THE AI TOOLCHAIN

NO. 041

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 29, 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 338 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 60 of 64 tools

▾ // THE BRIEF HIDE

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

Three unrelated things stand out. Anthropic's Claude Opus 5.5 and Sonnet 5.5 land with 1M-token context and always-on adaptive thinking, iro already exposes Opus 5.5 at that window. AutoRound can now

vim ? help

quantize a model with no calibration dataset at all. And Braintrust's `bt trace` strips secrets out of coding-agent spans locally, before they leave the machine.

BUILD

Fine-tune, distill and align without writing your own training loop

torchtune now ships recipes for LoRA/QLoRA, knowledge distillation, DPO with RSO loss and distributed QAT, plus dataset builders for chat, preference and multimodal data and multi-node SLURM runs; Tinker's SDK adds SFT and RL APIs with checkpoint save/resume and playground export. The bespoke glue between a raw dataset and a running job — formatting, packing, restart-after-crash — is now library code.

torchtune · Tinker

DEPLOY

Quantize a model when you have no calibration data and no time to collect it

AutoRound's new NeUQI grid-search flag finds weight scales without a calibration set, removing the step where you scrape representative prompts before you can compress anything. Model Optimizer adds composed HF AutoQuantize recipes, an `iq2_xxs` weight-only format and a DFlash2 speculative-decoding draft variant — drafting tokens with a small model and verifying them in one pass of the big one, for the same output in less wall time.

AutoRound · NVIDIA Model Optimizer

GOVERN

Redact secrets out of agent traces before they reach the vendor

Braintrust's `bt trace` CLI redacts coding-agent spans locally and adds ingestion-side redaction for project logs, so a leaked API key in a tool call never becomes a stored artifact you have to go delete. The MCP TypeScript SDK binds machine-to-machine OAuth credentials and tokens to the specific issuing authorization server, closing the door on a token minted for one server being replayed against another.

Braintrust · MCP TypeScript SDK

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.121 NOTES ↗ RANK

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

> xalgorix's latest releases rework the pentest agent's orchestration and reasoning layers, add a single-agent mode alongside new authentication and rate-limiting controls, and introduce an environment variable to toggle individual specialists.

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

01 Per-specialist toggle via XALGORIX_DISABLED_SPECIALISTS NEW

70

Adds the `XALGORIX_DISABLED_SPECIALISTS` environment variable to the dashboard's environment settings, letting operators selectively disable individual specialists at deploy time, e.g. `XALGORIX_DISABLED_SPECIALISTS=specialist-a,specialist-b`.

Disable specific specialists in the dashboard at deploy time by setting the environment variable in your deployment configuration.

env

```
XALGORIX_DISABLED_SPECIALISTS=specialist-a,specialist-b
```

— Exact env var and runnable example make it directly usable v4.6.114 v4.6.116

THINNER COVERAGE BELOW

02 Applicability-driven agent orchestration overhaul IMPROVED

45

Introduces applicability-driven orchestration for the pentest agent, adding directives, typed dispositions, and honest completion signalling to how the agent decides what to run and reports when it's done.

— Names mechanism (directives, typed dispositions) but no config surface v4.6.119

03 Application, cloud, and API security skill layers rebuilt

IMPROVED

35

Rebuilds the application-security skill as a black-box application reasoning layer, rebuilds the cloud-security skill as an offensive cloud pentest knowledge layer, and consolidates API security skills into evidence-driven offensive playbooks.

— Names three rebuilt skill areas but no mechanism detail per area v4.6.119

04 **Single-agent mode with root-owned methodology** NEW 35

Adds a single-agent mode where the root agent owns the full pentest methodology directly, running with zero specialist sub-agents.

— Describes behaviour but no config or invocation path v4.6.120

05 **Machine-to-machine API tokens and login rate limiting** NEW 30

Adds machine-to-machine API token authentication to the web interface, and adds a configurable login rate limiter.

— Named capabilities but no endpoint, flag, or config key given v4.6.121

v4.6.121 NOTES ↗ LATEST STABLE

xalgorix v4.6.121 adds machine-to-machine API token authentication and a configurable login rate limiter.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.121
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.121
```

v4.6.120 NOTES ↗ STABLE

xalgorix v4.6.120 adds single-agent mode where root owns the full methodology with zero specialists.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.120
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.120
```

v4.6.119 NOTES ↗ STABLE

xalgorix v4.6.119 overhauls agent orchestration and rebuilds application, cloud, and API security reasoning layers.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.119
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.119
```

Introduces applicability-driven orchestration for the agent, with directives, typed dispositions, and honest completion signalling.

Rebuilds the application-security skill as a black-box application reasoning layer.

Rebuilds the cloud-security skill as an offensive cloud pentest knowledge layer.

Consolidates API security skills into evidence-driven offensive playbooks.

v4.6.116 NOTES ↗ STABLE

Adds `XALGORIX_DISABLED_SPECIALISTS` environment variable to control specialist visibility in the dashboard.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.116
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.116
```

└──▷ USE IT

Disable specific specialists in the dashboard at deploy time by setting the environment variable in your deployment configuration.

.env

```
XALGORIX_DISABLED_SPECIALISTS=specialist-a,specialist-b
```

v4.6.114 NOTES ↗ STABLE

xalgorix v4.6.114 adds `XALGORIX_DISABLED_SPECIALISTS` env var to toggle individual specialists on/off.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.114
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.114
```

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

OpenAI Codex CLI

Feedback ⓘ rust-v0.159.0 NOTES ↗ CODE ↗

RANK

OpenAI Codex CLI runs an agent in the terminal that reads, changes, and tests code in local repositories.

> Codex CLI adds an opt-in `instant_interrupt` setting for steering the agent mid-response, paginated thread history via the app-server API, and substantially expanded native Mermaid flowchart syntax support, while removing prompt-suggestion settings and the bundled plugin-creator skill.

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

01 Expanded native Mermaid flowchart rendering IMPROVED

90

Native Mermaid flowchart rendering now supports more edge types, labels, punctuation/semicolons in labels, and node groups, plus solid and dashed edges with directed, undirected, or bidirectional endpoints (`-->` , `---` , `<-->` , `-.->` , `-.-` , `<-.->`), including pipe labels and spaced directed infix labels such as `A -- send --> B` and `A -. retry .-> B` . `&` groups now expand into all source-to-target edges (e.g. `A & B --> C & D` creates four edges), enforced with a 24-edge limit after expansion and a cap of 24 node references per group, and `flowchart` / `graph` directives default to top-down layout when no direction is specified.

– Full syntax, limits and defaults given, directly usable in diagrams. rust-v0.159.0 rust-v0.159.0-alpha.13

02 Paginated thread history in app-server API IMPROVED

67

Adds item-anchor pagination to the `thread/items/list` app-server API so clients can fetch thread history starting from a specific item.

– Names exact endpoint but not pagination parameters. rust-v0.159.0

03 Removal of prompt-suggestions setting BREAKING

60

The `tui.prompt_suggestions` setting and automatic follow-up prompt suggestions have been removed; any config referencing `tui.prompt_suggestions` will silently have no effect.

– Names the exact config key and its silent-no-op behavior. rust-v0.159.0

THINNER COVERAGE BELOW

04 **Instant interrupt during agent responses** NEW 58

Adds an opt-in `instant_interrupt` setting that lets new user input steer Codex mid-response or during long-running code-mode calls, so a user can redirect the agent without waiting for the current turn to finish.

— Names the config key but not where it lives or defaults. rust-v0.159.0

05 **Warnings viewer keep action** NEW 45

Adds a warnings viewer action bound to the `k` key to keep a reviewed warning for later instead of dismissing it.

— Names the keybinding but little else about scope. rust-v0.159.0

06 **Removal of plugin-creator skill** BREAKING 35

The bundled `plugin-creator` skill has been removed.

— Names the removed skill but no migration guidance. rust-v0.159.0

└──▷ BREAKING ON UPGRADE

! The `tui.prompt_suggestions` setting and automatic follow-up prompt suggestions have been removed; any config referencing `tui.prompt_suggestions` will silently have no effect.

! The bundled `plugin-creator` skill has been removed.

rust-v0.159.0 NOTES ↗ CODE ↗ LATEST STABLE

Codex CLI v0.159.0 adds opt-in instant interrupt, paginated thread history, and expanded native Mermaid flowchart support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch rust-v0.159.0
https://github.com/openai/codex.git
# already have the repo? check out this version:
$ git checkout rust-v0.159.0
```

Adds `instant_interrupt` opt-in setting that lets new user input steer Codex mid-response or during long-running code-mode calls.

Adds item-anchor pagination to the `thread/items/list` app-server API so clients can fetch thread history starting from a specific item.

Expands native Mermaid flowchart rendering to support more edge types, labels, punctuation/semicolons in labels, and node groups.

Adds a warnings viewer action (`k`) to keep a reviewed warning for later instead of dismissing it.

Adds a refreshed compact welcome screen and consistent borderless session headers with turn tips for new TUI sessions.

└──▷ **BREAKING ON UPGRADE**

! The `tui.prompt_suggestions` setting and automatic follow-up prompt suggestions have been removed; any config referencing `tui.prompt_suggestions` will silently have no effect.

! The bundled `plugin-creator` skill has been removed.

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.408 CODE ↗ LATEST STABLE

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 v0.1.408 adds a per-request Requests tab and `magpie usage --csv` to export every API call of a period as CSV.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.408 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.408
```

└──▷ TRY IT

Export every request for the current month as CSV to reconcile against your provider's bill.

```
$ magpie usage --csv
```

Adds a Requests tab to the usage UI that lists every API call of the selected period (newest first), with columns for requested model, served model, provider, host, session, effort, status, tokens, and cost.

Adds `Filter` support on the ledger via `agent`, `failed`, and `q` query parameters, letting you narrow the Requests view to a specific agent, failed calls only, or a keyword matched against model, provider, host, or session.

Adds `Swapped` field on each ledger row flagging calls where the reply named a different model than the one requested.

NEW SUPPORTED PLATFORM Claude Sonnet 5.5 added to Bedrock model list

▾ 23 more releases in this issue · 2026-09-29

> v0.1.406 CODE ↗ STABLE
>
> Adds a Routing tab to the tray panel showing the gateway's latest requests, and renames the Usage tab to Allowances.
>
> └──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.406 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.406
```

v0.1.404 CODE ↗ STABLE

magpie v0.1.404 adds Antigravity CLI agent support, live RMB cost display, and model-routing transparency.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.404 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.404
```

Adds Antigravity CLI (`agy`) as an agent on the Agents page, letting users pick any of magpie's models when invoking it through magpie.

Displays usage costs in RMB at a live exchange rate (fetched from `https://open.er-api.com/v6/latest/USD`, cached for 12 hours, falling back to 7.2 CNY/USD) across the Usage page, tray panel, TUI, and CLI when the CNY currency option is enabled in Settings.

Marks any response served by a different model than the one requested, identifying the actual model from the vendor's reply, so routing substitutions are visible to the user.

v0.1.402 CODE ↗ STABLE

Automatic account switching now covers Claude Code (not just Codex), and newly signed-in provider accounts appear in listings immediately without a manual Refresh.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.402 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.402
```

Extends automatic account switching (when an account's allowance is spent) to Claude Code, in addition to the existing Codex support — per-account affinity, backoff, and usage are preserved across switches.

Lists a provider account's models immediately after sign-in during a live session, without requiring a manual Refresh.

Preserves the provider label on user-customised model names in the model list.

v0.1.401 CODE ↗ STABLE

magpie v0.1.401 lets you import existing CLIProxyAPI, Codex CLI, and Claude Code auth files directly as accounts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.401 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.401
```

Supports importing ChatGPT (Codex) and Claude accounts from CLIProxyAPI auth files (`{type: "codex", id_token, access_token, refresh_token, account_id, email, …}` or `{type: "claude", access_token, refresh_token, email, …}`), Codex CLI's `auth.json`, Claude Code's `.credentials.json`, or a bare refresh token, eliminating the need to sign in again.

└──▷ BREAKING ON UPGRADE

! Importing an account from a CLIProxyAPI, Codex CLI, or Claude Code auth file rotates the refresh token — the source file stops working for the original tool after the import.

v0.1.397 CODE ↗ STABLE

Agents page gains per-CLI version display and one-click updates via npm, bun, pnpm, Homebrew, or the agent's own updater.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.397 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.397
```

Adds per-agent CLI version tracking and one-click update on the Agents page, detecting the install method (npm, bun, pnpm, Homebrew, or the agent's own updater such as `claude update`, `codex update`, or `opencode upgrade`) to run the correct update command.

Moves agents to an available model automatically when their provider is disabled, removed, or its last account is signed out.

Adds a safeguard for custom-provider balance tokens that prevents them from being left unqueried, and surfaces actionable messaging when a new-api relay's System Access Token is refused.

v0.1.395 CODE ↗ STABLE

Magpie now routes Amazon Bedrock GPT model requests to the Responses API (`/openai/v1/responses`) when function tools with reasoning are in use.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.395 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.395
```

Automatically retries tool-bearing requests with `reasoning_effort` set to `'none'` on the chat completions path when no Responses API endpoint is available for the model.

v0.1.394 CODE ↗ STABLE

Adds Qoder subscription integration routed through `api3.qoder.sh`'s agent_chat_generation SSE, plus ultra-effort support for routing groups with ChatGPT models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.394 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.394
```

Adds Qoder as a subscription provider, speaking the endpoint its own desktop client uses: `api3.qoder.sh`'s `agent_chat_generation` SSE, wrapped in the COSY envelope (RSA-encrypted key, signed request line), with OAuth device flow authentication (`qoder.com` -> `dt-` device token -> `jt-` job token) and a credential store in `internal/qoder`.

Adds `provider/qoder.go` exposing signed-in account, available models, and sign-in flow; `gateway/qoder.go` translating requests into Qoder bodies and reading SSE back, including tool-call folding.

Routing groups that include a ChatGPT model now offer Codex's `ultra` effort level, forwarding `max` to members without their own `ultra`.

Collapses the Connect section in the Gateway tab UI so the less-frequently-used setup

area no longer dominates over the recent-calls view.

v0.1.392 CODE ↗ STABLE

Adds Qoder subscription integration and NVIDIA NIM / ModelScope provider presets; usage.jsonl now records time-to-first-token (TTFT).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.392 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.392
```

Adds NVIDIA NIM (`integrate.api.nvidia.com/v1` , key from `build.nvidia.com`) and 魔搭 ModelScope as built-in provider presets.

Records time-to-first-token (TTFT) per streamed reply in `usage.jsonl` , enabling real generation-speed measurement alongside existing total-duration and output-token fields.

v0.1.391 CODE ↗ STABLE

magpie now supports Kimi Code (Moonshot's kimi CLI) as an agent, wiring catalog models into ~/.kimi/config.toml

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.391 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.391
```

└──▷ TRY IT

Override the Kimi config directory (useful in CI or multi-user setups) before launching magpie so it writes to a non-default location.

```
$ KIMI_SHARE_DIR=/opt/kimi magpie
```

Strips the `thinking` field (Kimi Code's thinking-mode switch) alongside the existing `enable_thinking` field before forwarding requests to OpenAI endpoints, preventing unknown-argument rejections.

v0.1.389 CODE ↗ STABLE

Magpie v0.1.389 adds StepFun Step Plan usage visibility — 5-hour, weekly, and credit windows shown on the Usage page via a browser sign-in bookmarklet flow.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.389 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.389
```

Adds `stepfun-auth.json` session store (beside `providers.json`, user-readable only) to persist StepFun platform credentials for up to 30 days, enabling magpie to autonomously renew the session and poll plan limits without re-authentication.

Adds `POST <platform>/api/step.openapi.devcenter.Dashboard/QueryStepPlanRateLimit` and `POST <platform>/api/step.openapi.devcenter.Dashboard/GetStepPlanStatus` platform API calls to fetch Step Plan 5-hour window, weekly window, and credit (subscription and top-up) usage rates.

Adds `stepfun/<site>/signout` API endpoint to sign out of a StepFun platform session from within magpie.

Adds `stepPlan` field to the StepFun provider UI, rendering a `Step Plan usage` section on the Usage page showing the 5-hour, weekly, and credit windows for StepFun (international, app 20700) and StepFun China (app 10300) providers.

Adds a bookmarklet-based sign-in flow (`SaveStepFunPaste`) letting users bring a StepFun platform session into magpie by pasting the session cookie obtained from a signed-in browser window.

v0.1.388 CODE ↗ STABLE

Magpie v0.1.388 adds custom API-key redaction rules and cross-provider conversation routing fixes.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.388 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.388
```

Adds user-defined masking rules in `Settings → Privacy` with `Kind`, `Prefix`, and `Regex` (RE2) fields, up to `MaxRules` (32) rules, to redact secrets magpie's built-in rules don't cover.

Automatically masks gateway API keys matching patterns like `oc_sk_…` or `or_sk_…` in fields such as `key`, `apiKey`, `api-key`, and `token` in outbound traffic.

Supports continued conversations when a session moves from one provider (e.g. Grok) to another (e.g. Codex/GPT via ChatGPT sign-in), handling cross-vendor sealed reasoning items that previously caused routing failures.

v0.1.387 CODE ↗ STABLE

Adds account balance display for StepFun (China) and StepFun (international) providers via their `/v1` /accounts endpoints.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.387 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.387
```

Adds balance reading for StepFun (China) via `https://api.stepfun.com/v1/accounts`, reporting remaining credit in ¥.

Adds balance reading for StepFun (international) via `https://api.stepfun.ai/v1/accounts`, reporting remaining credit in $.

v0.1.386 CODE ↗ STABLE

magpie v0.1.386 adds multi-account support for Kiro and Devin providers

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.386 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.386
```

Supports multiple Kiro accounts simultaneously, each stored in its own home directory under a `kiro/` folder alongside `logins.json`, with per-account `kiro-auth-token.json` refresh token isolation.

Supports multiple Devin accounts simultaneously, each stored in its own home directory under a `devin/` folder alongside `logins.json`, with per-account `credentials.toml` laid out so the Devin CLI (`devin auth status`, `devin models list`) can operate against each home independently.

v0.1.384 CODE ↗ STABLE

Adds Baidu Qianfan Token Plan (个人版) provider preset and fixes image-model test probes to use the images API.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.384 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.384
```

Image models are now probed via the images API (not chat) when running a provider test, matching how the gateway actually draws with them.

v0.1.382 CODE ↗ STABLE

Routing UI now shows verbatim vendor error messages and lets you jump to the newest failed request in one click.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.382 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.382
```

Adds verbatim vendor error text (It said: …) inline beneath each failed routing attempt in the request detail view, truncated at 600 characters.

Makes the 'errors your agent saw' stat clickable, jumping directly to the newest request whose agent received an HTTP 400+ error.

v0.1.380 CODE ↗ STABLE

Magpie's Devin integration now surfaces fast-run model variants and expands effort suffixes to include `none`, `minimal`, and `priority`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.380 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.380
```

Expands the Devin effort suffix regex to recognise `none`, `minimal`, and `priority` in addition to the existing `min`, `low`, `medium`, `high`, `xhigh`, `max`, and `fast` values.

Adds fast-run model variants for each Devin family to the model catalog and declared-model map, e.g. `claude-opus-5-5-fast`, so the Devin agent picker can offer them alongside standard family and variant IDs.

v0.1.379 CODE ↗ STABLE

Group routing members can now be pinned to a specific reasoning effort level using `provider/model:low` syntax

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.379 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.379
```

└──▷ USE IT

Pin a routing group member to a fixed low reasoning effort so cost-sensitive turns always use low compute, regardless of what the agent or classifier requested.

```text
provider/model:low
```

Supports `provider/model:<effort>` syntax for group members, pinning a member to a fixed reasoning effort level (one of `none`, `minimal`, `low`, `medium`, `high`, `xhigh`, `max`) regardless of what the agent or classifier requests.

