml
execution:
  lane_tools: <tool-recipe-registry-entry>
```

— Names exact config key, file, and fallback condition with example v1.60.0

02 Release pipeline: planner, rollback, incident lanes NEW

70

ForgeDock's release pipeline adds a release planner (exclusions, a dependency DAG, lanes, a plan hash, and approvals), a rollback ladder and incident flow, a gate-only fix lane that re-evaluates gates at HEAD without triggering a new merge cycle, a GitHub conductor workflow for host-less recovery, a managed P0 lane and a live-base rule, merge-back with validation after fast lanes, credential split and a workflow-computed release gate, a service-scoped bug-fix train lane, and multi-wave release start, flag policy, and a milestone lane.

— Many named release-pipeline additions but no commands or config keys v1.60.0

THINNER COVERAGE BELOW

03 Self-recovery for review and conformance findings IMPROVED

55

ForgeDock adds self-recovery to review and conformance findings — builder-fixable defects now exit `needs-human` after one resume.

— Describes before/after state change but no invocation detail v1.60.0

04 **Per-lane wait and cold-start telemetry** NEW

45

ForgeDock now records per-lane `waits.start` and cold-start seconds in bench-adversarial rows.

— Names recorded fields but no access method given v1.60.0

05 **Passive triage mode for issuewatch** NEW

40

Issue-watch gains a passive triage mode that classifies and labels issues without ever claiming them, for `issuewatch`.

— States behavior but no flag or command to enable it v1.60.0

06 **Structured classification record for investigations** NEW

40

ForgeDock adds a structured classification record with mechanical `exists_in` evidence to the investigation flow.

— Names the field but gives no usage path v1.60.0

WAS THIS USEFUL? 👍 👎

flow-next

Feedback ⓘ flow-next-v9.10.0 NOTES ↗ RANK

flow-next is a workflow plugin that takes a bug, idea or ticket to a verified pull request with specs, cross-model review, live QA, and receipts

> flow-next adds end-to-end Bitbucket Cloud support, bringing make-pr, resolve-pr, and land to Bitbucket-hosted repositories with full auto-detection and pilot status reporting.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Bitbucket Cloud support across make-pr, resolve-pr, and land**

NEW 95

Authenticate and configure Bitbucket Cloud with `bkt auth login https://bitbucket.org --kind cloud --username <Atlassian email> --web-token`, then `bkt context create bitbucket --host https://api.bitbucket.org/2.0 --workspace <workspace>` to enable all three Bitbucket commands. The full spec-to-merge loop runs unattended via `flow --auto --until=merge`, with the forge auto-detected from the origin host (`bitbucket.org`, `ssh.bitbucket.org`, or `altssh.bitbucket.org`) with no extra setting. `make-pr` opens, updates, and configures Bitbucket Cloud pull requests, supporting `--draft`, `--ready`, and `--update`, and stacked specs target the parent branch while its pull request is open. `land` squash-merges Bitbucket Cloud pull requests after checking build statuses, merge checks, and review approvals, naming both heads if a push slips in between the readiness check and the merge. `flowctl pilot snapshot` now reports `forge: "bitbucket"` and the open pull request's readiness for Bitbucket Cloud origins.

Authenticate bkt against Bitbucket Cloud, then set up a workspace context so make-pr, resolve-pr, and land can all operate on a Bitbucket-hosted repository.

```
$ bkt auth login https://bitbucket.org --kind cloud --username
you@example.com --web-token
bkt context create bitbucket --host https://api.bitbucket.org/2.0
--workspace my-workspace
```

— Names exact commands, flags, hosts, and merge mechanics for the new forge.

flow-next-v9.10.0

WAS THIS USEFUL? 👍 👎

Tester Army e2e

Feedback ⓘ @e2e-dev/e2b@0.0.1 NOTES ↗

RANK

e2e is an end-to-end testing framework that automates UI tests across web and mobile applications.

> e2e shipped a self-contained HTML test reporter, an 8–15x faster native PNG codec for screenshots, new E2B sandbox and Cloudflare Browser Run providers, diff-scoped exploration via `e2e explore --changed-since`, Bun runtime support, multi-tab browser control, and relevance-scored test ordering, alongside a breaking change to origin-scoped credential routing.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 Diff-scoped exploration with `--changed-since` NEW

95

Explore only what a branch changed with `e2e explore --changed-since <ref>`, which reads the commit messages and diff since the merge base, plans steps for affected screens, and records the change as `run.explore.change`; explore() on `e2e/runner` accepts `changedSince` too. A change that no user can see (docs-only, tests-only) is skipped when no goal is given — the run passes with no step, `run.explore.ended` reads `skipped`, and the comment headline reads `nothing to explore`.

Scope an exploration run to only the screens a branch could affect, so CI reports issues caused by that change and flags unrelated ones only as warnings.

```
$ npx e2e explore --changed-since origin/main
```

— Runnable command plus exact skip-state field names e2e@0.23.1

02 Origin-scoped credential routing for web engine BREAKING

90

web({ headers }) and web({ basicAuth }) can be scoped to additional origins by listing them in `credentialOrigins: ['https://api.example.com']`, covering both headers and basic-auth credentials. Breaking: headers now go only to the app's own origin (scheme + host + port of `app.url`), not every host on the site, and basicAuth answers 401 only from the app's own origin; other origins must be added to `credentialOrigins`, and apps whose `app.url` redirects to a different origin must list the destination or update `app.url`. Engines now receive the app origin as `EngineAppInfo.origin` in `prepare` and `init`, scoped to the run's port when

`app.url` requests a free one, so an engine can scope an injected credential to the exact origin.

Allow a separate API origin to receive the same auth headers and basic-auth credentials that go to the app origin.

```typescript
web({
  basicAuth: { username: 'admin', password: 'secret' },
  headers: { 'X-Api-Key': 'tok_xyz' },
  credentialOrigins: ['https://api.example.com']
})
```

— Config key, migration note and before/after fully specified `e2e@0.23.1` `@e2e-dev/web@0.16.0`

---

03 **Self-contained HTML test report** NEW 90

Call htmlReporter() in the e2e config to write the full test run as a single HTML page at `.e2e/report.html`, openable from disk with no server required. Set `media: 'embed'` to inline screenshots and videos directly into the report, or `media: 'none'` to omit all media. The report lists tests by file with status tabs, search, and per-test attempts, errors, steps, screenshots, and video, with each step linked to its timestamp in the video; agent steps show their explanation, actions taken, model, token count, cost, and whether the step replayed from cache. A failed assertion shows expected and received values with the diff marked, and when the test file is in the project the report shows the code around the failing call with each step linked to its source; a `copy prompt` button on a failed test copies its failure message, steps, and file paths for pasting into a coding agent.

— Exact config call, file path and media keys, no live example

`@e2e-dev/html-reporter@0.0.1`

---

04 **Relevance-scored test ordering for changed-since runs** NEW 90

decisionFilter() now attaches a `relevance` score (0–1) to each kept test, so `--changed-since` runs the likeliest-to-fail tests first. A custom filter can surface partial relevance on failure by attaching `relevance` to the error it throws, so the run still warns and runs every test with touched ones first. Tests are ordered in bands at 75%, 50%, and 25% relevance within each target, with targets ordered by their best file's band; ties break by config order, and within a band the file the previous run spent longest in starts first. decisionFilter() requires e2e 0.23.1 or newer; an older runner refuses the new `relevance` key and drops the whole filter.

Run only the tests most likely affected by recent changes, with the highest-relevance files starting first to surface failures early.

```
$ npx e2e --changed-since main --max-failures 1
```

— Mechanism, bands and version requirement fully specified `@e2e-dev/decision@0.3.3`

05 **Scroll-exploring assertions with inconclusive and policy controls**

IMPROVED 90

`agent.assert` gains opt-in scroll-exploring behaviour with bounded action and model-call budgets. Pass `inconclusiveOnOverflow: true` to `createToolLoopExecutor` to return `ASSERTION_INCONCLUSIVE` instead of shortening history on context overflow; call createVerdictTool(kind) with an act kind to drop `ASSERTION_INCONCLUSIVE` from act verdicts. A read-only assertion policy is enforced so hover and mutating actions from custom executors fail with `POLICY_DENIED` during any assertion. `agent.assert`, `agent.waitFor`, and `agent.extract` now see the current URL location so URL-based instructions can hold or fail instead of ending inconclusive, with `agent.waitFor` re-judging on location-only changes except under `vision: 'only'`.

Fail a context overflow as `ASSERTION_INCONCLUSIVE` rather than silently truncating the agent's history during an assertion step.

typescript
```
import { createToolLoopExecutor } from '@e2e-dev/acp';

const executor = createToolLoopExecutor({ inconclusiveOnOverflow: true });
```

Restrict a verdict tool to act-only verdicts, excluding `ASSERTION_INCONCLUSIVE` from the options it can return.

typescript
```
import { createVerdictTool } from '@e2e-dev/acp';

const verdictTool = createVerdictTool('act');
```

— Named functions, error codes, and runnable snippets given `e2e@0.23.1` `@e2e-dev/acp@0.1.2`

06 **Multi-tab control in browser fixture** NEW 85

The `browser` fixture now drives tabs: a tab opened by the app joins browser.tabs() in the background, browser.waitForPopup(trigger) runs a trigger and makes the new tab

active, and browser.newTab() opens a tab in the same context; each `BrowserTab` exposes `url`, `title`, `isActive`, `isClosed`, `activate`, and `close`. The agent's `switch_tab` and `close_tab` verbs move between tabs and close them.

Handle an OAuth popup in a test — wait for the popup tab, complete the flow, then assert back in the original tab.

```typescript
test('OAuth login opens a popup', async ({ app, browser, screen
}) => {
  await app.open('/login');
  const popup = await browser.waitForPopup(() =>
    screen.getByRole('button', { name: 'Sign in with Google'
}).click()
  );
  // popup is now the active tab; original tab waits in
background
  await popup.activate();
  // ... complete OAuth flow ...
  await popup.close();
});
```

— Full API surface plus runnable popup-handling example `@e2e-dev/web@0.16.0`

07 **Native PNG codec for screenshots** BREAKING 80

PNG decoding and encoding now run on `@napi-rs/image` (a native addon with a prebuilt binary per platform), making decoding 8–15x faster, encoding 5–8x, and `toHaveScreenshot` captures 4–12x faster; a missing binding fails with `IMAGE_CODEC_UNAVAILABLE`. Gray PNGs with 1, 2, or 4 bits per sample now decode, and 16-bit gray or RGB PNGs with a transparent color now compare pixels at 16 bits. Screenshots written by the runner (new baselines or rewritten ones) are stored in different bytes than before: a Chromium capture is about 15% larger and a device screenshot about 20% smaller. The addon adds 15–21 MB to an install and has no prebuilt binary for Linux on ppc64, s390x, or riscv64; on those platforms any call that reads or writes a screenshot throws `IMAGE_CODEC_UNAVAILABLE`.

— Numbers and error name given but no command to act on `@e2e-dev/decision@0.3.3` `@e2e-dev/mobile@0.12.7`

08 **Run e2e under Bun without Node.js** NEW 80

Run `e2e` without Node.js using Bun 1.4.3 or newer via `bunx --bun e2e run`; `e2e init` registers the MCP server with `bunx` in a Bun project.

Run the full test suite on Bun without installing Node.js — useful in Bun-native CI environments.

```
$ bunx --bun e2e run
```

— Runnable command and version requirement given `e2e@0.23.1`

09 **Configless MCP session startup** NEW 75

Open an MCP session with no config file by passing `engine` and `app` to `open_session`, where `engine` is a package name or a factory descriptor `{package, export, options}` and nested options can reference other engines the same way.

Start an MCP session against an iOS app without a config file on disk, specifying the engine and device provider inline.

```
$ open_session {
  engine: {
    package: "@e2e-dev/mobile",
    options: {
      platform: "ios",
      device: { package: "@e2e-dev/limrun", export: "limrun" }
    }
  },
  app: { url: "https://example.com/MyApp.app" }
}
```

— Config shape explained with a runnable session example `@e2e-dev/mobile@0.12.7` `@e2e-dev/web@0.16.0`

10 **Raw driver session access for custom tools** NEW 75

Access the raw driver session from custom tools via unstable_deviceSession(...engines) from `@e2e-dev/mobile/tools`, sharing `DeviceSession` with `device.unstable_raw`; the client is typed as `DeviceSessionClient`, exported from `@e2e-dev/mobile`.

Access the raw driver session in a custom tool to call a capability e2e does not yet wrap.

```typescript
import { unstable_deviceSession } from '@e2e-dev/mobile/tools';
import type { DeviceSessionClient } from '@e2e-dev/mobile';

export const myTool = async (...engines: Parameters<typeof
unstable_deviceSession>) => {
  const session: DeviceSessionClient = await
unstable_deviceSession(...engines);
  // call raw driver methods here
};
```

— Exact import path and runnable tool stub given `@e2e-dev/mobile@0.12.7`

11 **Skip post-gesture stabilization hold** NEW 70

Send `postGestureStabilization: false` with test-owned swipe and scroll to skip agent-device's post-gesture hold (up to 3.5 s); the first action on a control within 2 s of such a gesture re-reads it until two reads agree.

— Named flag and timing mechanism, no runnable snippet `@e2e-dev/mobile@0.12.7`

12 **Sandboxed and hosted browser/device providers** NEW 65

`@e2e-dev/e2b` adds e2b() for running Android 16 tests in isolated E2B sandboxes, installing the target `.apk` into one sandbox per worker slot, and e2bBrowsers() from `@e2e-dev/e2b/web` for starting a Chromium instance per worker slot or per attempt in E2B sandboxes. `@e2e-dev/browser-run` adds a browser provider supplying hosted Chromium leases via Cloudflare Browser Run for the e2e web engine.

— Names packages and functions but no runnable setup example `@e2e-dev/browser-run@0.1.0` `@e2e-dev/e2b@0.0.1`

13 **Soft keyboard occlusion handling in mobile queries** IMPROVED 65

Nodes hidden under the soft keyboard are now excluded from `screen.scrollUntilVisible`, toBeVisible(), and `getByRole`; a blocked tap reports `ACTION_FAILED` with the name of the covering element instead of an engine failure.

— Behavior and error code named, no usage example `@e2e-dev/mobile@0.12.7`

14 **INVALID_CONFIG validation for device and timeout settings** IMPROVED 65

mobile({ device: null }) now fails with `INVALID_CONFIG` instead of a raw `TypeError`. Any config timeout above 2147483647 ms is rejected with `INVALID_CONFIG`, covering `timeout`, `launchTimeout`, `actionTimeout`, `assertionTimeout`,

`cleanupTimeout`, `app.command.startupTimeout`, `shutdownTimeout`, and `agents.<name>.judgmentTimeout`.

— Every affected config key named, no runnable example `e2e@0.23.1` `@e2e-dev/mobile@0.12.7`

15 **JUnit reporter screenshot attachments for GitLab CI** NEW

60

The `junit` reporter now links failure screenshots in GitLab CI using the `[[ATTACHMENT|<path>]]` format in `<system-out>`, so screenshots appear directly in pipeline test reports.

— Exact format string named, no setup snippet `e2e@0.23.1`

16 **App command must stay in foreground** BREAKING

60

An app command that exits with code 0 after readiness (e.g. `docker compose up -d --wait`) now fails the `e2e run`; keep the app in the foreground (e.g. `docker compose up` without `-d`) instead.

— Clear before/after example of required config change `e2e@0.23.1`

THINNER COVERAGE BELOW

17 **Per-step device prepare timing logs** NEW

50

Mobile now logs device prepare timing per step — lines like `iPhone 17 Pro: boot 8.1s, iOS runner 120.3s, open Settings 3.2s` — including when a step fails or times out.

— Sample log line given but no config or usage path `@e2e-dev/mobile@0.12.7`

18 **Automatic Codex MCP registration via e2e init** NEW

40

`e2e init` now registers the MCP server in Codex project settings automatically, preserving existing settings and comments.

— One-line addition with no further mechanism detail `e2e@0.23.1`

└──▷ **BREAKING ON UPGRADE**

! web({ headers }) now sends headers only to the app's own origin (scheme + host + port of `app.url`), not to every host on the site. Any other origin that needs headers must be listed in `credentialOrigins`.

! web({ basicAuth }) now answers a 401 only from the app's own origin, not from any third-party origin the app embeds. Add other origins that need the credential to `credentialOrigins`.

! An app whose `app.url` redirects to a different origin (e.g. `http:` to `https:`, apex to `www`) must either list the destination origin in `credentialOrigins` or update `app.url` to point to

it directly.

! decisionFilter() requires e2e 0.23.1 or newer; an older runner refuses the new `relevance` key and drops the whole filter.

! Screenshots written by the runner (new baselines or rewritten ones) are stored in different bytes than before: a Chromium capture is about 15% larger and a device screenshot about 20% smaller.

! The `@napi-rs/image` native addon adds 15–21 MB to an install and has no prebuilt binary for Linux on ppc64, s390x, or riscv64; on those platforms any call that reads or writes a screenshot will throw `IMAGE_CODEC_UNAVAILABLE`.

! An app command that exits with code 0 after readiness (e.g. `docker compose up -d --wait`) now fails the `e2e run`; keep the app in the foreground (e.g. `docker compose up` without `-d`) instead.

WAS THIS USEFUL? 👍 👎

ai-git-bot

Feedback ⓘ 1.31.0 NOTES ↗ RANK

Ai-git-bot is the self-hosted AI workflow automation platform for Git repositories.

> ai-git-bot 1.31.0 improves its pr-diff agent tool with line numbers and multi-file queries, adds native Yarn support, and changes commit authorship to match the integration token's account.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Enhanced pr-diff agent tool** IMPROVED 62

The `pr-diff` agent tool now includes line numbers in its output and accepts up to 8 file paths per call, making diff queries more precise and batch-friendly.

— Names exact limit and tool but no usage example 1.31.0

THINNER COVERAGE BELOW

02 **Commits authored as integration account** IMPROVED 35

Commits are now authored as the account the integration token authenticates as, rather than a generic identity.

— Explains behaviour change but no config or flag named 1.31.0

03 **Native Yarn package manager support** NEW 25

ai-git-bot now supports Yarn natively as a package manager.

— Bare statement, no mechanism or config details given 1.31.0

WAS THIS USEFUL? 👍 👎

octo-agent

Feedback ⓘ v1.17.24 NOTES ↗ RANK

Octo-agent is an open-source, single-binary, self-hosted AI agent — your models and data stay on your machine. A coding agent on par with Claude Code and a personal assistant lighter than OpenClaw, across eight interfaces (TUI, CLI, Web, desktop, IM, VS Code, Obsidian, Mobile), all in one MIT-licensed Go binary running on any model.

> Octo-agent's web UI now surfaces timing information, adding dates on older messages and timestamps on every reply.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Timestamps in web UI chat history** IMPROVED 35

The web UI now shows the date on older messages and a timestamp on every reply, making it easier to track when messages were sent.

— Describes the change but no UI path or config detail v1.17.24

WAS THIS USEFUL? 👍 👎

zot

Feedback ⓘ v0.4.77 NOTES ↗ RANK

zot manages coding agent workflows from the command line.

> zot adds xAI as a supported model provider, with model discovery and authentication support for current Grok releases.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **xAI provider support** NEW 60

zot now supports the xAI provider, including model discovery and support for current Grok releases, authenticated via the `XAI_API_KEY` environment variable.

— Names env var and provider but no commands or discovery details v0.4.77

WAS THIS USEFUL? 👍 👎

Alibaba OpenCodeReview

Feedback ⓘ v1.13.9 NOTES ↗

RANK

OpenCodeReview is a secure, fast, efficient, battle-tested at Alibaba's scale. Hybrid architecture code review tool: deterministic pipelines + LLM Agent, precise line-level comments, built-in multi-language ruleset (NPE, thread-safety, XSS, SQL injection), OpenAI & Anthropic compatible.

> OpenCodeReview's run coverage manifest now records which files were excluded from a review run.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Excluded files tracked in run coverage manifest** IMPROVED

35

The run coverage manifest now records excluded files, making it visible which files were skipped during a review run.

— Describes what changed but no manifest field names or access method v1.13.9

WAS THIS USEFUL? 👍 👎

◆ Model Training & Fine-Tuning

fizgig

Feedback ⓘ v9.7.1 NOTES ↗ RANK

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's v9.7.1 release reworks the training UI around per-job views with safer run handling — resumable interrupted runs, duplicate-name protection, and starred checkpoints — alongside a reworked Likeness/Style scoring system and a MiniMax H3 adapter upgrade.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 Per-job tab switching in training UI NEW

65

Each job's training folder, captions, masks, sample prompts, settings, and previews now live in their own view, switchable from a pill in the top right (bottom right on phone) so you can move between the active run, queued runs, and a new setup without losing place. The pill's list on the Training tab also lets you set up a new run by copying the current job into a new setup with the same training folder, named with `_2` (or the next free number) after the original name.

— Names UI location and naming convention but no screenshot v9.7.1

02 Interrupted-run recovery on restart NEW

65

Fizgig now detects a run cut off mid-training and, on the next start, asks whether to carry on from the last saved state, start the next queued run, or clear the queue and start fresh — before anything else runs. A line above Start appears when the Resume Training box holds a saved state, naming which state Start would continue from, with a button to clear it.

— Explains the three-option recovery flow and UI cue v9.7.1

03 Per-run, per-folder Likeness and Style scoring BREAKING

65

Likeness and Style scores are now stored per run and per training folder, so reference photos are picked once per training folder and reused across every run on those pictures, and two runs sharing a samples folder never mix or overwrite each other's scores. Scores recorded before v9.7.1 are rescored the first time you open a run, and you must pick the three reference photos once more per training folder.

— Explains mechanism and migration impact on upgrade v9.7.1

04 Duplicate LoRA name guards NEW

60

Start, Add to queue, and Check settings now refuse a LoRA name that already has files in the same output folder and tell you which name is free, preventing two runs from overwriting each other's files.

— Names the three actions guarded and the mechanism v9.7.1

THINNER COVERAGE BELOW

05 **VRAM-free app window option** NEW 55

Add a Draw the window without the graphics card option in Preferences > App window so the window holds none of the VRAM a tight run needs.

— Names exact preference and location to enable it v9.7.1

06 **Starred preview checkpoints** NEW 50

You can star any preview in the Samples gallery, the grid, or a run's card to mark its epoch's checkpoint with a star in the checkpoint lists, making it easy to find the right LoRA later.

— Clear UI path but no deeper mechanism detail v9.7.1

07 **Model download progress indicators** IMPROVED 45

Fizgig now shows download progress for model files with speed and time remaining beside the size, and logs each filename as it starts.

— Names what's shown but no UI path v9.7.1

08 **MiniMax H3 training adapters v3** IMPROVED 45

MiniMax H3 now uses Ostris's training adapters v3; the updater migrates Preferences from v1 or v2 and leaves the old file on disk.

— Names version and migration behavior, no further detail v9.7.1

09 **Resizable caption boxes in Captions tab** IMPROVED 35

Caption boxes in the Captions tab now grow to fit their text in cards, the table, and the big view; on a wide window the big view puts the caption beside the picture.

— Describes UI change only, no deeper mechanism v9.7.1

10 **Epoch step countdown on run card** IMPROVED 35

Fizgig now shows steps remaining in the current epoch on the run card, beside the step counter.

— Bare description of a small UI addition v9.7.1

└──▷ **BREAKING ON UPGRADE**

! Likeness and Style scores recorded before v9.7.1 are rescored the first time you open a run, and you must pick the three reference photos once more per training folder.

WAS THIS USEFUL?  

Hugging Face TRL

Feedback ⓘ v1.18.0 NOTES ↗ RANK

TRL trains transformer language models with reinforcement learning, covering supervised fine-tuning, reward modeling, PPO, DPO, and GRPO.

> TRL's headline change is graduating AsyncGRPOTrainer to a stable, top-level trainer that decouples generation from training via a vLLM server, paired with a large overhaul of AsyncGRPOConfig's timeout and concurrency controls, LoRA adapter weight sync, customizable rollout loops, agent training via Harbor/OpenEnv, new training metrics, VLM fine-tuning, and new AMD ROCm and AWS Trainium backend support.

└──▷ **WHAT SHIPPED · 9 FEATURES** most completely described first ? what's the number?

01 AsyncGRPOTrainer graduates to stable API NEW 95

`AsyncGRPOTrainer` and `AsyncGRPOConfig` are now importable directly from `trl`, decoupling generation from training by streaming rollouts from a vLLM server while the trainer updates the model concurrently. The old `trl.experimental.async_grpo` import path (`async_grpo_config`, `async_rollout_worker`, `packing`, `vllm_client`, `weight_transfer`) still works but is deprecated with a `FutureWarning` until v2.0.0, and the vLLM client is renamed `AsyncVLLMClient`, now at `trl.generation.async_vllm_client`.

Run async GRPO training with generation on one GPU and training on another, overlapping the two for higher throughput.