Adds CodeBuddy CN (`copilot.tencent.com`) compatibility by correcting empty `finish_reason` and duplicate empty tool-call name fields in relayed streams.

v0.1.375 CODE ↗ STABLE

Magpie now returns an actionable 400 when a Codex sealed subagent handoff lands on a Magpie-served model

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.375 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.375
```

Returns a descriptive HTTP 400 error when an OpenAI-sealed `agent_message` (matching `^gAAAAA[A-Za-z0-9_-]+={0,2}$`) is posted to a Magpie-served model via the `CodexPath/responses` endpoint, guiding users to either use a Magpie-served lead

or an OpenAI subagent — with no decryption, hidden ChatGPT request, or upstream call attempted.

v0.1.374 CODE ↗ STABLE

magpie v0.1.374 adds native Kiro sign-in support via `app.kiro.dev` (Google, GitHub, AWS Builder ID, IAM Identity Center).

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.374 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.374
```

v0.1.370 CODE ↗ STABLE

Amazon Bedrock gains GPT-6 model support and fixes Anthropic endpoint auth to send only `x-api-key`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.370 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.370
```

Adds `global.openai.gpt-6-astra`, `global.openai.gpt-6-sol`, and `global.openai.gpt-6-luna` to the `bedrock` preset's model list.

Bedrock's Anthropic endpoint (`/anthropic/v1/messages`) now sends `x-api-key` alone, dropping the duplicate `Bearer` header that caused endpoint test failures.

v0.1.368 CODE ↗ STABLE

magpie now detects a stale Codex background daemon after account switches and offers a one-command restart

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.368 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.368
```

└──▷ TRY IT

After switching Codex accounts in magpie, restart the background daemon so new sessions use the correct account.

```
$ curl -s -X POST http://localhost:<port>/api/codex/daemon/restart
-H 'Content-Type: application/json' -d '{}'
```

Adds detection of a running Codex `--managed-daemon` (`codex app-server --managed-daemon`) after an account switch, warning users that sessions still run on the previous account until the daemon is restarted.

Adds `POST /api/codex/daemon/restart` endpoint and the `RestartCodexDaemon` action, which runs `codex app-server daemon restart` scoped to the switched account's `CODEX_HOME`.

v0.1.367 CODE ↗ STABLE

Magpie v0.1.367 marks free-tier models in the picker and adds USED/LEFT usage indicators to the panel.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.367 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.367
```

The model picker now badges models whose name or ID contains the word `free` (e.g. OpenRouter's `foo/bar:free` pattern) with a green FREE label, distinct from models that are free because they don't consume plan credits.

The usage panel now displays USED/LEFT pills on each usage group so you can see consumption at a glance.

v0.1.364 CODE ↗ STABLE

Adds Amazon Bedrock as a model provider, routing Claude via Anthropic messages and other models via OpenAI chat completions.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.364 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.364
```

> Adds `noList` field (`noList: true`) to `PresetDef` config to mark providers whose runtimes expose no `/models` listing endpoint, with a static `models` array used instead.
>
> Supports per-region Bedrock endpoints (e.g. `us-east-1` , `ap-southeast-1` , `eu-*` , `apac-*` , `jp-*` , `au-*`) selectable from the preset's `Regions` list.

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ sdk/sdk/v0.0.87 NOTES ↗ CODE ↗ RANK

Autonomous coding agent as an SDK, IDE extension, or CLI assistant.

> Cline shipped a native desktop app for macOS and Windows with parallel sessions and remote SSH support, alongside SDK updates that expand the model catalog, add PHP code search, cloud session handoff, and daemon diagnostics.

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

01 **PHP code search support in fallback search** NEW 80

Adds `php`, `phtml`, `inc`, and `twig` extensions to the fallback code search allowlist, making Laravel and other PHP projects visible in `search_codebase`, and excludes Composer's `vendor` directory from both search and the workspace file index.

– Names exact extensions, function, and excluded directory sdk/sdk/v0.0.87

02 **Cline Desktop native app with remote SSH support** NEW 75

Cline ships a native desktop app for macOS and Windows (Windows in beta), available at `cline.bot/desktop`, for running agent sessions outside a code editor. It supports running sessions in parallel, scheduling recurring tasks, and importing sessions from other coding agents, plus Cline Desktop SSH for managing sessions on remote machines — running the agent, its tools, and code on a remote dev box, build server, or cloud VM over SSH while the UI stays local.

– Names product URL and SSH feature but no CLI walkthrough product docs

03 **Hub daemon logging and startup telemetry** IMPROVED 75

The hub daemon log now records one line per socket close, including the close code, reason, and whether the heartbeat sweep dropped it. `hub.daemon.startup` telemetry and the daemon log report the PID, command line, lock state, and health-probe result when startup fails with `EADDRINUSE`.

– Names telemetry key and error code but is diagnostic-only sdk/sdk/v0.0.87

04 **Model catalog expansion with new providers and recommended models**

IMPROVED 70

Adds two new providers to the model catalog: `bee` (Bee by HEOSSI) and `pareto` (Pareto Inference), the latter configured via the new `PARETO_API_KEY` environment variable, bringing the total to 211 providers and 6,447 models. The Cline recommended model list now includes Claude Sonnet 5.5, Claude Opus 5.5, and `stealth/pixel-canary`, while Claude Opus 5 has been removed.

— Names env var, providers, counts but no setup steps sdk/sdk/v0.0.87

05 **Cloud session handoff and persistent workspaces** NEW

60

Adds shared cloud session handoff to the `./cloud` export, enabling a local session to be moved to a cloud agent and back, backed by persistent cloud workspaces that can be suspended and resumed.

— Names export path but no exact command syntax sdk/sdk/v0.0.87

THINNER COVERAGE BELOW

06 **Session metadata refresh in LocalRuntimeHost** IMPROVED

55

`LocalRuntimeHost.updateSession` now returns persisted metadata and refreshes the active session's in-memory metadata immediately after an update, including title changes.

— Names exact method but limited practical usage detail sdk/sdk/v0.0.87

sdk/sdk/v0.0.87 NOTES ↗ CODE ↗ LATEST STABLE

Cline SDK v0.0.87 adds two new model providers, PHP code search, cloud session handoff, and persistent cloud workspaces.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk/sdk/v0.0.87
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout sdk/sdk/v0.0.87
```

Adds `PARETO_API_KEY` environment variable support for the new `pareto` (Pareto Inference) provider in the model catalog.

Adds `php`, `phtml`, `inc`, and `twig` extensions to the fallback code search allowlist, making Laravel and other PHP projects visible in `search_codebase`; excludes Composer's `vendor` directory from search and the workspace file index.

Adds shared cloud session handoff to the `./cloud` export, enabling a local session to be moved to a cloud agent and back, with persistent cloud workspaces that can be suspended and resumed.

Adds two new providers to the model catalog: `bee` (Bee by HEOSSI) and `pareto` (Pareto Inference), bringing the total to 211 providers and 6,447 models.

> Adds Claude Sonnet 5.5, Claude Opus 5.5, and `stealth/pixel-canary` to the Cline recommended model list; removes Claude Opus 5.
>
> +2 more

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.33.6 NOTES ↗ RANK

Letta Code is a terminal coding agent with persistent memory and identity across development sessions.

> Letta Code shipped a live Cloud desktop sharing subcommand, native coding-agent session capture inside subagents, local backend worker support, and a Windows-specific environment variable to enable automatic client-side dreaming.

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

01 Windows opt-in for automatic client-side dreaming NEW

75

Adds the `LETTA_ENABLE_WINDOWS_AUTO_REFLECTION=1` environment variable to opt native Windows into automatic client-side dreaming, which is otherwise disabled by default on that platform even when `/sleeptime` settings are active.

Enable automatic client-side dreaming on native Windows when you rely on `/sleeptime` schedules and need the same background reflection behaviour as macOS/Linux.

```
$ set LETTA_ENABLE_WINDOWS_AUTO_REFLECTION=1 && letta-code
```

— Names exact env var, platform gap, and gives runnable command. product docs

02 Cloud desktop sharing via CLI NEW

60

Adds `letta sandbox desktop-link` CLI subcommand to share a live Cloud desktop session with a collaborator directly from the terminal.

Share a live Cloud desktop with a collaborator directly from the CLI.

```
$ letta sandbox desktop-link
```

— Exact subcommand given but no mechanism detail. v0.33.6

THINNER COVERAGE BELOW

03 Local backend worker support NEW

20

Letta Code now supports local backend workers for workflow execution.

— Only a one-line description, no config or usage detail. v0.33.6

04 Native session capture in subagents NEW

20

Captures native coding-agent sessions in subagents.

— Bare description with no mechanism or usage path. v0.33.6

v0.33.6 NOTES ↗ LATEST STABLE

Letta Code v0.33.6 adds native coding-agent session capture, a live Cloud desktop share link, and local backend worker support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.33.6 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.33.6
```

└──▷ TRY IT

Share a live Cloud desktop with a collaborator directly from the CLI.

```
$ letta sandbox desktop-link
```

Adds `letta sandbox desktop-link` CLI subcommand to share a live Cloud desktop session.

Supports local backend workers for workflow execution.

Captures native coding-agent sessions in subagents.

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.89 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 v1.0.89 release adds Claude Code rule file support, plugin enable/disable controls, several new models, and a batch of terminal UX and sandboxing refinements.

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

01 **Configurable indexed-search activation threshold** NEW 75

Adds the `TGREP_FILE_COUNT_THRESHOLD` environment variable to configure when automatic indexed file search kicks in, useful for tuning behavior in large monorepos.

Tune when Copilot activates indexed file search — useful in large monorepos where you want search to kick in only above a specific file count.

```
$ TGREP_FILE_COUNT_THRESHOLD=5000 copilot
```

— Names exact env var with a runnable example v1.0.89

02 **Plugin enable and disable controls** NEW 75

Adds `copilot plugin enable` to re-enable a previously disabled direct plugin install; direct plugin installs can now be individually enabled and disabled, and a plugin recorded as disabled stops loading.

Re-enable a direct plugin that was previously disabled so it loads again on next startup.

```
$ copilot plugin enable
```

— Named command plus behavior and runnable example v1.0.89

03 **Chat input prompt recall and undo** NEW 70

In local sessions, Esc Esc in an empty chat input takes back a prompt whose turn the model has not started answering and removes it from the conversation, while Up in an empty chat input recalls the pending message and leaves queued prompts in the queue.

— Names exact key sequences and precise behavior v1.0.89

04 **Claude Code rule files as custom instructions** NEW 65

Adds support for Claude Code rule files in `.claude/rules` as custom instructions, letting projects supply agent-specific guidance that the CLI picks up automatically.

Provide project-specific agent instructions by placing Claude Code rule files where the CLI will pick them up automatically.

```
$ mkdir -p .claude/rules && echo '# Always prefer async functions' > .claude/rules/style.md
```

— Names exact path with example of usage v1.0.89

05 **MCP tool filters match slash-containing names** IMPROVED 60

`server/tool` and `server/*` tool filters now match MCP tool names that contain a slash.

— Names exact filter syntax fixed v1.0.89

THINNER COVERAGE BELOW

06 **Sandboxed localhost access on Windows** IMPROVED 55

Sandboxed commands on supported Windows versions can access localhost when Local network access is on.

— Names the setting and platform scope v1.0.89

07 **Model ID autocomplete in /model pickers** IMPROVED 55

Autocompletes model IDs in `/model` and `/model plan` argument pickers.

— Names exact commands affected v1.0.89

08 **PR creation follows repository templates** IMPROVED 50

PR creation now follows repository pull request templates, preserving required sections and checklist structure.

— Describes behavior but no exact config surface v1.0.89

09 **Connector consent progress display** NEW 40

Shows managed Connector consent progress with a copyable authorization URL during connect and reconnect.

— Describes UI feedback without exact navigation v1.0.89

10 **Sandbox access to session files and logs** IMPROVED 40

Agent shell commands in sandbox can access session files and logs.

— States scope but no mechanism detail v1.0.89

11 **New models in model picker** NEW 35

Adds GPT-6 Sol and GPT-6 Luna to the model picker when available, and adds `claude-opus-4.5` to the supported model list.

— Names models but no usage detail v1.0.89

12 **Auto-suggested routing tier** NEW 30

The CLI auto suggests a routing tier and lets you switch with a shortcut or click.

— Vague on the mechanism and exact shortcut v1.0.89

13 **Queued exit commands during agent responses** IMPROVED 30

The CLI now queues exit commands while an agent response is still running, instead of exiting immediately.

— Bare behavior change, no further detail v1.0.89

14 **ACP sessions resilient to paused clients** IMPROVED 30

ACP sessions stay connected when clients pause reading large responses.

— Names protocol but limited mechanism detail v1.0.89

v1.0.89 NOTES ↗ LATEST STABLE

GitHub Copilot CLI v1.0.89 adds Claude rule files, configurable indexed-search threshold, new models, and plugin enable/disable controls.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.0.89 https://github.com/github/copilot-cli.git
# already have the repo? check out this version:
$ git checkout v1.0.89
```

└──▷ TRY IT

Tune when Copilot activates indexed file search — useful in large monorepos where you want search to kick in only above a specific file count.

```
$ TGREP_FILE_COUNT_THRESHOLD=5000 copilot
```

Re-enable a direct plugin that was previously disabled so it loads again on next startup.

```
$ copilot plugin enable
```

Provide project-specific agent instructions by placing Claude Code rule files where the CLI will pick them up automatically.

```
$ mkdir -p .claude/rules && echo '# Always prefer async functions'
> .claude/rules/style.md
```

Adds `TGREP_FILE_COUNT_THRESHOLD` environment variable to configure automatic indexed search activation.

Adds `copilot plugin enable` command to re-enable a previously disabled direct plugin install.

Adds support for Claude Code rule files in `.claude/rules` as custom instructions.

Adds GPT-6 Sol and GPT-6 Luna to the model picker when available.

Adds `claude-opus-5.5` to the supported model list.

+12 more

WAS THIS USEFUL? 👍 👎

Superset

Feedback ⓘ cli-v1.31.0 NOTES ↗ RANK

Superset is an agentic IDE to orchestrate 100+ coding agents in parallel. Run any agent with your own subscription.

> Superset's desktop client picked up GPT-6 and Opus 5.5 agent support, Slack page unfurling, cloud agent status in the sidebar, environment secrets management, per-environment regions, and mobile PR handling, while removing Google Calendar triggers; the CLI gained an `--audience` flag and a scrollable version history submenu.

└──▷ **WHAT SHIPPED · 13 FEATURES** most completely described first ? what's the number?

01 **`--audience` flag for CLI command visibility** NEW 70

Adds `--audience` flag to the CLI to hide unreleased or cloud-incompatible commands from users who cannot run them.

– Named flag with clear purpose, runnable as-is desktop-v1.31.0

THINNER COVERAGE BELOW

02 **GPT-6 and Opus 5.5 agent support** NEW 55

Adds GPT-6 and Opus 5.5 as supported models in the agents panel, synced with the live GPT-6 rollout.

– Names both models and panel but not selection mechanics desktop-v1.31.0

03 **Workspace attribution in page.list results** IMPROVED 50

Surfaces which workspace a page came from in paginated `page.list` results.

– Names the exact endpoint/field but not response shape desktop-v1.31.0

04 **Regional assignment per environment** NEW 50

Assigns a region to each environment, chosen nearest to the person, with all boxes running in that region.

– Explains selection logic and scope, no config surface named desktop-v1.31.0

05 **Stop watching a page from watcher menu** NEW 50

Enables stopping an agent from watching a page directly from its row in the watcher menu.

– Names UI location, gives a clear starting point desktop-v1.31.0

06 **Slack unfurling for Superset page links** NEW 45

Unfurls Superset page links in Slack as the posting user, with Open Graph-ratio thumbnails.

— Describes behaviour but not how to enable it `desktop-v1.31.0`

07 **Stable pull request panes keyed by repo and number** IMPROVED 45

Keys pull request panes by repository and number so each PR gets a stable, addressable pane.

— Explains mechanism but no UI path given `desktop-v1.31.0`

08 **Mobile PR support in workspace sheet** NEW 45

Adds a PR section to the full-height workspace sheet on mobile and captures `pull_request_merged` events on mobile.

— Two mobile PR additions named, no further detail `desktop-v1.31.0`

09 **Google Calendar triggers removed from integrations** BREAKING 40

Removes Google Calendar triggers from integrations, retaining Gmail.

— Clear before/after but no migration guidance `desktop-v1.31.0`

10 **Scrollable version history submenu in CLI** IMPROVED 35

The CLI's version history submenu now caps its height and scrolls, making long histories navigable without losing context.

— UI behaviour described, no exact path or command given `cli-v1.31.0`

11 **Cloud agent status in desktop sidebar** NEW 35

Shows agent status reported by a cloud box directly in the desktop sidebar.

— Brief description, no navigation detail `desktop-v1.31.0`

12 **Sign-in to usage accounts from embedded terminals** NEW 35

Supports signing in to usage accounts from within embedded terminals.

— States capability only, no mechanism `desktop-v1.31.0`

13 **Environment secrets management in CLI** NEW 30

Adds environment secrets management to the CLI.

— Only a name, no commands or scope given `desktop-v1.31.0`

└──▷ **BREAKING ON UPGRADE**

! Google Calendar triggers are removed from integrations; only Gmail triggers remain.

cli-v1.31.0 NOTES ↗ STABLE

Superset CLI v1.31.0 adds a scrollable version history submenu.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch cli-v1.31.0 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout cli-v1.31.0
```

Version history submenu now caps its height and scrolls, making long histories navigable without losing context.

desktop-v1.31.0 NOTES ↗ LATEST STABLE

Superset desktop v1.31.0 adds GPT-6/Opus 5.5 agent support, Slack page unfurling, CLI audience flags, and cloud agent status in the sidebar.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v1.31.0 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout desktop-v1.31.0
```

Adds `--audience` flag to the CLI to hide unreleased or cloud-incompatible commands from users who cannot run them.

Adds GPT-6 and Opus 5.5 as supported models in the agents panel, synced with the live GPT-6 rollout.

Unfurls Superset page links in Slack as the posting user, with Open Graph-ratio thumbnails.

Shows agent status reported by a cloud box directly in the desktop sidebar.

Supports signing in to usage accounts from within embedded terminals.

+8 more

└──▷ BREAKING ON UPGRADE

! Google Calendar triggers are removed from integrations; only Gmail triggers remain.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.284 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.284 makes Claude Sonnet 5.5 the default model, switches interactive terminal and VS Code sessions to auto mode by default, and adds an MCP reconnect-all command, Google Cloud OTLP telemetry, certificate-based gateway authentication, and clearer rate-limit spend visibility.

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

01 **Google Cloud OTLP telemetry destination** NEW 85

Claude apps gateway `telemetry.forward_to` destinations gain `auth: { google: {} }`, enabling telemetry export to a Google Cloud OTLP endpoint using the gateway's own Google Cloud credentials.

Route Claude apps gateway telemetry to a Google Cloud OTLP endpoint using the gateway's own GCP credentials.

yaml

```yaml
telemetry:
  forward_to:
    - auth:
        google: {}
```

— Exact config key with a working YAML example. v2.1.284

02 **Effort slider keybinding actions** NEW 80

New keybinding actions `effortSlider:decreaseEffort`, `effortSlider:increaseEffort`, and `effortSlider:toggleUltracode` let the `/effort` slider's arrow and Tab keys be rebound in `keybindings.json`.

Remap the effort slider controls and Ultracode toggle to custom keys in your keybindings config.

```json
[
  { "key": "ctrl+shift+,", "action":
"effortSlider:decreaseEffort" },
  { "key": "ctrl+shift+.", "action":
"effortSlider:increaseEffort" },
  { "key": "ctrl+shift+u", "action":
"effortSlider:toggleUltracode" }
]
```

— Named config keys with a runnable rebinding example. v2.1.284

03 **Ultracode as independent effort toggle** IMPROVED 75

Ultracode is now its own toggle within `/effort` (Tab, or `/effort ultracode [on|off]`): it no longer forces xhigh effort and stays enabled at any effort level.

— Clear before/after behavior with exact command. v2.1.284

04 **Rate limit spend visibility** NEW 75

Status line's `rate_limits.spend_limit` gains `used_usd`, `limit_usd`, and `period` fields, and `/usage` plus the status line now show dollar amounts (e.g. '$271.40 / $500.00 spent this month'); claude.ai subscribers also get a new `/rate-limit-options` entry in `/help` and the command menu.

— Names exact fields and a discoverable command entry. v2.1.284

05 **Claude Sonnet 5.5 default model** NEW 70

`claude-sonnet-5-5` is now the default Sonnet model on the Anthropic API, with a 1M context window, $2/$10 per Mtok input/output pricing, and $0.20/Mtok cache reads.

— Exact pricing and context window given, but no user action required. v2.1.284

06 **MCP reconnect-all command** NEW 70

The interactive terminal gains `/mcp reconnect all`, which retries every MCP server that failed to connect or needs authentication in one command.

Retry all failed or unauthenticated MCP servers at once without restarting your session.

```
$ /mcp reconnect all
```

— Exact runnable command supplied. v2.1.284

07 **Auto mode default behavior and prompts** BREAKING 70

Interactive terminal and VS Code sessions now start in auto mode by default when no permission mode is configured, across every plan and provider (still overridden by `permissions.defaultMode`); auto mode's prompt before reading outside the working directories also gains a `Yes, but ask again next time` answer.

— Clear behavior change and override key, no migration steps. v2.1.284

08 **Certificate client authentication for gateway** NEW 65

The Claude apps gateway now supports certificate client authentication (`private_key_jwt`) with its identity provider, for identity providers that issue certificate credentials instead of client secrets.

— Names the auth mechanism but no setup steps given. v2.1.284

THINNER COVERAGE BELOW

09 **Gateway startup warnings for managed policy models** NEW 55

The Claude apps gateway now warns at startup when a managed policy's `availableModels` is empty or omits the startup model, unless `model` or `enforceAvailableModels` is set.

— Names the config fields but not a direct action. v2.1.284

v2.1.284 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.284 adds Claude Sonnet 5.5 as default, `/mcp reconnect all`, Google Cloud OTLP telemetry, and new keybinding actions for the effort slider.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.1.284 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.284
```

└──▷ TRY IT

Retry all failed or unauthenticated MCP servers at once without restarting your session.

```
$ /mcp reconnect all
```

Remap the effort slider controls and Ultracode toggle to custom keys in your keybindings config.

keybindings.json

```
[
  { "key": "ctrl+shift+,", "action": "effortSlider:decreaseEffort"
},
  { "key": "ctrl+shift+.", "action": "effortSlider:increaseEffort"
},
  { "key": "ctrl+shift+u", "action": "effortSlider:toggleUltracode"
}
]
```

Route Claude apps gateway telemetry to a Google Cloud OTLP endpoint using the gateway's own GCP credentials.

yaml

```
telemetry:
  forward_to:
    - auth:
        google: {}
```

Adds `claude-sonnet-5-5` as the new default Sonnet model on the Anthropic API — 1M context window, $2/$10 per Mtok input/output with $0.20/Mtok cache reads.

Adds `/mcp reconnect all` command in the interactive terminal to retry every MCP server that failed to connect or needs authentication at once.

Adds `effortSlider:decreaseEffort`, `increaseEffort`, and `toggleUltracode` keybinding actions so the `/effort` slider's arrow and Tab keys can be rebound in `keybindings.json`.

Adds `auth: { google: {} }` for Claude apps gateway `telemetry.forward_to` destinations, enabling telemetry export to Google Cloud's OTLP endpoint using the gateway's Google Cloud credentials.

Adds certificate client authentication (`private_key_jwt`) between the Claude apps gateway and its identity provider, for identity providers that issue certificate credentials instead of client secrets.

+6 more

WAS THIS USEFUL? 👍 👎

Vercel v0

Feedback ⓘ changelog · Sep 29, 2026 CHANGELOG ↗ RANK

The v0 generative UI platform builds web applications from natural-language prompts.

> v0 introduced a browsable AI Gateway model picker with token pricing and Thinking/Fast Mode toggles, and reworked how Enterprise teams control access to custom models.

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

01 Enterprise-owner controlled custom model access

BREAKING

80

Enterprise team owners can restrict access to custom AI Gateway models by disabling Allow custom models in Workspace settings, while individual model restrictions and isolated tenant restrictions continue to apply. Existing Enterprise teams that had previously turned off Restrict to v0 Models must now explicitly re-enable Allow custom models in Workspace settings to regain custom model access under this new scheme.

Enable custom AI Gateway models for your Enterprise team so members can use models beyond the default v0 set.

> 📍 In v0, go to Workspace settings and enable Allow custom models.

— Names exact setting and required migration step in Workspace settings.

changelog-20260929-14a3cce8

└──▷ BREAKING ON UPGRADE

! Existing Enterprise teams that previously turned off Restrict to v0 Models must still explicitly opt in to Allow custom models in Workspace settings to gain access to custom models under the new scheme.

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

v0 adds a browsable AI Gateway model picker with token pricing, Thinking/Fast Mode toggles, and Enterprise owner-controlled custom model access.

└──▷ HOW TO FIND IT

Enable custom AI Gateway models for your Enterprise team so members can use models beyond the default v0 set.

> 📍 In v0, go to Workspace settings and enable Allow custom models.

Enterprise teams with custom model access can be restricted again by owners disabling Allow custom models in Workspace settings; individual model restrictions and isolated

tenant restrictions continue to apply.

└─▷ **BREAKING ON UPGRADE**

! Existing Enterprise teams that previously turned off `Restrict to v0 Models` must still explicitly opt in to `Allow custom models` in Workspace settings to gain access to custom models under the new scheme.

WAS THIS USEFUL?  

Amazon Kiro

Feedback ⓘ changelog · Sep 29, 2026 CHANGELOG ↗

RANK

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

> Kiro's biggest update centers on agentic infrastructure: durable work queues survive Gateway restarts, AWS Fargate joins SSH and SSM as a remote-crew connection method, and Claude Opus 5.5 arrives with a 1M-token context window, alongside expanded PR watches, new kirocrew CLI commands, and session-wide tool approval.

└──▷ WHAT SHIPPED · 17 FEATURES most completely described first ? what's the number?

01 **Durable crew session log** NEW 90

Setting `KIROCREW_CREW_LOG=1` in `~/.kiro/crew/.env` enables an append-only session trail with status, usage, timeline, tools, and approvals views, backed by the read-only `kirocrew-crew-log` MCP server for agent access.

Enable the durable crew log so long-running agentic sessions survive Gateway restarts and expose a session ledger with timeline, tools, and approvals.

env

```
KIROCREW_CREW_LOG=1
```

— Named config key, file path, and MCP server included changelog-20260929-c030f782

02 **AWS Fargate remote-crew support** NEW 90

AWS Fargate joins SSH and AWS Systems Manager Session Manager (SSM) as a remote-crew connection method, requiring no inbound network rule. Task lifetime is configured via `fargate.task_ttl_seconds` in `cloud.json` (default six hours, max ten concurrent tasks).

Set a custom Fargate task lifetime (e.g. 12 hours) when default 6-hour tasks expire before long-horizon work completes.

json

```
{
  "fargate": {
    "task_ttl_seconds": 43200
  }
}
```

— Config key, file, defaults, and limits all named `changelog-20260929-c030f782`

---

03 **kirocrew pod lifecycle subcommands** NEW 80

Adds `kirocrew pod up`, `down`, `ls`, `status`, `token`, `url`, `logs`, `prune`, and `provision` subcommands to manage the lifecycle of an isolated pod for a Kiro Crew source worktree on Windows via Task Scheduler.

— All nine subcommands named verbatim `changelog-20260929-c030f782`

---

04 **PR watches for GitLab, Bitbucket, Azure DevOps** NEW 80

Pull request watches extend to GitLab (via `glab`), Bitbucket Cloud (via `BITBUCKET_EMAIL` and `BITBUCKET_API_TOKEN`), and Azure DevOps Services (via `az` with the azure-devops extension and `az login` or `AZURE_DEVOPS_EXT_PAT`).

— Every provider, tool, and credential env var named `changelog-20260929-c030f782`

---

05 **.env files no longer auto-load** BREAKING 75

Project `.env` files no longer load automatically into chat sessions, MCP servers, or tools; Kiro now starts with the environment inherited from the terminal. Workflows relying on automatic loading must export required variables in the shell before starting Kiro or reference them in MCP server configuration.

— Explains before/after and migration steps `changelog-20260929-c030f782`

---

06 **Session-wide tool approval** NEW 70

`/tools trust-all` automatically approves all tool calls for the remainder of the current V3 session.

Auto-approve all tool calls for the current V3 session to reduce interruptions during a trusted, hands-off automation run.

```
$ /tools trust-all
```

— Exact command given, scope limited to session `changelog-20260929-c030f782`

---

07 **Auto (Jev) model-picker routing** NEW 70

A new Auto (Jev) option in the model picker, under `Settings > Developer > Feature Previews`, routes each turn by complexity through `decisions.model_route`, with per-message feedback controls.

— Names config path and routing key, UI path given `changelog-20260929-c030f782`

---

08 **Side Chat for ephemeral questions** NEW 65

Side Chat, opened via `/side` or `/btw`, provides an ephemeral question thread beside the main conversation, with read-only lookup support on Kiro CLI backend sessions.

— Exact commands named, scope defined changelog-20260929-c030f782

09 **kirocrew doctor and restart commands** NEW 65

`kirocrew doctor` reports missing installation components and backend limitations (e.g. Pi session MCP tool restrictions); `kirocrew restart` reloads the latest model list for Crew.

— Both commands and their exact purposes named changelog-20260929-c030f782

10 **Model and reasoning settings UI** IMPROVED 65

Kiro Web adds a reasoning effort selector for supported models and folds consecutive reasoning steps and tool calls into a collapsible timeline block. A new `/model` panel lets users switch models and configure thinking and effort in one place, becoming the primary interface for reasoning settings.

— Names panel command and UI consolidation changelog-20260929-c030f782

11 **Session dashboard filter combining** IMPROVED 65

The session dashboard adds a filter combining `current workspace`, `main sessions`, and `bookmarked` filters, with an `includes empty` option to reveal hidden empty sessions.

— Names all filter options and toggle changelog-20260929-c030f782

12 **Durable work queue across Gateway restarts** IMPROVED 60

Accepted subagent work, Task Runner steps, and workflow calls are now durable across Gateway restarts via a persistent queue, visible on the System page.

— Names affected surfaces and where to view queue changelog-20260929-c030f782

THINNER COVERAGE BELOW

13 **Turn minimap in desktop chat** NEW 55

A turn minimap in the left gutter of desktop chat shows one marker per loaded turn with hover previews and click-to-jump navigation.

— Describes mechanism but no config surface changelog-20260929-c030f782

14 **OpenCode and goose agent backends** NEW 50

OpenCode and goose join Kiro CLI, Claude Code, Codex, and KAS as selectable agent backends in the Agent Backend selector (Preview).

— Names selector and all backend options changelog-20260929-c030f782

15 **Cloud sessions without repo connection** IMPROVED 45

V3 cloud sessions can now start before connecting a repository via `/repo`.

— Names command but minimal mechanism detail changelog-20260929-c030f782

16 **Automatic Gateway update mechanism** IMPROVED 45

Gateway now updates automatically, waiting for active turns, scheduled runs, subagents, Task Runner steps, and workflows to become idle before applying the update.

— Describes mechanism but no config control named changelog-20260929-c030f782

17 **Configurable terminal command timeout** IMPROVED 30

IDE 1.1.70 adds a configurable default timeout for local agent terminal commands.

— No config key or default value given changelog-20260929-c030f782

└──▷ ALSO FROM THESE RELEASES



└──▷ BREAKING ON UPGRADE

! Project `.env` files no longer load automatically into chat sessions, MCP servers, or tools. Workflows that relied on automatic `.env` loading must instead export the required variables in the shell before starting Kiro, or reference exported values in the MCP server configuration.

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

Kiro gains Claude Opus 5.5, durable work queues, PR watches for GitLab/Bitbucket/Azure DevOps, Fargate remote crews, and session-wide tool approval.

└──▷ USE IT

Enable the durable crew log so long-running agentic sessions survive Gateway restarts and expose a session ledger with timeline, tools, and approvals.

~/.kiro/crew/.env

```
KIROCREW_CREW_LOG=1
```

Set a custom Fargate task lifetime (e.g. 12 hours) when default 6-hour tasks expire before long-horizon work completes.

cloud.json

```
{
  "fargate": {
    "task_ttl_seconds": 43200
  }
}
```

Auto-approve all tool calls for the current V3 session to reduce interruptions during a trusted, hands-off automation run.

```
$ /tools trust-all
```

Adds `KIROCREW_CREW_LOG=1` in `~/.kiro/crew/.env` to enable an append-only session trail with status, usage, timeline, tools, and approvals views, backed by the `kirocrew-crew-log` MCP server for read-only agent access.

Adds `fargate.task_ttl_seconds` in `cloud.json` to configure the lifetime of AWS Fargate remote-crew tasks (default six hours, max ten concurrent tasks).

Adds `kirocrew pod up`, `down`, `ls`, `status`, `token`, `url`, `logs`, `prune`, and `provision` subcommands to manage isolated-pod lifecycle for a Kiro Crew source worktree on Windows via Task Scheduler.

Extends pull request watches to GitLab (via `glab`), Bitbucket Cloud (via `BITBUCKET_EMAIL` and `BITBUCKET_API_TOKEN`), and Azure DevOps Services (via `az` with azure-devops extension and `az login` or `AZURE_DEVOPS_EXT_PAT`).

Adds `/tools trust-all` command to automatically approve all tool calls for the remainder of the current V3 session.

+16 more

└──▷ BREAKING ON UPGRADE

! Project `.env` files no longer load automatically into chat sessions, MCP servers, or tools. Workflows that relied on automatic `.env` loading must instead export the

required variables in the shell before starting Kiro, or reference exported values in the MCP server configuration.

WAS THIS USEFUL? 👍 👎

Cognition Devin Desktop

Feedback

changelog · Sep 29, 2026 | CHANGELOG ↗ | RANK

Devin Desktop is an agent-native code editor for developing software locally, remotely, and in containerized environments.

> Devin Desktop shipped diagnostic profiling commands, MCP server support over ACP session endpoints, repository context files via AGENTS.md, and a Microsoft Teams integration for starting sessions from chat.

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

01 **Language server profiling commands** NEW 80

Adds `Devin: Save Language Server Memory Profile` and `Devin: Save Language Server CPU Profile` commands to capture a diagnostic `.zip` that can be shared with the Devin support team when diagnosing performance issues.

Capture a language-server CPU profile to send to support when diagnosing a sluggish extension host.

> 📍 Open the Command Palette (Ctrl+Shift+P / Cmd+Shift+P), type `Devin: Save Language Server CPU Profile`, press Enter, then share the downloaded .zip with the Devin support team.

— Names exact commands and gives a runnable capture-and-share workflow.

changelog-20260929-0242431e

02 **Repository context via AGENTS.md** NEW 80

Supports `AGENTS.md` files to supply Devin with repository context and instructions, with a 16 KiB automatic injection limit for structuring longer guidance.

Provide Devin with repo-specific coding standards and toolchain instructions so every session inherits the right context automatically.

```yaml
# Place at the root of your repository
# File: AGENTS.md

## Coding Standards
- Use Black for Python formatting
- All PRs require passing unit tests

## Toolchain
- Build: `make build`
- Test: `pytest tests/`

## Notes
Keep individual guidance sections under the 16 KiB auto-
injection limit.
```

— Exact file name, limit, and a runnable config example provided. product docs

---

03 **Microsoft Teams session integration** NEW 70

Adds Microsoft Teams integration, enabling sessions via `@Devin AI` mentions in channels or chats, with inline keyword controls to pause, resume, or stop sessions and in-context replies.

Start a Devin session directly from Microsoft Teams without leaving the channel where work is being discussed.

> 📍 In your Teams channel or chat: 1. Type a message and tag @Devin AI. 2. Describe the task in the same message (e.g., '@Devin AI refactor the auth module to use OAuth2'). 3. Devin replies in-thread; use inline keywords to pause, resume, or stop the session.

— Clear starting steps given, mechanism of keyword control described briefly.

product docs

---

04 **MCP servers via ACP session endpoints** NEW 65

MCP servers passed by an ACP client in `session/new` and `session/load` are now usable by the agent and listed in `/mcp`, including HTTP and SSE MCP servers.

— Names endpoints and protocols but no usage steps given. changelog-20260929-0242431e

└──▷ ALSO FROM THESE RELEASES

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

Devin Desktop adds language-server profiling commands and MCP server support via ACP session endpoints

└──▷ HOW TO FIND IT

Capture a language-server CPU profile to send to support when diagnosing a sluggish extension host.

📍 Open the Command Palette (Ctrl+Shift+P / Cmd+Shift+P), type `Devin: Save Language Server CPU Profile`, press Enter, then share the downloaded .zip with the Devin support team.

Adds `Devin: Save Language Server Memory Profile` and `Devin: Save Language Server CPU Profile` commands to capture a diagnostic `.zip` for support.

MCP servers passed by an ACP client in `session/new` and `session/load` are now usable by the agent and listed in `/mcp`, including HTTP and SSE MCP servers.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ desktop-v0.24.7 NOTES ↗

RANK

An open-source AI coding agent that lives in your terminal.

> Qwen Code shipped screen sharing in Qwen Live voice calls, an agent-prepared `/batch-api` workflow and a `/commit` slash command, plus Managed Runtime attestation and protocol support, ARM64 Linux desktop builds, remote desktop control via a `node_repl` relay, and notable startup performance gains.

└──▷ **WHAT SHIPPED · 19 FEATURES** most completely described first ? what's the number?

01 Opt-out of desktop update check NEW

80

Adds a `QWEN_DESKTOP_DISABLE_UPDATES` environment variable that lets a desktop installation opt out of the automatic update check performed at startup, useful for locked-down or air-gapped environments.

Opt a desktop installation out of the automatic update check at startup, useful in locked-down or air-gapped environments.

```
$ export QWEN_DESKTOP_DISABLE_UPDATES=1
qwen
```

— Exact env var and runnable example; brief on why it matters desktop-v0.24.7

02 /batch-api workflow in CLI NEW

70

Adds an agent-prepared `/batch-api` workflow, invocable as a slash command inside a Qwen Code session, to fan out large-scale code tasks through the Batch API.

Kick off a Batch API workflow from within a Qwen Code session to fan out large-scale code tasks.

```
$ qwen -p '/batch-api process all TODO comments in src/'
```

— Named command and example; underlying batch mechanism unspecified desktop-v0.24.7

03 Managed Runtime protocol and Java SDK support NEW

70

Adds a managed runtime attestation client to the Java SDK that requires attestation before adopting a Managed Runtime, the `W0a` Managed Workspace binding contract in the Java SDK, the `managed-context/1` envelope contract in the CLI, and a v2 `execute` / `status` / `cancel` Managed Runtime contract in the CLI with v2 tool

operations mounted on the Managed Runtime worker. Also adds a Durable Managed Session journal and failover, a Spring control plane and dual-path WebShell for managed agents, and durable workspace collaboration state for agents.

— Names every contract and component; mostly infra, low end-user actionability `v0.24.7` `desktop-v0.24.7`

04 **List managed sessions with qwen sessions ps** NEW 65

Adds `qwen sessions ps` to list all managed Agent View sessions, letting users monitor concurrent headless agent runs such as in CI.

List all managed Agent View sessions — useful for monitoring concurrent headless agent runs in CI.

```
$ qwen sessions ps
```

— Exact command and example; brief on what output provides `desktop-v0.24.7`

05 **/commit slash command** NEW 65

Adds a `/commit` slash command that drafts an AI-generated commit message directly inside the CLI session after changes are staged.

Let the AI draft your commit message after staging changes, instead of writing it manually.

```
$ qwen
# inside the session:
/commit
```

— Exact command and example; drafting logic not explained `v0.24.7`

06 **Session and endpoint configuration in qwen serve** IMPROVED 60

`qwen serve` now supports choosing the model and reasoning mode at session creation time, and the Qwen Live endpoint and model can be set from the serve/web-shell settings UI; the release also adds guarded Hosted Runtime foundations underneath `qwen serve`.

— Names `qwen serve` and UI paths; three thin additions grouped `desktop-v0.24.7`

07 **Git worktree support in web shell** IMPROVED 60

Adds worktree management to the repository's git dialog in the web shell, and supports optional Git worktrees for branch sessions, allowing isolated per-branch workspaces.

— Mechanism (isolated workspaces) and UI path named across two releases `v0.24.7`

desktop-v0.24.7

08 **Review auditing improvements** IMPROVED 60

Adds review auditing of the applied `--fix` for unpinned new assumptions, and records per-file content verdicts in review that survive a rebase.

— Names `--fix` and rebase behavior; concrete but narrow scope desktop-v0.24.7

THINNER COVERAGE BELOW

09 **Trajectory overview enhancements in web shell** IMPROVED 55

The trajectory overview in the web shell gains time-range selection to narrow the session table, zoom and pan controls, and a real-time display of the trajectory including idle periods; a later release adds inline inspection of trajectory records in the web shell.

— Names concrete UI controls across two releases; clear navigation path v0.24.7 desktop-v0.24.7

10 **ARM64 Linux desktop support** NEW 50

Adds a `linux-aarch64` leg to the Desktop release matrix, extending Desktop support to ARM64 Linux.

— Names build target `linux-aarch64`; clear but minimal detail v0.24.7

11 **Remote desktop control via node_repl relay** NEW 50

Enables remote sessions to control your local desktop through a `node_repl` relay, supporting computer-use workflows.

— Names `node_repl` relay and use case; no exact command given v0.24.7

12 **Screen sharing in Qwen Live voice calls** NEW 45

Streams a shared screen into browser voice calls via Qwen Live.

— Names Qwen Live; no navigation or technical detail given desktop-v0.24.7

13 **Web shell workflow additions** NEW 40

Adds in-conversation search, host controls for model management, per-prompt inspection of session tool calls, and selective host artifact integration in the web shell.