```
$ # Start the vLLM server on GPU 0
CUDA_VISIBLE_DEVICES=0 VLLM_SERVER_DEV_MODE=1 vllm serve
Qwen/Qwen3.5-2B \
    --weight-transfer-config '{"backend": "nccl"}' \
    --logprobs-mode processed_logprobs \
    --api-server-count 8 \
    --max-model-len 4096

# Train on GPU 1
CUDA_VISIBLE_DEVICES=1 python train_async_grpo.py
```

— Clear mechanism, import paths, and a runnable command example. v1.18.0

02 Customizable async rollout loop NEW 95

Subclass `AsyncRolloutLoop` (from `trl.trainer.async_grpo_trainer`) and override `filter_prompt`, `generate_one`, `generate_one_turn`, `compute_advantages`, `setup`, or `close` to customize rollout behavior, then set the

subclass as `loop_cls` on an `AsyncGRPOTrainer` subclass. This replaces the removed `rollout_worker=` constructor argument.

Override only the rollout generation step (for example, to add custom filtering) while keeping the rest of the async training loop intact.

```python
from trl import AsyncGRPOTrainer
from trl.trainer.async_grpo_trainer import AsyncRolloutLoop,
Rollout

class MyRolloutLoop(AsyncRolloutLoop):
    async def filter_prompt(self, prompt):
        # custom prompt-level filtering logic
        return await super().filter_prompt(prompt)

class MyAsyncGRPOTrainer(AsyncGRPOTrainer):
    loop_cls = MyRolloutLoop

trainer = MyAsyncGRPOTrainer(
    model="Qwen/Qwen3.5-2B",
    reward_funcs=my_reward_fn,
    train_dataset=dataset,
)
trainer.train()
```

— Exact classes, hook methods, and a working code example. v1.18.0

03

AsyncGRPOConfig concurrency and timeout overhaul

BREAKING

95

`AsyncGRPOConfig` now controls generation staleness with `max_staleness` (in tokens) and a fixed concurrent rollout ceiling with `max_concurrent_rollouts` (default 512, renamed from `max_inflight_tasks` and no longer derived from `per_device_train_batch_size`); `heartbeat_stale_after_s` is renamed `rollout_worker_timeout` (still 300 s default); and all vLLM timeouts are merged into `vllm_server_timeout` (default 1800 s, replacing the former separate `request_timeout` of 240 s and `weight_sync_timeout` of 600 s). `queue_maxsize`, `fork_threshold_tokens`, and non-positive `token_budget` (fixed-count batching) are removed — multi-turn rollouts always fork on re-tokenization drift and the token-budget batcher is the only mode — `repetition_penalty` is removed because it biased the importance ratio, and `remove_unused_columns` now defaults to `False` (`True` now raises an error since every dataset column is forwarded to reward functions).

— Enumerates every renamed and removed config key with defaults. v1.18.0

04 **LoRA adapter weight sync for vLLM** NEW 80

Enable LoRA adapter sync by starting vLLM with `--enable-lora`; each weight sync then sends only the adapter (~160 MB at `r=32`) instead of merged weights. The `weight_transfer=` constructor argument on `AsyncGRPOTrainer` is removed — weight sync (NCCL or LoRA adapter sync) is now configured entirely through `AsyncGRPOConfig`.

— Exact flag and transfer size given, but no full command. v1.18.0

05 **Score-centering objective and top_k sampling** NEW 75

`AsyncGRPOConfig` defaults to score centering with masked importance sampling via `score_centering=True` (set `score_centering=False` for GRPO's clipped objective), and `top_k` can now be used alongside `top_p` or `min_p` to renormalize trainer log-probs over vLLM's sampled candidate set.

— Named config flags but no usage example given. v1.18.0

06 **Agent training via HarnessAsyncGRPOTrainer** NEW 70

Use `HarnessAsyncGRPOTrainer` in `trl.experimental.async_grpo.openenv_harness` for OpenEnv / Harbor agent training instead of manually constructing a `HarnessRolloutWorker`. The Harbor example's `--max-inflight` flag is renamed to `--max-concurrent-rollouts`, and the OpenEnv / Harbor environment server's `MAX_CONCURRENT_ENVS` must now be at least `max_concurrent_rollouts + 1`.

— Named classes and flags but no full command shown. v1.18.0

07 **New async GRPO training metrics** NEW 65

Zero-advantage group filtering is tracked via the new `frac_zero_std_groups` metric (groups whose rewards are all equal are dropped before the forward and backward pass), alongside new async-only metrics: `num_tokens`, `completions/{mean,min,max}_length`, `completions/{mean,min,max}_terminated_length`, `completions/clipped_ratio`, `reward_zero_std_groups`, `completions/duplicate_frac`, and `sampling/importance_sampling_ratio/{mean,min,max}`.

— All metric names listed but no usage guidance. v1.18.0

THINNER COVERAGE BELOW

08 **VLM training on text-only data** NEW 45

Train VLM checkpoints (Qwen3.5, Qwen3.6, Qwen3-VL) on text-only data with the vision tower frozen via `AsyncGRPOTrainer`.

— States capability only, no configuration detail given. v1.18.0

09 **AMD ROCm and AWS Trainium backend support** NEW 40

TRL's trainers now run on AMD Instinct GPUs through PyTorch's ROCm build, and on AWS Trainium through TorchNeuron, PyTorch's native Neuron backend.

— Names backends but gives no setup instructions. v1.18.0

└──▷ BREAKING ON UPGRADE

! The `trl.experimental.async_grpo` submodule layout is gone: `from trl.experimental.async_grpo.async_grpo_config import AsyncGRPOConfig` and imports from `async_rollout_worker`, `packing`, `vllm_client`, or `weight_transfer` in that package no longer work. Import `AsyncGRPOTrainer` and `AsyncGRPOConfig` from `trl`, `AsyncRolloutLoop` and `Rollout` from `trl.trainer.async_grpo_trainer`, and `AsyncVLLMClient` (renamed from the old vLLM client) from `trl.generation.async_vllm_client`.

! The `weight_transfer=` constructor argument on `AsyncGRPOTrainer` is removed; weight sync is now configured entirely through the config (NCCL or LoRA adapter sync).

! The `rollout_worker=` constructor argument on `AsyncGRPOTrainer` is removed; to change rollout behavior, subclass `AsyncRolloutLoop` and pass it as `loop_cls`.

! `AsyncGRPOConfig.max_inflight_tasks` is renamed to `max_concurrent_rollouts` (fixed ceiling, default 512, no longer derived from `per_device_train_batch_size`).

! `AsyncGRPOConfig.heartbeat_stale_after_s` is renamed to `rollout_worker_timeout` (same 300 s default).

! `AsyncGRPOConfig.request_timeout` and `weight_sync_timeout` are removed; all vLLM timeouts are now covered by `vllm_server_timeout` (default 1800 s, was 240 s for readiness and 600 s for requests).

! `AsyncGRPOConfig.queue_maxsize` is removed; generation backpressure is now controlled by `max_staleness` in tokens.

! `AsyncGRPOConfig.fork_threshold_tokens` is removed; multi-turn rollouts always fork on re-tokenization drift.

! `AsyncGRPOConfig.token_budget <= 0` (fixed-count batching) is removed; the token-budget batcher is the only mode. Use `None` (reads `max_model_len` from the server) or a positive integer.

! `AsyncGRPOConfig.repetition_penalty` is removed because the sampler's log-probs include it while the trainer's do not, biasing the importance ratio.

! `AsyncGRPOConfig.remove_unused_columns` now defaults to `False`; setting it to `True` raises an error because every dataset column is forwarded to the reward functions.

! The Harbor example's `--max-inflight` flag is renamed to `--max-concurrent-rollouts`; the OpenEnv / Harbor environment server's `MAX_CONCURRENT_ENVS` must be at least `max_concurrent_rollouts + 1`.

WAS THIS USEFUL? 👍 👎

ModelScope MS-SWIFT

Feedback ⓘ v4.6.1 NOTES ↗ RANK

MS-SWIFT provides training, fine-tuning, reinforcement learning, evaluation, and deployment workflows for language and multimodal models.

> MS-SWIFT v4.6.1 adds Entropack-based flexible QLoRA quantization and expands decision-model training to new model families, alongside Megatron preflight validation, Ascend NPU environment checks, and security hardening of checkpoint transport and template media loading; several breaking changes affect MTP defaults, dependency minimums, and plugin restoration on resume.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 Explicit control over MTP layer loading and defaults

BREAKING

95

`--mtp_num_layers` now defaults to `auto` (follows checkpoint) and `--mtp_shared_weights` defaults to automatic detection; add MTP to a base checkpoint with `--mtp_num_layers N --mtp_allow_random_init true` (requires mcore-bridge >= 1.7.0). This is a breaking change: checkpoints that already ship MTP layers will now train them on upgrade unless you pass `--mtp_num_layers 0 --mtp_shared_weights false`; a checkpoint declaring repeated MTP weights will resolve `mtp_shared_weights=True`, which requires `mtp_num_layers > 0`.

Add MTP heads to a base checkpoint that ships none, using random initialisation for the new layers.

```
$ swift sft --model <model_id_or_path> --mtp_num_layers 2 --
mtp_allow_random_init true
```

— Exact flags, defaults, migration command, and requirement given v4.6.1

02 Ascend NPU environment check and Megatron-Core 0.18 support

NEW

78

Audit Huawei Ascend NPU environments before training with the new standalone script `examples/ascend/check_env.py`; Megatron training on Ascend now targets Megatron-Core 0.18 via `MegatronAdaptor` + TENPU.

— Names script path and backend version target v4.6.1

03 Entropack flexible quantization for QLoRA NEW

75

Enables flexible QLoRA-style bitrate quantization via Entropack, with a best-practice guide at `docs/source_en/BestPractices/Flexble-Quantization-With-`

`Entropack.md` and a ready-to-run recipe at `examples/train/qlora/entropack.sh`.

Run QLoRA fine-tuning with Entropack flexible quantization to stay within a VRAM budget on a 7B model.

```
$ bash examples/train/qlora/entropack.sh
```

— Names doc and script paths; limited mechanism detail v4.6.1

---

04 **Decision model training extended to new model families**

IMPROVED 73

Train and infer decision models (typed-decision System-1 scoring) using `--task_type decision`, now extended with four new model families — THX-02, `jaredpalmer/kev-*`, Laya, and StartLux — plus shell recipes under `examples/train/decision/*.sh`.

— Names flag and model families but no example command v4.6.1

---

05 **Path-traversal protection for template media loading** IMPROVED

73

Block path-traversal attacks on template media loading: `load_file` now applies `SafeMediaPath` before opening any local path, so `SWIFT_MEDIA_ALLOWED_DIRS` can no longer be bypassed through image or audio template loading.

— Names function and env var fixed; no user action required v4.6.1

---

06 **Minimum dependency version requirements raised** BREAKING

65

Minimum `megatron-core` raised from `>= 0.16` to `>= 0.18`; `deepspeed` raised to `>= 0.19.3,< 0.20` (was `< 0.19`); `transformers` upper bound raised to `< 5.23` (was `< 5.20`); and `mcore-bridge` raised from `>= 1.6.5` to `>= 1.7.0`.

— Exact version ranges given but no migration mechanism beyond upgrading v4.6.1

---

07 **external_plugins no longer auto-restored on resume** BREAKING

65

`external_plugins` listed in a checkpoint's `args.json` is no longer restored on resume; pass `--external_plugins` explicitly on the command line if you depend on it.

— Names flag and file but limited rationale given v4.6.1

THINNER COVERAGE BELOW

08 **Megatron parallelism preflight validation** NEW 50

Catch Megatron TP/PP/CP/EP/ETP divisibility mismatches at argument entry before process-group initialization, with actionable error messages.

— Describes behavior but no flag or command to invoke v4.6.1

09 **HMAC authentication for checkpoint ZMQ channel** IMPROVED 50

Harden the checkpoint ZMQ control channel with HMAC authentication, covering both HCCL and NCCL transports.

— States mechanism but no config key or flag to enable v4.6.1

10 **Support for eight new models** NEW 50

MS-SWIFT now supports eight new models: NVIDIA-Nemotron-3.5-Super-VL-120B-A12B, NVIDIA-Nemotron-3.5-Transcribe-30B-A4B, Unlimited-OCR-2, MiniCPM-OCR-1.0, Olmo-3-7B-Think, flint, and EmbeddingGemma-2 (multimodal embedding, text/image/video/audio into a 768-d vector).

— Lists model names but no usage detail beyond naming v4.6.1

11 **Packing and padding_free for Gemma 4 on Megatron backend** NEW 45

Support packing and `padding_free` for Gemma 4 text-only data on the Megatron backend.

— Names flag and backend but no example v4.6.1

└──▷ BREAKING ON UPGRADE

! `--mtp_num_layers` now defaults to `auto` instead of `None`: checkpoints that ship MTP layers will train them on upgrade. To restore v4.6.0 behaviour, pass `--mtp_num_layers 0 --mtp_shared_weights false` together.

! `--mtp_shared_weights` now defaults to automatic detection (`None`) instead of `False`: a checkpoint declaring repeated MTP weights will resolve `mtp_shared_weights=True`, which requires `mtp_num_layers > 0`.

! `external_plugins` listed in a checkpoint's `args.json` is no longer restored on resume; pass `--external_plugins` explicitly on the command line if you depend on it.

! Minimum `megatron-core` raised from `>= 0.16` to `>= 0.18`.

! Minimum `deepspeed` raised to `>= 0.19.3,< 0.20` (was `< 0.19`).

! `transformers` upper bound raised to `< 5.23` (was `< 5.20`).

! Minimum `mcore-bridge` raised from `>= 1.6.5` to `>= 1.7.0`.

WAS THIS USEFUL? 👍 👎

◆ Speech & Voice AI

LiveKit Agents

Feedback ⓘ livekit-agents@1.8.12 NOTES ↗

RANK

LiveKit Agents builds realtime voice and video AI agents on LiveKit's WebRTC infrastructure, with turn detection, interruption handling, and telephony.

> LiveKit Agents adds delegation recovery for GPT-Live, brings in a wave of new STT/TTS/LLM provider support, updates the local turn-detector model, and tightens NVIDIA TTS encoding requirements.

└──▷ WHAT SHIPPED · 4 FEATURES most completely described first ? what's the number?

01 **Delegation recovery in GPT-Live** NEW 87

Pass `delegation_recovery=True` to `GPTLiveModel` to recover caller turns that were transcribed but never delegated, which starts backend delegation after a quiet grace period — useful when callers speak over the agent and turns get dropped. As a side effect, `delegation_created` listeners now also receive backend delegations, so code that previously assumed only response-mode delegations must filter on `mode="responses"`.

Re-engage a caller whose answer GPT-Live transcribed but never delegated — useful when users speak over the agent and turns get dropped.

python

```python
from livekit.plugins import openai

model = openai.GPTLiveModel(delegation_recovery=True)
```

— Names exact flag and class with a runnable example and behavior change

livekit-agents@1.8.12

02 **New STT/TTS/LLM provider and model support** NEW 63

LiveKit Agents adds support for Lokutor (Conv 2.0 STT and Versa 2.0 TTS across nine languages), xAI's `xai/stt-2` speech-to-text model through Inference with word alignment for adaptive interruption, Google's Gemini 4 Carbon, Gemini 4 Hydrogen (also through Inference), and `gemini-decision-1` through the Decisions API, Perplexity Flash and standard Decider v1.3, v1.2, and latest aliases, AssemblyAI's `assemblyai-stt-3-7-realtime-agent` for streaming STT, and Baseten's `streaming-v2` transcription protocol.

— Lists many named models/providers but no setup mechanism shown livekit-agents@1.8.12

THINNER COVERAGE BELOW

03 **NVIDIA TTS encoding restricted to LINEAR_PCM** BREAKING

45

NVIDIA TTS output is now restricted to `LINEAR_PCM`; other encoding configurations will need to be updated.

– Names the breaking constraint but no migration steps given `livekit-agents@1.8.12`

04 **Turn-detector model updated to v1-mini** IMPROVED

35

The local turn-detector model is updated to v1-mini, which adds turn-length input support.

– Names new model version but no usage detail beyond that `livekit-agents@1.8.12`

└──▷ **BREAKING ON UPGRADE**

! `delegation_created` listeners in GPT-Live now also receive backend delegations; filter on `mode="responses"` where you previously assumed only response-mode delegations.

! NVIDIA TTS output is restricted to `LINEAR_PCM`; other encoding configurations will need to be updated.

WAS THIS USEFUL? 👍 👎

AGENT

◆ Web Search & Scraping for Agents

Perplexity API

Feedback ⓘ updated · Nov 14, 2026 NOTES ↗

RANK

Perplexity API provides language models with real-time web search and retrieval for building applications that research and answer questions.

> Perplexity's Agent API adds support for Google's Gemini 4 Hydrogen model while quietly dropping documentation references to three older Gemini Flash model IDs.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Gemini 4 Hydrogen model in Agent API** NEW 90

The Agent API now supports `google/gemini-4-hydrogen`, priced at $1.50 per million input tokens, $0.075 per million cached input tokens, and $7.50 per million output tokens (reasoning tokens included). It can be invoked via the standard `/chat/completions` endpoint by setting `"model": "google/gemini-4-hydrogen"`.

Call the Agent API with the new Gemini 4 Hydrogen model to run a reasoning-heavy agent task.

```
$ curl https://api.perplexity.ai/chat/completions \
  -H 'Authorization: Bearer <your_api_key>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "google/gemini-4-hydrogen", "messages": [{"role": "user", "content": "<your prompt>"}]}'
```

— Names model id, full pricing tiers, and runnable curl example. snapshot-20261114

02 **Gemini 3.6/3.7/3.8 Flash removed from Agent API docs** BREAKING 60

`google/gemini-3.6-flash`, `google/gemini-3.7-flash`, and `google/gemini-3.8-flash` are no longer listed in the Agent API model docs, pricing calculator, or examples; integrations referencing these model IDs should be updated.

— Names three removed model ids but gives no migration path. snapshot-20261114

└──▷ **BREAKING ON UPGRADE**

! `google/gemini-3.6-flash`, `google/gemini-3.7-flash`, and `google/gemini-3.8-flash` are no longer listed in the Agent API model docs, pricing calculator, or examples — any

integration referencing these model IDs should be updated.

WAS THIS USEFUL? 👍 👎

◆ AI Agent Frameworks

Arcade

Feedback ⓘ updated · Nov 14, 2026 NOTES ↗

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

Arcade hardens toolkit process isolation, adds a Snipe-IT hardware-asset toolkit, and expands Linear, Slack, Google Docs, Drive, and Forms toolkit capabilities.

Toolkit process isolation now prevents any toolkit from reading the Engine's secret or another toolkit's secret, signing its own tokens, or calling another toolkit's runtime directly — each toolkit authenticates with its own secret.

A gateway's resource list now shows which toolkit or server each resource came from in its display title.

Linear toolkit now supports `archived_at` to include archived issues, can skip descriptions to shrink large pages, and `GetIssue` can return recent state with names instead of IDs for pull-request or project references.

Linear toolkit adds a tool to list a project's or initiative's change history, and `ListProjects` / `ListInitiatives` rows now report health, owner, and key dates.

Slack toolkit's `ListMessages` now reports when each thread last received a reply and which users reacted to each message, and `SearchContent` accepts `*` to match every message in the window.

+2 more

NEW INTEGRATION Snipe-IT toolkit

└─▷ BREAKING ON UPGRADE

! Google Forms `CopyForm` and Google Slides `CreatePresentationFromTemplate` with no destination now place copies in the caller's My Drive instead of beside the source — automation that depended on the previous placement or inherited sharing will need review.

WAS THIS USEFUL? 👍 👎

claw-orchestrator

Feedback ⓘ v7.23.0 NOTES ↗ RANK

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 adds three Gemini 4 model registrations for Antigravity sessions, explicit permission-mode passthrough for Antigravity, and cache-aware turn-count pricing.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Gemini 4 model registrations for Antigravity** NEW 85

Registers `gemini-4-carbon`, `gemini-4-argon` and `gemini-4-hydrogen` for Antigravity sessions with Gemini API pricing and a 1M context window; the highest-effort `gemini-4-hydrogen` model is reached via `effort: 'xhigh'`.

Target Gemini 4's highest-effort model for a complex long-context Antigravity session.

yaml

```yaml
model: gemini-4-hydrogen
effort: xhigh
```

— Names exact models, pricing basis, context window and config trigger. v7.23.0

02 **Explicit permission-mode passthrough for Antigravity sessions**

IMPROVED 80

Antigravity sessions now pass agy permission modes explicitly: `default`, `manual` and `dontAsk` sessions use agy's Default mode which refuses any unpermitted tool call, while `acceptEdits`, `auto` and `plan` sessions pass the matching `--mode` flag.

— Names every mode and the flag mechanism but no runnable example given. v7.23.0

THINNER COVERAGE BELOW

03 **Cache-aware turn counting for Antigravity** IMPROVED 40

Antigravity turn counts now include cache reads, priced at each model's cached rate.

— States the behaviour change but no concrete numbers or usage path. v7.23.0

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.9.44 NOTES ↗ RANK

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

> HyperFrames shipped a debugging environment variable for tracing orphaned browser processes, a new safe-area warning in its `check` command for 9:16 social video, and live opaque-preview overlays for drag, resize, and layer operations in Studio.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **HYPERFRAMES_OWNER environment variable for process tracing**

NEW 70

Setting `HYPERFRAMES_OWNER` tags which run started a leftover browser process, making it easier to trace orphaned browsers back to their originating CLI invocation.

— Names exact env var and purpose, but no usage example v0.9.44

02 **Safe-area text warning in check command** NEW 60

HyperFrames' `check` command now warns when text elements fall outside the safe areas of a 9:16 social video format.

— Names command and format but no example of output v0.9.44

THINNER COVERAGE BELOW

03 **Opaque live preview for Studio editing operations** IMPROVED 45

HyperFrames Studio now draws drag and resize operations live over an opaque channel preview instead of a transparent one, including the layers panel and marquee selection.

— Describes the UI change but no navigation path or demo v0.9.42

WAS THIS USEFUL? 👍 👎

cotal

Feedback ⓘ v0.164.7 NOTES ↗ RANK

Cotal is the open standard for agent coordination

> Cotal's biggest change this window is owner-scoped publishing of tool inputs and results to the console and dashboard, alongside issuer-endorsed sign-in, automatic macOS background-service updates, a channel cleanup flag, and status warnings for mismatched extension versions.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 Tool payload publishing with owner-scoped visibility NEW

95

Cotal now publishes bounded tool inputs and results to permitted event readers and shows them in interactive console details and dashboard Work turns. Publishing can be suppressed while keeping tool names and times via `--no-event-payloads` on the command line or `eventPayloads: false` in a persona's config; payload visibility uses a separate owner-scoped rail, with owners and space admins able to read by default and other owners requiring an explicit event-channel grant.

Restrict payload publishing for a specific persona by setting `eventPayloads: false` in its config.

```yaml
eventPayloads: false
```

— Names exact flag, config key, and permission mechanism. v0.164.0

02 Channel cleanup with --purge NEW

75

Remove a channel's saved registry entry even when its chat stream has already been deleted, using `cotal channel remove --purge`, avoiding an unconfirmed error.

Clean up a channel whose broker stream has already been deleted, avoiding an unconfirmed error.

```
$ cotal channel remove --purge
```

— Exact command and the error state it fixes. v0.164.4

03 macOS background-service auto-update via launchd IMPROVED

70

The macOS background service now installs updates automatically when automatic updates are on, handing the updater to a separate one-shot launchd job so restarting the service does not interrupt installation or extension reconciliation. Users on a version before v0.164.6 need to run `cotal update --self` once to enable this behavior going forward.

Trigger the one-time self-update needed on a Mac running a version before v0.164.6 to enable fully automatic future updates via launchd.