— Four thin one-line UI additions; no mechanism or exact path desktop-v0.24.7

14 **Startup performance improvements** IMPROVED 40

Makes the bundled browser-use runtime self-contained and skips interactive-only probes on headless startup, reducing cold-start overhead.

— Two named performance fixes; no measured numbers given desktop-v0.24.7

15 **Landlock sandboxing fallback** NEW 40

Adds a Landlock execution fallback for sandboxed process isolation on Linux.

— Names Landlock mechanism; no configuration or flag given v0.24.7

16 **Native advisor tool** NEW 35

Adds a native advisor tool with consultation behavior and usage limits.

— Names capability only; no flag, limit value or usage path given desktop-v0.24.7

17 **Structured on-demand memory recall** NEW 35

Adds structured on-demand memory recall, giving the agent bounded retrieval from its memory store.

— Explains bounded retrieval mechanism; no named config or limit v0.24.7

18 **Channel access policies and message routing** IMPROVED 30

Separates private and group access policies in channels, and routes messages to prefix-specific sessions in channels.

— Two thin one-line additions; no mechanism or scope given desktop-v0.24.7

19 **Scoped document selection on mobile** NEW 25

Supports scoped native document selection on mobile (Phase 2).

— Thin one-line note; unclear what Phase 2 actually changed v0.24.7

desktop-v0.24.7 NOTES ↗ LATEST STABLE

Qwen Code desktop-v0.24.7 adds screen sharing in voice calls, a `/batch-api` workflow, managed runtime attestation, and major startup performance gains.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v0.24.7
https://github.com/QwenLM/qwen-code.git
# already have the repo? check out this version:
$ git checkout desktop-v0.24.7
```

└──▷ TRY IT

Opt a desktop installation out of the automatic update check at startup, useful in locked-down or air-gapped environments.

```
$ export QWEN_DESKTOP_DISABLE_UPDATES=1
qwen
```

List all managed Agent View sessions — useful for monitoring concurrent headless agent runs in CI.

```
$ qwen sessions ps
```

Kick off a Batch API workflow from within a Qwen Code session to fan out large-scale code tasks.

```
$ qwen -p '/batch-api process all TODO comments in src/'
```

Adds `QWEN_DESKTOP_DISABLE_UPDATES` environment variable to opt out of the desktop startup update check.

Adds `qwen sessions ps` listing of managed Agent View sessions.

Adds `/batch-api` agent-prepared Batch API workflow in the CLI.

Supports model and reasoning selection at session creation time in `qwen serve`.

Sets the Qwen Live endpoint and model from the serve/web-shell settings UI.

+24 more

v0.24.7 NOTES ↗ STABLE

Qwen Code: Adds `/commit` slash command that drafts AI-generated commit messages directly in the CLI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.24.7 https://github.com/QwenLM/qwen-
code.git
# already have the repo? check out this version:
$ git checkout v0.24.7
```

└──▷ TRY IT

Let the AI draft your commit message after staging changes, instead of writing it manually.

```
$ qwen
# inside the session:
/commit
```

Adds a `linux-aarch64` leg to the Desktop release matrix, extending Desktop support to ARM64 Linux.

Enables remote sessions to control your local desktop through a `node_repl` relay for computer-use workflows.

Supports optional Git worktrees for branch sessions in the web shell, allowing isolated per-branch workspaces.

Adds structured on-demand memory recall, giving the agent bounded retrieval from its memory store.

Adds a Landlock execution fallback for sandboxed process isolation on Linux.

+3 more

WAS THIS USEFUL? 👍 👎

herdr

Feedback ⓘ v0.9.2 NOTES ↗ RANK

Herdr is a coding-agent runtime that manages persistent terminal sessions across local and SSH machines.

> Herdr v0.9.2 broadens coding-agent support to more than 20 agents via screen-state detection, native state reporting, and a new custom-agent API, adds `herdr machine status` / `reconnect` commands with finer SSH connection control, and replaces a removed pane-overlay socket API with standard Kitty graphics protocol support — rounded out by a workspace-grouped Goto picker, rolling session snapshots, and a batch of new TUI/config keys.

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

01 **Machine connection commands and SSH tuning** NEW 95

Adds `herdr machine status [<label-or-id>] [--json]` for fresh, non-interactive SSH and remote Herdr checks, and `herdr machine reconnect <label-or-id>` to complete SSH authentication (including MFA) in a terminal. `herdr machine add` gains a `--remote-session <name>` flag to skip interactive session discovery and target a specific remote session, and a `--label "Build machine"` flag to set a custom sidebar name (optional, defaulting to the SSH hostname); interactive `machine add` now also discovers existing Herdr sessions on the host for selection. When `remote.manage_ssh_config=true`, a single approved OpenSSH connection (with `ControlPersist 600`) is shared across machine add, the sidebar, and saved-machine API commands. SSH compression is applied via `-C` to reduce screen-update traffic on limited connections (overriding `Compression no` in your SSH config), agent forwarding now uses a stable address across reconnects on updated Linux/macOS servers, and clicking `! auth`, `! error`, or a failed connection's reconnecting status in the sidebar shows the latest error and the relevant CLI command.

Check SSH reachability of a remote machine non-interactively in a script or CI pipeline.

```
$ herdr machine status my-server --json
```

Quickly check whether a saved machine is reachable from a script or CI step without triggering interactive prompts.

```
$ herdr machine status my-build-server --json
```

Recover SSH authentication for a machine showing `! auth` in the sidebar — run this in an interactive terminal so prompts can be answered.

```
$ herdr machine reconnect my-build-server
```

Add a machine targeting a specific named remote session, skipping interactive discovery, with a human-friendly sidebar label.

```
$ herdr machine add workbox --remote-session agents --label
"Build machine"
```

— Exact flags and runnable commands for every new subcommand v0.9.2

02 **New TUI and session config keys** NEW 95

Adds `ui.confirm_close`: when enabled, the TUI prompts before closing a workspace's last tab via a keybinding or the tab menu (CLI/API tab closes remain immediate). Adds `keys.clear_pane` (e.g. `clear_pane = "prefix+ctrl+k"`) to clear the focused pane's screen and scrollback — keeping the current cursor line and soft-wrapped rows — without sending input to the running program; unset by default. Adds `[session] startup_per_agent_delay_ms` (default 100ms, settable to `0` to launch all agents simultaneously) to space out restored agents at startup. `keys.prefix` now accepts a single key string (e.g. `"ctrl+b"`, `"f12"`, `"esc"`) or an array of keys (e.g. `prefix = ["ctrl+space", "ctrl+s"]`) so one config can share prefix-mode entry across machines, and `prefix+?` lists all active prefixes. A `default_shell` config key sets the executable used for new interactive panes, defaulting to `SHELL` then `/bin/sh`; on Windows it resolves `pwsh.exe` (PowerShell 7) on `PATH` before falling back to the inbox `powershell.exe` (Windows PowerShell 5.1), with an explicit `default_shell` still taking precedence. Adds a `terminal.mouse` controller event accepting `action`, `button` (`left` / `middle`), `column`, `row`, and optional `modifiers` (`Shift=1`, `Ctrl=2`, `Alt=4`, default 0) at zero-based cell coordinates of the controller's viewport.

Require confirmation before accidentally closing the last workspace tab from a keybinding.

```yaml
ui.confirm_close: true
```

Send a left mouse button down event to a running pane at cell column 12, row 5 via the terminal controller API.

```
$
{"type":"terminal.mouse","action":"down","button":"left","column
":12,"row":5}
```

Bind a key to wipe a pane's screen and scrollback instantly during an incident triage session without disturbing the running process.

```toml
clear_pane = "prefix+ctrl+k"
```

Stagger agent restores after a server restart to avoid thundering-herd load on reconnect.

```toml
[session]
startup_per_agent_delay_ms = 250
```

Disable staggered agent startup to launch all agents simultaneously on a low-contention host.

```ini
startup_per_agent_delay_ms = 0
```

Share one herdr config between macOS and a remote Linux machine by listing multiple prefix keys so either works.

```toml
[keys]
prefix = ["ctrl+space", "ctrl+s"]
```

Bind multiple prefix keys so both Ctrl+Space and Ctrl+S enter prefix mode — useful when one conflicts with another app.

```toml
prefix = ["ctrl+space", "ctrl+s"]
```

Stagger agent restores to avoid hammering the system after a server restart — set a longer delay between each agent coming back.

```toml
[session]
startup_per_agent_delay_ms = 500
```

— Every config key named with exact values, examples, and defaults v0.9.2

03 Coding agent integrations and custom-agent API NEW

90

Herdr detects agent state by reading live bottom-of-pane output against a detection manifest for `copilot`, `cursor`, `opencode`, `pi`, `omp`, `droid`, `devin`, `kimi`, `kilo`, `hermes`, `qodercli`, `qwen`, `letta`, `mastracode`, `grok`, `antigravity-cli`, Gemini CLI, and Cline, and supports native Herdr state reporting with no install required for Crush, Command Code, Muse Code, and Prime Agent. New direct-reporting integrations for Pi, OMP, Kimi Code CLI, OpenCode, Kilo Code CLI, and MastraCode let Herdr use reported state instead of screen detection, and a custom-agent API lets third-party agents report their own state and resume command without native Herdr support, so `herdr integration install <agent>` can restore the exact session after a server restart. The OpenCode integration additionally preserves existing V1 TUI preferences and plugins on install, reuses an existing V1 registration, and removes only the Herdr registration on uninstall — after updating it, existing OpenCode servers and their TUIs must be restarted so the old server reporter cannot overwrite pane-local state.

Restore a `cursor` agent session to the same state after a Herdr server restart by installing its integration.

```
$ herdr integration install cursor
```

— Names every supported agent and the custom API; restart caveat buried in prose v0.9.2

04 **Kitty graphics protocol replaces pane-overlay API** BREAKING

88

Adds Kitty graphics protocol support for rendering pane images in Kitty-compatible outer terminals, controlled by a boolean config key; in remote sessions the server config controls parsing while the local client config controls outer-terminal output. A local Ghostty client optimization sends image data through temporary files rather than encoding it into terminal output, with automatic fallback to inline delivery, and Linux gains copy-on-write snapshot support for whole-file uncompressed RGBA uploads using filesystem cloning into Herdr's private `/var/tmp` store, capped at 16 MiB per snapshot and 64 MiB total. This replaces the external socket pane-overlay API, which is removed with no replacement: `pane.graphics.info`, `pane.graphics.set`, `pane.graphics.clear`, and `pane.graphics.stream` now return `unknown_method`, and applications must write standard Kitty graphics directly to their terminal output instead.

— Removed methods and replacement mechanism both named exactly v0.9.2

05 **Terminal environment identity for new panes** NEW

70

New panes set `TERM_PROGRAM=herdr` and `TERM_PROGRAM_VERSION`, and Herdr strips inherited outer-terminal session markers from iTerm2, WezTerm, Kitty, Windows Terminal, tmux, screen, and Zellij — plus iTerm2's `LC_TERMINAL`, Claude Code, Codex,

and OMP markers — before applying launch environment values, so new panes don't inherit the originating terminal or agent session identity. It also exposes `HERDR_ENV` so shell startup scripts can detect Herdr panes and skip automatic tmux attachment logic.

Prevent recursive tmux shrink/flicker when your shell auto-attaches to tmux — skip attachment inside Herdr panes by checking `HERDR_ENV`.

```
$ if [[ -z ${TMUX:-} && ${HERDR_ENV:-} != 1 ]]; then
  tmux new-session -A -s my-session
fi
```

— Concrete env vars and terminals named; no config key to tune v0.9.2

06 **Goto picker for agents and terminals** NEW 70

A Goto picker lists every agent and terminal grouped by workspace; pressing `/` enters search mode supporting names, agent kinds, tabs, workspaces, branches, and paths, while `b` / `w` / `i` / `d` filter to blocked, working, idle, or done agents respectively and `a` shows all terminals again. Left/Right jump to the first agent or terminal in the previous/next workspace section (or move the text cursor while searching), the selected destination's location and path show below the list, and the list supports mouse-wheel scrolling, scrollbar-track clicks, and thumb dragging when it overflows.

— Full keybinding behavior described but navigation is UI-only v0.9.2

THINNER COVERAGE BELOW

07 **Rolling session layout snapshots** NEW 55

Keeps up to 48 rolling layout snapshots in `session-snapshots/` on all platforms, written at most once every 15 minutes, with the first saved layout copied immediately. Herdr prefers the live shell's OS-reported working directory when available, and retains the last confirmed directory after the shell exits.

— Concrete numbers given but no config key or command to act on product docs

08 **Live terminal color re-reading on resize** IMPROVED 45

Sending `SIGWINCH` to a Herdr client on Unix, or resizing, now re-reads host terminal colors, letting theme switchers that use OSC apply new palettes without restarting panes or requiring a light/dark notification.

— Mechanism named but nothing for reader to configure directly v0.9.2

09 **Event subscription backpressure signal** IMPROVED 45

Event subscriptions now deliver an `events_lost` error to readers that fall too far behind instead of silently dropping events.

— Names the error but gives no handling example v0.9.2

╰──▷ BREAKING ON UPGRADE

- ! After updating the OpenCode integration, shared OpenCode servers and their TUIs must be restarted so the old server reporter cannot overwrite pane-local state.
- ! The external socket pane-overlay API for Kitty graphics is no longer available; applications must now use the standard Kitty graphics protocol through pane terminal output.
- ! The external socket pane-overlay API has been removed; there is no replacement socket image API.
- ! The Herdr-specific pane graphics API is removed: `pane.graphics.info`, `pane.graphics.set`, `pane.graphics.clear`, and `pane.graphics.stream` now return `unknown_method`. Apps must write standard Kitty graphics directly to their terminal instead.

v0.9.2 NOTES ↗ LATEST STABLE

herdr v0.9.2 adds multi-prefix keys, self-reported agent resume, new machine subcommands, and per-agent startup delay config

╰──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.2 https://github.com/herdrdev/herdr.git
# already have the repo? check out this version:
$ git checkout v0.9.2
```

╰──▷ USE IT

Bind multiple prefix keys so both Ctrl+Space and Ctrl+S enter prefix mode — useful when one conflicts with another app.

~/.config/herdr/config.toml

```
prefix = ["ctrl+space", "ctrl+s"]
```

Stagger agent restores to avoid hammering the system after a server restart — set a longer delay between each agent coming back.

~/.config/herdr/config.toml

```
[session]
startup_per_agent_delay_ms = 500
```

Supports multiple prefix keys via `prefix = ["ctrl+space", "ctrl+s"]` in config, with `prefix+?` listing all active prefixes.

Adds `[session] startup_per_agent_delay_ms` config key to control the spacing between restored agents at startup (default 100 ms).

New `herdr machine status` subcommand checks saved machines without prompting; `herdr machine reconnect` completes SSH authentication including MFA in the

terminal.

Interactive `machine add` now discovers existing Herdr sessions on the host for selection; `--label` is optional and defaults to the SSH hostname.

Agents can self-report their resume command so Herdr reopens their exact session after a server restart, with no built-in integration required.

+7 more

└──▷ **BREAKING ON UPGRADE**

! The Herdr-specific pane graphics API is removed: `pane.graphics.info`, `pane.graphics.set`, `pane.graphics.clear`, and `pane.graphics.stream` now return `unknown_method`. Apps must write standard Kitty graphics directly to their terminal instead.

WAS THIS USEFUL? 👍 👎

Trailhq Graft

Feedback ⓘ v0.21.1 CODE ↗ RANK

Graft builds a code knowledge graph that gives coding agents compact, repository-specific context through a CLI and MCP server.

> Graft's v0.21 release overhauls `graft trail push` with chunked gzip uploads, agent-aware context-file targeting, SSE-based build tracking, and agentless sign-up completion, while `graft trail pull` now writes every context-file type and surfaces pending suggestion counts via a new telemetry event.

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

01 **Agentless sign-up completion via `graft trail push`** NEW

90

Enables sign-up completion when a coding agent (Claude Code, Cursor, CI) runs `graft trail push` on an unlinked repo: the first run saves state to `.graft/config.json` and opens the sign-up page; the next run claims the handoff via `POST /api/public/graft-handoffs/claim` and continues the push.

— Names exact config path and claim endpoint, fully runnable flow. v0.21.0

02 **Chunked gzip upload with retry backoff** IMPROVED

80

`graft trail push` uploads the full digest as gzip in 1 MiB chunks, 4 at a time, each retried up to 4 times with 0.5/1/2/4 s backoff on network errors, 5xx, and 429; the `complete` call is retried the same way.

— Precise mechanism and numbers but not a direct user action. v0.21.0

03 **Wider `sources` coverage in push payload** IMPROVED

75

Widens `sources` in the `graft trail push` payload to include folder `CLAUDE.md` / `AGENTS.md` files (depth up to 4, max 30) and `.claude/skills/*/SKILL.md` (max 30), all sent as `agent_instructions` with graft blocks stripped.

— Concrete limits and paths named, mechanism-only otherwise. v0.21.0

04 **`graft trail pull` writes all context file types** IMPROVED

73

Extends `graft trail pull` (and `graft claude-md pull`) to write accepted changes for every context file type — `agents_md`, `folder_claude_md`, `cursor_rule`, and `skill` — not only the root `CLAUDE.md`.

— Names exact command and all four covered file types. v0.21.0

05 **Agent-aware context-file targeting in `graft trail push`**

IMPROVED

71

`graft trail push` now sends `agents` (the wired agent set from `graft init` plus disk state) on both the early and full upload so Trail surfaces only the context-file pages those agents read; the review link now points to the `/brain/<id>/context-files` overview.

— Explains mechanism and names the new review link path. v0.21.0

06 **`trail_pulled` telemetry event** IMPROVED 66

Adds a `trail_pulled` telemetry event emitted when `graft trail pull` finishes, with properties `outcome` (`written` / `already_present` / `nothing_accepted` / `skipped` / `error` / `dry_run`), `kinds` (sorted list of context-file kinds written), `files_bucket`, `changes_bucket`, and `skipped_bucket`. A later release adds a `suggested_bucket` property to differentiate pulls where suggestions were waiting and none were accepted.

— Fully enumerated properties but telemetry isn't user-actionable. v0.21.1 v0.21.0

07 **Build status via SSE with polling fallback** NEW 65

`graft trail push` follows the build over `GET /events` (SSE) with up to 3 reconnects before falling back to polling.

— Names endpoint and reconnect count, thin on broader behaviour. v0.21.0

THINNER COVERAGE BELOW

08 **Suggestion summary line in `graft trail pull`** NEW 58

`graft trail pull` now prints a summary line such as `● 21 suggested so far for the files claude read, 2 of them accepted`, followed by the review link, scoped to the files the wired agents actually read.

— Names exact output string and command, no deeper mechanism given. v0.21.1

v0.21.1 CODE ↗ LATEST STABLE

graft trail pull now shows pending suggestion counts and which agent files they apply to

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.21.1 https://github.com/trailhq/Graft.git
# already have the repo? check out this version:
$ git checkout v0.21.1
```

Adds a suggestions summary line to `graft trail pull` output: prints `● 21 suggested so far for the files claude read, 2 of them accepted`, followed by the review link, cut to the files the wired agents actually read.

Adds `suggested_bucket` property to the `trail_pulled` telemetry event, enabling differentiation between pulls where suggestions were waiting and none were accepted.

v0.21.0 CODE ↗ STABLE

graft trail push gains chunked upload, agent-aware context-file push, and agentless sign-up; trail pull now covers all context file types

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.21.0 https://github.com/trailhq/Graft.git
# already have the repo? check out this version:
$ git checkout v0.21.0
```

Extends `graft trail pull` (and `graft claude-md pull`) to write accepted changes for every context file type — `agents_md`, `folder_claude_md`, `cursor_rule`, and `skill` — not only the root `CLAUDE.md`.

Sends `agents` (the wired agent set from `graft init` plus disk state) on both the early and full upload so Trail surfaces only the context-file pages those agents read; review link now points to the `/brain/<id>/context-files` overview.

Uploads the full digest as gzip in 1 MiB chunks, 4 at a time, each retried up to 4 times with 0.5/1/2/4 s backoff on network errors, 5xx, and 429; `complete` is retried the same way.

Follows the build over `GET /events` (SSE) with up to 3 reconnects before falling back to polling.

Widens `sources` in the push payload to include folder `CLAUDE.md` / `AGENTS.md` files (depth up to 4, max 30) and `.claude/skills/*/SKILL.md` (max 30), all sent as `agent_instructions` with graft blocks stripped.

+2 more

WAS THIS USEFUL? 👍 👎

Charm Crush

Feedback ⓘ v0.97.0 NOTES ↗ CODE ↗ RANK

Terminal-based AI coding assistant that integrates LLMs, LSPs, and MCP servers into your development workflow.

> Crush v0.97.0 adds a sidebar toggle shortcut, the ability to enable or disable MCP servers from the UI, and git branch display in the TUI.

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

01 **Sidebar toggle shortcut** NEW 55

Adds the `ctrl+b` keyboard shortcut to toggle the sidebar on and off.

— Exact keybinding named, immediately usable v0.97.0

02 **MCP server toggling from the UI** NEW 40

MCP servers can now be toggled on/off either globally or locally from within the UI.

— Names the capability and scope but no exact UI path v0.97.0

03 **Git branch display in TUI** NEW 23

The TUI now displays the current git branch.

— Bare description of what changed, no mechanism v0.97.0

v0.97.0 NOTES ↗ CODE ↗ LATEST STABLE

Crush v0.97.0 adds MCP server toggling, git branch display in the TUI, and a ctrl+b sidebar shortcut.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.97.0
https://github.com/charmbracelet/crush.git
# already have the repo? check out this version:
$ git checkout v0.97.0
```

Adds `ctrl+b` keyboard shortcut to toggle the sidebar on and off.