```
$ cotal update --self
```

— Explains mechanism and one-time migration command. v0.164.6

04 **cotal status version-mismatch warning** IMPROVED 70

`cotal status` now shows a warning naming affected packages when first-party extension versions differ from the CLI, and recommends running `cotal update --self` to reconcile the installation.

— Names the command and the fix it recommends. v0.164.7

WAS THIS USEFUL? 👍 👎

CopilotKit

Feedback ⓘ v1.88.0 NOTES ↗ RANK

CopilotKit connects AI agents to application interfaces with shared state, tools, and generative UI components.

> CopilotKit v1.88.0 improves Learning data capture for Continual Learning and changes thread naming behavior in a way that may break existing logic on upgrade.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Richer Learning data capture** IMPROVED 20

CopilotKit now captures richer Learning data to improve Continual Learning from real use.

— No detail on what data or mechanism changed v1.88.0

02 **Thread naming behavior change** BREAKING 20

Thread naming behavior has changed in v1.88.0; existing thread-naming logic may break on upgrade. The release notes flag this as breaking but do not specify the old or new values.

— Breaking change noted but no specifics given v1.88.0

└──▷ **BREAKING ON UPGRADE**

! Thread naming behavior has changed in v1.88.0; existing thread-naming logic may break on upgrade (the release notes name this as a breaking change but do not detail the old or new values).

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.173 NOTES ↗ RANK

The Vercel AI SDK provides TypeScript APIs for model generation, structured output, tool use, and streaming application interfaces.

> Vercel AI SDK added approval gating and configurable timeouts to `WorkflowAgent`, plus a way to keep tool call errors out of the model's message history.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Structured timeout limits in WorkflowAgent** NEW 90

`WorkflowAgent` accepts a `timeout: { totalMs, stepMs, firstChunkMs, chunkMs }` object passed to generate() or stream() to cap total run time, per-step model-call duration, and streaming chunk waits; invalid timeout values are rejected with an `InvalidArgumentError`.

Cap total run time, each model-call step, and streaming chunk waits to avoid runaway `WorkflowAgent` executions.

typescript

```typescript
const result = await agent.stream({
  prompt: 'Summarise the data.',
  timeout: {
    totalMs: 60_000,
    stepMs: 15_000,
    firstChunkMs: 5_000,
    chunkMs: 3_000,
  },
});
```

— Names exact config fields and error behaviour with runnable example. ai@7.0.173

02 **Hide tool call errors from model history** NEW 80

Passing `redactToolErrors: true` together with an `onError` callback to `generateText` replaces tool call errors with a sanitized message before they reach the model's message history, keeping internal error detail out of context sent to the model.

Keep tool call errors hidden from the model's message history by replacing them with a sanitized message via `onError`.

typescript

```typescript
const result = await generateText({
  model: 'openai/gpt-6-astra',
  prompt: 'Run the analysis tool.',
  tools: { ... },
  redactToolErrors: true,
  onError: (error) => 'Tool encountered an error.',
});
```

— Flag and callback named with a runnable code example. ai@5.0.290

03 **Tool-call approval workflow in WorkflowAgent** NEW 70

`WorkflowAgent` supports a `toolApproval` configuration passed to the constructor or to a per-call generate()/stream() invocation to gate tool execution: approved calls run within the same step, while denied calls return an `execution-denied` result without pausing the workflow.

— Mechanism and config key named but no usage example given. ai@7.0.173

WAS THIS USEFUL? 👍 👎

AWS Strands Agents

Feedback ⓘ python/v1.65.0 NOTES ↗

RANK

Strands Agents is AWS's open-source SDK for production agents in Python and TypeScript, with a model-driven loop, tool integration, and deployment targets.

> Strands Agents added async multi-agent pipeline builders, an auto-decide mode for HumanInTheLoop, broad Anthropic/OpenAI/Bedrock model-provider improvements including pre-built vendor client support, and TypeScript-only context summarization and SageMaker-based decision routing — alongside breaking changes to conversation-history representation, offload granularity, and default harness plugins.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 SageMakerDecisionModel for decision routing NEW 60

A new `SageMakerDecisionModel` routes agent decisions through a System One model hosted on SageMaker AI instead of the default decision model, configured with an `endpointName`.

Route decisions through a System One model on SageMaker AI instead of the default decision model.

typescript

```typescript
import { SageMakerDecisionModel } from '@strands-agents/sdk'

const decisionModel = new SageMakerDecisionModel({ endpointName:
'<your-sagemaker-endpoint>' })
```

— Only known from a code example, no release-note text typescript/v1.25.0

THINNER COVERAGE BELOW

02 Async multi-agent pipeline builders NEW 45

New async agent builders let you build multi-agent pipelines asynchronously, and child agents that are content-filtered or refused are now reported as tool errors rather than failing silently.

— No code example or exact API name for the builders python/v1.65.0

└──▷ **BREAKING ON UPGRADE**

! Injected text is now written into conversation history as an `InjectedBlock` (Python and TypeScript) rather than plain text — existing code that reads or inspects history entries may encounter the new block type.

- ! The todos plugin is no longer included in the default built-in harness plugins; agents relying on it must add it back explicitly.
- ! Offload conditions in the TypeScript SDK now decide context-offload granularity rather than the previous behaviour — TypeScript context offload configuration may need review.
- ! Offload conditions now decide context granularity (TypeScript) — existing offload condition logic may behave differently after upgrade.
- ! Injected text is now written into history as an `InjectedBlock` (TypeScript) — code that reads raw injected strings from conversation history will need updating.
- ! The todos plugin is no longer included in the default built-in harness plugins — agents relying on it must add it explicitly.

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ `langchain-openai==1.8.0` NOTES ↗ RANK

LangChain is an open-source framework that orchestrates applications powered by language models.

> langchain-openai 1.8.0 adds support for OpenAI's Decisions API via a new classifier class, while raising its minimum Python and langchain-core requirements.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Raised minimum Python and langchain-core versions** BREAKING

65

Support for Python 3.10 is dropped; the minimum supported Python version is now 3.11, with users on Python 3.10 continuing to resolve `langchain-openai` 1.7.x. The minimum required `langchain-core` version is now 1.8.0.

— Exact version numbers given, migration path noted `langchain-openai==1.8.0`

THINNER COVERAGE BELOW

02 **OpenAIClassifier for OpenAI Decisions API** NEW

55

The new `OpenAIClassifier` class lets LangChain call OpenAI's now generally available Decisions API directly.

— Names the class and API but no usage example or parameters `langchain-openai==1.8.0`

└──▷ **BREAKING ON UPGRADE**

! Support for Python 3.10 is dropped; the minimum supported Python version is now 3.11. Users on Python 3.10 will continue to resolve `langchain-openai` 1.7.x.

! The minimum required `langchain-core` version is now 1.8.0.

WAS THIS USEFUL? 👍 👎

open-multi-agent

Feedback ⓘ v1.24.0 NOTES ↗ RANK

open-multi-agent is a self-hosted TypeScript agent runtime with durable approvals and verifiable run records.

> open-multi-agent ships an experimental image-generation tool call, adds an NVIDIA NIM provider, and lands several breaking changes to finish-reason handling, Claude sampling params, approval gates, and message read state.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 NVIDIA NIM provider NEW

70

Adds the `nvidia` provider for NVIDIA NIM models, defaulting to the hosted NIM endpoint and falling back to the `NVIDIA API key` environment variable, with base URL support for self-hosted deployments. Adding `nvidia` to the exported supported-provider union can require a new branch in exhaustive TypeScript switches over provider ids.

Point the nvidia provider at a self-hosted NIM deployment instead of the public endpoint.

```
$ npm i @open-multi-agent/core@1.24.0
```

— Names provider id and env var, example shows install only v1.24.0

02 Guardrail finish reasons mapped to content_filter BREAKING

70

Provider safety and guardrail finish reasons now map to `content_filter`, covering Bedrock `guardrail-intervened` and `content-filtered`, and Gemini `safety`, `recitation`, `blocklist`, `prohibited-content`, `SPII`, and `image-safety` finishes. Callers branching on stop reason must fold the old provider-specific strings into `content_filter` and re-check logic that assumed a tool-use turn cannot report it.

— Lists every mapped finish reason but no migration command v1.24.0

03 Claude models drop sampling parameters BREAKING

70

Claude Opus 4.7, Sonnet 5, and later models now omit `temperature`, `top_p`, and `top_k` on the Anthropic adapter and on Bedrock (foundation-model ids, inference profiles, and ARNs) since those models reject them; values passed in `extraBody` are still forwarded.

— Names exact params and models affected, no migration snippet v1.24.0

04 **Approval gate abort-signal behavior** BREAKING 60

Gate and approval code must now expect the gate not to be invoked when the abort signal is already aborted, an abort-error tool result instead of a decision when the signal fires mid-wait, and no approval record created for aborted calls; journal error handlers must also tolerate more than one callback per run.

— Describes behavior changes but no API names or code sample v1.24.0

THINNER COVERAGE BELOW

05 **Experimental image generation tool call** NEW 50

Adds `createImageTool`, which lets a model invoke `runImage` to generate and edit images as part of agent tool calls. The feature is marked experimental; callers depending on its options or result shape should pin an exact core version.

— Names the tool and function but gives no usage example v1.24.0

06 **Message read-state id handling** BREAKING 35

Consumers of message read state should expect ids that do not resolve to retained, addressed messages to be ignored; snapshots written by earlier versions still restore.

— Thin behavioral note with no named surface v1.24.0

07 **Gemini thinking token usage recalibration** BREAKING 30

Token budgets, cost-estimate assertions, and eval totals for Gemini thinking models must be recalibrated because output usage now includes thinking tokens.

— States the change but no numbers or mechanism v1.24.0

└──▷ BREAKING ON UPGRADE

- ! Adding `nvidia` to the exported supported-provider union can require a new branch in exhaustive TypeScript switches over provider ids.
- ! Callers branching on stop reason must fold the old provider-specific strings into `content_filter`, and should re-check logic that assumed a tool-use turn cannot report `content_filter`.
- ! Sampling values (`temperature`, `top_p`, `top_k`) set for affected Claude models are no longer sent to Anthropic or Bedrock; values passed in `extraBody` are still forwarded.
- ! Token budgets, cost-estimate assertions, and eval totals for Gemini thinking models must be recalibrated because output usage now includes thinking tokens.
- ! Gate and approval code must expect: the gate not to be invoked when the abort signal is already aborted, an abort-error tool result instead of a decision when the signal fires mid-wait, and no approval record for aborted calls; journal error handlers must tolerate more than one callback per run.

! Consumers of message read state should expect ids that do not resolve to retained, addressed messages to be ignored; snapshots written by earlier versions still restore.

WAS THIS USEFUL? 👍 👎

Mastra

Feedback ⓘ @mastra/core@1.83.0 NOTES ↗ RANK

Mastra is a TypeScript framework for AI applications and agents, providing workflows, agent memory, RAG, tool calling, and evals.

> Mastra's core package adds resumable agent streams with reconnect/cancel semantics, an expanded built-in MCP server with OAuth support, and delta-write workflow storage for Postgres and libSQL, alongside smaller additions to scheduling, memory chunking, observability sorting, agent context, skills, and Slack generative UI — plus a breaking change to how `durable` behaves on a plain `Agent`.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 Resumable agent streams and chat routes NEW 95

Every `Agent` now supports resumable streams so clients can reconnect after a disconnect, follow a live run with agent.observe(runId), and cancel it with observer.abort() or agent.abortRunStream(runId), with replay caching and TTL enforcement built in. chatRoute({ resumable: true }) builds resumable AI SDK chat routes that generate a POST + reconnect GET endpoint pair replaying active runs via agent.observe(runId), with route arrays configurable via `server.apiRoutes: [routes]`.

Reconnect to an in-progress agent run from a second client without restarting it — useful in long-running chat sessions or multi-tab UIs.

typescript

```typescript
const run = await agent.generate('Analyze the sales report', {
runId: 'run-abc123' });
// In another process or tab, observe the same run:
const observer = agent.observe('run-abc123');
for await (const chunk of observer) {
  process.stdout.write(chunk.text);
}
// Cancel at any time:
observer.abort();
```

— Full mechanism, named APIs, and a runnable reconnect example @mastra/core@1.83.0

02 Built-in MCP server at /api/mcp with OAuth NEW 95

Configure the built-in MCP server with the new `server.mcp` option — toggle it on or off, change its path, pick toolsets including `memory`, `observability`, `observability_analytics`, `stored_agents`, feature agents and workflows, serve read-only tools, and include or exclude agents, workflows, and tools by id. OAuth

discovery and token audience checks arrive via `server.mcp.oauth` fields (`resource`, `requireAudience`, `authorizationServers`, `scopesSupported`), with auth providers implementing getOAuthMetadata() and verifyAudience(), and CompositeAuth.getAuthenticatedProvider(user) returning the provider that accepted the token. Introspect the server via the new `GET /api/mcp/builtin` route.

Enable the built-in MCP server for existing agents and tools with OAuth audience enforcement, so MCP clients can call your backend without a separate MCPServer definition.

typescript

```typescript
import { Mastra } from '@mastra/core';

const mastra = new Mastra({
  server: {
    auth: myAuthProvider,
    mcp: {
      toolsets: ['memory', 'observability'],
      oauth: {
        resource: 'https://api.example.com',
        requireAudience: true,
        authorizationServers: ['https://auth.example.com'],
        scopesSupported: ['read', 'write'],
      },
    },
  },
});
```

— Named endpoint and config fields with a runnable setup example `@mastra/core@1.83.0`

03 **Breaking change: durable flag scope on Agent** BREAKING

83

On a plain `Agent`, `durable` now controls resumable streams only, not checkpoint-backed execution. To keep checkpoint-backed durable execution, migrate to createDurableAgent({ agent }); pass `true` to enable resumable streams everywhere, `false` to disable them, or an object with `pubsub`, `cache`, `shouldCache`, and `cleanupTimeoutMs` to configure stream behavior.

— Exact migration path and config fields spelled out `@mastra/core@1.83.0`

04 **Delta-write workflow storage (storageVersion 2)** NEW

78

`@mastra/pg` and `@mastra/libsql` support a new experimental `storageVersion: 2` option that reduces workflow storage writes by persisting only per-step changes; step history loads lazily via an async steps() function.

– Named config key and mechanism, no runnable example given `@mastra/core@1.83.0`

05 **Generative UI cards in Slack channel adapter** NEW 68

An experimental `generativeUI` option on the Slack channel adapter gives the agent a `render_card` tool that posts json-render specs as Block Kit cards (fields, tables, charts, images, link buttons); it requires the optional `@json-render/chat` package.

– Named option, tool, and package, no usage example `@mastra/core@1.83.0`

06 **One-off and bounded agent schedules** NEW 65

Schedules can fire once at a specific time via `runAt`, or run on a recurring `cron` schedule capped with `endAt`, avoiding orphan cron jobs that never stop.

Schedule a one-time agent run at a specific time, or cap a recurring schedule so it stops automatically — avoids leaving orphan cron jobs.

typescript

```
import { Mastra } from '@mastra/core';

const mastra = new Mastra({
  schedules: [
    // Fire once at a specific moment:
    { agent: myAgent, runAt: new Date('2025-09-01T09:00:00Z') },
    // Run every hour but stop after the deadline:
    { agent: myAgent, cron: '0 * * * *', endAt: new Date('2025-
12-31T23:59:59Z') },
  ],
});
```

– Runnable schedule example but not described in prose bullets `@mastra/core@1.83.0`

07 **maxChunkChars limit for semantic recall** NEW 65

The new `maxChunkChars` semantic-recall option (default 16,384 characters) splits long messages into individually embedded vector rows instead of failing or truncating.

– Named config key and default, no usage example `@mastra/core@1.83.0`

08 **Span query sorting by duration, model, and provider** IMPROVED 65

Completed span queries can now be sorted by duration with `orderBy: [{ field: 'durationMs', direction: 'desc' }]` to surface the slowest spans, and `model` and `provider` are added as new sort fields.

– Named sort syntax and fields, grouped from same observability area `@mastra/core@1.83.0`

09 **User context in dynamic agent config callbacks** IMPROVED

65

Dynamic agent configuration callbacks (`instructions` , `model` , `tools` , `memory` , and others) now receive an optional `user` argument; when server auth is configured, the authenticated user arrives as a `MastraUser` shape (`id` , `name?` , `email?` , `avatarUrl?` , `role?`).

— Named shape and fields, no runnable example `@mastra/core@1.83.0`

---

10 **Skill catalog hidden from system message in SkillsProcessor**

NEW

65

Setting `injectCatalog: false` on `SkillsProcessor` keeps skills out of the agent system message; the agent instead gets a short instruction to find skills via the `skill_search` tool and load them with the `skill` tool.

— Named flag and tool names, no example `@mastra/core@1.83.0`

THINNER COVERAGE BELOW

11 **getFeedbackById storage lookup** NEW

50

storage.getFeedbackById(feedbackId) looks up a single feedback record, returning the record or `null` .

— Exact method signature but minimal context `@mastra/core@1.83.0`

└──▷ **BREAKING ON UPGRADE**

! On a plain Agent , `durable` now controls resumable streams only, not checkpoint-backed execution. To keep checkpoint-backed durable execution, migrate to createDurableAgent({ agent }); pass `true` to enable resumable streams everywhere, `false` to disable them, or an object with `pubsub` , `cache` , `shouldCache` , and `cleanupTimeoutMs` to configure stream behavior.

WAS THIS USEFUL? 👍 👎

◆ Browser & Computer Use

browser-use

Feedback ⓘ 1.0.5 NOTES ↗ RANK

The browser-use Python library automates web browsers for AI agents.

> browser-use shipped automatic post-run judgement with prompt-injection and bot-block detection, plus a screenshot-pruning optimization that roughly halves input token cost on long screenshot-heavy runs.

└──▷ WHAT SHIPPED · 3 FEATURES most completely described first ? what's the number?

01 **Automatic run judgement with gotcha tracking** NEW 87

Passing `judge_context=` with evaluation criteria triggers automatic post-run evaluation (or disable it with `judge=False`); results land in `result.judgement` with fields `success`, `failure_reason`, `description`, `prompt_injection_detected`, `bot_block_or_captcha_detected`, and `gotchas`. The judge now records site-pushback events — form-format rejections, errors, blocks, and rate-limits — in `gotchas` even on successful runs, giving visibility into fragile paths the agent navigated.

— Names exact flags and result fields, fully actionable 1.0.5

02 **Batched screenshot pruning cuts token cost** IMPROVED 65

browser-use now prunes old screenshots in one batch once 20 tool results carry one, instead of on every new screenshot, preserving provider prompt-cache hits and cutting input token cost — a 705-step task falls from 27.8M to roughly half as many input tokens.

— Concrete mechanism and numbers but no config surface to invoke 1.0.4

THINNER COVERAGE BELOW

03 **Dedicated Chromium instance for BU_NAME callers** IMPROVED 43

`BU_NAME` callers now get their own Chromium instance when Chrome is unavailable, so named-caller sessions no longer fail silently.

— Names the env var but little mechanism detail 1.0.4

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.34.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

> This window centers on MCP server security controls — OAuth client associations with consolidation previews, an explicit `mcp:connect` grant, and connection log exports to OTEL — alongside real-time risk policy enforcement in Claude and a round of admin tooling covering platform credentials, hook visibility, and Organizational Memory evals.

└──▷ WHAT SHIPPED · 24 FEATURES most completely described first ? what's the number?

01 **MCP connection log exports to OTEL** NEW 90

Enable the MCP connection logs data export to send failed connections and server problems as `speakeasy.mcp_connection` OTEL events to the project's OTEL destination (blocked or declined connections excluded by default). Set the minimum severity with `set_data_export_min_severity` in Platform MCP or via API: `error` for failed connections (default), `warn` to also send blocked connections, or `info` to also send declined ones.

— Names the exact event type, tool, and severity values. server@2.33.0 dashboard@0.147.0

02 **Audit log origin filter and Team changes view** IMPROVED 86

Filter by `origin` (`user` or `system`) plus date range and resource on the `auditlogs.list` and `auditlogs.listFacets` endpoints, and a new default Team changes view hides Speakeasy's background activity, which now lives under Automated changes; existing `?category=configuration` links redirect to Team changes.

Filter the audit log to only user-originated changes when reviewing who on the team made configuration changes.

```
$ curl -X GET 'https://<your-gram-host>/api/auditlogs.list?origin=user'
```

Retrieve only user-originated audit log entries, filtering out Speakeasy background activity, over a date range.

```
$ GET /auditlogs?origin=user&from=2025-01-01&to=2025-07-01
```

— Named endpoints with runnable example queries. `server@2.33.0`

03 **Session and chat access scoped to caller** BREAKING 85

Restricts session and chat data access to the caller's own sessions and agents they own across several surfaces: the four raw log endpoints (readable without `log:read`), `telemetry.listSessions` (scoped to projects the caller holds `chat:read` on, with other-project totals flagged `scoped: true`) and `telemetry.searchChats`, Platform MCP's `list_chats`, `search_chats`, `read_chat`, `search_tool_calls`, and `list_recent_tool_calls`, and the ability to reveal the matched value of a risk finding — all without needing organization-wide `chat:read`.

— Names every affected endpoint and tool; no example call given. `server@2.33.0`

04 **OAuth client association and consolidation previews for MCP servers**

NEW 83

Attach an upstream OAuth client to an MCP server with `remoteSessions.attachServerClientAssociation`, which also binds the client to issuers of fronting gateways; detaching now supports undo, and the audit log records the new `remote-session-client:attach-mcp-server` action. Consolidating one authorization server into another now shows a preview of how each affected MCP server's OAuth clients would change before confirming, and blocks the consolidation when a server would end up with two OAuth clients for one provider or with a client its upstream does not accept.

— Named API method and audit action, but no sample call shown. `server@2.34.0`

05 **Organizational Memory eval runs in admin** NEW 80

Staff can start Organizational Memory eval runs against a demo organization's synthetic corpus, choosing answer, judge, summary, and digest models, track runs to completion, and compare runs side by side by accuracy, valid citations, leaks, tokens, cost, and latency on a new Organizational Memory evals page. Runs and results are also available through a new admin API, serialized one at a time and capped per rolling 24 hours, using the seeded demo corpus.

— Names metrics and rate cap, but no endpoint name given. `admin@0.13.0`

06 **Customer-managed encryption keys via orgEncryption API** NEW 78

Manage customer-managed encryption keys through the new `orgEncryption` management API: register a GCP KMS symmetric key, verify it, switch the organization to it, complete or roll back the switch, rotate a data key, and resume after a suspension.

— Named API and full action list, but no sample call. server@2.33.0

07 **Look up policy bypass request target** NEW 78

Look up a risk policy bypass request's target without filing a request using the new `risk.getPolicyBypassRequestTarget` endpoint, which surfaces the server name, URL or agent-supplied identifier, and the tool being requested.

Retrieve a risk policy bypass request's target — including the server name and tool — before deciding whether to file the request.

```
$ curl -X GET 'https://<your-gram-host>/api/risk.getPolicyBypassRequestTarget?link=<request-link>'
```

— Named endpoint with a runnable example call. server@2.33.0 dashboard@0.147.0

08 **Gemini chat import via Google Vault** NEW 75

Import Gemini web chats into Agent Sessions through Google Vault: organization admins paste a service account key and delegated Workspace admin, pick an organizational unit or Entire organization, and Speakeasy syncs completed UTC days every hour.

— Named steps and sync cadence, UI path only implied. server@2.33.0

09 **Platform credential lock and GCP credential management in admin** 75

BREAKING

A platform credential's identity can no longer be changed, and it cannot be deleted while Speakeasy-managed organization encryption keys still reach KMS through it — operations that previously may have succeeded are now refused. Staff can now manage platform GCP credentials from a new Platform credentials section in admin: list, create, view, run a live identity check, edit, and delete, with the same guards enforced.

— Names the new admin UI actions and the exact guard condition. server@2.33.0 admin@0.13.0

10 **Explicit mcp:connect permission required** BREAKING 70

Connecting to an MCP server now requires an explicit `mcp:connect` grant — `mcp:read` and `mcp:write` no longer imply it, so viewing or managing a server no

longer permits calling its tools. Existing grants were backfilled, but new grants for viewing or managing a server will not automatically allow calling its tools.

— Clear permission names and migration note, no UI path given. server@2.33.0

11 **Hooks visibility tab in admin** NEW 70

Staff can now view an organization's hook installs on a new Hooks tab in admin and through Admin MCP, showing version drift against the current release, install method, hooks endpoints in use, quiet machines, and traffic still reaching legacy endpoints.

— Lists exact data shown, UI path only. admin@0.13.0

12 **Real-time risk policy enforcement in Claude and Claude Cowork** NEW 65

Risk policies now enforce in real time directly inside Claude: Anthropic inference hooks scan each tool call when Validate tool calls is on in Claude, with a block, warn, or quarantine match denying the frame before any tools run; in Claude Cowork, hooks send prompt and tool events, warned prompts lead with an approval link, warned tool calls trigger Claude's own permission prompt, and block messages mask the matched value.

— Describes mechanism and setting, no API surface named. Behavior flow described, no config name or endpoint.

server@2.33.0

13 **Team memory access permissions** NEW 65

Grant `memory:read` and `memory:write` permissions for upcoming team memory via a custom role, the Access page, or the new Platform MCP `configure_memory_access_role` tool.

— Names permissions and tool though feature is still upcoming. dashboard@0.147.0

14 **Custom domain DNS diagnostics** IMPROVED 60

Custom domain setup now shows the designated CNAME target and ownership TXT record, with separate DNS configuration and infrastructure diagnostics, one recheck action, and independent DNS freshness; organization admins can read saved diagnostics through Platform MCP.

— Names DNS record types and recheck action, UI-level only. dashboard@0.147.0

15 **Unified role editor with per-tool MCP grid** IMPROVED 60

The role editor now shows all permissions on one page, with MCP access as a project-and-server grid supporting per-tool checkboxes, an explicit deny flow, role templates for new roles, and a Reset button to revert unsaved changes.

— UI elements described, navigable but no API/config name. dashboard@0.147.0

16 **Assistant identity upgrade endpoint removed** BREAKING

60

The `assistants.upgradeIdentity` endpoint and the Platform MCP `upgrade_assistant_workload_identity` tool are removed; Speakeasy now automatically moves each assistant to its own agent identity.

— Names removed endpoint and tool, no manual migration steps. server@2.33.0 dashboard@0.147.0

THINNER COVERAGE BELOW

17 **Gateways filter and source badges on MCP servers list**

IMPROVED 55

Add a one-click Gateways quick filter to the MCP servers list, and give each source badge (Gateway, Hosted, Remote, Tunneled, Unproxied) its own color on MCP cards and table rows.

— UI change named exactly, navigable but no deeper mechanism. dashboard@0.148.0

18 **Independent personal MCP server sign-ins** IMPROVED 50

Personal MCP server sign-ins always create a new, independent sign-in; the dashboard now offers New from scratch or New from existing instead of reusing a stored sign-in.

— Names the two UI options, limited mechanism detail. dashboard@0.147.0

19 **externalKeys API restricted to signing keys** BREAKING

50

The `externalKeys` API now only ever returns signing keys; an encryption key is refused from a JSON Web Key Set.

— Names endpoint and behavior change, no example given. server@2.33.0

20 **Organizational Memory on/off toggle replaces extraction cap**

IMPROVED 50

The Business Memory Extraction staff control is now an Organizational Memory on/off toggle with no cap input, defaulting to 100 when switched on; organizations with a cap of 0 now count as off for Organizational Memory.

— Names default value and toggle, limited further mechanism. admin@0.13.0

21 **Configurable assistant reasoning effort** NEW 45

Set the assistant's reasoning effort to Off, Low, Medium (default), or High, persisted through runtime inference requests.

— Options named but no config key or UI path given. server@2.34.0

22 **Tool Logs visible to all members** IMPROVED 45

Tool Logs now opens for every member: without the View tool logs permission, members see their own sessions and their agents' activity with a notice and a way to request wider access.

— Behavior change described, no config or endpoint named. dashboard@0.147.0

---

23 **Withdrawn tier reported on workload subject removal** IMPROVED 40

Withdrawing an admitted workload subject now returns `withdrawn_tier`, indicating whether the organization or project admission was removed.

— Names returned field but no endpoint or example call. dashboard@0.147.0

---

24 **Okta, MCP approval, MCP research and cost analytics GA** IMPROVED 35

Okta connections, MCP approval, MCP research, and employee identity folding in cost analytics are now enabled for every organization, with rollout flags removed.

— Lists four now-GA features with no usage detail. server@2.34.0

└──▷ BREAKING ON UPGRADE

! Connecting to an MCP server now requires an explicit `mcp:connect` grant; `mcp:read` and `mcp:write` no longer imply it. Existing grants were backfilled, but new grants for viewing or managing a server will not automatically allow calling its tools.

! The `assistants.upgradeIdentity` endpoint and the Platform MCP `upgrade_assistant_workload_identity` tool are removed.

! The `externalKeys` API now only ever returns signing keys; an encryption key is refused from a JSON Web Key Set.

! A platform credential's identity can no longer be changed, and it cannot be deleted while Speakeasy-managed organization encryption keys still reach KMS through it.

! `telemetry.listSessions` now returns per-session rows only for projects the caller holds `chat:read` on, with totals flagged `scoped: true` for others; `telemetry.searchChats` narrows to the caller's own sessions and agents they own instead of listing every member's.

! The `assistants.upgradeIdentity` endpoint and the Platform MCP `upgrade_assistant_workload_identity` tool are removed; Speakeasy now automatically moves each assistant to its own agent identity.

! A platform credential's identity can no longer be changed while Speakeasy-managed organization encryption keys still reach KMS through it, and it can no longer be deleted under the same condition — operations that previously may have succeeded are now refused.

WAS THIS USEFUL? 👍 👎

proxmoxmcp-plus

Feedback ⓘ v0.7.0 NOTES ↗ RANK

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 improves reliability of task monitoring, tool discovery, and API rate limiting: task status now persists when logs are unreadable, read-only tools and target discovery advertise richer metadata, and OpenAPI rate limits are measured by elapsed time instead of wall clock.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Task status persistence when logs unreadable IMPROVED

48

When a Proxmox task log cannot be read, the last observed task progress now stays available instead of being lost, and the job audit records `polled_without_log` to flag the degraded read.

— Names the audit marker but no config or command to invoke it v0.7.0

02 OpenAPI rate-limit window timing fix IMPROVED

48

OpenAPI rate-limit windows are now measured using elapsed time rather than wall-clock time, so system clock corrections no longer freeze or incorrectly clear a client's request allowance.

— Explains before/after mechanism but no tunable setting named v0.7.0

03 Idempotency and target discovery advertisements IMPROVED

37

Read-only tools and Code Mode lookups now advertise idempotency, and target discovery advertises network access and retries after encountering malformed inventory without caching the bad result.

— Describes behavior change but no interface for reader to act on v0.7.0

WAS THIS USEFUL? 👍 👎

memwhale

Feedback ⓘ v0.27.0 NOTES ↗ RANK

Memwhale is a persistent, local memory for developers and their coding agents. Records commands, output, errors, and the fixes that worked into SQLite and serves them over MCP.

> Memwhale v0.27.0 adds a full terminal workflow for reviewing and approving agent-saved lessons, plus a portable file format for sharing approved lessons across machines and teammates.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Terminal lesson review workflow NEW 85

Lessons queued by agents can now be reviewed entirely from the terminal with `mw memory pending`, `mw memory approve <id>`, and `mw memory reject <id>`, so agents only act on approved lessons without needing the dashboard or TUI pane. `mw doctor` now also warns when lessons are waiting for review and points to `mw memory pending`.

After an agent saves a lesson via MCP, approve it from the terminal so it appears in notes, warnings, and search.

```
$ mw memory pending
mw memory approve 1
```

— Names exact commands and doctor integration; runnable end to end v0.27.0

02 Lesson sharing via publish and adopt NEW 85

`mw memory publish` writes approved lessons to a repository-portable file at `.memorywhale/lessons.md`, and `mw memory adopt` loads that file into the local review queue, letting teammates on another machine pull the file and adopt the lessons into their own queue.

Publish approved lessons to a file you can commit, so teammates on a fresh clone can adopt them into their own review queues.

```
$ mw memory publish
git add .memorywhale/lessons.md && git commit -m 'share lessons'
# On another machine after pull:
mw memory adopt
```

— Exact commands and file path given, with a full commit/pull workflow v0.27.0

THINNER COVERAGE BELOW

03 **Claude Code lesson labeling** IMPROVED 35

Lessons saved from Claude Code's shell are now labeled `claude-code` instead of the previous `human` label; only newly saved lessons receive the updated label.

— Describes the change but gives no action for readers to take v0.27.0

WAS THIS USEFUL? 👍 👎

kubernetes-mcp-server

Feedback ⓘ v0.1.3 NOTES ↗ RANK

Model Context Protocol (MCP) server for Kubernetes and OpenShift

> kubernetes-mcp-server added a config-driven toggle and discovery mechanism for Kiali service mesh tools, opt-in control over which tools are exposed to MCP clients, namespace-scoped resource creation, NetObserv target filtering, and structured Kubernetes Status objects for resource tool errors.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Kiali tool-filter toggle and in-cluster URL discovery NEW

95

The `enable_target_compatibility_tool_filters` setting (default: on) controls whether Kiali tools are registered based on reachability checks and in-cluster URL discovery; setting it to `false` always registers Kiali tools and skips both the `/api/status` probe and CR-based discovery. Setting `[toolset_configs.kiali].url` or installing a Kiali CR lets the server discover the in-cluster Service URL when Kiali tools are missing from `tools/list`.

Disable reachability checks so Kiali tools always appear in `tools/list`, bypassing the `/api/status` probe and CR-based discovery.

toml

```toml
enable_target_compatibility_tool_filters = false

[toolset_configs.kiali]
url = "http://kiali.istio-system.svc:20001"
```

— Names exact config keys, probe endpoint, and gives runnable config example product docs

02 Opt-in tool exposure via config NEW

60

Adding `opt_in_tools` to the server config controls which tools are exposed to MCP clients, letting a deployment restrict its exposed tool surface to a specific list rather than exposing everything.

Restrict an MCP server deployment to only expose specific tools rather than the full tool surface.

yaml

```yaml
opt_in_tools:
  - resources_create_or_update
  - resources_get
```

— Names config key with runnable example but no further mechanism v0.1.3

03 **Namespace targeting for resource creation** IMPROVED

60

Passing `-n namespace` to `resources_create_or_update` applies the resource to a specific namespace, matching `kubectl apply -n namespace` behavior.

— Names exact tool and flag but no example shown v0.1.3

THINNER COVERAGE BELOW

04 **Structured Status objects for resource tool errors** IMPROVED

50

Resource tool errors now return a structured Kubernetes `Status` object instead of plain text, giving clients machine-readable error details.

— States before/after behavior but no example or schema given v0.1.3

05 **NetObserv target compatibility filtering** NEW

25

NetObserv now supports target compatibility filtering to restrict which targets are eligible for network observation.

— Thin one-line description with no mechanism or usage detail v0.1.3

WAS THIS USEFUL? 👍 👎

Prefect FastMCP

Feedback ⓘ v4.2.0 NOTES ↗ RANK

FastMCP is a Python framework for building MCP servers and clients, handling transport, schema generation, authentication, and server composition.

> FastMCP 4.2.0 introduces server-hosted agent skills with a catalog provider, client-side skill verification and installation, dependency injection via Annotated type hints, and raises the minimum Python version while reshaping the skills API.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 Skill hosting, verification, discovery, and CLI support NEW

85

Introduces a full skill system: host skills on the server via `SkillCatalogProvider` and Skill.from_directory() so a single deploy updates every connected agent; verify and install skills client-side with Client.read_skill() (checking each file's size and digest) and to_directory(); discover skills across a `ClientGroup` and through mounted servers and proxies for a unified multi-server inventory; and enumerate or serve skills via the new `--skills` flag on `fastmcp list` and the lightweight `SkillManifestProvider` in `fastmcp.skills.provider`.

Serve a skill directory from your MCP server so every agent gets updates on the next deploy without reinstalling locally.

python

```python
from fastmcp import FastMCP, Skill
from fastmcp.skills.provider import SkillCatalogProvider

deploy = Skill.from_directory("skills/deploy")

mcp = FastMCP("Team Skills")
mcp.add_provider(SkillCatalogProvider([deploy]))
```

List all skills exposed by a running server directly from the terminal.

```
$ fastmcp list --skills http://localhost:8000
```

— Named classes plus a runnable server example Mechanism named but no runnable example given Named surface but no command or example Flag and provider named with a runnable list command

v4.2.0

02 Annotated dependency injection for tool parameters NEW

80

Declare dependencies directly in `Annotated` type hints — for example `Annotated[dict, Depends(get_config)]` — so a parameter keeps its real type while the dependency can still read the tool argument through `CallArgument`.

— Concrete syntax named, no standalone runnable example v4.2.0

---

03 **Skill class split into storage-neutral and wire model**

65

BREAKING

The `Skill` class is now a storage-neutral object; the wire model previously named `Skill` is now `MCPSkill`. The old import path continues to work until 4.3.

— Names old and new types with migration window v4.2.0

---

04 **4.1 skills API deprecated** DEPRECATED

65

The 4.1 skills API — `fastmcp.server.providers.skills` and `fastmcp.utilities.skills` — is deprecated and will be removed in 4.3; existing code still runs but emits deprecation warnings.

— Exact module paths and removal timeline given v4.2.0

THINNER COVERAGE BELOW

---

05 **npm registry override for fastmcp dev** NEW

55

Override the npm registry used during `fastmcp dev` app builds with the `npm_config_registry` environment variable.

— Exact env var named, thin on context v4.2.0

---

06 **Python 3.11 minimum version requirement** BREAKING

40

Python 3.11 or newer is now required to run FastMCP; Python 3.10 and earlier are no longer supported.

— Clear but minimal detail beyond the version cutoff v4.2.0

└──▷ BREAKING ON UPGRADE

! Python 3.11 or newer is now required; Python 3.10 and earlier are no longer supported.

! The `Skill` class is now a storage-neutral object — the wire model previously named `Skill` is now `MCPSkill`. The old import continues to work until 4.3.

! The 4.1 skills API (`fastmcp.server.providers.skills` , `fastmcp.utilities.skills`) is deprecated and will be removed in 4.3; existing code still runs but emits deprecation warnings.

WAS THIS USEFUL?

api-mcp-server

Feedback ⓘ v3.42.1 NOTES ↗ RANK

MCP server that exposes Hostinger's platform API for hosting deployment, CDN control, domain management, and WordPress operations.

> api-mcp-server added a full set of GPU instance and GPU exposed-service management operations, plus campaign update and send operations for Reach campaigns.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **GPU instance and exposed service operations** NEW 59

New GPU compute management operations: `gpu_instances_create`, `gpu_instances_list`, `gpu_instances_get`, `gpu_instances_delete`, and `gpu_instances_metrics` for creating, listing, inspecting, deleting GPU instances and fetching their runtime metrics; plus `gpu_exposed-services_create`, `gpu_exposed-services_list`, and `gpu_exposed-services_delete` for managing GPU exposed services.

– Names all eight operations but no mechanism or behavior detail. v3.41.0 v3.42.0

02 **Campaign update and send operations** NEW 40

Added `reach_campaigns_update` to update an existing campaign and `reach_campaigns_send` to send a campaign.

– Names two operations but gives no further mechanism. v3.42.1

WAS THIS USEFUL?

pgedge-postgres-mcp

Feedback ⓘ CVE-2026-126199 CVE ↗

STABLE RANK

Pgedge-postgres-mcp is a pgEdge MCP Server. A PostgreSQL MCP server with a Natural Language Agent CLI and Web UI.

`CVE-2026-126199`: pgEdge Postgres MCP Server write confirmation skipped for writes that begin with a reading keyword (CVSS 6.8, affects `<= 1.0.0`)

▾ **2 more releases in this issue** · 2026-09-03 → 2026-11-11

> CVE-2026-126197 CVE ↗ STABLE
>
> `CVE-2026-126197`: pgEdge Postgres MCP Server SQL injection through custom pl-do tool parameters (CVSS 8.8, affects `<= 1.0.0`)

> CVE-2026-126196 CVE ↗ STABLE
>
> `CVE-2026-126196`: pgEdge Postgres MCP Server read-only mode can be bypassed by SQL statement smuggling through query_database (CVSS 8.1, affects `<= 1.0.0`)

WAS THIS USEFUL? 👍 👎

◆ AI Agent Communication

AG-UI

Feedback ⓘ release/2026-11-13 NOTES ↗ RANK

AG-UI: the Agent-User Interaction Protocol. Bring Agents into Frontend Applications.

> AG-UI's LangGraph adapter gained native interrupt response schema support and now declares protocol version on run-started events, alongside a breaking bump in peer dependency requirements.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Native interrupt response schema in LangGraph adapter

IMPROVED 68

The `@ag-ui/langgraph` adapter now reads LangGraph's native `Interrupt.response_schema` field (requires langgraph >= 1.2.12), falling back to value-level `responseSchema` or `response_schema` keys when the native field is absent.

– Names exact fields and version requirement but no usage example release/2026-11-13

02 Peer dependency bump to 1.0.0

BREAKING 60

`@ag-ui/client` and related peer dependencies now require >=1.0.0, because `utils.ts` uses `contentToText` and `contentHasMedia`, which only exist in 1.0.

– Names exact functions and version requirement causing the break release/2026-11-13

THINNER COVERAGE BELOW

03 Protocol version on run-started events NEW 21

Run-started events now declare the protocol version, per the release summary; no further mechanism is described.

– Mentioned only in summary, no detail on format or usage release/2026-11-13

└──▷ **BREAKING ON UPGRADE**

! `@ag-ui/client` and related peer dependencies now require >=1.0.0 because `utils.ts` uses `contentToText` and `contentHasMedia`, which only exist in 1.0.

WAS THIS USEFUL? 👍 👎

◆ Agent Memory

Volcengine OpenViking

Feedback ⓘ react@0.1.11 NOTES ↗

RANK

OpenViking is a context database that manages agent memory, knowledge retrieval, and skills.

> OpenViking's v0.5.5 release adds pgvector/PostgreSQL backends and multi-node coordination to scale beyond Redis, new batch-tagging across API/CLI/SDKs, and Gateway permission controls, alongside sweeping breaking changes to memory types, Kafka config, and git security; the react@0.1.11 release reworks Console into a multi-mode UI with error-isolating slots and migrates FileBrowser to a new permission field.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 pgvector and PostgreSQL backends for multi-host deployments

NEW 95

Add a pgvector vector backend via `storage.vectordb.backend: "pgvector"` (supports exact and HNSW search) and PostgreSQL path-lock leases via `storage.agfs.pathlock.provider: "postgres"`, enabling multi-process and multi-host deployments to share vector data without Redis; install with `openviking[pgvector]`.

Switch a multi-host deployment from Redis to PostgreSQL for both vector search and path-lock leases

yaml

```yaml
storage:
  vectordb:
    backend: "pgvector"
  agfs:
    pathlock:
      provider: "postgres"
```

— Exact config keys, install command, and runnable example. v0.5.5

02 Batch tag operations across API, CLI, and SDKs NEW

95

New endpoint `POST /api/v1/content/batch_set_tags` accepts 1–50 URIs per request and returns successful `items` and `failed` entries, with partial failures still returning HTTP 200; the same bulk operation is available from the CLI as `ov attrs batch-set-tags` and in the Python, TypeScript, and Go SDKs. In `set_tags` `append`

mode, `k+=number` or `k-=number` arithmetic now maintains usage counters on tags (a missing key starts at 0); non-numeric values or malformed tags (e.g. missing `=`) now fail the request instead of being silently dropped.

Tag 50 content URIs in one call and check for partial failures in the response

```
$ curl -X POST https://<host>/api/v1/content/batch_set_tags \
  -H 'Content-Type: application/json' \
  -d '{"uris":["viking://user/doc1","viking://user/doc2"],"tags":
{"reviewed":"true"},"mode":"append"}'
```

Increment a per-document access counter atomically using the new arithmetic tag syntax

```
$ ov attrs batch-set-tags --uris viking://user/doc1 --tags
'access_count+=1' --mode append
```

— Endpoint, CLI command, limits, and two runnable examples. v0.5.5

---

03 **Gateway tool permission policy defaults and narrowing**

85

BREAKING

The Gateway Context Profile `disabled_tools` default is now empty, so all tools — including the 8 write tools `remember`, `write`, `edit`, `add_resource`, `add_skill`, `forget`, `set_acl`, and `cancel_watch` — are open by default; `PUT /admin/policies/{id}` is a full replacement, so omitting `disabled_tools` re-enables write tools on a saved Profile. The `X-OpenViking-Context` request header's `disabled_tools` can now only narrow permissions further, and `gateway_tools: true` can no longer re-enable tools the Profile has disabled.

— Exact endpoint, header, and tool list but no sample request. v0.5.5

---

04 **Console multi-mode UI with async task tracking** BREAKING

80

Console now accepts a `modes` prop set to `['conversation']`, `['commands']`, or both, with independent `command.onOpenUri` routing and async task tracking via status snapshots, notifications, and optional query adapters. This replaces the removed `@openviking/react/modules/viking-bot` hook entry — existing conversation-only usages should switch to `<Console modes={['conversation']} />`.

Restrict Console to conversation-only mode as a drop-in replacement for the removed viking-bot hook entry.

```text
<Console modes={['conversation']} />
```

— Names prop, routing, and migration path from removed hook. react@0.1.11

05 **Derived path history index for snapshot log queries** BREAKING

75

Speed up `snapshot/log?paths=...` queries with a derived path history index — SQLite by default, MySQL for multi-instance deployments via `git.snapshot.log.provider: "mysql"` and `OPENVIKING_SNAPSHOT_LOG_DSN`, or `none` to keep the original scan. The index is enabled by default, creating a SQLite database at `{storage.workspace}/_system/snapshot/log/snapshot_log.db` per node; multi-instance deployments must set the MySQL provider and DSN or the service will not start.

— Exact config keys and failure condition, no example command. v0.5.5

06 **Kafka reporting config moved into ov.conf** BREAKING

75

Kafka reporting config must now live in `ov.conf` under `reporting.kafka`; the `config_file` and `nacos_*` fields are removed from `server.observability.reporting`, and the environment variables `NACOS_CONFIG_FILE` and `NACOS_SERVER_ADDR` are no longer read. Deployments previously configured via Nacos or external JSON must migrate `kafka` and `telemetry` sections into `ov.conf` and restart.

— Exact config path and removed env vars with migration note. v0.5.5

07 **React package icon, label, and component renames** BREAKING

70

Imports of `TabImpactIcon`, `TabSourceIcon`, and `TabTestIcon` no longer exist and must be removed; `FileBrowserLabels.searchPreview`, `searchPreviewEmpty`, and `searchPreviewFailed` overrides are removed. WikiDashboard actions moved to the host application, and FileTree's `renderDashboard` prop was renamed to `renderRowTrailing`.

— Names every removed export and renamed prop for migration. v0.5.5 react@0.1.11

08 **Multi-node coordination config for Watch leadership** NEW

70

Control multi-node Watch leadership and semantic task versioning with the new `storage.agfs.coordination` config key, accepting `memory`, `sqlite`, or `cache` as values; per-key state expires after one day. The older `watch_leader_provider` config key is deprecated and will be removed next version — migrate to `storage.agfs.coordination.provider`.

— Names config key and values but no runnable example. v0.5.5

09 **Git redirect and private network blocking hardening** BREAKING

65

Git pre-flight checks, clone, and fetch no longer follow HTTP redirects — callers must use the final repository URL — and loopback, link-local, unspecified, multicast, reserved, and known cloud-metadata addresses are blocked and cannot be allowlisted via the code-hosting whitelist or `allow_private_networks=true`.

— Specific blocked address classes and flag, no example request. v0.5.5

10 **Error boundary slots for rendering failures** NEW

60

The new `slots.errorFallback` prop isolates rendering failures in blocks, file previews, command results, and conversation regions, with local reset callbacks and retryable lazy content so a failure in one region does not take down the page.

Provide a custom error fallback UI for rendering failures in blocks or conversation regions without taking down healthy parts of the page.