Enables toggling MCP servers on/off globally or locally from the UI.

Displays the current git branch in the TUI.

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ Opus 5.5 (2026-09-28) BLOG ↗

RANK

Amp is Sourcegraph's agentic coding tool for the terminal and editor, running multi-step edits with subagents and shared team threads.

> Amp's biggest addition this window is live desktop sharing for runners, alongside new orb secret-handling (ID token injection and Tailscale credential exchange), GitHub commit-author identity controls, and a default model switch to Claude Opus 5.5 with new mid-task steering.

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

01 **Live desktop sharing in Amp runners** NEW 98

Runners started with `amp --no-tui --desktop` expose a Desktop tab to threads: on macOS this captures the real display via an installed `Amp Desktop Helper.app` (at `~/.amp/bin/amp-desktop-helper/<version>`), and on Linux it spins up a headless `labwc`/Waymote virtual desktop at 1280x720 using auto-downloaded `waymote-gateway` and `waymote-streamd` binaries in `~/.amp/bin/waymote/<waymote-version>`. `amp runner desktop status` inspects which desktop-sharing dependencies are detected, and `amp runner desktop allow screen-recording` / `amp runner desktop allow accessibility` re-check or re-request macOS permissions. On Linux, threads served by a desktop-enabled runner inherit `WAYLAND_DISPLAY`, `DISPLAY`, and `PULSE_SERVER` so launched graphical programs appear in the Desktop tab without capturing the host session, and each runner isolates its compositor, logs, and sockets under `~/.cache/amp/runner-desktop/<runner-id>` so multiple runners can run separate desktops on one machine.

Start an Amp runner that exposes a live Desktop tab to all threads it serves — on macOS this captures your real screen; on Linux it creates an isolated 1280x720 virtual desktop.

```
$ amp --no-tui --desktop
```

Check which desktop-sharing dependencies (labwc, waymote, PulseAudio, etc.) the runner can find before starting a desktop session.

```
$ amp runner desktop status
```

Re-prompt macOS for Screen Recording permission after initially denying it, so the desktop helper can capture and stream the display.

```
$ amp runner desktop allow screen-recording
```

— Full flags, subcommands, paths and env vars given `product docs`

02 **ID token injection and Tailscale exchange for orbs** NEW

78

HTTP MCP servers configured under `amp.mcpServers` can now interpolate `${amp:id-token}` in request headers, giving the server an Amp ID token scoped to its own origin. Orbs can also exchange that Amp ID token for a temporary Tailscale node registration credential, connecting the orb to a tailnet.

Inject an Amp ID token into an MCP server request header so the server can authenticate against its own origin without a stored secret.

```yaml
mcpServers:
  my-server:
    url: "https://my-mcp-server.example.com"
    headers:
      Authorization: "Bearer ${amp:id-token}"
```

— Named config key and interpolation syntax with example `product docs`

03 **Mid-task steering with Enter key** NEW

68

Enter steers is now the default behavior: sending a correction, constraint, or course change while the agent is working reaches the running agent immediately, which folds the update into the current task without stopping and restarting or dropping progress.

— Explains behavior and default clearly, no exact command shown `Opus 5.5 (2026-09-28)`

04 **Commit author identity settings for GitHub commits** NEW

60

A new Commit Author setting under Personal Settings → Project Defaults controls how Amp credits commits in new personal projects, offering You + Amp (user's identity as author, Amp as co-author) or You (user's identity only, no co-author). When Amp is credited, a co-author trailer is added to commits, which can be disabled by setting a named variable as a personal, project, or workspace switch.

— UI path given but toggle variable name omitted `product docs`

05 **Claude Opus 4.5 as default medium mode model** IMPROVED

60

Medium mode's default model switches from GPT-5.6 Sol to Claude Opus 5.5, configurable via Tune Modes. The `ChatGPT Only` preset keeps medium pinned to GPT-5.6 Sol billed to the ChatGPT subscription, and switching away requires changing medium in Tune Modes.

— Names models and settings path, no deeper mechanism Opus 5.5 (2026-09-28)

Opus 5.5 (2026-09-28) BLOG ↗ LATEST STABLE

Amp's medium mode switches to Claude Opus 5.5 by default, with improved steerability and high reasoning effort.

Switches medium mode default model from GPT-5.6 Sol to Claude Opus 5.5, configurable via Tune Modes.

Adds Enter steers behavior by default, so mid-task corrections reach the running agent immediately without stopping and restarting.

Supports mid-task steering: send a correction, constraint, or course change while the agent works and it folds the update into the current task without dropping progress.

ChatGPT Only preset keeps medium pinned to GPT-5.6 Sol billed to the ChatGPT subscription; switching requires changing medium in Tune Modes.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

OpenClaw

Feedback ⓘ v2026.8.33 NOTES ↗ CODE ↗ RANK

OpenClaw is a personal AI assistant platform that connects messaging channels, tools, and model providers through a gateway.

> This release expanded the extended-stable model line with four new models while confirming full retention of the existing GPT-5.6 family's behavior.

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

01 **New models in extended-stable line** NEW 40

Adds `Meta Muse Spark 1.3`, `Anthropic Fable 5.1`, `OpenAI GPT-6 Astra`, and `GPT Image 2.5` to the extended-stable line.

— Names four new models but no usage detail or mechanism. v2026.8.33

02 **GPT-5.6 family behavior preserved** IMPROVED 40

Preserves the complete GPT-5.6 family (Sol, Terra, Luna) catalog, routing, migration, image, vision, and thinking behavior — including Ultra reasoning across model-runtime boundaries and Bedrock tool-result images — in the extended-stable line.

— Describes retained behaviors but no new action for readers. v2026.8.33

v2026.8.33 NOTES ↗ CODE ↗ LATEST STABLE

Adds Meta Muse Spark 1.3, Anthropic Fable 5.1, OpenAI GPT-6 Astra, and GPT Image 2.5 to the extended-stable line.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.8.33
https://github.com/openclaw/openclaw.git
# already have the repo? check out this version:
$ git checkout v2026.8.33
```

Preserves complete GPT-5.6 family (Sol, Terra, Luna) catalog, routing, migration, image, vision, and thinking behavior — including Ultra reasoning across model-runtime boundaries and Bedrock tool-result images — in the extended-stable line.

WAS THIS USEFUL? 👍 👎

CrewAI

Feedback ⓘ 1.15.23 NOTES ↗ RANK

Open-source Python framework for building multi-agent workflows using role-based AI agent Crews and event-driven Flows.

> CrewAI 1.15.23 adds Gemini 2.0 Flash model support, AMP-backed run evaluation via `crewai eval`, and richer task-span tracing.

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

01 **AMP-backed `crewai eval` run evaluation** IMPROVED 60

The `crewai eval` command now evaluates the last traced run through AMP, recording the run instead of merely printing it to the console.

— Names the exact command and mechanism change 1.15.23

THINNER COVERAGE BELOW

02 **Richer task tracing spans** IMPROVED 35

Task tracing spans now include the declared output format and results, enhancing visibility into task execution.

— States what changed but no config/API detail 1.15.23

03 **Gemini 2.0 Flash model support** NEW 25

CrewAI 1.15.23 adds support for Gemini 2.0 Flash, per the release summary.

— Mentioned only in summary, no further detail given 1.15.23

1.15.23 NOTES ↗ LATEST STABLE

CrewAI 1.15.23 adds Gemini 2.0 Flash support, AMP-backed `crewai eval` run evaluation, and richer task-span tracing.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 1.15.23
https://github.com/crewAIInc/crewAI.git
# already have the repo? check out this version:
$ git checkout 1.15.23
```

Implements `crewai eval` evaluation of the last traced run through AMP, recording the run instead of printing it.

Enhances task tracing spans to include declared output format and results.

WAS THIS USEFUL? 👍 👎

LangChain

Feedback ⓘ langchain==1.4.3 NOTES ↗ RANK

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

LangChain 1.4.3 is a patch release; the sole new capability is Bedrock Mantle support in `init_chat_model`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch langchain==1.4.3 https://github.com/langchain-ai/langchain.git
# already have the repo? check out this version:
$ git checkout langchain==1.4.3
```

NEW INTEGRATION Bedrock Mantle chat model support in init_chat_model

WAS THIS USEFUL? 👍 👎

CopilotKit

Feedback ⓘ v1.75.0 NOTES ↗ RANK

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

> CopilotKit v1.75.0 adds a message transformation hook for chat rendering and bumps a virtualization dependency that requires a manual update to avoid a scroll glitch.

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

01 Upgraded @tanstack/react-virtual dependency requirement

75

BREAKING

`@copilotkit/react-core` now requires `@tanstack/react-virtual` `^3.14.13`, up from `^3.13.0`. Lockfiles pinned to the old range must be updated (e.g. `pnpm update @tanstack/react-virtual`) or a one-frame scroll twitch persists in long chats.

– Gives exact package, version range and migration command v1.75.0

02 transformMessages prop on CopilotChatMessageView NEW

60

`@copilotkit/react-core` adds a `transformMessages` prop to `CopilotChatMessageView` for intercepting and transforming chat messages before they are rendered.

– Names exact prop and component but no usage example given v1.75.0

└──▷ BREAKING ON UPGRADE

! `@copilotkit/react-core` now requires `@tanstack/react-virtual` `^3.14.13`; lockfiles pinned to the old `^3.13.0` range must be updated (e.g. `pnpm update @tanstack/react-virtual`) or a one-frame scroll twitch persists in long chats.

v1.75.0 NOTES ↗ LATEST STABLE

CopilotKit v1.75.0 adds `transformMessages` to `CopilotChatMessageView` for runtime message transformation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.75.0
https://github.com/CopilotKit/CopilotKit.git
# already have the repo? check out this version:
$ git checkout v1.75.0
```

Adds `transformMessages` prop to `CopilotChatMessageView` in `@copilotkit/react-core` for intercepting and transforming chat messages before rendering.

└──▷ BREAKING ON UPGRADE

! `@copilotkit/react-core` now requires `@tanstack/react-virtual` `^3.14.13`; lockfiles pinned to the old `^3.13.0` range must be updated (e.g. `pnpm update @tanstack/react-virtual`) or a one-frame scroll twitch persists in long chats.

WAS THIS USEFUL?  

Vercel AI SDK

Feedback ⓘ ai@7.0.119 NOTES ↗ RANK

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

> AI SDK 7.0.119 refines image model capability declarations, confirming file and mask input support for several providers and correcting one unsupported case.

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

01 **Image model input capability declarations updated** IMPROVED

50

Advertises file and mask input support for `Together AI FLUX.2 Pro` and Flex (single-image editing) via confirmed model ID capability declarations, and file-input support for `QuiverAI Arrow 2` and Arrow 2 Telos image models. Also marks Together AI Gemini image inputs as unsupported under the current single-image request mapping.

— Names exact models but no usage or code example given ai@7.0.119

ai@7.0.119 NOTES ↗ LATEST STABLE

AI SDK 7.0.119 adds image model file and mask input capability declarations for Together AI FLUX.2 and QuiverAI Arrow 2.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch ai@7.0.119 https://github.com/vercel/ai.git
# already have the repo? check out this version:
$ git checkout ai@7.0.119
```

Advertises image model file and mask input support for `Together AI FLUX.2 Pro` and Flex (single-image editing) via confirmed model ID capability declarations.

Advertises file-input support for `QuiverAI Arrow 2` and Arrow 2 Telos image models.

Marks Together AI Gemini image inputs as unsupported under the current single-image request mapping.

WAS THIS USEFUL? 👍 👎

Databricks Omnigent

Feedback ⓘ v0.15.0 NOTES ↗ RANK

Open-source meta-harness that orchestrates Claude Code, Codex, Cursor, Pi and custom agents behind one interface, with policy controls, OS-level sandboxing and live session sharing.

> Across four releases Omnigent expanded its harness lineup (Devin, Jcode, Antigravity, Grok Build), added native Kubernetes and libkrun microVM sandbox providers, a built-in Canvas board, deeper GitHub PR tracking, and reverse-proxy deployment support, alongside breaking changes to host launch behavior and terminal transport.

└──▷ **WHAT SHIPPED · 25 FEATURES** most completely described first ? what's the number?

01 New agent harness integrations NEW

85

Adds `omni devin` as a native harness, mirroring the real Devin TUI inside the Omnigent web UI alongside Claude Code and Codex. Adds Jcode as a first-class ACP harness, and an `EXTRA_HARNESS_CLIS` build-arg to bake extra harness CLIs (goose, jcode, cursor, kimi, ...) into the host image without forking the Dockerfile. Adds an `omni agy` shortcut and `agy-native` aliases for the new Antigravity harness, adds Devin and Grok Build to the New Chat model picker, and surfaces ACP sub-agents (e.g. Devin) as child sessions with their tool calls visible in the transcript.

Launch a Devin session as a native harness to get the full Devin TUI experience inside the Omnigent web UI.

```
$ omni devin
```

Bake Jcode and Goose CLIs into a custom host image at build time so agents can invoke them without modifying the upstream Dockerfile.

```
$ docker build --build-arg EXTRA_HARNESS_CLIS='jcode goose' -t my-omnigent-host .
```

— Multiple named commands and a build-arg, but spans several thin additions. v0.15.0 v0.13.0 v0.12.0

02 Local libkrun microVM sandbox provider NEW

85

Adds `sandbox.provider: microsandbox` to run each agent sandbox in a local libkrun microVM with sub-second boots and in-place resume.

Run agent sandboxes in lightweight local microVMs instead of containers — useful when you want fast, isolated environments without a full Kubernetes cluster.

```yaml
sandbox.provider: microsandbox
```

— Exact config key with concrete performance behaviour and example. v0.13.0

03 **Host service and launch behavior changes** BREAKING 85

Adds `omni host enable` and `omni host disable` commands to install a per-user macOS or Linux background service for the Omnigent host, plus `omnigent host` and `omnigent start` flags to auto-open the web UI on launch. Bare `omni` now starts the background host instead of launching the polly agent — run `omni run` to launch the default agent instead.

Install Omnigent as a persistent per-user service on macOS or Linux so the host starts automatically on login.

```
$ omni host enable
```

— Exact commands plus clear before/after migration instruction. v0.12.0

04 **Reverse-proxy subpath support** NEW 80

Adds the `OMNIGENT_WEB_BASE_PATH` environment variable and `omnigent server --base-path` flag to serve the web UI and API behind a reverse-proxy subpath.

Run Omnigent behind an nginx reverse proxy at a subpath so the UI and API are accessible at `/omnigent` without port exposure.

```
$ OMNIGENT_WEB_BASE_PATH=/omnigent omnigent server --base-path /omnigent
```

— Exact env var, flag and runnable command given. v0.15.0

05 **Built-in Canvas page** BREAKING 80

Adds a built-in Canvas page — enable with `OMNIGENT_FEATURES=canvas` — providing a per-project board of draggable session cards with live status, worktree branches, and PR links. The standalone `omnigent-canvas` extension is removed; teams must switch to this built-in flag.

Enable the built-in Canvas board for a team that was previously using the standalone omnigent-canvas extension.

```env
OMNIGENT_FEATURES=canvas
```

— Exact enabling flag and clear migration path from removed extension. v0.13.0

06 **Kubernetes agent-sandbox infrastructure** NEW 65

Adds self-reclaiming Kubernetes agent-sandbox resources that stay alive while an agent runs, suspend when idle, and resume with their workspace intact; offline sandboxes can be reaped automatically. Runs Kubernetes sandboxes on dedicated node pools using tolerations.

— Explains full sandbox lifecycle but no exact config key shown. v0.13.0 v0.14.0

07 **GitHub panel in session workspace** NEW 65

Adds a GitHub panel in the session workspace rail showing the branch's PR summary, comments, CI checks, and a stacked diff with a file-tree navigator, refreshing live as the agent pushes.

— Detailed panel contents but only a UI path, no command. v0.13.0

08 **Composer and session-finding UI improvements** IMPROVED 60

Adds image preview for attached files in the Composer before sending, broader file-type attachment support for Claude Code and Codex sessions, and skill discovery via `/` (or `$` for Codex) from the new session page. Also adds fuzzy search for finding sessions and a keyboard-accessible model picker in the redesigned composer.

— Names trigger characters but mostly describes UI, not a command. v0.14.0 v0.15.0

09 **Claude SDK context-window usage and /compact** NEW 60

Shows context-window usage in Claude SDK sessions and supports `/compact` for native compaction, including with Claude Pro/Max logins.

— Names the exact `/compact` command and login scope. v0.14.0

10 **Pi harness setup and control options** NEW 60

Adds a `/effort` mid-session command for the pi harness to change reasoning-effort without starting a new session, and a Pi — use own login option during setup for the pi harness.

Switch the pi harness reasoning effort mid-session without restarting, useful when a task changes complexity.



— Exact command with runnable example plus a setup option. v0.12.0

THINNER COVERAGE BELOW

11 **Side chats inside sessions** NEW 55

Adds side chats inside sessions, started from the '+' button in the chat pane's right panel; the side chat sees session context without polluting the main session.

— Clear UI path and behaviour, no config surface named. v0.15.0

12 **Curated model shortlists across harnesses** NEW 55

Adds curated model shortlists (catalogs) to Pi, custom ACP agents, and managed Kubernetes / Agent Sandbox sessions, with per-harness gateways and optional live gateway discovery.

— Names several surfaces but no concrete config or command. v0.15.0

13 **Session workspace organization polish** IMPROVED 55

Adds session export from the actions menu and manual project ordering. New sessions are later auto-named from the first prompt (toggle in Settings), long transcripts virtualize for fast switching, and a session can be moved into a project via its title's folder tag.

— Several small UI conveniences named individually, no single command. v0.15.0 v0.13.0 v0.12.0

14 **GitHub PR tracking in sessions** IMPROVED 55

Tracks multiple GitHub PRs in one session — switch between summaries and diffs, and link or unlink PRs manually — and shows PR titles in the session PR dropdown.

— Describes mechanism for switching/linking PRs, no API named. v0.14.0 v0.15.0

15 **Terminal transport requirements and changes** BREAKING 55

Managed terminals now require tmux 3.3 or newer — Omnigent warns during setup and fails at terminal launch on older versions. Web terminals moved to a single control-mode attach path; legacy PTY transport configuration has been removed.

— States the version requirement and transport change but no migration command. v0.12.0 v0.13.0

16 **Project-scoped session defaults** NEW 55

Adds a `project_id` parameter at session creation to inherit the project's agent, workspace, and git defaults.

— Names the parameter and what it inherits, no full API shown. v0.12.0

17 **Import of existing local CLI sessions** NEW 50

Adds import of existing local CLI sessions (Claude Code, Codex, and others) into Omnigent from the web UI, preserving original titles.

— Names source CLIs and behaviour, clear UI entry point. v0.12.0

18 **Message deeplinks** NEW 45

Adds message deeplinks so any message can be copied as a direct link, with jump-to-cited-file-line support.

— Describes behaviour but no exact command or endpoint. v0.15.0

19 **Managed sandbox provisioning enhancements** NEW 45

Supports provisioning managed sandboxes through Gensee, and starts managed sandboxes with multiple repositories and branches cloned in parallel.

— Names Gensee provider but no config key or command. v0.14.0

20 **Enterprise authentication and credential storage options** NEW 45

Adds an enterprise opt-in OAuth 2.0 client-credentials grant for authentication, and a HashiCorp Vault Transit credential-store backend for secret management.

— Names the auth grant and Vault backend but no config keys given. v0.13.0

21 **Session forking with independent settings** NEW 45

Adds ability to fork a session with its own model, reasoning effort, and approval mode.

— States what can be changed on fork but no command shown. v0.12.0

22 **UI customization and mobile UX additions** NEW 45

Adds a configurable Enter/⌘`-Enter` send shortcut, appearance export/import, a heavier code-font option, a markdown table of contents, a ChatGPT-style mobile drawer, and full-screen mobile search to the UI.

— Lists several UI tweaks by name but no config paths. v0.12.0

23 **Per-token custom pricing for self-hosted models** NEW

40

Adds per-token custom pricing configuration for self-hosted models such as Ollama and vLLM.

— Names supported models but no config key or command. v0.12.0

24 **Task badge for background commands** NEW

35

Adds a task badge for inspecting running background commands from the chat interface.

— Bare description of a UI element, no mechanism detail. v0.14.0

25 **CockroachDB application database support** NEW

30

Supports CockroachDB as Omnigent's application database.

— One-line mention with no configuration detail. v0.14.0

└──▷ BREAKING ON UPGRADE

! Managed terminals now require tmux 3.3 or newer — Omnigent warns during setup and fails at terminal launch on older versions.

! The standalone `omnigent-canvas` extension is removed; enable the built-in Canvas page with `OMNIGENT_FEATURES=canvas` instead.

! Bare `omni` now starts the background host instead of launching the polly agent — run `omni run` to launch the default agent.

! Web terminals moved to a single control-mode attach path; legacy PTY transport configuration has been removed.

v0.15.0 NOTES ↗ STABLE

Omnigent v0.15.0 adds a Devin native harness, side chats, message deeplinks, and reverse-proxy subpath support via `OMNIGENT_WEB_BASE_PATH`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.15.0 https://github.com/omnigent-ai/omnigent.git
# already have the repo? check out this version:
$ git checkout v0.15.0
```

└──▷ TRY IT

Run Omnigent behind an nginx reverse proxy at a subpath so the UI and API are accessible at `/omnigent` without port exposure.

```
$ OMNIGENT_WEB_BASE_PATH=/omnigent omnigent server --base-path
/omnigent
```

Launch a Devin session as a native harness to get the full Devin TUI experience inside the Omnigent web UI.

```
$ omni devin
```

Adds `OMNIGENT_WEB_BASE_PATH` environment variable and `omnigent server --base-path` flag to serve the web UI and API behind a reverse-proxy subpath.