```text
<Console slots={{ errorFallback: ({ reset }) => <button onClick={reset}>Retry</button> }} />
```

— Concrete prop and example but limited mechanism detail. react@0.1.11

11 **FileBrowser permission field migration to access** BREAKING

60

FileBrowser now reads effective permissions from the `access` field in filesystem `ls` / `stat` responses instead of `permission`; backends must upgrade or adapt custom gateway responses to keep supplying the read-only indicator and effective ACL level.

— Clear before/after but no example of the response shape. react@0.1.11

12 **Gateway Context Profile recall scope and memory capture controls**

NEW

60

Restrict Gateway recall scope to the current peer with `recall_peer_scope` (`all` or `actor`) and limit which memory types capture extracts with `capture_memory_types` in Context Profiles.

— Named fields and values but no example or endpoint. v0.5.5

13 **Partial import status reporting** BREAKING 60

First non-strict imports now report `status="partial"` and `import_complete=false` on partial success instead of failing entirely. Callers that check only task status must also check `import_complete`; partial imports do not advance the source checkpoint, and the Watch round is recorded as `partial`.

— Named status fields and caller impact, no example payload. v0.5.5

14 **Immediate conflict response for privacy-config writes** IMPROVED 60

Privacy-config write operations now return HTTP 409 (`CONFLICT`) with `conflict_type: path_busy` and `retryable: true` immediately on a lock conflict, instead of waiting up to 30 seconds.

— Exact status code and fields but no sample response. v0.5.5

THINNER COVERAGE BELOW

15 **Retirement of soul and identity memory types** BREAKING 55

The `soul` and `identity` memory types are retired and ignored even if present in persisted policies or deployment overrides; automatic memory updates can no longer modify `soul.md` or `identity.md`, and batches that include such operations are rejected.

— Clear before/after but no migration step given. v0.5.5

16 **ExperienceDashboard visual and pagination updates** IMPROVED 40

ExperienceDashboard now shows Updated tags, loading skeletons, one-decimal injection rates, scroll-based trajectory pagination, sliding tab indicators, and configurable rounded outcome bars.

— Lists several UI changes but no mechanism or config. react@0.1.11

17 **Skill updates no longer auto-recover on failure** BREAKING 35

Skill updates no longer back up or roll back on failure; a failed or cancelled update leaves files, indexes, and privacy config in their failed state with no recovery to the previous version.

— States behavior change only, no named surface or recovery step. v0.5.5

---

18 **Parent directory preview links for abstract/overview files**

IMPROVED 25

Clicking `.abstract.md` and `.overview.md` links now opens the parent directory's L0 and L1 preview.

— Bare behavior description, no further detail. react@0.1.11

└─▷ **BREAKING ON UPGRADE**

! The public `@openviking/react/modules/viking-bot` hook entry is removed; replace it with Console using `modes={['conversation']}` and its conversation services.

! Remove imports of `TabImpactIcon`, `TabSourceIcon`, and `TabTestIcon` — these exports no longer exist.

! Remove `FileBrowserLabels.searchPreview`, `searchPreviewEmpty`, and `searchPreviewFailed` overrides — these label keys are removed.

! FileBrowser reads permissions from the `access` field in `ls` / `stat` responses; returning only the old `permission` field no longer supplies the read-only indicator or effective ACL level — upgrade the backend or adapt custom gateway responses.

! The Gateway Context Profile `disabled_tools` default is now empty, meaning all tools — including the 8 write tools `remember`, `write`, `edit`, `add_resource`, `add_skill`, `forget`, `set_acl`, and `cancel_watch` — are open by default. Accounts with no saved `default` Profile and deployments without an explicit `disabled_tools` in `default_profile` now get all tools. `PUT /admin/policies/{id}` is a full replacement: omitting `disabled_tools` from the request body re-enables write tools on a saved Profile.

! The `X-OpenViking-Context` request header's `disabled_tools` can now only narrow permissions further; `gateway_tools: true` can no longer re-enable tools that the Profile has disabled. Clients that relied on `{"disabled_tools":[]}` or `{"gateway_tools":true}` to gain extra tools must now modify the Profile instead.

! The `soul` and `identity` memory types are retired and ignored even if present in persisted policies or deployment overrides. Automatic memory updates can no longer modify `soul.md` or `identity.md`; batches that include such operations are rejected.

! Kafka reporting config must now live in `ov.conf` under `reporting.kafka`; the `config_file` and `nacos_*` fields are removed from `server.observability.reporting`, and the environment variables `NACOS_CONFIG_FILE` and `NACOS_SERVER_ADDR` are no longer read. Deployments previously configured via Nacos or external JSON must migrate `kafka` and `telemetry` sections into `ov.conf` and restart.

! Skill updates no longer back up or roll back on failure; a failed or cancelled update leaves files, indexes, and privacy config in their failed state with no recovery to the previous version.

- ! First non-strict imports now report `status="partial"` and `import_complete=false` on partial success instead of failing entirely. Callers that check only task status must also check `import_complete`; partial imports do not advance the source checkpoint, and the Watch round is recorded as `partial`.
- ! Git pre-flight checks, clone, and fetch no longer follow HTTP redirects; use the final repository URL. Loopback, link-local, unspecified, multicast, reserved, and known cloud-metadata addresses are blocked and cannot be allowlisted via the code-hosting whitelist or `allow_private_networks=true`.
- ! Privacy-config write operations now return HTTP 409 (`CONFLICT`) with `conflict_type: path_busy` and `retryable: true` immediately on a lock conflict instead of waiting up to 30 seconds.
- ! The snapshot path history index is enabled by default, creating a SQLite database at `{storage.workspace}/_system/snapshot/log/snapshot_log.db` per node. Multi-instance deployments must configure `git.snapshot.log.provider: "mysql"` and set `OPENVIKING_SNAPSHOT_LOG_DSN`; the service will not start if the DSN variable is missing.
- ! The `watch_leader_provider` config key is deprecated and will be removed in the next version. Migrate to `storage.agfs.coordination.provider` and remove the old key.
- ! In `set_tags` `append` mode, keys ending in `+` or `-` are now interpreted as arithmetic increment/decrement operators (e.g. `count+=1` increments `count`), not as literal tag names. Using a non-numeric value (e.g. `x+=abc`) or patching a non-numeric existing value returns `InvalidArgumentError`. Malformed tags (e.g. missing `=`) in `append` mode now fail the request instead of being silently dropped.
- ! The `@openviking/react` package has multiple breaking interface changes, including WikiDashboard actions moved to the host, `FileTree`'s `renderDashboard` renamed to `renderRowTrailing`, Console `modes` changes, and removal of the `@openviking/react/modules/viking-bot` entry point.

WAS THIS USEFUL? 👍 👎

EverMind AI EverOS

Feedback ⓘ v1.7.0 NOTES ↗ RANK

EverOS provides a local-first memory layer that stores and evolves context for AI agents.

> EverOS v1.7.0 ships a live memory Purge API, per-task LLM model routing, configurable boundary-detection and tokenizer strategies, and new profile and benchmark tooling, alongside several breaking config changes.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 Live memory Purge API NEW 95

New `POST /api/v2/memory/purge`, `POST /api/v2/memory/purge/confirm`, and `GET /api/v2/memory/purge/{job_id}` endpoints let operators start, confirm and monitor memory purges without stopping the server. Writes into the purged range return `409 PURGE_IN_PROGRESS`, cross-site requests are blocked with `403 CROSS_SITE_REFUSED`, and job runtime can be bounded with the new `[purge]` config section.

Purge a single project's memory while the server keeps serving other apps, then confirm the deletion only after verifying the job is correct.

```
$ # Start a purge scoped to one project
curl -X POST https://everos.example.com/api/v2/memory/purge \
  -H 'Content-Type: application/json' \
  -d '{"app_id": "myapp", "project_id": "proj-42"}'

# Poll progress
curl https://everos.example.com/api/v2/memory/purge/<job_id>

# Confirm deletion when ready
curl -X POST
https://everos.example.com/api/v2/memory/purge/confirm \
  -H 'Content-Type: application/json' \
  -d '{"job_id": "<job_id>"}'
```

— Full endpoints, status codes and config section named. v1.7.0

02 Per-task LLM model and parameter settings NEW 95

Each of twelve LLM tasks (`boundary` , `episode` , `summary` , `atomic_fact` , `foresight` , `profile` , `reflection` , `agent_case` , `agent_skill` , `knowledge` , `search` , `decider`) can get its own model via `[llm.<task>]` config sections or `EVEROS_LLM__<TASK>__<FIELD>` environment variables. Per-call defaults `[llm]` `temperature` , `max_tokens` , and `sdk_max_retries` apply only when a call doesn't

specify its own values, and are overridable per task via `[llm.<task>]`; run `everos config show` to see what each task resolves to and where each field came from.

Use a cheap small model for fast boundary detection and a larger reasoning model for profile extraction, keeping costs low without sacrificing profile quality.

toml

```toml
[llm]
model = "gpt-4.1-mini"

[llm.boundary]
model = "qwen3-4b-instruct-2507"

[llm.profile]
model = "o3"
timeout_seconds = 240

[boundary_detection]
strategy = "batch"

[profile]
max_episodes_per_call = 10
```

— Lists all tasks, env var pattern, and an inspection command. v1.7.0

---

03 **Configurable BM25 keyword tokenizer** NEW 90

`[tokenizer] name` sets the keyword (BM25) tokenizer to `jieba` (default, suited to Chinese) or `nltk` (suited to English, installed via `pip install "everos[nltk]"` plus `python -m nltk.downloader punkt_tab stopwords`). Changing `name` on an existing store requires `everos cascade rebuild`.

Switch to the nltk tokenizer for more accurate English keyword search, then rebuild the existing store's index to match.

```
$ # 1. Update the config
# everos.toml:
#   [tokenizer]
#   name = "nltk"

# 2. Install the extra
pip install "everos[nltk]"
python -m nltk.downloader punkt_tab stopwords

# 3. Rebuild the existing store's index
everos cascade rebuild
```

— Exact config key, install command, and rebuild step given. v1.7.0

04 **Benchmark dataset download and configuration** NEW 85

Fetch benchmark datasets with `bash benchmarks/download_data.sh <benchmark>`, including LoCoMo's gold audit, LongMemEval's upstream file, and SubtleMemory's nested personas (re-running the script resumes interrupted downloads). The new `flush_scope` setting controls how dataset sessions are flushed to EverOS, accepting `"session"` (default) or `"conversation"`, and `benchmarks/configs/locomo-step.toml` (selectable as `DATASET=locomo-step`) runs LoCoMo with step boundary detection and prompt-only replies, matching the configuration behind the published LoCoMo score.

— Names script, setting, and dataset config file/selector. v1.7.0

05 **Configurable boundary-detection strategy** NEW 83

`[boundary_detection] strategy` selects between `batch` (default, one call per batch of messages, roughly one-tenth the calls and 5x faster than `step`) and `step` (one decision per message, scoring about 0.3 points higher on LoCoMo; use with `[memorize] mode = "chat"`).

Use a cheap small model for fast boundary detection and a larger reasoning model for profile extraction, keeping costs low without sacrificing profile quality.

toml

```toml
[llm]
model = "gpt-4.1-mini"

[llm.boundary]
model = "qwen3-4b-instruct-2507"

[llm.profile]
model = "o3"
timeout_seconds = 240

[boundary_detection]
strategy = "batch"

[profile]
max_episodes_per_call = 10
```

— Names config key, modes, and measured tradeoffs. v1.7.0

06 **Episode cap per profile extraction call** NEW 60

`[profile] max_episodes_per_call` (default 10) limits how many episodes a single profile extraction call folds together.

Use a cheap small model for fast boundary detection and a larger reasoning model for profile extraction, keeping costs low without sacrificing profile quality.

```toml
[llm]
model = "gpt-4.1-mini"

[llm.boundary]
model = "qwen3-4b-instruct-2507"

[llm.profile]
model = "o3"
timeout_seconds = 240

[boundary_detection]
strategy = "batch"

[profile]
max_episodes_per_call = 10
```

— Names config key and default, no further mechanism. v1.7.0

THINNER COVERAGE BELOW

07 Do-not-record and pending fields in user profiles NEW

55

User profiles now keep `do_not_record` and `pending` lists in `user.md` frontmatter: items the user asked not to be remembered are excluded from the profile, and information seen only once waits in `pending` until a later conversation confirms it.

— Names frontmatter fields and behavior but no command. v1.7.0

08 Decider settings moved to `[llm.decider]` DEPRECATED

55

Retrieval decider model settings move from `[decider]` to `[llm.decider]`; the old `[decider]` section still works but logs a startup warning.

— Names old and new config paths and migration behavior. v1.7.0

09 Startup validation for `[llm]` config keys BREAKING

55

Unknown keys under `[llm]`, or an `EVEROS_LLM__*` environment variable naming no valid key, now stop server startup with a list of valid keys; fix them before upgrading.

— Names exact failure trigger and required fix. v1.7.0

10 **Per-episode summaries for search results** NEW 35

Each episode now gets its own summary so a search result shows what the episode is about at a glance; this costs one additional model call per episode.

— Explains benefit and cost but no configurable surface. v1.7.0

11 **Provider-building Python API changed** BREAKING 25

The Python API for building providers changed in this release; HTTP and CLI interfaces are unchanged apart from additions.

— No specifics given on what in the API changed. v1.7.0

└──▷ **BREAKING ON UPGRADE**

! Unknown keys under `[llm]`, or an `EVEROS_LLM__*` environment variable naming no valid key, now stop server startup with a list of valid keys — fix them before upgrading.

! The Python API for building providers changed; HTTP and CLI are unchanged apart from additions.

! Move the retrieval decider's model settings from `[decider]` to `[llm.decider]`; the old `[decider]` section still works but logs a startup warning.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Inference Providers & GPU Clouds

Nebius AI Cloud

Feedback ⓘ changelog · Nov 14, 2026

CHANGELOG ↗

Nebius AI Cloud provides GPU compute, managed Kubernetes, distributed training infrastructure, and serverless AI workloads.

Nebius AI Cloud adds an MCP server (preview) for coding agents, Serverless AI resource cloning, tiered Compute quotas, and uncapped spot GPU pricing with priority allocation.

Connect a coding agent such as Claude Code, Codex, or Cursor to Nebius AI Cloud via the Nebius MCP server (preview) to run CLI commands, provision GPUs, run Serverless AI workloads, and control costs.

Clone Serverless AI resources in the web console, with the creation page prefilled from the original resource's settings.

Compute now has default and full quota tiers: new accounts start with default quotas, which Nebius may raise automatically based on usage history, with individual increases still requestable.

Spot prices for preemptible GPU platforms are now uncapped and can exceed on-demand rates; each platform has a minimum price, and setting a max price in a pricing policy grants higher allocation priority when capacity is tight.

NEW INTEGRATION Dropbox integration for Object Storage

NEW SUPPORTED PLATFORM Managed Service for MLflow in eu-west1

└─▷ BREAKING ON UPGRADE

! Spot prices for preemptible GPU platforms are no longer capped at on-demand rates and can now exceed them; existing workloads relying on the previous cap may incur higher costs unless a max price is set in a pricing policy.

WAS THIS USEFUL? 👍 👎

Anyscale

Feedback ⓘ Version 0.26.117 release notes NOTES ↗ RANK

Anyscale provides a managed Ray platform for distributed AI development, training, batch processing, and model serving.

> Anyscale's latest CLI/SDK release automates GPU metrics collection on KubeRay clusters and changes the working directory for workspace command execution on Kubernetes-stack workspaces.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Automatic DCGM exporter install for KubeRay** NEW 85

Passing `--yes` to `anyscale cloud setup --stack kuberay` installs the DCGM exporter without prompting, even on clusters with no GPU nodes. The bundled exporter only activates on nodes carrying a GPU label, so GPU nodes added later are automatically covered.

Set up KubeRay Connect on a cluster that has no GPU nodes yet, automatically installing the DCGM exporter so GPU metrics are captured when GPU nodes are added later.

```
$ anyscale cloud setup --stack kuberay --yes
```

— Names exact flag and command, explains activation mechanism.

Version 0.26.117 release notes

02 **Workspace command execution directory change** BREAKING 80

On Kubernetes-stack workspaces, `anyscale workspace_v2 run_command` and anyscale.workspace.run_command() now run commands in the workspace project directory (e.g. `/home/ray/default`) instead of `/home/ray`. Commands relying on starting in `/home/ray` must now use an absolute path or an explicit `cd`.

— Names exact APIs and old/new paths with migration guidance.

Version 0.26.117 release notes

└──▷ **BREAKING ON UPGRADE**

! On Kubernetes-stack workspaces, `anyscale workspace_v2 run_command` and anyscale.workspace.run_command() now run commands in the workspace project directory (e.g. `/home/ray/default`) instead of `/home/ray`; any command that relied on starting in `/home/ray` must now use an absolute path or an explicit `cd`.

WAS THIS USEFUL?  

◆ ML Platforms & SDKs

Snowflake Cortex AI

Feedback ⓘ changelog · Nov 14, 2026

CHANGELOG ↗ RANK

Snowflake Cortex AI provides managed language-model functions, search, and agents that work with data governed in Snowflake.

> Snowflake Cortex AI's latest release adds page-level filtering to document AI extraction, expands semantic views with many-to-many joins and materialization, rolls out new CoCo capabilities including Tableau-to-Streamlit conversion and automations, and adds Anthropic Claude 5.5 and OpenAI GPT-6 as available foundation models.

└──▷ WHAT SHIPPED · 5 FEATURES most completely described first ? what's the number?

01 **Page filtering for document AI extraction** NEW 70

Pass `FL_FILTER_PAGES` to `AI_EXTRACT` and `AI_PARSE_DOCUMENT` to scope document AI calls to specific pages rather than processing entire files.

— Names exact parameter and two functions, no further mechanism given

changelog-20261114-7c470485

02 **CoCo workflow and connectivity additions** NEW 60

CoCo gains several capabilities: convert Tableau workbooks to Streamlit in Snowflake apps (Preview), build automated workflows with CoCo automations (now generally available), connect CoCo to Microsoft Excel and Google Sheets (Preview), get skill recommendations and suggestions to create and share skills inside CoCo, and manage CoCo administration through the `CoCo Admin` interface in Snowsight (Preview).

— Names five distinct CoCo surfaces but each described only briefly

changelog-20261114-7c470485

THINNER COVERAGE BELOW

03 **Many-to-many semantic view relationships and materialization**

IMPROVED 45

Semantic views now support many-to-many relationship definitions (Preview) and materialization for persisted results (generally available), improving join patterns and query performance.

— States effect but no config keys or concrete numbers given changelog-20261114-7c470485

04 **Claude 5.5 and GPT-6 models in Cortex AI** NEW 45

Cortex AI now supports Anthropic Claude 5.5 and OpenAI GPT-6 as generally available foundation models, adding two new models to the Cortex model roster.

— Names specific models but no function or usage detail changelog-20261114-7c470485

05 **AI_TRANSCRIBE accuracy improvements** IMPROVED 15

`AI_TRANSCRIBE` received accuracy improvements, per the release summary; no further mechanism or scope is described.

— Only mentioned in headline summary, no detail bullet changelog-20261114-7c470485

WAS THIS USEFUL? 👍 👎

Tracer Cloud OpenSRE

Feedback ⓘ v0.1.2026.11.14 NOTES ↗

RANK

OpenSRE is an open-source toolkit for building AI SRE agents.

> OpenSRE's single release this window let the Kubernetes operator target specific namespaces within a configured allowlist, scoping agent actions.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Namespace targeting in Kubernetes operator allowlist** NEW 45

OpenSRE's Kubernetes operator now lets the model target namespaces inside a configured operator allowlist, scoping agent actions to permitted namespaces.

— Describes behaviour but no config key or command shown v0.1.2026.11.14

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.57.0 NOTES ↗

RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> OpenAI TypeScript SDK updated its TypeScript type definitions across several API surfaces in v7.57.0.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Updated types for Agents, Live, Images, admin APIs** IMPROVED

30

The SDK now carries updated TypeScript types for the Agents, Live, Images, and admin APIs.

— Names affected API areas but no specific types or fields changed v7.57.0

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.51.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK v3.51.0 adds expanded agent, image, Live, and external key capabilities, though specifics were not detailed.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Expanded agent, image, Live, external key support** NEW

15

OpenAI Python SDK now supports expanded agent, image, Live, and external key capabilities via API updates, per the v3.51.0 release notes. No specific endpoints, flags, or mechanisms were described.

— No concrete surfaces, mechanisms, or examples given v3.51.0

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback ⓘ

Harness: IAM authentication with remote MCP servers (2026-11-13) BLOG ↗ RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore's Harness component now supports IAM authentication when connecting to remote MCP servers.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 IAM authentication for remote MCP servers in Harness NEW

37

Agents using AgentCore Harness can now connect to IAM-protected remote MCP servers without managing separate credentials for each server.

— Explains benefit but no config keys, flags, or setup steps given.

Harness: IAM authentication with remote…

WAS THIS USEFUL? 👍 👎

Hugging Face Transformers

Feedback ⓘ v5.23.0

NOTES ↗ RANK

Hugging Face Transformers is a model-definition framework for machine learning models in text, vision, audio, and multimodal tasks, for both inference and training.

> Transformers v5.23.0 introduces a unified ExportedModel API covering ONNX, OpenVINO, Dynamo and ExecuTorch exports, an experimental JEV decision pipeline, selectable expert-parallelism implementations, and six new model architectures (Inkling2, LFM3, LFM3-VL, Dots 3 Note Preview, IQuestCoder, InternLM3), alongside breaking changes to ONNX export return types, audio extras and `ModelOutput`.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 **Unified ExportedModel exporter API** NEW 98

A new `ExportedModel` API exports any model to ONNX Runtime, OpenVINO, Dynamo or ExecuTorch via OnnxExporter().export(), can be pushed with exported.push_to_hub(), and reloaded with ExportedModel.from_pretrained(), which accepts the same kwargs and output class as the original model. Export-time ONNX Runtime graph fusions are configurable via OnnxConfig(ort_fusions='cpu' | 'cuda' | 'webgpu' | 'dml'), rewriting the graph into fused attention, `GroupQueryAttention`, rotary, norm and MoE ops, and a single backend registry now lets teams register custom export backends with `register_backend`.

Export a model to ONNX with CUDA-fused ops and push the result to the Hub for reuse by any team member.

python

```python
from transformers import AutoModel, AutoTokenizer
from transformers.exporters import ExportedModel, OnnxConfig,
OnnxExporter

model = AutoModel.from_pretrained('google/embeddinggemma-2')
tokenizer = AutoTokenizer.from_pretrained('google/embeddinggemma-
2')
inputs = tokenizer(['Exporters are here!'], return_tensors='pt')

exported = OnnxExporter().export(model, inputs,
OnnxConfig(dynamic=True, ort_fusions='cuda'))
exported.push_to_hub('my-org/embeddinggemma-2-onnx')
```

Load a previously exported ONNX model from the Hub and run inference with the same interface as the original model.

```python
from transformers.exporters import ExportedModel
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained('my-org/embeddinggemma-2-onnx')
inputs = tokenizer(['Hello world'], return_tensors='pt')

exported_model = ExportedModel.from_pretrained('my-org/embeddinggemma-2-onnx', device='cuda')
outputs = exported_model(**inputs)
```

— Full runnable export/reload example and named flags, classes and config options.

v5.23.0

02 **New model additions** NEW 78

Adds Inkling2, a 594B/17B-active multimodal mixture-of-experts model accepting text and image inputs via a Perception Encoder vision transformer projecting one embedding per 40×40-pixel patch; LFM3, the third-generation Liquid Foundation Model family combining gated short convolutions with grouped-query attention and full/sliding-window attention layers; LFM3-VL, a vision-language model built on LFM3 that accepts images and videos at native resolution using a hybrid vision encoder with 2D rotary embeddings; Dots 3 Note Preview, a multimodal mixture-of-experts model accepting text, image, video and audio inputs with BF16 and FP8 inference; IQuestCoder, a code language model family available in 7B, 14B and 40B sizes with base, instruct and thinking variants, loaded via `LlamaModel` and `LlamaForCausalLM` with `IQuestCoderConfig`; and InternLM3, an 8B instruction-tuned language model loaded via `LlamaModel` and `LlamaForCausalLM` with `InternLM3Config`, mapping `qkv_bias` to `attention_bias` and `bias` to `mlp_bias`.

— Detailed architecture and numbers but no loading snippet or class name. Architecture described but no config class or load example named. Mechanism described but no loading class or config named. Inference precisions named but no loading class or example. Sizes, variants and exact loading classes named. Named config mapping and loading classes make it directly usable.

v5.23.0

03 **Selectable expert-parallelism implementation** BREAKING 75

Expert parallelism now exposes a selectable `ep_implementation` — all-reduce for CUDA-graph-friendly inference or all-to-all for training — configured via

`ep_dispatch_experts`; the previous `ep_router` expert-parallel style has been removed.

— Names exact config flags and removed option but no usage example. v5.23.0

---

04 **JEV decision pipeline for single-token scoring** NEW

65

A new `JevForDecision` model class and experimental `decision` pipeline (or the `decide` method) score single-token labels, reading head files `head.safetensors` and `judge_config.json` alongside the model weights.

— Named classes and weight files but no runnable example given. v5.23.0

---

05 **Audio extras trimmed and LM decoding deprecated** BREAKING

60

The `[audio]` extra now only installs `torchaudio`, `soundfile` and `soxr`; `librosa` and `phonemizer` are dropped. `Wav2Vec2ProcessorWithLM` and pyctcdecode/kenlm LM decoding are deprecated.

— Named packages and deprecated class but no migration guidance. v5.23.0

THINNER COVERAGE BELOW

---

06 **Rope deltas removed from ModelOutput** BREAKING

35

Rope deltas are removed from `ModelOutput`.

— Bare removal note with no explanation or migration path. v5.23.0

└──▷ BREAKING ON UPGRADE

! The `ep_router` expert-parallel style is removed.

! export() now returns an `ExportedModel` instead of an ONNX program; an ONNX program is now an `onnx_ir.Model`.

! `OnnxConfig` and `OpenVINOConfig` no longer have an `output_path` option — use `save_pretrained` instead.

! The `[audio]` extra now only installs torchaudio, soundfile and soxr; librosa and phonemizer are dropped.

! `Wav2Vec2ProcessorWithLM` and pyctcdecode/kenlm LM decoding are deprecated.

! Rope deltas are removed from `ModelOutput`.

WAS THIS USEFUL? 👍 👎

skillhub

Feedback ⓘ v0.2.27 NOTES ↗ RANK

Skillhub is a self-hosted, open-source agent skill registry for enterprises. Publish & version skill packages, govern with RBAC and audit logs, deploy on-premise with Docker or Kubernetes.

> SkillHub's v0.2.27 release gives super-admins tools to curate the public homepage, brand the instance, and manage configuration through a reorganized, safer settings UI.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **Instance branding and site information settings** NEW

65

Super administrators can set the site name, upload PNG/JPEG logos and browser icons, edit multilingual homepage titles and descriptions, configure documentation and repository links, and control section visibility.

— Names concrete formats (PNG/JPEG) but no exact settings path. v0.2.27

02 **Curated skills and topics for homepage** NEW

60

Super administrators can now curate publicly visible skills and topics — selecting skills, organizing topics, setting covers, adjusting display order, and saving drafts for preview before publishing independently; visitors see only published content.

— Describes workflow but no exact UI path or config key. v0.2.27

03 **Reorganized system configuration with previews and conflict feedback**

IMPROVED

60

System configuration now has dedicated sections for site information, curated skills and topics, sign-in and review, and notifications and maintenance, with homepage previews, unsaved-change warnings, and configuration-conflict feedback.

— Names the four sections and UX safeguards but no exact locations. v0.2.27

THINNER COVERAGE BELOW

04 **Search and filtering for curated listings** NEW

55

Curated listings and topic links now support search by name or purpose and filtering by existing tags, with topics expanding in place and browsing state restored when returning from skill details.

— Clear behavior described, no API or flag named. v0.2.27

WAS THIS USEFUL?  

◆ Inference Serving & Optimization

Microsoft ONNX Runtime

Feedback ⓘ plugin-ep-cuda/v0.4.0

NOTES ↗ LATEST STABLE

ONNX Runtime runs machine-learning models across hardware and operating systems with an open-source inference engine.

ONNX Runtime CUDA Plugin EP adds CPU/GPU MoE expert offloading, native GGUF block-packed weight operators, and INT2/INT3/INT4 Tensor-Core QMoE kernels for large-model inference.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch plugin-ep-cuda/v0.4.0
https://github.com/microsoft/onnxruntime.git
# already have the repo? check out this version:
$ git checkout plugin-ep-cuda/v0.4.0
```

Session-scoped CPU/GPU hybrid expert offloading for `com.microsoft::QMoE` nodes, configured via `session.moe.offload.*` options with no model changes required.