Adds `omni devin` native harness, mirroring the real Devin TUI into the web UI alongside the existing Claude Code and Codex harnesses.

Adds side chats inside sessions, started from the '+' button in the chat pane's right panel — the side chat sees session context without polluting it.

Adds message deeplinks so any message can be copied as a direct link, with jump-to-cited-file-line support.

Adds curated model shortlists (catalogs) to Pi, custom ACP agents, and managed Kubernetes / Agent Sandbox sessions, with per-harness gateways and optional live gateway discovery.

+2 more

v0.14.0 NOTES ↗ STABLE

Omnigent v0.14.0 adds multi-repo PR tracking, Gensee-provisioned sandboxes, Kubernetes node-pool tolerations, and CockroachDB support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.14.0 https://github.com/omnigent-
ai/omnigent.git
# already have the repo? check out this version:
$ git checkout v0.14.0
```

Supports provisioning managed sandboxes through Gensee.

Runs Kubernetes sandboxes on dedicated node pools using tolerations.

Supports CockroachDB as Omnigent's application database.

Tracks multiple GitHub PRs in one session — switch between summaries and diffs, and link or unlink PRs manually.

Starts managed sandboxes with multiple repositories and branches, cloned in parallel.

+3 more

v0.13.0 NOTES ↗ STABLE

Omnigent v0.13.0 adds a GitHub PR panel, built-in Canvas board, libkrun/Kubernetes sandbox providers, and OAuth 2.0 + Vault credential backends.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.13.0 https://github.com/omnigent-ai/omnigent.git
# already have the repo? check out this version:
$ git checkout v0.13.0
```

└──▷ USE IT

Run agent sandboxes in lightweight local microVMs instead of containers — useful when you want fast, isolated environments without a full Kubernetes cluster.

```yaml
sandbox.provider: microsandbox
```

Bake Jcode and Goose CLIs into a custom host image at build time so agents can invoke them without modifying the upstream Dockerfile.

```
$ docker build --build-arg EXTRA_HARNESS_CLIS='jcode goose' -t my-omnigent-host .
```

Enable the built-in Canvas board for a team that was previously using the standalone omnigent-canvas extension.

```.env
OMNIGENT_FEATURES=canvas
```

Adds `sandbox.provider: microsandbox` to run each agent sandbox in a local libkrun microVM with sub-second boots and in-place resume.

Adds self-reclaiming Kubernetes agent-sandbox resources that stay alive while an agent runs, suspend when idle, and resume with their workspace intact; offline sandboxes can be reaped automatically.

Adds `EXTRA_HARNESS_CLIS` build-arg to bake extra harness CLIs (goose, jcode, cursor,

kimi, ...) into the host image without forking the Dockerfile.

Adds an enterprise opt-in OAuth 2.0 client-credentials grant for authentication.

Adds a HashiCorp Vault Transit credential-store backend for secret management.

+4 more

└─▷ **BREAKING ON UPGRADE**

! Managed terminals now require tmux 3.3 or newer — Omnigent warns during setup and fails at terminal launch on older versions.

! The standalone `omnigent-canvas` extension is removed; enable the built-in Canvas page with `OMNIGENT_FEATURES=canvas` instead.

v0.12.0 NOTES ↗ STABLE

Omnigent v0.12.0 adds CLI session import, project-scoped session defaults, Antigravity/Devin/Grok harnesses, and host-as-a-service commands.

└─▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.12.0 https://github.com/omnigent-ai/omnigent.git
# already have the repo? check out this version:
$ git checkout v0.12.0
```

└─▷ **TRY IT**

Install Omnigent as a persistent per-user service on macOS or Linux so the host starts automatically on login.

```
$ omni host enable
```

Switch the pi harness reasoning effort mid-session without restarting, useful when a task changes complexity.

```
$ /effort high
```

Adds `omni host enable` and `omni host disable` commands to install a per-user macOS or Linux background service for the Omnigent host.

Adds `omni run` as the explicit command to launch the default agent (previously triggered by bare `omni`).

Adds `omni agy` shortcut and `agy-native` aliases for the new Antigravity harness.

Adds `project_id` parameter at session creation to inherit the project's agent, workspace, and git defaults.

Adds `/effort` mid-session command for the pi harness to change reasoning-effort without starting a new session.

+9 more

└──▷ BREAKING ON UPGRADE

- **!** Bare `omni` now starts the background host instead of launching the polly agent — run `omni run` to launch the default agent.
- **!** Web terminals moved to a single control-mode attach path; legacy PTY transport configuration has been removed.

WAS THIS USEFUL? 👍 👎

A2A Python SDK

Feedback ⓘ v1.2.0 NOTES ↗ RANK

The A2A Python SDK implements the Agent2Agent protocol for discovering agents and exchanging tasks and messages.

> A2A Python SDK v1.2.0 adds horizontally-scaled cluster mode, opt-in media-type validation against the agent card, an `application/a2a+json` REST content type, and caching headers on the agent card endpoint.

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

01 **REST responses use application/a2a+json content type** BREAKING

52

HTTP+JSON REST responses are now served with the `application/a2a+json` content type instead of `application/json`.

– Names exact before/after content-type values v1.2.0

02 **Caching headers on agent card endpoint** IMPROVED

37

Caching headers were added to the agent card endpoint to reduce redundant fetches.

– Names endpoint but not specific header values v1.2.0

03 **Opt-in media type validation against agent card** NEW

33

The server can now validate incoming message media types against the agent card, as an opt-in check.

– Describes behaviour but not the enabling flag v1.2.0

04 **Multi-replica cluster mode for servers** NEW

23

The server gains multi-replica cluster mode intended for horizontally scaled deployments.

– No config keys or flags named for enabling it v1.2.0

v1.2.0 NOTES ↗ LATEST STABLE

A2A Python SDK v1.2.0 adds multi-replica cluster mode, media-type validation, caching headers, and application/a2a+json REST responses.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v1.2.0 https://github.com/a2aproject/a2a-python.git
# already have the repo? check out this version:
$ git checkout v1.2.0
```

Adds multi-replica cluster mode to the server for horizontally scaled deployments.

Adds opt-in validation of message media types against the agent card on the server.

Serves HTTP+JSON REST responses with `application/a2a+json` content type instead of `application/json`.

Adds caching headers to the agent card endpoint to reduce redundant fetches.

WAS THIS USEFUL? 👍 👎

Stacklok Mecatl

Feedback ⓘ v0.0.42 NOTES ↗ RANK

Mecatl is an open-source agent harness for running the same provider-agnostic agent loop locally, remotely, or on Kubernetes, with durable state, permissions, and gRPC/HTTP-SSE APIs.

> Mecatl v0.0.42 introduces microVM-isolated execution environments with fine-grained egress controls, alongside a draft native Kubernetes execution provider, operator-level scheduling limits, and a handful of UI and SDK updates.

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

01 **MicroVM execution environments and controls** NEW 95

Adds microVM execution environments that isolate filesystem tools and Bash inside a VM while keeping model providers and credentials on the host (Linux amd64 only). Includes `mecated microvm doctor` and `mecated microvm status` read-only administration commands, `execution.default_placement: microvm-local` set in `~/.config/mecatl/settings.yaml`, `mecated serve --headless --default-placement microvm-local` for unattended headless operation, and the `execution.microvm.guest_egress` config key (values `deny-all`, `allowlist`) alongside `mecated serve` flags `--microvm-guest-egress=deny-all|allowlist` and repeatable `--microvm-guest-allow=HOST:PORT/tcp|udp` to control outbound network policy from microVM guests.

Verify your host is ready for local microVM execution before starting an isolated session environment on Linux amd64.

```
$ mecated microvm doctor
```

Configure all sessions to run inside local microVMs by default, so filesystem tools and Bash are isolated from the host.

```yaml
execution:
  default_placement: microvm-local
```

Run a headless microVM server that restricts guest egress to a specific host, preventing agents from reaching arbitrary external endpoints.

```
$ mecated serve --headless --default-placement microvm-local --microvm-guest-egress=allowlist --microvm-guest-allow=api.example.com:443/tcp
```

— Full command, config key, and flag set named with mechanism v0.0.42

THINNER COVERAGE BELOW

02 **UI improvements in mecatui and Studio** IMPROVED 40

Bounds sessions, agent definition, and skills inventory viewports in `mecatui` to improve navigation across large inventories, and shows session delegation activity in Studio.

— Names UI areas affected but no exact navigation path v0.0.42

03 **Operator scheduling profile constraints** NEW 30

Constrains executor scheduling to operator profiles, limiting which execution profiles a given operator deployment may schedule.

— Describes effect but no config keys or flags named v0.0.42

04 **Native Kubernetes execution provider (draft)** NEW 25

Adds a draft native Kubernetes execution provider for running agent sessions natively on Kubernetes.

— Only named as a draft, no mechanism or config given v0.0.42

05 **Configurable commit co-author guidance** NEW 25

Adds configurable Mecatl commit co-author guidance to the prompt subsystem.

— No config surface or example given for how to configure it v0.0.42

06 **TypeScript SDK v0.4.0** NEW 20

Releases TypeScript SDK v0.4.0.

— Bare version bump with no changelog detail v0.0.42

v0.0.42 NOTES ↗ LATEST STABLE

Mecatl v0.0.42 adds microVM execution environments, a native Kubernetes execution provider draft, operator scheduling profiles, and configurable commit co-author guidance.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.0.42 https://github.com/stacklok/mecatl.git
# already have the repo? check out this version:
$ git checkout v0.0.42
```

└──▷ TRY IT

Verify your host is ready for local microVM execution before starting an isolated session environment on Linux amd64.

```
$ mecated microvm doctor
```

Configure all sessions to run inside local microVMs by default, so filesystem tools and Bash are isolated from the host.

~/.config/mecatl/settings.yaml

```
execution:
  default_placement: microvm-local
```

Run a headless microVM server that restricts guest egress to a specific host, preventing agents from reaching arbitrary external endpoints.

```
$ mecated serve --headless --default-placement microvm-local --
microvm-guest-egress=allowlist --microvm-guest-
allow=api.example.com:443/tcp
```

Adds `mecated microvm doctor` and `mecated microvm status` read-only administration commands for the local microVM execution environment.

Adds `execution.default_placement: microvm-local` to `~/.config/mecatl/settings.yaml` to route sessions into isolated local microVM environments.

Adds `mecated serve --headless --default-placement microvm-local` for unattended headless microVM server operation.

Adds `execution.microvm.guest_egress` config key (values: `deny-all`, `allowlist`) and `mecated serve` flags `--microvm-guest-egress=deny-all|allowlist` and repeatable `--microvm-guest-allow=HOST:PORT/tcp|udp` to control outbound network policy from microVM guests.

Adds microVM execution environments, isolating filesystem tools and Bash inside a VM while keeping model providers and credentials on the host (Linux amd64 only).

+6 more

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

MCP TypeScript SDK

Feedback ⓘ v2.2.0 NOTES ↗ RANK

The MCP TypeScript SDK provides TypeScript clients, servers, and transports for Model Context Protocol.

> MCP TypeScript SDK hardened machine-to-machine OAuth by binding credentials and tokens to a specific issuing authorization server, and added auto-pagination for list calls across three releases in this window.

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

01 **Issuer binding for M2M OAuth providers** BREAKING 95

Adds an `expectedIssuer` parameter to `ClientCredentialsProvider`, `PrivateKeyJwtProvider`, `StaticPrivateKeyJwtProvider`, and `CrossAppAccessProvider` to bind M2M OAuth credentials to a specific authorization server; fetchToken() now throws `AuthorizationServerMismatchError` when the provider's client information is bound to a different authorization server. An `issuer` field was added to `OAuthTokensSchema` and `OAuthClientInformationSchema` so stored tokens and client information retain the binding. Omitting `expectedIssuer` when constructing these providers is now deprecated and logs a `console.warn`; as a breaking change, storage backends that reject unknown fields must be updated to allow the new `issuer` field.

Bind M2M credentials to a specific authorization server so mismatched tokens are rejected before any network request is sent.

typescript

```
import { ClientCredentialsProvider } from
'@modelcontextprotocol/client';

const provider = new ClientCredentialsProvider({
  clientId: 'my-service',
  clientSecret: process.env.CLIENT_SECRET,
  expectedIssuer: 'https://auth.example.com'
});
```

Bind an OAuth credential provider to a specific authorization server so fetchToken() rejects mismatched servers before sending credentials.

```typescript
import { ClientCredentialsProvider } from
'@modelcontextprotocol/client';

const provider = new ClientCredentialsProvider({
  clientId: 'my-client',
  clientSecret: 'my-secret',
  expectedIssuer: 'https://auth.example.com'
});
```

Prevent OAuth token reuse across authorization servers by supplying `expectedIssuer` when building a credentials provider.

```typescript
import { ClientCredentialsProvider } from
'@modelcontextprotocol/client';

const provider = new ClientCredentialsProvider({
  clientId: 'my-client-id',
  clientSecret: 'my-client-secret',
  expectedIssuer: 'https://auth.example.com'
});
```

— Names every provider, error class, schema field and deprecation path v2.2.0

@modelcontextprotocol/client@2.2.0 1.31.0

02 **Auto-pagination for list calls** IMPROVED 80

listTools(), listPrompts(), listResources(), and listResourceTemplates() called without a cursor now auto-paginate by following `nextCursor` until exhausted, with duplicate-page detection and a `listMaxPages` cap to bound the walk.

Retrieve the full tool list from a server with many pages without manually tracking cursors.

```typescript
import { Client } from '@modelcontextprotocol/client';

// listTools() now follows nextCursor automatically;
listMaxPages caps the walk
const allTools = await client.listTools();
console.log(allTools.tools.length);
```

Retrieve all tools across a paginated server without manually tracking cursors — the client now auto-paginates until `nextCursor` is absent.

typescript

```typescript
import { Client } from '@modelcontextprotocol/client';

const client = new Client(/* transport */);
const allTools = await client.listTools(); // follows nextCursor automatically
```

— Names methods and the capping parameter with a runnable example

`@modelcontextprotocol/client@2.2.0` `v2.2.0`

THINNER COVERAGE BELOW

03 **.localhost loopback for OAuth HTTPS guard** NEW 55

Treats hostnames ending in `.localhost` as loopback for the OAuth token-endpoint HTTPS guard (per RFC 6761 §6.3), enabling host-based multi-tenant local OAuth flows.

— Names the mechanism but no runnable example given `@modelcontextprotocol/client@2.2.0`

└──▷ BREAKING ON UPGRADE

! Stored OAuth tokens and client information now include an `issuer` field — storage backends that reject unknown fields must be updated to allow it.

! Constructing `ClientCredentialsProvider`, `PrivateKeyJwtProvider`, or `StaticPrivateKeyJwtProvider` without `expectedIssuer` is deprecated.

v2.2.0 NOTES ↗ LATEST STABLE

MCP TypeScript SDK v2.2.0: auto-paginating list calls and stricter machine-to-machine OAuth issuer validation

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.2.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout v2.2.0
```

└──▷ USE IT

Bind M2M credentials to a specific authorization server so mismatched tokens are rejected before any network request is sent.

```typescript
import { ClientCredentialsProvider } from
'@modelcontextprotocol/client';

const provider = new ClientCredentialsProvider({
  clientId: 'my-service',
  clientSecret: process.env.CLIENT_SECRET,
  expectedIssuer: 'https://auth.example.com'
});
```

Retrieve the full tool list from a server with many pages without manually tracking cursors.

```typescript
import { Client } from '@modelcontextprotocol/client';

// listTools() now follows nextCursor automatically; listMaxPages
caps the walk
const allTools = await client.listTools();
console.log(allTools.tools.length);
```

Adds `expectedIssuer` parameter to `ClientCredentialsProvider`, `PrivateKeyJwtProvider`, `StaticPrivateKeyJwtProvider`, and `CrossAppAccessProvider` to bind M2M OAuth credentials to a specific authorization server; omitting it is now deprecated and logs a warning.

fetchToken() now throws `AuthorizationServerMismatchError` when the provider's client information is bound to a different authorization server, preventing cross-server credential misuse.

`@modelcontextprotocol/client@2.2.0` NOTES ↗ STABLE

MCP TypeScript SDK 2.2.0 hardens OAuth provider security with issuer binding and `AuthorizationServerMismatchError`

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @modelcontextprotocol/client@2.2.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout @modelcontextprotocol/client@2.2.0
```

└──▷ USE IT

Bind an OAuth credential provider to a specific authorization server so fetchToken() rejects mismatched servers before sending credentials.

```typescript
import { ClientCredentialsProvider } from
'@modelcontextprotocol/client';

const provider = new ClientCredentialsProvider({
  clientId: 'my-client',
  clientSecret: 'my-secret',
  expectedIssuer: 'https://auth.example.com'
});
```

Retrieve all tools across a paginated server without manually tracking cursors — the client now auto-paginates until `nextCursor` is absent.

```typescript
import { Client } from '@modelcontextprotocol/client';

const client = new Client(/* transport */);
const allTools = await client.listTools(); // follows nextCursor
automatically
```

Adds `issuer` stamp support to `OAuthTokensSchema` and `OAuthClientInformationSchema`, so a provider that reads storage back through these schemas retains the issuer binding.

Deprecates constructing `ClientCredentialsProvider`, `PrivateKeyJwtProvider`, `StaticPrivateKeyJwtProvider`, or `CrossAppAccessProvider` without `expectedIssuer`; the constructor now emits a `console.warn` for the deprecated call signature.

listTools(), listPrompts(), listResources(), and listResourceTemplates() called without a cursor now auto-paginate by following `nextCursor` until exhausted, with duplicate-page detection and `listMaxPages` cap.

Treats hostnames ending in `.localhost` as loopback for the OAuth token-endpoint HTTPS guard (per RFC 6761 §6.3), enabling host-based multi-tenant local OAuth flows.

1.31.0 NOTES ↗ STABLE

MCP TypeScript SDK 1.31.0 binds stored OAuth tokens to their issuing authorization server via a new `issuer` field and `expectedIssuer` parameter.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 1.31.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout 1.31.0
```

└──▷ USE IT

Prevent OAuth token reuse across authorization servers by supplying `expectedIssuer` when building a credentials provider.

```typescript
import { ClientCredentialsProvider } from
'@modelcontextprotocol/client';

const provider = new ClientCredentialsProvider({
  clientId: 'my-client-id',
  clientSecret: 'my-client-secret',
  expectedIssuer: 'https://auth.example.com'
});
```

Adds `issuer` field to stored OAuth tokens and client information to record which authorization server issued them.