New `GatherPacked`, `MatMulPacked`, and `MoEPacked` operators consume GGUF block-packed quantized weights in native layout without offline repacking.

New offline K-packed NVFP4 QMoE weight path (`weights_prepacked=1`) with register-decoded GEMV and single-CTA fused softmax/top-k routing for up to 512 experts.

NEW SUPPORTED PLATFORM New fast INT3 and mixed-width QMoE kernels with BF16 Tensor-Core decode GEMVs for INT4, INT3, and INT2 experts.

Shared MoE GEMM tactic-profiling cache across identically shaped layers reduces session-creation time for large MoE models.

+1 more

NEW SUPPORTED PLATFORM Linux aarch64 CUDA 13 GPU support

RETIRED SUPPORTED PLATFORM ACL (Arm Compute Library) execution provider removed

RETIRED SUPPORTED PLATFORM In-tree NvTensorRTRTXExecutionProvider provider bridge removed

└──▷ BREAKING ON UPGRADE

! Using `session.moe.offload.*`, pipelined external-weight loading, or `OrtApi::EpDevice_GetMemoryCapacity` requires ONNX Runtime core 1.32.0.

! Models using `GatherPacked`, `MatMulPacked`, `MoEPacked`, or ONNX opset 28 require ONNX Runtime core 1.32.0.

! With ONNX Runtime 1.32.0 core and `session.resource_cuda_partitioning_settings`, `GroupQueryAttention` nodes are declined by the plugin EP.

! Runtime fusions `ep.cuda.runtime_fusion.unit_swish` and `ep.cuda.runtime_fusion.causal_conv_with_state` are not available in CUDA plugin builds.

! Core and the CUDA plugin must be built from the same ONNX Runtime revision to avoid contributed-operator schema mismatches.

▾ **1 more release in this issue** · 2026-11-13 → 2026-11-14

v1.32.0 NOTES ↗ STABLE

ONNX Runtime adds CPU/GPU hybrid MoE offloading, new memory-capacity and weightless-compilation APIs, zero-copy buffer borrowing, and sub-second session init on large graphs.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.32.0
https://github.com/microsoft/onnxruntime.git
# already have the repo? check out this version:
$ git checkout v1.32.0
```

Query device total and available memory with the new `OrtApi::EpDevice_GetMemoryCapacity` (C++ EpDevice::GetMemoryCapacity()) and `OrtEpFactory::GetDeviceMemoryCapacity` APIs, with built-in and plugin CUDA EP support.

Control weightless compilation mode explicitly via `ModelCompilationOptions_SetWeightlessMode` or the `ep.enable_weightless_mode` session option (`'0'` = NONE, `'1'` = `EXTERNAL_ONLY`, `'2'` = ALL), replacing the deprecated `ModelCompilationOptions_SetWeightlessEnabled` and `ep.enable_weightless` options; compiled EPContext models now record `weightless_mode` and `external_dependencies` in model metadata.

Enable zero-copy CPU buffer borrowing in `AddExternalInitializers` via `session.use_external_initializer_file_buffers_directly="1"` for naturally aligned buffers larger than 127 bytes on little-endian hosts.

Accelerate external initializer loading into GPU memory with lock-free multi-worker pinned-memory staging, configurable via `session.external_data_loader.*`

session options.

Configure CPU/GPU MoE expert offloading and memory budgets via `session.moe.offload.*` session options, with an optional kernel dispatch policy via `session.kernel_dispatch_policy` (`auto`, `latency`, `memory`, or `safe`).

+3 more

└──▷ **BREAKING ON UPGRADE**

- ! The `ACLExecutionProvider` (Arm Compute Library EP), its build options, and its standalone C registration API have been removed; get_all_providers() no longer lists `ACLExecutionProvider` and applications must remove it from explicit provider lists.
- ! The in-tree provider-bridge `NvTensorRTRTXExecutionProvider` sources have been removed; callers must migrate to the standalone NVIDIA TensorRT RTX plugin EP registered via `OrtApi::RegisterExecutionProviderLibrary`, and legacy `AppendExecutionProvider` calls now return a deprecation error.
- ! The DLPack `is_bool_tensor` low-level override has been removed; `uint8` DLPack metadata now always imports as `uint8`.
- ! `OrtCompileApi::ModelCompilationOptions_SetWeightlessEnabled` and the `ep.enable_weightless` / `ep.enable_weightless_ep_context_nodes` session options are deprecated; migrate to `ModelCompilationOptions_SetWeightlessMode` and `ep.enable_weightless_mode`.
- ! This is the last release distributed via CocoaPods; future releases will use Swift Package Manager.

WAS THIS USEFUL? 👍 👎

◆ AI / LLM Gateways

Mozilla AI Otari

Feedback ⓘ v0.47.0 NOTES ↗ RANK

Mozilla AI Otari is an open-source AI gateway you run yourself. One endpoint for 40+ providers, with virtual keys, budgets, and usage tracking.

> Otari v0.47.0 adds cost transparency for tool calls, per-environment sandbox pricing for code execution, broader search tool visibility, model catalog overrides, and invitation origin tracking for organizations.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 **Tool cost reporting and settlement** IMPROVED 55

Usage reporting now shows the tool fee separately from the token cost, and tools return their settled cost alongside results for direct search, fetch, and code execution calls.

– Names the tool types but no API field or endpoint given v0.47.0

02 **Per-environment sandbox pricing for code execution** NEW 50

Code execution now supports sandbox environments chosen per run, with pricing set per environment.

– Explains mechanism but no config key or command shown v0.47.0

03 **Broader search tool visibility for API keys** IMPROVED 45

Search tools are now listed to any API key, with deployment-funded direct searches marked accordingly.

– Describes behaviour but no endpoint or flag named v0.47.0

04 **Model catalog overrides** NEW 25

Operators and organizations can now override a model's details in the catalog.

– No mechanism, UI path, or field named v0.47.0

05 **Invitation origin storage for organizations** NEW 15

Organizations now store invitation and membership origin data.

– Bare description, no fields or mechanism given v0.47.0

WAS THIS USEFUL? 👍 👎

sample-connector-for-bedrock

Feedback ⓘ v0.0.45

NOTES ↗ RANK

Sample-connector-for-bedrock is a seamlessly invoke Amazon Bedrock or your custom models, enabling a smooth experience with AWS GenAI services.

> This release adds image-to-image painting powered by eight Stability model capabilities, tightens per-key model visibility across the model-list APIs, moves database password handling to AWS Secrets Manager, and ships several breaking changes to painter configuration and the CloudFormation deployment templates.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 Painter model and region restrictions

BREAKING 90

`regions` no longer applies to `painter` configs — remove it from the model config, since painter now always calls us-west-2 for Stability models (and outside us-east-1, us-east-2, and us-west-2, all pictures and prompts go to us-west-2). `localLlmModel` in a `painter` config must now support function calling and accept images — Claude Sonnet (`claude-sonnet-5-5-chat`) is recommended — and the `nova-canvas` provider is removed, so any model configured on it returns HTTP 400; migrate to a `painter` model using `stability.stable-image-ultra-v1:1` , `stability.stable-image-core-v1:1` , or `stability.sd3-5-large-v1:0` . A `painter` model whose `paintModelId` is not one of those three now returns an error immediately without calling Bedrock.

— Exact config fields, model ids, and migration path all named. v0.0.45

02 RDS provisioning hardening in CloudFormation templates

90

BREAKING

When `PGSQL_PASSWORD_SECRET_ARN` is set, BRConnector reads the database password from AWS Secrets Manager instead of `PGSQL_PASSWORD` . The PGPassword parameter in `quick-build-brconnector.yaml` and `brconnector-ecs-fargate.yaml` can now be left empty; the stack generates the password into Secrets Manager and exposes it as the `PGPasswordSecretArn` output. A new `PGStorageEncrypted` parameter in both templates enables RDS storage encryption for new stacks. RDS databases created by the templates now have deletion protection enabled, and stack updates that replace the database keep the old one as a snapshot rather than deleting it. Updating an existing stack to the new templates prompts for `PGStorageEncrypted` — choosing `true` on an existing stack replaces its database with a new, empty one, so choose `false` to preserve the existing database — and a new `lambda` or `lambda-web` stack with an empty PGPassword requires the 0.0.45

BRConnector image to read the generated password from Secrets Manager; using an older image or `SourceRef` prevents the stack from reaching its database.

— Every CFN parameter and env var named with precise upgrade caveats. v0.0.45

03 **Per-model image billing via image_prices** BREAKING 84

A new `image_prices` field in a `painter` config sets per-model image billing; without it, painter now bills at built-in list prices instead of zero (previously a `painter` config with no `price_per_image` or `image_prices` billed at 0).

Bill each Stability model at its own rate in a painter config — set `image_prices` so heavy models like SD3.5 Large cost more than Core.

```json
{
  "localLlmModel": "claude-sonnet-5-5-chat",
  "paintModelId": "stability.stable-image-ultra-v1:1",
  "image_prices": {
    "stability.stable-image-ultra-v1:1": 0.08,
    "stability.stable-image-core-v1:1": 0.04,
    "stability.sd3-5-large-v1:0": 0.065
  }
}
```

— Named config field plus a runnable example config with real prices. v0.0.45

04 **Per-key model visibility across model-list endpoints** IMPROVED 80

A key not in a group now sees only its own bound models in `GET /v1/models`, `/user/model/list`, `/user/model/list-for-brclient`, and the Claude Code Web (`/cc`) model menu, preventing exposure of group models the key cannot call.

— Four exact endpoints named with the exact access-control behavior change. v0.0.45

05 **Image-to-image painting via Stability models** NEW 70

The `painter` provider now supports image-to-image workflows: users can upload up to 5 pictures (JPEG, PNG, or WebP, 10 MB each) in one request, and the chat model routes across eight Stability capabilities — text to image, image to image, search and replace, remove background, style guide, style transfer, control structure, and control sketch. Uploaded pictures are stored under `<s3Prefix>/uploads/<date>/` in S3 with no lifecycle rule, so they persist until explicitly deleted, and painter now replies in the language of the user's latest message even when earlier messages used a different language.

— Full capability list and limits given, but no command or flag to invoke it. v0.0.45

06 BRClient served at /brclient/ in lambda-web stacks NEW

62

`lambda-web` stacks now serve BRClient at `/brclient/`, built from source commits pinned in `scripts/brclient.source`; previously `/brclient/` returned 404.

— Named path and pinning file, but limited mechanism detail. v0.0.45

THINNER COVERAGE BELOW

07 Presigned picture URL expiry extended to 4 hours BREAKING

48

Presigned picture URLs now last up to 4 hours instead of 1; URLs signed with temporary credentials stop working when those credentials expire, so older pictures can become inaccessible sooner.

— Clear before/after duration but no action a reader takes. v0.0.45

└──▷ BREAKING ON UPGRADE

- ! `localLlmModel` in a `painter` config must now support function calling and accept images; Claude Sonnet (`claude-sonnet-5-5-chat`) is recommended. Configs using models that do not meet both requirements will break.
- ! `regions` no longer applies to `painter`; remove it from the model config. painter always calls us-west-2 for Stability models, and outside us-east-1, us-east-2, and us-west-2 all pictures and prompts go to us-west-2.
- ! A `painter` config with no `price_per_image` or `image_prices` is now billed at built-in list prices instead of 0.
- ! Presigned picture URLs now last up to 4 hours instead of 1; URLs signed with temporary credentials stop working when those credentials expire, so older pictures can become inaccessible sooner.
- ! The `nova-canvas` provider is removed; any model configured on it returns HTTP 400. Migrate to a `painter` model using `stability.stable-image-ultra-v1:1`, `stability.stable-image-core-v1:1`, or `stability.sd3-5-large-v1:0`.
- ! A `painter` model whose `paintModelId` is not one of the three supported Stability models now returns an error immediately without calling Bedrock.
- ! Updating an existing stack to the new `quick-build-brconnector.yaml` or `brconnector-ecs-fargate.yaml` prompts for `PGStorageEncrypted`; choosing `true` on an existing stack replaces its database with a new, empty one — choose `false` to preserve the existing database.
- ! A new `lambda` or `lambda-web` stack with an empty `PGPassword` requires the 0.0.45 BRConnector image to read the generated password from Secrets Manager; using an older image or `SourceRef` prevents the stack from reaching its database.

WAS THIS USEFUL?  

◆ Local LLM Runtimes

llama.cpp

Feedback ⓘ v1.0.0 NOTES ↗ CODE ↗ RANK

Llama CPP is a C/C++ library and set of command-line tools that runs large language models locally.

> llama.cpp v1.0.0 ships FP8 KV cache quantization, a host-memory MoE expert cache, new OpenAI-compatible TTS endpoints, support for several new text and vision architectures, and default SSRF/CORS/DNS-rebinding hardening in llama-server, alongside several breaking API removals and renames.

└──▷ WHAT SHIPPED · 12 FEATURES most completely described first ? what's the number?

01 **Audio/TTS generation and streaming support** NEW 85

Adds audio/TTS capabilities: stream raw PCM audio at the model sample rate via the new `mtmd_helper_resampler` function in `mtmd-helper.h`; call `llama-server`'s new `/v1/audio/speech` OpenAI-compatible endpoint, or set a `temperature` on the existing `/tts` endpoint, to generate speech from text; and generate drama-3 TTS audio (fishaudio) using `MTMD_GEN_AUDIO_TYPE_DRAMA3` in `mtmd_helper_gen_audio_inp`.

— Endpoint, header file and function names all given verbatim. v1.0.0

02 **FP8 KV cache quantization** NEW 80

Enable FP8 KV caches with `-ctk f8_e4m3` or `-ctv f8_e4m3`, with optional static per-tensor scales, to cut memory use on KV storage.

Run inference with FP8 KV caches to reduce memory pressure when serving large models with long contexts.

```
$ llama serve -hf ggml-org/Qwen3.5-0.8B-GGUF -ctk f8_e4m3 -ctv
f8_e4m3
```

— Named flags with a runnable command example. v1.0.0

03 **Host-memory MoE expert cache** NEW 78

Control host-memory MoE expert caching via `moe_cache_size` and `moe_cache_cpu_share` in `llama_context_params`, using LRFU/ClassLRU replacement policies with no CPU wait on cached experts.

— Named config params and replacement mechanism, but no usage example. v1.0.0

04 **External PLE/engram shard loading** NEW 75

Load external PLE/engram shards from separate GGUF files using `--shard` or `--ple`, splitting large model components across files.

Load a model with external PLE shards split into separate GGUF files, useful for distributing large specialised components.

```
$ llama serve -hf ggml-org/Qwen3.5-0.8B-GGUF --ple ./expert-shard.gguf
```

— Flags and a runnable loading command are given. v1.0.0

05 **Server-side SSRF/CORS/DNS-rebinding hardening** NEW

65

`llama-server` hardens against SSRF, CORS and DNS-rebinding issues by default; operators can opt back into private media URLs or unauthenticated tool use via `--media-allow-private-urls` or `--allow-unauthenticated-tools`.

— Opt-out flags named but hardening mechanism only summarized. v1.0.0

06 **Speculative decoding API promoted and renamed** BREAKING

65

The NextN/speculative decoding API is promoted from `llama-ext.h` into `llama.h`, and `llama_set_embd_state_src` is renamed to `llama_set_ctx_state_src`; callers must update to the new name.

— Exact header and function names given with a migration note. v1.0.0

07 **Deprecated API functions removed** BREAKING

65

35 deprecated functions, `llama_set_warmup`, `defrag_thold`, and the `--defrag-thold` CLI flag are removed from `llama.h`; code or scripts using them will break.

— Named removed symbols and flag with clear breaking impact. v1.0.0

THINNER COVERAGE BELOW

08 **New text model architecture support** NEW

58

llama.cpp now supports three new text architectures: Kimi-K3.1-Pro (DSA replacing full-attention MLA), Ripple-65B-Preview (`ricursive` dense 65B), and xing4_0 (mHC, MLA, MoE, MTP).

— Architectures named with technical detail but no usage instructions. v1.0.0

09 **Chat organization and editor in web UI** IMPROVED

55

The `llama-server` web UI now organises chats into named groups, adds a live-preview markdown editor for chat input, and gains a `search_chats` tool.

— UI features named but no navigation detail given. v1.0.0

10 **`clip.h` removed for mtmd migration** BREAKING 50

`clip.h` is removed; multimodal code that included it must migrate to the `mtmd` subsystem.

— Names removed header and replacement subsystem, no migration steps. v1.0.0

11 **New vision multimodal model support** NEW 30

llama.cpp now supports MageVL-Cowork and Kimi-K3.1 vision multimodal models.

— Bare model names with no mechanism or usage detail. v1.0.0

12 **Gemma 4 streaming audio input** IMPROVED 30

Gemma 4 now supports streaming audio input via the multimodal subsystem.

— Minimal description, no mechanism or flags given. v1.0.0

└──▷ **BREAKING ON UPGRADE**

! `llama_set_embd_state_src` is renamed to `llama_set_ctx_state_src` in `llama.h`; callers must update to the new name.

! 35 deprecated functions, `llama_set_warmup`, `defrag_thold`, and the `--defrag-thold` CLI flag are removed from `llama.h`; code or scripts using them will break.

! `clip.h` is removed; multimodal code that included it must migrate to the `mtmd` subsystem.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Pinecone

Feedback ⓘ changelog · Nov 14, 2026 CHANGELOG ↗ RANK

Pinecone is a managed vector database that stores and queries embeddings for AI applications.

> Pinecone shipped one-click pg_pinecone deployment templates for Render and Railway, plus a semantic search capability that embeds text columns directly without requiring a vector column.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **One-click pg_pinecone deployment on Render and Railway** NEW

67

A one-click template deploys `pg_pinecone` on Render or Railway, running PostgreSQL 18 with `pg_pinecone` preinstalled and preloaded, requiring no server of your own.

— Names platforms, PostgreSQL version, but no CLI steps changelog-20261114-1e5de11b

THINNER COVERAGE BELOW

02 **Semantic search without vector column in pg_pinecone** NEW

54

`pg_pinecone` v0.12.1 adds the ability to embed a text column with a hosted model, enabling semantic search without needing a dedicated vector column.

— Names version and mechanism, lacks exact syntax changelog-20261114-1e5de11b

WAS THIS USEFUL? 👍 👎

uteke

Feedback ⓘ v0.24.0 NOTES ↗ RANK

Uteke is a long-term memory for AI agents (MCP). 2% recall@5 on LongMemEval-S.

> uteke's v0.24.0 release adds a full skill package management system plus document similarity/duplicate detection, topic-based room search, and alias-preserving room renames, alongside several breaking changes to MCP error handling, HTTP status codes, skill-file access, and request concurrency limits.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 **Skill package library and management** NEW 90

Index a local skill directory with `uteke skill index <dir>`, then search, list, show, find similar, lint, and approve skills via CLI (`uteke skill search`, `uteke skill list`, `uteke skill show`, `uteke skill similar`, `uteke skill lint`, `uteke skill approve <slug>@<hash>`), HTTP API (`POST /skill/search`, `POST /skill/list`, `POST /skill/show`, `POST /skill/similar`), and MCP tools (`uteke_skill_search`, `uteke_skill_list`, `uteke_skill_show`, `uteke_skill_similar`).

Index a local skill package so agents can recall only the skills a task needs, without reading every document.

```
$ uteke skill index ./skills
```

— Names full set of CLI, HTTP and MCP surfaces plus a runnable example v0.24.0

02 **Document similarity and duplicate detection** NEW 90

Adds `uteke doc similar <slug>` (or `--text` / `--file` for an unstored draft), `uteke doc duplicates` (with optional `--subtree` or `--namespace`), `POST /doc/similar`, and the MCP tool `uteke_doc_similar`, configurable via new `[doc_similarity]` keys `similar_threshold` and `duplicate_threshold` in `uteke.toml` (defaults 0.80 and 0.92).

Tune when two documents are flagged as duplicates or as similar, to reduce false positives in large stores.

```toml
[doc_similarity]
similar_threshold = 0.85
duplicate_threshold = 0.95
```

— Full surface list, defaults, and a runnable config example v0.24.0

03 **New room capabilities: search, naming convention, and rename alias**

NEW 88

Adds `uteke room search <QUERY>` (with optional `--limit N` or `--namespace NS`), `POST /room/search`, and the MCP tool `uteke_room_search`, fusing keyword and semantic signals so rooms with no title are still found by their content; enforces the `category:title` room-id convention (for example `project:uteke`) via `[rooms] require_category_title = true` in `uteke.toml`, returning a warning by default and a 400 `invalid_room_id` error when enabled; and room rename now records the old id as an alias via `uteke room rename`, `POST /room/rename`, and `uteke_room_rename`, so agents and config files that reference the old id continue to work.

Find rooms relevant to a topic without reading a full room list — useful when hundreds of rooms exist.