└──▷ BREAKING ON UPGRADE

- **!** Stored OAuth tokens and client information now include an `issuer` field — storage backends that reject unknown fields must be updated to allow it.
- **!** Constructing `ClientCredentialsProvider`, `PrivateKeyJwtProvider`, or `StaticPrivateKeyJwtProvider` without `expectedIssuer` is deprecated.

WAS THIS USEFUL? 👍 👎

gram

Feedback ⓘ server@2.13.0 NOTES ↗ RANK

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

> Gram shipped public/private network traffic visibility for MCP servers, giving admins hourly telemetry and a dashboard panel to verify client routing before switching servers between public and private.

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

01 **Public/private traffic visibility for MCP servers** NEW

78

A new `mcp_network_traffic_hourly_summaries` table stores hourly totals for 90 days, tracking observed public vs. private inbound MCP request counts per server, fed by an `mcp_network_request` telemetry log written on each resolved inbound MCP request to hosted, remote, tunneled, or stored gateway endpoints (carrying server ID and network surface, but no tool URN, so it never counts as a tool call). The `telemetry.getMcpNetworkTraffic` API returns zero-filled hourly points for a 24h or 7d window plus the last-seen timestamp for each network route, and this data surfaces in the Network access panel for MCP servers and gateways so admins can verify client routing before switching between public and private.

— Names table, API method, and log fields; no UI path given. server@2.13.0 dashboard@0.127.0

server@2.13.0 NOTES ↗ LATEST STABLE

Adds public/private traffic visibility for MCP servers with `mcp_network_traffic_hourly_summaries` table and `telemetry.getMcpNetworkTraffic` API.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch server@2.13.0 https://github.com/speakeasy-api/gram.git
# already have the repo? check out this version:
$ git checkout server@2.13.0
```

Adds `mcp_network_traffic_hourly_summaries` table storing hourly totals for 90 days, tracking observed public vs. private inbound MCP request counts per server.

Adds `telemetry.getMcpNetworkTraffic` API returning zero-filled hourly points for a

24h or 7d window plus the last-seen timestamp for each network route.

Shows observed public versus private traffic in the Network access panel for MCP servers and gateways, so admins can verify client routing before switching between public and private.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

HeyGen HyperFrames

Feedback ⓘ v0.8.89 NOTES ↗ RANK

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

> HyperFrames expanded its Studio embedding API for host apps, added a `hyperframes clean` CLI command, player range looping, typed video file support, and Enter-key drill-down into sub-compositions.

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

01 **`hyperframes clean` CLI command** NEW 77

A new `hyperframes clean` command finds and removes files that HyperFrames left on disk, useful for reclaiming disk space after heavy rendering sessions.

Reclaim disk space after heavy rendering sessions by clearing all HyperFrames-generated files.

```
$ hyperframes clean
```

— Runnable command with clear purpose and example v0.8.87

02 **Player range looping via range-start and range-end** NEW 65

The Player now accepts `range-start` and `range-end` parameters to play and loop a specific range of a film.

— Named parameters given but no full usage example v0.8.87

THINNER COVERAGE BELOW

03 **Studio embedding API expansions for host apps** IMPROVED

56

Host and embedding apps gained a growing set of ways to reuse Studio internals: saving canvas moves, resizes, and rotations made through Studio's handles; mounting Studio's move, resize, and rotate canvas handles directly over a host's own preview; hiding the timeline's audio effects controls; saving style edits on a selection through Studio's own style-save mechanism; reusing Studio's DOM editing outside the Studio editor; keeping paint order in step with timeline rows; reusing Studio's snapping and marquee selection in a host's own preview; and importing Studio's UI and Player without the full editor to reduce a host's initial load.

— Names many embedding capabilities but no concrete API syntax given v0.8.89 v0.8.87 v0.8.83

04 **Enter-key drill-down into sub-compositions** NEW

50

Pressing Enter on a focused composition clip now opens the sub-composition, matching the existing double-click behaviour, and a host app can now own the drill-down breadcrumb, which follows the Studio theme.

— Describes behaviour and host control but no config surface v0.8.85

05 **Studio Server history-cache project listing endpoint** NEW

50

Studio Server adds an endpoint that lists the projects the history cache knows about, along with their folders.

— Endpoint named but no method or path given v0.8.83

06 **Player support for typed video files** NEW

40

The Player now supports video files that declare an explicit MIME type, such as `video/mp4`.

— Names the format but little mechanism detail v0.8.86

v0.8.89 NOTES ↗ LATEST STABLE

HyperFrames v0.8.89 lets embedding apps save Studio canvas moves, resizes, and rotations made via handles.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.89 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.89
```

v0.8.87 NOTES ↗ STABLE

HyperFrames v0.8.87 adds `hyperframes clean`, player range looping, and new Studio embedding controls for hosts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.87 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.87
```

└──▷ TRY IT

Reclaim disk space after heavy rendering sessions by clearing all HyperFrames-generated files.

```
$ hyperframes clean
```

Adds `hyperframes clean` CLI command to find and remove files HyperFrames left on disk.

Adds `range-start` and `range-end` player parameters to play and loop a specific range of a film.

Enables embedding apps to mount Studio's move, resize, and rotate canvas handles over their own preview.

Enables embedding apps to hide the timeline's audio effects controls in Studio.

Enables embedding apps to save style edits on a selection through Studio's own style-save mechanism.

v0.8.86 NOTES ↗ STABLE

HyperFrames v0.8.86 adds Player support for typed video files such as video/mp4.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.86 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.86
```

v0.8.85 NOTES ↗ STABLE

HyperFrames v0.8.85 lets the Enter key drill into a focused sub-composition, with host-app-owned breadcrumb support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.85 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.85
```

Pressing Enter on a focused composition clip now opens the sub-composition, matching the existing double-click behaviour, and a host app can own the drill-down breadcrumb which follows the Studio theme.

v0.8.83 NOTES ↗ STABLE

HyperFrames v0.8.83 lets host apps reuse Studio's DOM editing, snapping, marquee selection, and paint-order sync, and adds a history-cache project listing to Studio Server.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.83 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.83
```

Enables host apps to reuse Studio's DOM editing outside the Studio editor.

Enables host apps to keep paint order in step with timeline rows.

Enables host apps to reuse Studio's snapping and marquee selection in their own preview.

Adds a Studio Server endpoint that lists the projects the history cache knows, with their folders.

> Allows a host's first screen to import Studio's UI and Player without the editor, reducing initial load.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback ⓘ

AWS Agent Registry: Custom metadata schemas (2026-09-28) BLOG ↗ RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Amazon Bedrock AgentCore's AWS Agent Registry now supports custom metadata schemas, letting teams define typed fields and search records by those values.

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

01 **Custom metadata schemas in AWS Agent Registry** NEW 86

AWS Agent Registry adds custom metadata schemas that let you define structured, typed fields (`text` , `enum` , `URL` , or `boolean`) with optional `required` constraints for validated information on every record. It supports a default schema applied to all record types plus record-type-specific schema overrides, and records can be searched or filtered by custom metadata values using the new `customMetadata.{fieldName}` filter syntax, e.g. `customMetadata.team = 'security-ops'` .

Filter registry records by a custom metadata field — useful when you want to find all agents tagged with a specific team or environment value.

```
$ customMetadata.team = 'security-ops'
```

— Names field types, filter syntax and example query

AWS Agent Registry: Custom metadata sch…

AWS Agent Registry: Custom metadata schemas (2026-09-28) BLOG ↗ LATEST

STABLE

AWS Agent Registry gains custom metadata schemas with typed fields and `customMetadata.{fieldName}` filter syntax for record search.

└──▷ TRY IT

Filter registry records by a custom metadata field — useful when you want to find all agents tagged with a specific team or environment value.

```
$ customMetadata.team = 'security-ops'
```

Adds custom metadata schemas to AWS Agent Registry, letting you define structured, typed fields (`text` , `enum` , `URL` , or `boolean`) with optional `required` constraints for

validated information on every record.

Supports a default schema that applies to all record types, plus record-type-specific schema overrides.

Enables searching and filtering registry records by custom metadata values using the new `customMetadata.{fieldName}` filter syntax.

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v1.9.0 NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> Anthropic Python SDK v1.9.0 adds streaming tool execution and `between_tools` thinking, promotes cache diagnostics to GA, and expands workspace rate-limit and Managed Agents filtering fields, plus a new `claude-sonnet-5-5` model ID.

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

01 **Streaming tool execution during replies** NEW 68

Tool calls can now optionally run concurrently while the model's reply is still streaming, rather than waiting for the stream to end, cutting round-trip latency in agentic pipelines using client.messages.stream(...).

Run tool calls concurrently while the model's reply is still streaming, cutting round-trip latency in agentic pipelines.

python

```
with client.messages.stream(
    model="claude-sonnet-5-5",
    max_tokens=1024,
    tools=[...],
    messages=[{"role": "user", "content": "What is the weather
in Paris?"}],
) as stream:
    for event in stream:
        # tool calls are executed as they arrive, not after the
stream ends
        ...
```

— Concrete mechanism and runnable streaming example given v1.9.0

02 **Cache diagnostics GA** IMPROVED 64

Promotes cache diagnostics to general availability: the `diagnostics` field is now available on both `Message` and `MessageCreateParams`, letting callers inspect prompt-cache hit/miss behaviour via `response.diagnostics`.

Inspect prompt-cache diagnostics on every message response to understand cache hit/miss behaviour in production.

```python
response = client.messages.create(
    model="claude-sonnet-5-5",
    max_tokens=1024,
    messages=[{"role": "user", "content": "Summarise the
document."}],
)
print(response.diagnostics)
```

— Named fields plus runnable inspection example v1.9.0

THINNER COVERAGE BELOW

03 **`between_tools` thinking type** NEW 40

Adds a `between_tools` thinking type to the API to support inter-tool reasoning by the model.

— Named field but no usage example or mechanism detail v1.9.0

04 **Workspace rate-limit inspection fields** NEW 40

Adds `include_inherited` and `source` fields to workspace rate limits for richer rate-limit inspection.

— Named fields but no usage example or navigation v1.9.0

05 **`claude-sonnet-5-5` model ID support** NEW 38

Adds `claude-sonnet-5-5` as a supported model ID.

— Named model ID, directly usable in calls v1.9.0

06 **Typed event filters for Managed Agents** IMPROVED 30

Adds typed event-type values to the Managed Agents events list filter.

— Bare description of a filter addition, no example v1.9.0

v1.9.0 NOTES ↗ LATEST STABLE

Anthropic Python SDK v1.9.0 adds streaming tool execution, between_tools thinking, cache diagnostics GA, and new workspace rate-limit fields.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.9.0
https://github.com/anthropics/anthropic-sdk-python.git
# already have the repo? check out this version:
$ git checkout v1.9.0
```

└──▷ USE IT

Run tool calls concurrently while the model's reply is still streaming, cutting round-trip latency in agentic pipelines.

```python
with client.messages.stream(
    model="claude-sonnet-5-5",
    max_tokens=1024,
    tools=[...],
    messages=[{"role": "user", "content": "What is the weather in
Paris?"}],
) as stream:
    for event in stream:
        # tool calls are executed as they arrive, not after the
stream ends
        ...
```

Inspect prompt-cache diagnostics on every message response to understand cache hit/miss behaviour in production.

```python
response = client.messages.create(
    model="claude-sonnet-5-5",
    max_tokens=1024,
    messages=[{"role": "user", "content": "Summarise the
document."}],
)
print(response.diagnostics)
```

Adds `include_inherited` and `source` fields to workspace rate limits for richer rate-limit inspection.

Adds typed event-type values to the Managed Agents events list filter.

Supports optionally running tool calls while the reply streams (parallel streaming tool execution).

Adds `between_tools` thinking type to the API for inter-tool reasoning.

> Promotes cache diagnostics to GA — `diagnostics` field now available on `Message` and `MessageCreateParams`.
>
> +1 more

WAS THIS USEFUL?  

Anthropic TypeScript SDK

Feedback ⓘ sdk-v0.129.0

NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> The TypeScript SDK adds typed client-side tool dispatch via `ClientToolUnion`, promotes cache diagnostics to general availability on `Message` and `MessageCreateParams`, and introduces the `claude-sonnet-5-5` model identifier alongside smaller additions to workspace rate-limit responses and Managed Agents event filtering.

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

01 **Cache diagnostics promoted to GA** IMPROVED 75

Promotes cache diagnostics to GA — `diagnostics` is now available on `Message` and `MessageCreateParams`, exposing prompt-cache hit/miss behaviour to callers inspecting a message response.

Inspect cache diagnostics on a message response to understand prompt-cache hit/miss behaviour in production.

typescript

```typescript
import Anthropic from '@anthropic-ai/sdk';

const client = new Anthropic();

const message = await client.messages.create({
  model: 'claude-sonnet-5-5',
  max_tokens: 1024,
  messages: [{ role: 'user', content: 'Summarise the codebase.'
}],
});

console.log(message.diagnostics);
```

— Named fields plus a runnable code example sdk-v0.129.0

THINNER COVERAGE BELOW

02 **ClientToolUnion for client-side tool dispatch** NEW 50

Adds `ClientToolUnion` type for client-executed tools, enabling typed dispatch of tool calls that run on the client side; the release also references streaming tool-call dispatch.

— Names the type but gives no usage example sdk-v0.129.0

03 **Workspace rate-limit response fields** NEW 45

Adds `include_inherited` and `source` fields to workspace rate-limit responses.

— Named fields but no usage example or behavior detail sdk-v0.129.0

04 **Typed event-type filter for Managed Agents** IMPROVED 40

Adds typed event-type values to the Managed Agents events list filter.

— Names the filter but not the specific values sdk-v0.129.0

05 **claude-sonnet-5-5 model identifier** NEW 35

Adds the `claude-sonnet-5-5` model identifier for use in API requests.

— New model name, no further detail given sdk-v0.129.0

06 **between_tools thinking type** NEW 15

Adds a `between_tools` thinking type, per the release summary; the changelog bullets do not elaborate on its behaviour.

— Mentioned only in summary, undocumented in bullets sdk-v0.129.0

sdk-v0.129.0 NOTES ↗ LATEST STABLE

Adds between_tools thinking type, ClientToolUnion, claude-sonnet-5-5, cache diagnostics GA, and streaming tool-call dispatch to the TypeScript SDK.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk-v0.129.0
https://github.com/anthropics/anthropic-sdk-typescript.git
# already have the repo? check out this version:
$ git checkout sdk-v0.129.0
```

└──▷ USE IT

Inspect cache diagnostics on a message response to understand prompt-cache hit/miss behaviour in production.

typescript

```typescript
import Anthropic from '@anthropic-ai/sdk';

const client = new Anthropic();

const message = await client.messages.create({
  model: 'claude-sonnet-5-5',
  max_tokens: 1024,
  messages: [{ role: 'user', content: 'Summarise the codebase.' }],
});

console.log(message.diagnostics);
```

Adds `ClientToolUnion` type for client-executed tools, enabling typed dispatch of tool calls that run on the client side.

Adds `include_inherited` and `source` fields to workspace rate-limit responses.

Adds typed event-type values to the Managed Agents events list filter.

Adds `claude-sonnet-5-5` model identifier.

Promotes cache diagnostics to GA — `diagnostics` is now available on `Message` and `MessageCreateParams`.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.20.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK v3.20.0 adds credential and session configuration for Agents and introduces incremental WebSocket streaming snapshots for the Responses API.

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

01 **WebSocket streaming snapshots in Responses API** NEW

41

The responses streaming path adds opt-in incremental WebSocket text and tool snapshots, while continuing to preserve the existing detailed WebSocket accumulator snapshots.

— Names the mechanism but no exact API or flag. v3.20.0

02 **Agent credential and session options** NEW

23

The Agents API surface gains new credential and session options, though the exact parameter names are not detailed.

— No specific field or flag names given. v3.20.0

v3.20.0 NOTES ↗ LATEST STABLE

OpenAI Python SDK v3.20.0 adds Agent credential/session options, Cyber access programs in Responses, and incremental WebSocket snapshots.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.20.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.20.0
```

Adds credential and session options to the Agents API surface.

Adds opt-in incremental WebSocket text and tool snapshots for the responses streaming path.

Preserves detailed WebSocket accumulator snapshots in the responses streaming path.

WAS THIS USEFUL? 👍 👎

Google Gemini Enterprise Agent Platform

Feedback

ⓘ September 27, 2026 (2026-09-27) BLOG ↗ RANK

Gemini Enterprise Agent Platform provides Google's managed model training, inference, evaluation, and agent deployment services, formerly offered under Vertex AI.

> Gemini Enterprise Agent Platform's CodeMender CLI picked up deeper environment and scan configuration controls — configurable file-extension scanning, exclude/include paths, external project access, per-session artifact storage, and sandboxed TMPDIR handling — while JupyterLab notebooks began forwarding client-side logs to Cloud Logging for easier debugging.

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

01 **Configurable file extensions for CodeMender scanning** NEW

83

CodeMender's `~/.codemender/config.yaml` (global) and per-repository `config.yaml` now support an `include` / `scan.extensions.include` list to extend scanning beyond the default language set. r0 documents the default as major programming languages (`.py` , `.java` , `.go` , `.js` , `.ts` , `.rs` , `.cs` , etc.), configurable to also cover IaC, shell scripts, XML, YAML, JSON, and properties files; r1 documents the default set as C/C++, C#/.NET, Go, Java, JavaScript/TypeScript, Kotlin, Python, Ruby, Rust, and PHP, extendable to any non-proprietary language understood by the underlying model.

Extend CodeMender to scan Swift and Scala repositories across all projects without modifying each repo individually.

```yaml
scan:
  extensions:
    include:
      # Default languages
      - .py
      - .java
      - .go
      - .js
      - .ts
      # Additional / custom languages
      - .swift
      - .scala
  exclude_dirs:
    - node_modules
    - vendor
    - dist
    - build
```

— Names exact config keys, file paths, and default language sets. product docs

02 **CodeMender config additions for excludes, project access, and artifacts**

NEW 72

Adds `exclude_dirs` in `~/.codemender/config.yaml`, defaulting to `["node_modules", "vendor", "dist", "bin", "target", "obj", "build", ".gradle"]` to skip large dependency and build folders. Adds `project_paths` (empty by default) to grant the agent access to files outside the scan target directory. Adds a per-session artifacts directory at `~/.codemender/artifacts/<session_id>/`, or `$CM_HOME/artifacts/<session_id>/` if `CM_HOME` is overridden, where the agent writes session output.

Allow the agent to read shared library code that lives outside the scan target directory during build or test analysis.

```yaml
project_paths:
  - /home/user/shared-libs
  - /home/user/common-utils
```

— Names each config key and default but describes them thinly. product docs

03 **Sandbox blocks and redirects host TMPDIR** NEW 60

When the CodeMender sandbox is enabled, the host `TMPDIR` is always blocked even if it is listed in `project_paths`; tools that respect `TMPDIR` are instead redirected to a temporary directory inside the per-session artifacts directory.

— Clear mechanism and scope, but nothing for a reader to configure. product docs

THINNER COVERAGE BELOW

04 **JupyterLab client-side logs surface in Cloud Logging** NEW

50

JupyterLab now forwards client-side logs — console errors, uncaught exceptions, unhandled promise rejections, and failed network requests — to the instance backend, where they surface in Cloud Logging for easier debugging.

— States what's logged but gives no navigation or query steps.

September 27, 2026 (2026-09-27)

> September 27, 2026 (2026-09-27) BLOG ↗ LATEST STABLE
>
> JupyterLab client-side logs (console errors, exceptions, failed requests) now surface in Cloud Logging for easier debugging.

WAS THIS USEFUL? 👍 👎

Thinking Machines Lab Tinker

Feedback ⓘ

changelog · Sep 29, 2026 | CHANGELOG ↗ | RANK

Tinker provides a managed API for training and sampling language models while users control their training algorithms and data.

> Tinker's SDK gained new SFT and RL fine-tuning training APIs, checkpoint save/resume and playground export, sparse token-level scoring, and chat-template formatting, alongside billing cost estimates in the CLI/API, Console UI additions, and a shift from browser-based login to API-key authentication.

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

01 Sparse target-position logprob scoring NEW

90

During sampling, `target_prompt_logprobs` lets you score specific token IDs at targeted prompt positions rather than every position — e.g. via tinker.TensorData.from_torch_sparse(target, pad_value=-1) — useful for evaluating token-level probabilities in an RL reward model without scoring every position.

Score specific token IDs at targeted prompt positions during sampling — useful for evaluating token-level probabilities in an RL reward model without scoring every position.

python

```
import torch
import tinker

# Shape: [len(prompt) - 1, K]; use -1 for cells you don't need
target = torch.full((prompt_len - 1, K), -1, dtype=torch.int64)
target[3, 0] = token_id_of_interest

response = sampler.sample(
    prompt=prompt_ids,

target_prompt_logprobs=tinker.TensorData.from_torch_sparse(targe
t, pad_value=-1),
)
print(response.target_prompt_logprobs)
```

— Runnable code example with exact API. changelog-20260929-9045f15c

02 Checkpoint save and resume APIs NEW

90

Adds `save_state_async` to save resumable training checkpoints and `create_training_client_from_state` to reload them, plus

`save_weights_for_sampler_async` to save sampling-ready weight checkpoints and `get_playground_url` to open them directly in the Tinker Playground.

Save a mid-training checkpoint so you can resume a fine-tuning run later without losing optimizer state.

```
$ training_checkpoint = save_state_async('my-checkpoint',
user_metadata={'step': 100})
# Later, resume from it:
client = create_training_client_from_state(training_checkpoint)
```

After training, export weights for sampling and jump straight into the Tinker Playground to test the model interactively.

```
$ sampling_checkpoint = save_weights_for_sampler_async('my-
sampler-checkpoint')
print(sampling_checkpoint.get_playground_url())
```

— All four checkpoint APIs named with runnable examples. product docs

03 **Estimated USD billing costs** NEW 90

Adds `estimated_cost_usd` and `effective_rate_usd_per_million_tokens` or `effective_rate_usd_per_gigabyte_month` fields to each token and storage event returned by RestClient.get_billing_usage() and the `tinker billing usage` CLI command. Also adds a `cost_data_through` field to `BillingUsageResponse` marking the exclusive UTC boundary up to which costs are invoice-reconciled, surfaced in the CLI's JSON output; the CLI table prints `null` for cost fields when no rate applies, while CSV output leaves those cells blank.

— All billing fields and CLI behavior named exactly. SDK 0.30.2

04 **API key authentication replaces WorkOS login** BREAKING 85

`tinker login` no longer supports WorkOS device authorization (browser-based sign-in); users must generate a key via the Tinker Console and paste it manually. The SDK also now supports the `TINKER_API_KEY` environment variable for authentication alongside the interactive `auth login` flow, useful for headless CI or server environments.

Authenticate without an interactive browser flow by setting your API key — useful in headless CI or server environments.

```
$ export TINKER_API_KEY=<your-api-key>
```

— Names exact env var and removed auth flow. changelog-20260929-9045f15c

05 **Auto-closing session context manager** NEW 80

tinker.ServiceClient() now supports use as a context manager (`with ... as session_client`), guaranteeing a training session is cleanly closed — even on crash or keyboard interrupt — without manually calling finish().

Guarantee a training session is always cleanly closed — even on crash or keyboard interrupt — without manually calling finish().

```python
import tinker

with tinker.ServiceClient() as service_client:
    session = service_client.start_session(model="my-model")
    # training loop here
    # session is finished automatically on exit
```

— Runnable snippet shows exact usage. changelog-20260929-9045f15c

06 **Surrogate loss metric renamed** BREAKING 70

In forward_backward_custom() metrics, the field previously reported as `loss:sum` (the server-computed surrogate) is now `surrogate_loss:sum` ; any code reading `loss:sum` for the surrogate must be updated.

— Exact renamed field with migration note. changelog-20260929-9045f15c

07 **RL fine-tuning loss modes and reward-weighted training** NEW 70

The forward/backward pass now supports `'ppo'` , `'cispo'` , and `'dro'` as named RL fine-tuning loss modes, alongside a reward-weighted RL training loop that uses per-token `'logprobs'` and `'advantages'` fields in training datums.

— Named loss modes and fields but no example. product docs

08 **SFT forward/backward and optimization API** NEW 60

Adds a forward/backward pass and optimization API for Supervised Fine-tuning (SFT) with per-token loss masking (0 to suppress, 1 to apply).

— Named masking values but no example. product docs

THINNER COVERAGE BELOW

09 **Console UI for sessions, checkpoints, and billing** NEW

50

The Tinker Console UI now exposes active sessions, saved checkpoints, and billing under Sessions, Checkpoints, and Billing sections.

— Exact UI section names, no further detail. product docs

10 **Chat template formatting for prompts** NEW

45

Adds `apply_chat_template` for formatting conversation-style prompts before sampling.

— Named function but no example or parameters. product docs

11 **Metadata visibility in sessions and checkpoints** NEW

40

The sessions and training-run tables can now show a `user_metadata` column, and checkpoints now display the metadata attached to them.

— No UI path or example given. changelog-20260929-9045f15c

└──▷ BREAKING ON UPGRADE

! In forward_backward_custom() metrics, the field previously reported as `loss:sum` (the server-computed surrogate) is now `surrogate_loss:sum`; any code reading `loss:sum` for the surrogate must be updated.

! `tinker login` no longer supports WorkOS device authorization (browser-based sign-in); users must generate a key via the Tinker Console and paste it manually.

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

Tinker: The sessions and training-run tables can now show a `user_metadata` column.

└──▷ USE IT

Score specific token IDs at targeted prompt positions during sampling — useful for evaluating token-level probabilities in an RL reward model without scoring every position.

```python
import torch
import tinker

# Shape: [len(prompt) - 1, K]; use -1 for cells you don't need
target = torch.full((prompt_len - 1, K), -1, dtype=torch.int64)
target[3, 0] = token_id_of_interest

response = sampler.sample(
    prompt=prompt_ids,

target_prompt_logprobs=tinker.TensorData.from_torch_sparse(target,
pad_value=-1),
)
print(response.target_prompt_logprobs)
```

Guarantee a training session is always cleanly closed — even on crash or keyboard interrupt — without manually calling finish().

```python
import tinker

with tinker.ServiceClient() as service_client:
    session = service_client.start_session(model="my-model")
    # training loop here
    # session is finished automatically on exit
```

Checkpoints now show the metadata attached to them.

└──▷ **BREAKING ON UPGRADE**

! In forward_backward_custom() metrics, the field previously reported as `loss:sum` (the server-computed surrogate) is now `surrogate_loss:sum`; any code reading `loss:sum` for the surrogate must be updated.

! `tinker login` no longer supports WorkOS device authorization (browser-based sign-in); users must generate a key via the Tinker Console and paste it manually.

SDK 0.30.2 NOTES ↗ STABLE

Tinker SDK 0.30.2 adds estimated USD cost fields to billing usage responses, including invoice-reconciled cost boundaries.

Adds `estimated_cost_usd` and `effective_rate_usd_per_million_tokens` or `effective_rate_usd_per_gigabyte_month` fields to each token and storage event

returned by RestClient.get_billing_usage() and the `tinker billing usage` CLI command.

Adds `cost_data_through` field to `BillingUsageResponse` marking the exclusive UTC boundary up to which costs are invoice-reconciled, also surfaced in the CLI's JSON output.

The `tinker billing usage` CLI table prints `null` for cost fields when no rate applies; CSV output leaves those cells blank.

WAS THIS USEFUL? 👍 👎

Snowflake Cortex AI

Feedback

changelog · Sep 29, 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 flagship addition this window is general availability of the OpenAI GPT-5.6 and GPT-6 Astra models, alongside a broad set of platform updates spanning Trust Center posture reporting, Snowpipe Streaming for Iceberg tables, Streamlit container runtime support, lineage tracking, and dynamic/hybrid table enhancements.

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

01 **Dynamic table incremental refresh expansions** IMPROVED 55

Dynamic table incremental refresh now supports the `VALUES` clause (generally available) and non-deterministic aggregate functions (generally available).

— Names exact SQL syntax supported, usable directly. changelog-20260929-baef9a02

02 **Hybrid table procedures and constraints** IMPROVED 55

Hybrid tables now support inline stored procedures (generally available) and `CHECK` constraints (generally available).

— Names exact constraint type and feature, both GA. changelog-20260929-baef9a02

03 **Snowpipe Streaming for Iceberg tables and event monitoring** NEW 50

Snowpipe Streaming now supports partitioned Apache Iceberg tables (generally available) and adds event table monitoring for Snowpipe Streaming.

— Names Iceberg and monitoring feature but no setup steps. changelog-20260929-baef9a02

04 **GPT-5.6 and GPT-6 Astra models in Cortex AI** NEW 45

Cortex AI now generally supports the OpenAI GPT-5.6 and GPT-6 Astra models.

— Names specific models but no invocation details or mechanism.

changelog-20260929-baef9a02

05 **Trust Center posture reporting and alerting** NEW 45

Adds account posture reporting in the Trust Center (Preview), surfacing a consolidated security posture view, plus programmatic notifications for Trust Center findings (Preview) enabling automated alerting on posture issues.

— Describes purpose but no config keys or endpoints given. changelog-20260929-baef9a02

06 **Container runtime for Streamlit apps in Native Apps** NEW

40

Adds container runtime support for Streamlit apps in Snowflake Native Apps (Preview).

— Names the surface but no usage detail. changelog-20260929-baef9a02

07 **Lineage preservation through temp tables and views** IMPROVED

30

Adds lineage preservation through temporary tables and views (generally available).

— Bare statement of scope with no mechanism. changelog-20260929-baef9a02

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

Snowflake Cortex AI adds GPT-5.6/GPT-6 Astra models, Trust Center posture reporting, and Snowpipe Streaming Iceberg table support.

Adds OpenAI GPT-5.6 and GPT-6 Astra models to Cortex AI (generally available).

Adds account posture reporting in the Trust Center (Preview), surfacing a consolidated security posture view.

Adds programmatic notifications for Trust Center findings (Preview), enabling automated alerting on posture issues.

Adds Snowpipe Streaming support for partitioned Apache Iceberg tables (generally available).

Adds Snowpipe Streaming event table monitoring.

+6 more

WAS THIS USEFUL?  

Cerebras Inference

Feedback ⓘ changelog · Sep 29, 2026

CHANGELOG ↗ RANK

Cerebras Inference provides hosted language-model inference through shared APIs and dedicated endpoints on Cerebras hardware.

> Cerebras Inference temporarily boosted the Qwen 3.8 27B rate limit for Developer-tier users and added new docs surfaces for managing and browsing models.

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

01 **Higher Qwen 3.8 27B Developer-tier rate limit** IMPROVED

63

The Developer-tier total token rate limit for Qwen 3.8 27B was temporarily raised from 450K to 750K tokens per minute, a 5× multiple of the uncached limit of 150K tokens per minute, up from the standard 3×.

— Names exact limits and multiplier but no config/API surface changelog-20260929-e7df5aa2

THINNER COVERAGE BELOW

02 **Management API for Dedicated Inference model weights** NEW

45

A new Management API allows programmatically uploading, versioning, and deploying custom model weights with Dedicated Inference.

— Names the API but no endpoints or usage detail product docs

03 **Model Catalog for Shared Inference** NEW

35

A new Model Catalog lets users browse all models available with Cerebras Shared Inference.

— Bare feature name, no navigation or scope details product docs

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

> Cerebras Inference temporarily raises the Qwen 3.8 27B Developer-tier total token rate limit from 450K to 750K tokens per minute.

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock

Feedback ⓘ changelog · Sep 29, 2026

CHANGELOG ↗ RANK

Amazon Bedrock provides managed foundation-model inference, customization, knowledge bases, and APIs for generative AI applications.

> Amazon Bedrock replaced its per-model daily token quota with a single cross-model, per-account quota scoped per Region.

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

01 **Cross-model daily token quota** BREAKING 65

The per-model Model invocation max tokens per day quota is replaced by a single Cross-Model Max Tokens Per Day quota, scoped per account, per Region across all supported Amazon Bedrock models. Any quota-increase requests or monitoring previously targeting the per-model quota must be updated to target the new cross-model quota.

— Names both old and new quota but no migration steps or numeric limits.

changelog-20260929-3f60a59a

└──▷ BREAKING ON UPGRADE

! The per-model Model invocation max tokens per day quota is replaced by a single Cross-Model Max Tokens Per Day quota scoped per account, per Region; any quota-increase requests or monitoring previously targeting the per-model quota must be updated to the new cross-model quota.

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

Amazon Bedrock replaces per-model token-per-day quotas with a single cross-model, per-account daily token quota.

Replaces the per-model Model invocation max tokens per day quota with a Cross-Model Max Tokens Per Day quota scoped per account, per Region across all supported Amazon Bedrock models.

└──▷ BREAKING ON UPGRADE

! The per-model Model invocation max tokens per day quota is replaced by a single Cross-Model Max Tokens Per Day quota scoped per account, per Region; any quota-increase requests or monitoring previously targeting the per-model quota must be updated to the new cross-model quota.

WAS THIS USEFUL? 👍 👎

Perplexity API

Feedback ⓘ changelog · Sep 29, 2026

CHANGELOG ↗ RANK

Perplexity API Platform - build with the Router, Agent, Search, and Embeddings APIs. Real-time, web-wide research and Q&A capabilities for your products.

> Perplexity API shipped a lower-latency Fast Search path for the Search and Agent APIs, added new directly addressable model IDs, opened an AWS Marketplace SaaS listing with shared credits across its APIs, and announced the sunset of several legacy GPT-5 model IDs.

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

01 **Fast Search path for Search and Agent APIs** NEW 90

Adds `search_type: "fast"` on the Search API and on the Agent API `web_search` tool, a lower-latency path roughly 800ms faster than the standard `search_type: "web"` web search, priced at $1.00 per 1,000 requests.

Cut search latency and cost in a Search API call by switching to the Fast Search path.

```
$ curl -X POST https://api.perplexity.ai/search \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"search_type": "fast", "query": "latest CVEs in OpenSSL"}'
```

Use Fast Search in an Agent API web_search tool call to keep agentic pipelines under latency budgets.

```
$ curl -X POST https://api.perplexity.ai/agent \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "openai/gpt-6-luna", "tools": [{"type": "web_search", "search_type": "fast"}], "messages": [{"role": "user", "content": "Summarise threat actor TTPs from the last 7 days."}]}'
```

— Named flag, price and latency, with runnable curl examples. changelog-20260929-d95eda13

02 **Deprecation of legacy GPT-5 model IDs** BREAKING 85

On 2026-10-24 at 00:00 UTC, the Agent API and Router API will stop accepting model IDs `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini`, both in requests and model-list endpoint responses; direct model selections and fallback chains using these IDs must be updated before the cutoff.

— Exact cutoff date and full list of affected model IDs. changelog-20260929-d95eda13

THINNER COVERAGE BELOW

03 **AWS Marketplace SaaS listing** NEW 55

Perplexity API Platform is now available as a SaaS listing on AWS Marketplace, allowing API credits to be purchased through an AWS account and applied to a shared balance across the Router API, Agent API, Search API, and Embeddings.

— Describes billing mechanism but no setup steps given. product docs

04 **New directly addressable model IDs** NEW 40

Adds `xai/grok-4.7`, `openai/gpt-6-luna`, and `openai/gpt-6-sol` as directly addressable model IDs.

— Names the model IDs but gives no usage detail. changelog-20260929-d95eda13

└──▷ BREAKING ON UPGRADE

! On 2026-10-24 at 00:00 UTC, the Agent API and Router API will stop accepting the model IDs `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini` — both in requests and model-list endpoint responses. Direct model selections and fallback chains using these IDs must be updated before the cutoff.

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

Perplexity API: Adds `search_type: "fast"` on the Search API and Agent API `web_search` tool for a lower-latency path (~800 ms faster) at $1.00 per 1,000 requests, versus `search_type: "web"` for standard web search.

└──▷ TRY IT

Cut search latency and cost in a Search API call by switching to the Fast Search path.

```
$ curl -X POST https://api.perplexity.ai/search \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"search_type": "fast", "query": "latest CVEs in OpenSSL"}'
```

Use Fast Search in an Agent API web_search tool call to keep agentic pipelines under latency budgets.

```
$ curl -X POST https://api.perplexity.ai/agent \
  -H 'Authorization: Bearer <token>' \
  -H 'Content-Type: application/json' \
  -d '{"model": "openai/gpt-6-luna", "tools": [{"type":
"web_search", "search_type": "fast"}], "messages": [{"role":
"user", "content": "Summarise threat actor TTPs from the last 7
days."}]}'
```

Adds `xai/grok-4.7` as a directly addressable model ID.

Adds `openai/gpt-6-luna` and `openai/gpt-6-sol` as directly addressable model IDs.

└──▷ BREAKING ON UPGRADE

! On 2026-10-24 at 00:00 UTC, the Agent API and Router API will stop accepting the model IDs `openai/gpt-5.4`, `openai/gpt-5.4-mini`, `openai/gpt-5.4-nano`, `openai/gpt-5.2`, `openai/gpt-5.1`, `openai/gpt-5`, and `openai/gpt-5-mini` — both in requests and model-list endpoint responses. Direct model selections and fallback chains using these IDs must be updated before the cutoff.

WAS THIS USEFUL? 👍 👎

Hugging Face PEFT

Feedback ⓘ v0.21.1 NOTES ↗ RANK

PEFT implements parameter-efficient fine-tuning methods including LoRA, QLoRA, prefix tuning, and adapters for large pretrained models.

> PEFT v0.21.1 adds compatibility with Tensor Parallel setups for PEFT adapters.

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

01 **Tensor Parallel compatibility for adapters** IMPROVED

40

PEFT adapters now work correctly with Tensor Parallel (TP), requiring `Transformers >= 5.17.0`.

— Names version requirement but no mechanism or usage detail v0.21.1

v0.21.1 NOTES ↗ LATEST STABLE

PEFT v0.21.1 enables Tensor Parallel (TP) compatibility, requiring Transformers ≥ 5.17.0.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.21.1
https://github.com/huggingface/peft.git
# already have the repo? check out this version:
$ git checkout v0.21.1
```

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

Apple MLX

Feedback ⓘ v0.32.3 NOTES ↗ RANK

MLX is Apple's array framework for machine learning on Apple silicon, with a NumPy-like API, unified memory, and lazy evaluation.

> MLX v0.32.3 adds Bessel's-correction control for standard deviation, a way to query array buffer sizes, new fused Metal kernels for cross-entropy, delta nets and attention, and CUDA-side additions like cuSOLVER Cholesky decomposition, alongside several Windows platform fixes and a breaking change to `unstack`'s return type.

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

01 **Bessel's correction for std/var** NEW 73

Adds a `correction` parameter to `std` and `var` to control Bessel's correction, letting callers request sample (`ddof=1`) vs population statistics explicitly, e.g. mx.std(x, correction=1).

Compute the sample standard deviation (Bessel's correction, `ddof=1`) the way NumPy does by default.

python

```
import mlx.core as mx

x = mx.array([2.0, 4.0, 4.0, 4.0, 5.0, 5.0, 7.0, 9.0])
print(mx.std(x, correction=1))  # sample std, not population std
```

— Runnable example given, but limited mechanistic detail beyond the parameter. v0.32.3

02 **Query array buffer size** NEW 68

Adds `get_array_buffer_size` to query the underlying Metal/CUDA buffer size occupied by an array, useful for memory-budget accounting.

Query how many bytes a large array's Metal/CUDA buffer actually occupies (useful for memory-budget accounting).

python

```
import mlx.core as mx

weights = mx.ones([4096, 4096], dtype=mx.float16)
buf_bytes = mx.get_array_buffer_size(weights)
print(f'Buffer size: {buf_bytes / 1024**2:.1f} MB')
```

– Named function with a runnable example, but no further mechanism described. v0.32.3

THINNER COVERAGE BELOW

03 **New Metal kernels for cross-entropy, delta nets, attention** NEW

50

Adds fused Metal kernels for `fast.cross_entropy` to accelerate loss computation, new Metal kernels for gated delta nets, and Metal SDPA support for `D96/V64` head dimensions.

– Names three specific kernel additions but gives no usage example or benchmark. v0.32.3

04 **Global-scale support in gather_qmm** NEW

50

Adds global-scale support to `gather_qmm` on both Metal and CUDA backends.

– Named function and backends, but no usage example given. v0.32.3

05 **Cholesky decomposition via cuSOLVER on CUDA** NEW

50

Implements Cholesky decomposition via `cuSOLVER` on the CUDA backend.

– Names the backend and underlying library but no code example. v0.32.3

06 **unstack now returns a tuple** BREAKING

50

`unstack` now returns a tuple instead of a list in Python; code relying on list-specific methods (e.g. `.append`) on the return value will break.

– Clear before/after behavior with explicit migration warning. v0.32.3

07 **Axis parameter ergonomics for indexing ops** IMPROVED

45

Makes `axis` optional in `put_along_axis` and defaults `axis` to `-1` in `take_along_axis` in Python, reducing boilerplate for common calls.

– Documents the behavior change to two named functions without a usage example. v0.32.3

08 **Explicit index