```
$ uteke room search "kubernetes deployment" --limit 5 --namespace default
```

— Named CLI/HTTP/MCP surfaces plus a runnable example Config key and exact error behavior named, no command example Explains mechanism and benefit, names surfaces, no example

v0.24.0

04 **Skip deprecated copies during import** NEW 81

Skip deprecated copies during import with `uteke import --skip-deprecated-copies` or the boolean field `skip_deprecated_copies` in the body of `POST`

`/import` (JSON/JSONL only), counting skipped rows in `duplicates_skipped`.

— Exact flag and field named with scope (JSON/JSONL only) v0.24.0

05 **Concurrency and size limits on remote requests** BREAKING

75

Remote requests that embed a client string are capped at 32768 characters with at most 2 running concurrently; additional requests are refused with HTTP 429 (`busy`). `POST /doc/search` in `hybrid` or `semantic` mode now takes a slot.

— Exact limits, error code, and affected endpoint named v0.24.0

06 **Room metadata on recall and search hits** IMPROVED 68

Recall, search, and room-recall hits now include an optional `rooms` list of `{room_id, room_name}` on `POST /recall`, `POST /search`, `POST /room/recall`, MCP tools `uteke_recall`, `uteke_search`, and `uteke_room_recall`, and the CLI `--json` output.

— Names every affected endpoint and field but no usage example v0.24.0

07 **Skill files no longer accessible as documents** BREAKING

63

Skill files are no longer readable or editable as documents: `doc get`, `POST /doc/get`, `uteke_doc_get`, `doc update`, and the children of a skill file answer `not found`.

— Names every affected command, endpoint and tool v0.24.0

08 **MCP tool errors return as tool results** BREAKING 60

A failed MCP tool call is now a tool result with `isError: true` instead of a JSON-RPC `-32603` error, and client error text no longer starts with `Failed:`.

— Describes exact before/after format change v0.24.0

09 **HTTP errors now respond by error class** BREAKING 60

HTTP routes that previously answered every non-validation failure with 500 now respond by error class (404, 429, 409, 400), including `POST /recall`.

— Names specific status codes and one affected endpoint v0.24.0

THINNER COVERAGE BELOW

10 **Standalone uteke-privacy crate and Agent Plugin** NEW 45

Publish or consume the `uteke-privacy` crate (released as a standalone crate containing the secret-detector logic) alongside the `uteke-mcp` portable Agent Plugin at `integrations/agent-plugin`.

— Names the crate and plugin path but no usage detail `v0.24.0`

└──▷ BREAKING ON UPGRADE

- ! A failed MCP tool call is now a tool result with `isError: true` instead of a JSON-RPC `-32603` error, and client error text no longer starts with `Failed:` .
- ! HTTP routes that previously answered every non-validation failure with 500 now respond by error class (404, 429, 409, 400), including `POST /recall` .
- ! Skill files are no longer readable or editable as documents: `doc get` , `POST /doc/get` , `uteke_doc_get` , `doc update` , and the children of a skill file answer not found .
- ! Remote requests that embed a client string are capped at 32768 characters with at most 2 running concurrently; additional requests are refused with HTTP 429 (`busy`). `POST /doc/search` in `hybrid` or `semantic` mode now takes a slot.

WAS THIS USEFUL?  

rabitq-library

Feedback ⓘ v0.8.5 NOTES ↗ RANK

Official library for RaBitQ vector quantization and approximate nearest-neighbor search, in C++ and Python.

> rabitq-library adds thread-based parallelism control for centroid assignment and begins transitioning toward v0.9 API naming with backward-compatible deprecation warnings.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Parallel centroid assignment via num_threads** NEW 80

A `num_threads` argument was added to `IvfIndex.set_centroids` to control parallelism and speed up centroid assignment on multi-core machines, e.g. index.set_centroids(centroids, num_threads=8).

Assign pre-computed centroids faster on a multi-core machine by parallelising the work across threads.

python

```
index.set_centroids(centroids, num_threads=8)
```

— Names exact parameter and shows runnable code v0.8.5

THINNER COVERAGE BELOW

02 **v0.9 API naming with deprecation warnings** DEPRECATED 40

The library now accepts v0.9 index and k-means API names alongside the older v0.8 spellings, emitting deprecation warnings to guide migration away from the older forms.

— No specific old/new names given v0.8.5

WAS THIS USEFUL? 👍 👎

liliang-cn CortexDB

Feedback ⓘ v2.151.1 NOTES ↗ RANK

CortexDB is a local-first AI agent memory with a knowledge graph, ontology constraints, and traceable reasoning—all in one SQLite file.

> CortexDB tightened its query planning and recall pipeline, adding a timeout-bound planner, automatic credential redaction on save, and duplicate-hit suppression on recall.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 **Timed query planner with lexical fallback** IMPROVED 40

CortexDB's live query planner now gets a 5-second window to produce an optimized plan before the lexical fallback plan takes over.

— Names mechanism and timeout but no config surface v2.151.1

02 **Credential redaction on save** NEW 40

CortexDB now redacts credentials in saved text, replacing them with `[REDACTED]` so secrets are never persisted to the brain.

— Describes behavior and replacement token, no config exposed v2.151.0

03 **Duplicate-hit suppression on recall** IMPROVED 35

CortexDB's recall now skips injecting a hit the current session was already given, preventing duplicate context from being re-injected.

— Explains behavior but no flag or metric given v2.151.0

WAS THIS USEFUL? 👍 👎

knowhere

Feedback ⓘ v1.2.56 NOTES ↗ RANK

Knowhere extracts, parses, and outputs structured chunks ready for AI Agents and RAG.

> Knowhere's latest release adds specialized parsing for engineering drawing sheets and extends its media capabilities to speech, audio, and video files with billing and citation features.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Speech, audio, and video parsing** NEW 53

Adds parsing support for speech, audio, and video files with duration billing, chapter outlines, and time-range citations.

— Names mechanism and outputs but no concrete flags or endpoints v1.2.56

02 **Atlas drawing-sheet title-block skills** NEW 42

Knowhere adds atlas drawing-sheet skills so table-of-contents calibration and fine tagging can use title-block drawing names and numbers when a page has no normal heading.

— Describes behavior but no config or API surface named v1.2.56

WAS THIS USEFUL? 👍 👎

◆ Document Parsing

pdf-mcp

Feedback ⓘ v3.8.0 NOTES ↗ RANK

Pdf-mcp is an agentic RAG over one PDF or a whole folder, with page-level evidence to check every answer: hybrid search, selective page reads, tables, OCR, charts, and CJK layouts.

> pdf-mcp adds OCR support so scanned PDF folders become searchable, and introduces a stricter confidence signal that flags when a corpus doesn't actually cover a query.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **OCR-backed corpus search for scanned PDFs** NEW 88

`pdf_corpus_warm` gains a new `ocr` field: `ocr="auto"` (the default when Tesseract is available) enables OCR, `ocr="off"` preserves prior behaviour, and `ocr_lang` sets the recognition language; `warm_complete` returns `true` once every page has been processed.

Warm a folder of scanned annual reports so corpus search can find answers across image-only PDFs.

```
$ pdf_corpus_warm(path="/data/annual-reports", ocr="auto",
ocr_lang="eng")
```

— Names exact field, values, and completion signal with a runnable example. v3.8.0

02 **Low-confidence flag on search results** BREAKING 82

`pdf_search` results now include `all_results_low_confidence: true` when the corpus doesn't appear to cover a query; the semantic confidence cutoff was raised from 0.5 to 0.65, and auto mode no longer treats a single-term page match as confident. This means off-topic queries that previously returned results now get flagged, so agents checking this field will report `not found` more often than in v3.7.0.

Guard against hallucinated answers by checking whether the corpus actually covers the question before returning a result to the user.

```python
results = pdf_search(query="capital expenditure 2022")
if results.get("all_results_low_confidence"):
    return "The documents don't appear to cover this question."
```

— Gives exact field, threshold change, and behavioural before/after. v3.8.0

└──▷ **BREAKING ON UPGRADE**

! Off-topic queries now return results flagged `low_confidence`; agents that check this flag will report `not found` more often than in v3.7.0.

WAS THIS USEFUL?  

kordoc

Feedback ⓘ v4.24.75 NOTES ↗ RANK

Kordoc is a 한국 문서를 읽고·고치고·새로 쓰는 AI 문서 도구 — HWP·HWPX·PDF·오피스·스캔(OCR)을 Markdown으로, 신구대조·서식 보존 편집·Markdown→HWPX 작성·HWP→HWPX 변환·양식 채우기·공문서 검수까지. 라이브러리·CLI·MCP | Read, diff, edit, write and convert Korean documents (HWP, HWPX, PDF, Office, scans) — library, CLI and MCP server

> Kordoc's HWPX generator gains fine-grained control over two-column layout spacing and divider lines.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Column gap and divider controls for HWPX generation** NEW

80

Generated HWPX documents can now control two-column spacing and divider lines via CLI flags `--column-gap` and `--column-line`, API options `page.columnGapMm` and `page.columnLine`, or MCP parameters `column_gap_mm` and `column_line`.

Generate a two-column exam sheet HWPX with a 3 mm column gap and a thin 0.12 mm black divider line.

```
$ npx kordoc generate 시험지.md --preset 보고서 --column-gap 3 --column-line -o 시험지.hwpx
```

— Names exact flags, API keys and MCP params with a runnable example. v4.24.75

NEW SUPPORTED PLATFORM Image file input: TIFF, GIF, AVIF, BMP, HEIC/HEIF, PCX, DCX

UPDATED SUPPORTED PLATFORM BMP decoder extended to 1/4/8-bit palette, RLE8, RLE4, 16-bit, V4, V5, OS/2

v4.24.74 NOTES ↗ STABLE

kordoc now normalizes images for LLM APIs (Claude, OpenAI, Gemini) automatically, converting BMP/GIF/TIFF/AVIF/HEIC/PCX/SVG to PNG unless `imageFormat` is set.

└──▷ **GET THIS VERSION** ▾

npm

```
$ npm install -g kordoc@4.24.74
```

```git
$ git clone --branch v4.24.74
https://github.com/chrisryugj/kordoc.git
# already have the repo? check out this version:
$ git checkout v4.24.74
```

named in the project's README · checked 14 Nov Wrong command?

Normalize images for LLM APIs without specifying `imageFormat`: PNG, JPEG, and WebP pass through as-is for Claude, OpenAI, and Gemini compatibility, while BMP, GIF, TIFF, AVIF, HEIC, PCX, SVG, and single-bitmap WMF are converted to PNG automatically; unconvertible images (vector WMF, EMF) are preserved and reported as `SKIPPED_IMAGE`.

Select pages from multi-page TIFF and DCX files using the `pages` option; each page is split individually and passed to OcrProvider as PNG.

Unsupported image formats (JPEG 2000, JPEG XL, PSD, WMF, EMF, EPS, DjVu) now return `UNSUPPORTED_FORMAT` with the format name in the error message.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ updated · Nov 14, 2026

NOTES ↗ RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith expanded LLM Gateway observability and Bedrock authentication, moved thread automations onto SmithDB queries, added new GPT-5.6 chat models, sandbox diagnostics, and several networking and export fixes.

└──▷ WHAT SHIPPED · 8 FEATURES most completely described first ? what's the number?

01 **LLM Gateway enhancements** NEW 77

The LLM Gateway now returns an `X-LangSmith-Gateway-Metadata` response header describing gateway routing, limits, guard activity, and upstream errors, which clients can attach to their own traces as `ls_gateway_info` metadata, and trace details now surface LLM Gateway outcomes, policy limits, selected models, and fallback attempts in a dedicated Gateway section, with errored runs blocked by the gateway showing a Blocked badge in the trace tree; it also now authenticates to Amazon Bedrock using an assumed IAM role or AWS access keys instead of requiring a static Bedrock API key, with existing Bedrock API keys still taking precedence when multiple credential types are configured.

— Names header, metadata key, and UI section but no exact navigation steps Explains mechanism and precedence but no config key named

snapshot-20261114

02 **SmithDB-backed thread automations** BREAKING 65

Thread annotation-queue and webhook automations now require SmithDB queries, replacing the previous run-filter approach, and show thread filters instead of run filters. Automations grouped by thread can now add one annotation queue item per matched conversation so reviewers can inspect the whole thread instead of only one run. Existing automations using run filters or the default `Is Trace = true` filter must be migrated.

— Describes behavior change and migration need but no exact query syntax

snapshot-20261114

03 **CSV export for colon-containing metadata keys** IMPROVED 62

Datasets whose input, output, or metadata keys contain a colon can now be exported to CSV, unblocking datasets built from runs with feedback stored under `source_run_feedback:<key>` metadata keys.

Export a dataset whose metadata keys contain colons (e.g. `source_run_feedback:score`) to CSV for offline analysis.

> 📍 In the LangSmith UI, open the dataset, click Export, and choose CSV. Datasets with `source_run_feedback:<key>` metadata keys now export correctly.

— Names exact metadata key format and includes a usage example `snapshot-20261114`

THINNER COVERAGE BELOW

04 **Turns view deprecated in thread detail** DEPRECATED 55

The Turns view in thread detail is deprecated and will be removed from LangSmith Cloud on October 31, 2026, and from self-hosted in v17; use the Messages or Details view instead.

— Gives exact removal dates and migration path `snapshot-20261114`

05 **WebSocket query string preserved in sandbox URLs** IMPROVED 54

WebSocket connections through a sandbox service URL now reach the service with their query string intact, unblocking dev servers that authenticate WebSocket connections with a query parameter such as Vite's HMR token.

— Names concrete use case (Vite HMR token) but no flag or config `snapshot-20261114`

06 **New GPT-5.6 models in LangSmith Chat** NEW 37

LangSmith Chat now offers GPT-5.6 Sol, GPT-5.6 Terra, and GPT-5.6 Luna alongside existing OpenAI models.

— Names the new models but no further detail `snapshot-20261114`

07 **Sandbox resource usage history** NEW 36

Sandboxes now retain recent CPU, memory, and network usage history to help diagnose resource consumption and performance bottlenecks.

— States purpose but not retention window or access method `snapshot-20261114`

08 **IPv4/IPv6 dual-stack support** NEW 23

LangSmith services now accept both IPv4 and IPv6 connections.

— Bare statement with no scope or configuration detail `snapshot-20261114`

└──▷ **BREAKING ON UPGRADE**

! The Turns view in thread detail is deprecated and will be removed from LangSmith Cloud on October 31, 2026, and from self-hosted in v17; use the Messages or Details view instead.

! Thread annotation-queue and webhook automations now require SmithDB queries; existing automations using run filters or the default `Is Trace = true` filter must be migrated.

WAS THIS USEFUL?  

Harbor Framework Harbor

Feedback ⓘ v0.26.0 NOTES ↗

RANK

Harbor is a framework for evaluating and improving AI agents through structured testing, metrics collection, and performance monitoring.

> Harbor v0.26.0 adds four new sandbox environment providers, an MCP tool server exposing Harbor operations, synchronous/asynchronous API clients, and new agent capabilities, alongside two breaking changes to runner env vars and job submission endpoints.

└──▷ **WHAT SHIPPED · 9 FEATURES** most completely described first ? what's the number?

01 New sandbox environment providers NEW

80

Adds four new environment providers: Snowflake Cortex Sandboxes (selectable via `--env snowflake`), Azure Container Apps Sandboxes for running Harbor jobs, `boxd` (requires `boxd>=0.3.0`, selectable with `--env boxd`, supporting a chosen region), and Superserve sandbox. Also adds typed options and schema discovery for environment providers generally, and a `namespace` option to the HuggingFace (`hf-sandbox`) environment provider.

Run a benchmark inside Snowflake-managed sandboxes instead of a local Docker environment.

```
$ harbor run --dataset terminal-bench@2.0 --agent claude-code --model anthropic/claude-opus-5-5 --n-concurrent 10 --env snowflake
```

Run a benchmark on boxd cloud machines in a chosen region.

```
$ harbor run --dataset terminal-bench@2.0 --agent claude-code --model anthropic/claude-opus-5-5 --n-concurrent 10 --env boxd
```

— Names flags and configs but little mechanism per provider v0.26.0

02 Token usage and cost reporting in copilot-cli NEW

70

The `copilot-cli` agent now reports token usage and cost via the new `--usage-output-file` flag.

— Exact flag named, ready to use v0.26.0

03 Job submission endpoint moved

65

BREAKING

Job submissions on the server now go to `/jobs/submit`; callers targeting the old endpoint must update their URLs.

— Exact new endpoint path given, old path unspecified v0.26.0

THINNER COVERAGE BELOW

04 **on_error judge policy in rewardkit** NEW 55

Adds an `on_error` judge policy to rewardkit that rejects out-of-range scores.

— Names policy and behavior but no config example v0.26.0

05 **ACP runner environment variable rename** BREAKING 55

ACP runner control environment variables are renamed to the `ACP_RUNNER_*` prefix; existing scripts or configs using the old names will break.

— Names prefix but no migration list of old vars v0.26.0

06 **MCP tool server for Harbor operations** NEW 50

Harbor's server now exposes Harbor operations and artifact downloads as MCP tools, automatically generated from API routes using FastMCP.

— Describes mechanism but no exact endpoint or usage shown v0.26.0

07 **Synchronous and asynchronous API clients** NEW 50

Adds synchronous and asynchronous Harbor API clients available via the new `client` module.

— Names the module but no usage example given v0.26.0

08 **New model support in native and openclaw agents** NEW 45

Adds support for NeMo Fabric 0.6 in the native agent and Gemini models in the `openclaw` agent.

— Names models and agents but no usage shown v0.26.0

09 **Scratch emptyDir volumes in cwsandbox** NEW 35

Adds scratch `emptyDir` volumes to `cwsandbox`.

— Bare mention with no further detail v0.26.0

└──▷ BREAKING ON UPGRADE

! ACP runner control environment variables are renamed to the `ACP_RUNNER_*` prefix — any existing scripts or configs using the old names will break.

! Job submissions on the server now go to `/jobs/submit` — callers targeting the old endpoint must update their URLs.

WAS THIS USEFUL? 👍 👎

evalscope

Feedback ⓘ v1.15.0 NOTES ↗ RANK

Evalscope is a streamlined and customizable framework for efficient large model (LLM, VLM, AIGC) evaluation and performance benchmarking.

> evalscope's v1.15.0 release adds nine new safety and agent benchmarks, remote sandbox execution via E2B and Aliyun Agent Sandbox, OpenAI-compatible text-to-image evaluation, and a warmup-exclusion flag for performance testing, alongside breaking changes including a Python 3.11 requirement and a new shared scoring version.

└──▷ **WHAT SHIPPED · 13 FEATURES** most completely described first ? what's the number?

01 Warmup exclusion flag for performance testing NEW

85

The new `--exclude-warmup-overlap` flag excludes warmup transition requests from closed-loop performance measurements; requests sent while warmup is still in flight are recorded separately and omitted from throughput/latency statistics.

Measure closed-loop throughput without warmup noise — useful when benchmarking a freshly started inference server where the first requests overlap with warmup.

```
$ evalscope perf --model your-model-name --api-url
http://localhost:8000/v1 --dataset-path your_dataset.jsonl --
exclude-warmup-overlap
```

— Named flag with mechanism and a runnable example command v1.15.0

02 Judge JSON output configuration and scoring version bump NEW

80

Configure custom judge JSON output via `judge.models[*].generation_config` for optional JSON mode in judge model settings. The shared scoring version is now `v1.3`; judge replies must satisfy their declared JSON schema or scored coverage is reduced.

— Config key and version number given, no worked example v1.15.0

03 New safety and agent benchmarks NEW

75

evalscope adds SALAD-Bench (open-ended and multiple-choice), XSTest (over-refusal), LongPIBench (long-context prompt injection), MASK (honesty under pressure), SLEIGHT-Bench (attack detection in coding-agent transcripts), MM-SafetyBench++ (answering, refusal behavior, and response quality across safe and unsafe multimodal contexts), AgentDojo (prompt-injection), AgentHarm (harmful and benign tasks), unguided Cybench CTF tasks with tool trace recording and cached

regrading, `SWE-Atlas QnA` for real-repository code comprehension reporting task resolve rate and rubric pass rate separately, native zero-shot `LawBench` covering 19 tasks and 9,500 samples (excluding document proofreading task `2-1`), and `FewCLUE` `CLUEWSC` for Chinese coreference resolution.

— All benchmark names and scope named, no run commands given v1.15.0

04 **Remote sandbox support for code and agent benchmarks** NEW

70

evalscope now supports running supported code and native-agent benchmarks on official E2B or Aliyun Agent Sandbox in addition to the default Docker engine; ordinary Python evaluations on Aliyun can use the built-in `base` template automatically. Sandbox extras now require `ms-enclave[docker,e2b]>=0.0.9`.

— Named sandboxes and dependency version, no command shown v1.15.0

THINNER COVERAGE BELOW

05 **KV cache hit rate metrics in performance testing** NEW

55

evalscope adds request and input-token coverage metrics for KV cache hit rates in performance testing, with warnings emitted when `cached_tokens` is absent; unknown values are no longer counted as misses.

— Mechanism and field named, but no usage example v1.15.0

06 **Text-to-image API evaluation support** NEW

55

evalscope now supports evaluating OpenAI-compatible text-to-image generation services, accepting base64 or URL image results; local Diffusers dependency checks apply only to local inference paths.

— Behavior described but no command or config shown v1.15.0

07 **Legacy versioning for VLMs Are Biased benchmark** IMPROVED

50

evalscope adds an explicit `v1-legacy` entry and separate cache identity for the VLMs Are Biased benchmark, which now defaults to revision-pinned v2 data.

— Names the versioning scheme but no run example v1.15.0

08 **Web dashboard run comparison and task settings** IMPROVED

45

evalscope's web dashboard now supports selecting specific historical runs for comparison, including multiple runs of the same model, and adds reasoning effort, thinking budget, thinking controls, and Extra Config options for top-level task settings.

— UI capabilities named without exact navigation path v1.15.0

09 **Prompt fixes requiring new predictions for three benchmarks** 45

BREAKING

Input/prompt fixes to `ArxivRollBench`, `AA-LCR`, and `CrossNER` require generating new predictions.

— Named benchmarks with clear required migration action v1.15.0

10 **Correct column in evaluation reports** NEW 40

evalscope adds a Correct column alongside Num in evaluation reports for binary, higher-is-better metrics, showing raw correct-sample counts.

— Small, self-contained report field addition v1.15.0

11 **Capped backoff retries for OpenAI-compatible requests** 40

IMPROVED

OpenAI-compatible Chat requests now use capped exponential backoff for retries and honor a numeric Retry-After header value as a minimum wait time.

— Mechanism named but not user-configurable v1.15.0

12 **Python 3.11 minimum requirement** BREAKING 40

evalscope now requires Python 3.11 or newer, up from the previous 3.10+ minimum.

— Clear version bump with obvious upgrade action v1.15.0

13 **Collection cache invalidation on version changes** BREAKING 35

Collection caches are now invalidated when member versions or configurations change; older collection caches require regrading.

— States behavior change but no concrete command v1.15.0

└──▷ **BREAKING ON UPGRADE**

! Python 3.11 or newer is now required (previously 3.10+).

! Sandbox extras now require `ms-enclave[docker,e2b]>=0.0.9`.

! The shared scoring version is now `v1.3`; judge replies must satisfy their declared JSON schema or scored coverage is reduced.

! Collection caches are invalidated when member versions or configurations change; older collection caches require regrading.

! ArxivRollBench, AA-LCR, and CrossNER input/prompt fixes require generating new predictions.

WAS THIS USEFUL?  

Langfuse

Feedback ⓘ v4.80.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse's v4.80.0 release adds Google AI Studio and Microsoft Azure as AI gateway providers, expands gateway key management and OpenRouter connectivity, introduces topic-based trace filtering, and makes skill size limits configurable.

└──▷ WHAT SHIPPED · 7 FEATURES most completely described first ? what's the number?

01 Gateway key scoping, editing, and masking NEW

65

Gateway keys can now be restricted to specific model patterns so teams can limit which models a key may access, existing gateway keys can be edited after creation to change their name, metadata, and allowed models, and API keys are now masked in the UI so credentials are not exposed in plain text.

Scope a gateway key to a subset of models to enforce access boundaries for a team or service.

> 📍 In the Langfuse UI, go to Connections & Models › Gateway Keys, open or create a key, and set the allowed model patterns in the Restrict to model patterns field before saving.

— Three named gateway-key capabilities with a UI walkthrough for one. v4.80.0

02 OpenRouter custom base URLs and connection status IMPROVED

65

The AI gateway now supports custom base URLs for `OpenRouter` connections and shows a connection test status indicator in the AI gateway connection dialog footer.

— Names OpenRouter and the exact UI element, but no example given. v4.80.0

03 Configurable skill size limits and draft persistence IMPROVED

60

Skill size and file count limits are now configurable via settings, and unsaved skill drafts are kept in `sessionStorage` so work survives a page reload.

— Names sessionStorage mechanism but no exact settings path. v4.80.0

THINNER COVERAGE BELOW

04 Google AI Studio and Microsoft Azure gateway providers NEW

55

Langfuse added Google AI Studio and Microsoft Azure as AI gateway providers, so LLM calls to these providers can be routed through and logged by Langfuse's AI gateway.

Connect Google AI Studio as an AI gateway provider so LLM calls are routed and logged through Langfuse.

> 📍 In the Langfuse UI, go to Connections & Models › Connections, click `Add connection`, select `Google AI Studio`, enter your API key, and confirm the connection test status in the dialog footer.

— Names two providers and gives UI steps, but no mechanism detail. v4.80.0

---

05 **Copy session link in Halo session menu** NEW 55

Added a `Copy session link` option to Halo's session menu, with session links hidden when HIPAA mode is active.

— Names the exact menu and conditional HIPAA behavior. v4.80.0

---

06 **Topic-based trace filtering and visualization** NEW 45

The tracing table can now be filtered by topic, with topics shown as a column in the trace list, and the topics list and treemap views now have a resizable split between panes.

— Describes three related topic features without a usage path. v4.80.0

---

07 **Remove prompt version labels via API and MCP** NEW 30

Labels can now be removed from a prompt version via the API and MCP.

— Names the surfaces but no endpoint, command, or mechanism. v4.80.0

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ updated · Nov 14, 2026 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust shipped voice-agent audio capture, a sweeping breaking overhaul of LiveKit and Pipecat tracing taxonomy, a Terraform provider update adding Patterns view management alongside new required secret/value versioning and resource removals, plus new spans and metadata across Strands, AI SDK, Temporal, OpenAI and Bedrock integrations.

└──▷ **WHAT SHIPPED · 7 FEATURES** most completely described first ? what's the number?

01 LiveKit Agents and Pipecat tracing overhauls (breaking)

90

BREAKING

LiveKit Agents tracing now requires `livekit-agents` v1.8.9 to v1.8.10 and renames spans (`livekit_agent_session` → `session`, `user_speaking` → `user_speech`, `agent_speaking` → `agent_speech`, `stt_processing` → `stt`, `tts_request` → `tts`, `llm_request_run` → `llm_response`), names tool spans after the function tool instead of `function_tool`, removes the `vad_endpointing` and `eou_detection` spans and per-component `livekit_metrics`, renames TTS time-to-first-token to `time_to_first_audio`, and attaches audio to the `session` span instead of speech spans — update saved queries, filters, and scorers accordingly. Pipecat tracing similarly renames spans (`pipecat_pipeline` → `session`, `pipecat_llm_response` → `llm_response`, `stt_transcription` → `stt`, `tts_response` → `tts`, `user_speaking` → `user_speech`), records audio on the `session` span instead of separate WAV or PCM files, moves token usage from `llm_response` to provider spans (set up the provider integration or use auto_instrument()), removes the `BraintrustPipecatObserver` export in favor of setup_pipecat(), wrap_pipeline_worker(), or auto_instrument(), and drops the `trace_turns` argument from setup_pipecat().

— Every renamed, removed span and version pin is named. Names every renamed span, removed export and migration path.

snapshot-20261114

02 Audio capture for voice-agent traces NEW

83

Set `BRAINTRUST_CAPTURE_USER_AUDIO_ATTACHMENTS` or `BRAINTRUST_CAPTURE_AGENT_AUDIO_ATTACHMENTS` to record caller and agent audio on voice-agent trace spans. Installing `braintrust[audio]` compresses recordings to Opus/Ogg (requires Python 3.12+ for PyAV); without it, Braintrust falls back to raw PCM.

Attach audio from both caller and agent to voice-agent trace sessions, with Opus/Ogg compression, so you can replay conversations alongside LLM spans.

```
$ export BRAINTRUST_CAPTURE_USER_AUDIO_ATTACHMENTS=true
export BRAINTRUST_CAPTURE_AGENT_AUDIO_ATTACHMENTS=true
pip install braintrust[audio]
```

— Exact env vars, install flag and fallback behavior all given. snapshot-20261114

03 **Continue traces with `--parent` in `bt trace run`** NEW

78

Pass `--parent` to `bt trace run` with the value returned by an SDK span's export() to attach a coding-agent session below that span instead of starting a new trace.

— Exact flag, source value and resulting behavior given. snapshot-20261114

04 **Patterns view management in Terraform provider** NEW

73

The `braintrust_view` Terraform resource now accepts `"patterns"` as its `view_type`, letting saved Patterns views be managed as Terraform code alongside other Braintrust resources.

Manage a Patterns saved view in Terraform so it is version-controlled alongside your other Braintrust resources.

```hcl
resource "braintrust_view" "my_patterns_view" {
  name      = "My Patterns View"
  view_type = "patterns"
}
```

— Names resource and field, includes a runnable config example. snapshot-20261114

05 **New spans and metadata across Strands, AI SDK and Temporal**

NEW

63

Braintrust adds Strands decision spans, AI SDK `ai.evaluate` spans, and `temporal.original_execution_run_id` on Temporal workflow spans. Temporal activity spans now record the retry attempt as `temporal.activity_attempt` and mark cancellations with `temporal.cancelled: true`.

— Names every new span/field but gives no usage guidance. snapshot-20261114

THINNER COVERAGE BELOW

06 **Span instrumentation improvements for OpenAI Responses and Bedrock**

IMPROVED

57

OpenAI Responses image-generation tool spans now include SVG image attachments and record `svg_usage` in metadata instead of output; Bedrock `invoke_model` spans now record token metrics, and `invoke_model` embedding spans log `{"count": N}` instead of the full embedding vectors.

— Names the field and change but no further mechanism. Names the endpoint and logging format but lacks usage detail.

snapshot-20261114

---

07 Optional telemetry argument in `bt mcp run` IMPROVED

50

`bt mcp run` no longer requires the telemetry argument, so you can call a Braintrust MCP server tool with only its own arguments.

— Names the command but no further mechanism given. snapshot-20261114

└──▷ 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`; you must add a version to each configured value and increment it to write a new value, because changing only the value no longer updates it, and references to `secret` or `value` 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`.
- ! LiveKit Agents tracing requires `livekit-agents` v1.8.9 to v1.8.10 and does not instrument other versions.
- ! LiveKit span names are renamed: `livekit_agent_session` → `session`, `user_speaking` → `user_speech`, `agent_speaking` → `agent_speech`, `stt_processing` → `stt`, `tts_request` → `tts`, `llm_request_run` → `llm_response`; update saved queries, filters, and scorers that reference old span names.
- ! LiveKit tool spans are now named after the function tool instead of `function_tool`.
- ! LiveKit `vad_endpointing` and `eou_detection` spans are removed, and per-component metrics (`livekit_metrics`) including end-of-utterance and silence-detection data are no longer recorded.
- ! LiveKit TTS time to first audio is renamed from `time_to_first_token` to `time_to_first_audio`.
- ! LiveKit audio now attaches to the `session` span instead of the speech spans.
- ! Pipecat span names are renamed: `pipecat_pipeline` → `session`, `pipecat_llm_response` → `llm_response`, `stt_transcription` → `stt`, `tts_response` → `tts`, `user_speaking` → `user_speech`.
- ! Pipecat audio is now recorded on the `session` span instead of as separate WAV or PCM files on individual spans.

- ! Pipecat token usage is now on provider spans instead of `llm_response`; set up the provider integration or use auto_instrument().
- ! `BraintrustPipecatObserver` is no longer exported; use setup_pipecat(), wrap_pipeline_worker(), or auto_instrument() instead, and remove the `trace_turns` argument from setup_pipecat().
- ! Bedrock `invoke_model` embedding spans now log `{"count": N}` instead of the embedding vectors.
- ! OpenAI Responses image generation tool spans record `svg_usage` in metadata instead of output.
- ! `client.agents`, `client.org_automations`, and their types are removed from the `braintrust.api` REST client.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI Runtime Defense & Guardrails

Doberman-Core

Feedback ⓘ v0.20.0 NOTES ↗ RANK

Doberman-Core is a runtime security tool that blocks unsafe or unintended AI agent inputs, outputs, and tool calls before execution.

> Doberman-Core v0.20.0 adds hook integrations for Claude Code, Codex and Cursor in `doberman dash`, a sandbox-isolation check in `doberman doctor`, a PATH-independent invocation method, a dedicated judge API key, and stronger TOTP replay protection.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Hook integration for Claude Code, Codex and Cursor NEW

75

`doberman dash` now answers `AUTH` prompts coming from Claude Code, Codex and Cursor hooks, and refuses to respond to non-loopback `Host` headers to prevent remote abuse of the dash endpoint.

— Names exact tools and the Host-header refusal mechanism, but no CLI example given

v0.20.0

02 Sandbox check for 2FA secret isolation NEW

60

`doberman doctor` now checks whether Claude Code's sandbox correctly keeps your 2FA secret isolated from the agent, so you can verify this before deploying.

Verify that Claude Code's sandbox is correctly isolating your 2FA secret from the agent before deploying.

```
$ doberman doctor
```

— Clear runnable command and purpose, but mechanism of the check itself unexplained

v0.20.0

03 Dedicated judge API key environment variable IMPROVED

60

The optional judge now prefers `DOBERMAN_JUDGE_API_KEY` over `ANTHROPIC_API_KEY` when both are set, letting users configure a separate key for the judge component.

— Names both env vars precisely but gives no setup example v0.20.0

THINNER COVERAGE BELOW

04 **Module invocation without PATH binary** NEW 50

Doberman-Core can now be run as `python -m doberman` in environments where the `doberman` command is not on PATH, such as a fresh CI container.

Run Doberman-Core in environments where the `doberman` binary is absent from PATH, such as a fresh CI container.

```
$ python -m doberman dash
```

— Exact runnable command given, but it's a minor, thinly described alternative entrypoint

v0.20.0

05 **One-time TOTP code enforcement** IMPROVED 50

TOTP codes are now accepted only once; a reused or older code within its validity window is refused, closing a replay window.

— Explains the security mechanism but offers no command or config to act on v0.20.0

WAS THIS USEFUL? 👍 👎

◆ Model & Agent Protection

ToolHive

Feedback ⓘ v0.65.0 NOTES ↗ RANK

ToolHive runs MCP servers in isolated containers and enforces identity and access policies for local and Kubernetes deployments.

> ToolHive's v0.65.0 release adds workload-identity-based client authentication via SPIFFE X.509-SVIDs, introduces ID-JAG token minting for delegated access, and ties audit logs into distributed tracing.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **ID-JAG token minting and delegate-client config** NEW

65

ToolHive adds ID-JAG (identity-augmented grant) token minting and delegate-client target configuration, including actor-token validator hooks and XAA client/server rules via `oauthproto/idjag`.

— Names concrete module path and sub-features though usage unclear v0.65.0

THINNER COVERAGE BELOW

02 **Clock skew tolerance storage for private_key_jwt** IMPROVED

52

The auth server now accepts caller-supplied storage for clock skew tolerance on `private_key_jwt` assertions.

— Names exact assertion type but lacks usage example v0.65.0

03 **SPIFFE X.509-SVID client authentication** NEW

45

Clients can now authenticate using SPIFFE X.509-SVIDs, enabling workload-identity-based access without shared secrets.

— Names the mechanism but no config or flag given v0.65.0

04 **Trace and span IDs in audit events** NEW

33

Audit events now record trace and span IDs, linking MCP request logs to distributed traces.

— Describes what changed but no mechanism or config detail v0.65.0

WAS THIS USEFUL? 👍 👎

Minimal Software Research minimal

Feedback ⓘ

release-dd7d8cf2 NOTES ↗ RANK

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 added `min login` for GitHub sign-in with keychain-backed git credentials for grant boxes, tightened sealed-box loadout validation, and rounded out box introspection with audit trails, JSON config output, configurable exec detach keys, and cleaner log streaming.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 **GitHub sign-in via `min login`** NEW 80

`min login` signs in to GitHub and stores credentials in the host keychain; grant boxes now get a git credential helper so agents no longer need a pasted PAT.

— Command and mechanism named but no flags documented. release-dd7d8cf2

02 **Tamper-evident audit trail via `min box audit`** NEW 65

`min box audit --parent <box>` reviews the tamper-evident audit trail for a box, merging its parent chain, to verify no decisions were edited.

Review the tamper-evident audit trail for a box, merging its parent chain, to verify no decisions were edited.

```
$ min box audit --parent my-agent-box
```

— Runnable command shown; no output format detailed. release-dd7d8cf2

03 **Stricter loadout validation for sealed boxes** IMPROVED 60

Sealed boxes now refuse a named loadout, a loadout that sets `loadout_mode`, and a contradicted constraint; sealed boxes also no longer read default loadouts.

— Names config key but gives no migration command. release-dd7d8cf2

04 **Machine-readable JSON output for `min box show`** NEW 60

`min box show -o json <box>` inspects a box's full configuration as machine-readable JSON, useful for scripting or CI diffing.

Inspect a box's full configuration as machine-readable JSON for scripting or CI diffing.

```
$ min box show -o json my-agent-box
```

— Runnable command shown; no schema detail given. release-dd7d8cf2

THINNER COVERAGE BELOW

05 **Clean `min box logs` output for non-terminal stdout** IMPROVED

50

`min box logs` now strips terminal control codes when stdout is not a terminal, keeping piped or redirected output readable.

— Named command but single-line fix with no options. release-dd7d8cf2

06 **Configurable detach keys for exec attach sessions** NEW

45

Detach from an exec attach session using configurable detach keys, implemented via `minimald` T44.

— No actual flag or key syntax provided. release-dd7d8cf2

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ changelog · Nov 14, 2026 CHANGELOG ↗ RANK

The Claude Developer Platform is the API for building applications on Anthropic's Claude models that generate, analyze, and automate work.

> Anthropic shipped two new Fable/Mythos 5.2 models with Fast mode acceleration for the Messages API, a new Compliance API endpoint for security posture, and self-hosted Managed Agents, alongside interoperability constraints governing how the new models handle thinking blocks and computer use.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Interoperability constraints for Fable/Mythos 5.2

BREAKING

83

Claude Fable 5.2 and Claude Mythos 5.2 now read thinking blocks produced by other models (Claude Opus 5.5, Claude Sonnet 5.5, Claude Haiku 5.5) on the Claude API and Claude Models on Google Cloud, preserving reasoning when a conversation switches to either new model, but thinking blocks produced by Fable 5.2 or Mythos 5.2 are silently dropped when the conversation moves to any model outside the Fable 5.2 / Mythos 5.2 pair, including Claude Opus 5.5 and Claude Fable 5.1. The prefix check is enforced for every account regardless of age, so editing the prefix or passing back an earlier turn's thinking block returns a 400 error. Computer use must be declared with the updated toolset; requests using the earlier tool return a 400 error on all platforms except Amazon Bedrock. In agentic workflows, a Fable 5.2 or Mythos 5.2 executor accepts only `claude-fable-5-2` or `claude-mythos-5-2` as its advisor, with any other advisor model returning a 400 error.

— Names exact models, error codes and the Bedrock exception changelog-20261114-e7aeba57

02 Security posture in Compliance API NEW

80

Organizations can query their current Security posture results via the new `GET /v1/compliance/organizations/{organization_id}/security_posture` endpoint, now in beta for Claude Enterprise organizations. Changes to attestations are recorded on the Security posture page in claude.ai Organization settings.

Retrieve the current Security posture results for your Claude Enterprise organization to audit attestation status programmatically.

```
$ curl -X GET
https://platform.claude.com/v1/compliance/organizations/{organization_id}/security_posture \
  -H 'x-api-key: $ANTHROPIC_API_KEY' \
  -H 'anthropic-version: 2023-06-01'
```

— Exact endpoint and method given, plus UI location `changelog-20261114-e7aeba57`

---

03 **`claude-fable-5-2` model** NEW 65

Anthropic adds `claude-fable-5-2`, a new model built for demanding reasoning and long-horizon agentic work, available on the Claude API and Amazon Bedrock (`bedrock-runtime`), with 128k max output tokens, always-on thinking, and pricing of $8 / $40 USD per MTok in/out.

— Specs and pricing given, no usage example `changelog-20261114-e7aeba57`

---

04 **Fast mode for Messages API** NEW 65

Fast mode can be enabled in Messages API requests on the Claude API by setting the flag on your request, with no beta header required, to get up to 2.5x higher output tokens per second; for `claude-fable-5-2` this costs $16 USD per MTok input / $80 USD per MTok output.

— Performance and pricing given, but exact flag name omitted `changelog-20261114-e7aeba57`

THINNER COVERAGE BELOW

05 **`claude-mythos-5-2` restricted model** NEW 50

Anthropic adds `claude-mythos-5-2`, a restricted-access variant of Fable 5.2 available only to organizations verified through programs such as the Cyber Verification Program, with the same capabilities and pricing as Fable 5.2.

— Access gated, no mechanism for verification process `changelog-20261114-e7aeba57`

└─▷ BREAKING ON UPGRADE

! Claude Fable 5.2 and Claude Mythos 5.2 enforce the prefix check for every account regardless of age; code that edits the prefix or passes a thinking block from an earlier turn back in a later turn will receive a 400 error.

! Claude Fable 5.2 and Claude Mythos 5.2 require computer use to be declared with the updated toolset; requests using the earlier tool return a 400 error on all platforms except Amazon Bedrock.

! In agentic workflows, a Claude Fable 5.2 or Claude Mythos 5.2 executor accepts only `claude-fable-5-2` or `claude-mythos-5-2` as its advisor; any other advisor model returns a 400 error.

! Thinking blocks produced by Claude Fable 5.2 or Claude Mythos 5.2 are silently dropped by the API when the conversation moves to any model outside the Fable 5.2 / Mythos 5.2 pair, including Claude Opus 5.5 and Claude Fable 5.1.

WAS THIS USEFUL? 👍 👎

OpenAI

Feedback ⓘ changelog · Nov 14, 2026 CHANGELOG ↗ RANK

OpenAI provides AI models and APIs for building applications that generate and reason with text, images, and audio.

> OpenAI expanded its image generation and editing API with new output formats, adjustable complexity, and markup-guided editing for newer image models.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **New formats and editing controls in image API** NEW

50

OpenAI's image generation and editing API now supports `SVG` output format and animated `GIF` output, accepts adjustable complexity for generated images, and supports markup-guided editing for `GPT-Image-2.5` and `GPT-Image-2.6` models.

— Names formats and models but no parameters, limits, or endpoint details.

changelog-20261114-75a723b5

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

Inference.net

Feedback ⓘ updated · Nov 14, 2026 NOTES ↗

RANK

net is an inference provider built for open weight models. Fast, stable, and affordable.

> Inference.net launched MiniMax M3.1 Flash as a Day 0 serverless model, added AutoEvals for automated model selection, released Schematron V2 for HTML-to-JSON extraction, and introduced an Inference Platform for monitoring and training self-improving models.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **MiniMax M3.1 Flash serverless model** NEW 65

Inference.net added MiniMax M3.1 Flash, a 428B-parameter Mixture-of-Experts model, available via serverless API at $0.10 per million input tokens and $0.50 per million output tokens.

— Pricing and model size given, but no API endpoint or flag named. snapshot-20261114

THINNER COVERAGE BELOW

02 **AutoEvals for model selection** NEW 50

AutoEvals replays production traffic across candidate models and returns quality, cost, and latency metrics to identify the best-fit model for a task.

— Mechanism described but no concrete usage surface given. snapshot-20261114

03 **Schematron V2 HTML-to-JSON models** NEW 45

Schematron V2 is a family of small purpose-built models that convert messy HTML into structured JSON at lower cost and latency than general-purpose LLMs.

— Describes purpose and benefit but no usage details or limits. snapshot-20261114

04 **Inference Platform for model lifecycle** NEW 30

Inference.net introduced an Inference Platform for monitoring, training, and deploying self-improving AI models built from production data.

— Only a brief prose description, no named features or surfaces. snapshot-20261114

WAS THIS USEFUL? 👍 👎

DeepSeek Models

Feedback ⓘ Date: 2026-11-13 NOTES ↗ RANK

DeepSeek Models is the DeepSeek API uses an API format compatible with OpenAI/Anthropic. By modifying the configuration, you can use the OpenAI/Anthropic SDK or softwares compatible with the OpenAI/Anthropic API to access the DeepSeek API.

> DeepSeek rolled out a wave of new API capabilities this window, headlined by a web search tool usable across its chat APIs and standalone endpoints for search, OCR, ASR, and TTS.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 **Web search tool across chat APIs** NEW 75

Enable web search in the Messages, Responses, and Chat Completions APIs by turning on the `deepseek_web_search` tool; with thinking disabled, the `deepseek-flash` model runs web or image searches and returns answers with citations. With thinking enabled alongside `deepseek_web_search`, the model performs deep thinking search, autonomously searching the web and reading links across multiple reasoning steps.

— Names the tool and model but no full request example. Date: 2026-11-13

02 **URL reading via deepseek_read_url tool** NEW 65

Enable URL reading by turning on the `deepseek_read_url` tool so the model detects links in user messages, fetches the pages, and answers with them in context.

— Names the exact tool flag, clear starting point. Date: 2026-11-13

THINNER COVERAGE BELOW

03 **File input for documents and images** NEW 55

Enable file input by uploading PDFs, Office files, images, and more; the service handles plain-text extraction or rich-layout parsing and adds the content to the conversation context.

— Describes mechanism but no upload API name or limits. Date: 2026-11-13

04 **Standalone Search, Fetch, OCR, ASR, and TTS APIs** NEW 45

Use standalone Search and Fetch, OCR, ASR, and TTS APIs as underlying capabilities independent of the conversation APIs.

— Names four API types but no endpoints or signatures. Date: 2026-11-13

05 **Safety check parameter for inputs and outputs** NEW 40

Enable model safety checks via the safety check parameter to get basic risk control: inputs and outputs are reviewed and a refusal marker is returned when a risk is detected.

— Parameter name itself not given, just its purpose. Date: 2026-11-13

06 **Open-sourced demo client for chat history import** NEW

30

Import DeepSeek chat history with an open-sourced demo chat client built on these APIs.

— Mentions the demo exists but no repo or usage detail. Date: 2026-11-13

WAS THIS USEFUL? 👍 👎

Tencent Hunyuan A13B Instruct Feedback ⓘ

retirement scheduled · Nov 13, 2026 MODEL ↗

Hunyuan A13B Instruct is a chat model from Tencent.

OpenRouter lists a retirement date of 15 Nov 2026 for tencent/hunyuan-a13b-instruct, first seen 13 Nov 2026.

WAS THIS USEFUL? 👍 👎

Xiaomi MiMo-V2.5-Pro Feedback ⓘ price cut · Nov 13, 2026

MODEL ↗

MiMo-V2.5-Pro is a chat model from Xiaomi.

Price cut on OpenRouter (xiaomi/mimo-v2.5-pro): $0.48 → $0.435 in, $1.80 → $0.87 out per million tokens (-52%), first seen 13 Nov 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 14B

Feedback ⓘ price increase · Nov 13, 2026

MODEL ↗

Qwen3 14B is a chat model from Alibaba.

Price increase on OpenRouter (qwen/qwen3-14b): $0.10 → $0.12 in, $0.22 → $0.24 out per million tokens (+20%), first seen 13 Nov 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 VL 30B A3B Thinking

Feedback ⓘ

retirement scheduled · Nov 13, 2026 MODEL ↗

Qwen3 VL 30B A3B Thinking is a reasoning model from Alibaba.

OpenRouter lists a retirement date of 15 Nov 2026 for qwen/qwen3-vl-30b-a3b-thinking, first seen 13 Nov 2026.

WAS THIS USEFUL? 👍 👎

StepFun Step 3.5 Flash

Feedback ⓘ

retirement scheduled · Nov 13, 2026 MODEL ↗

Step 3.5 Flash is a reasoning model from StepFun.

OpenRouter lists a retirement date of 15 Nov 2026 for stepfun/step-3.5-flash, first seen 13 Nov 2026.

WAS THIS USEFUL? 👍 👎

OTHER / UNCATEGORIZED

◆ UNCATEGORIZED

Tokenrip

Feedback ⓘ changes since 2026-10-15 CODE ↗

Tokenrip's command-line client.

Tokenrip adds named sign-ins, `rip auth add`, job/output/feedback commands, `rip task list --unassigned`, and workspace thread messaging via `rip message`

Tokenrip adds `rip auth add` command and named sign-ins so several agents can maintain separate sign-ins on a single account, keyed by name rather than account id, with `AMBIGUOUS_IDENTITY` errors surfaced on shared computers.

Tokenrip adds `rip actions` command to show this month's action count against the plan limit.

Tokenrip adds `rip task list --unassigned` flag to list tasks assigned to no one (`assignee=none`).

Tokenrip adds `rip workspace load` now shows the latest changes on each call.

Tokenrip adds `rip workspace load` support for looking up a workspace by its id or slug.

+2 more

└──▷ **BREAKING ON UPGRADE**

! The `rip comment` command is replaced by `rip message` for workspace threads.

! The `--audience`, `--membership`, and `folder share-contents` flags are removed as part of the flat workspace access change.

! `rip trash restore` syntax changed: the `<type>` positional argument is dropped, now `rip trash restore <id>` (the `type` field removed from the restore call).

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · November 14, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue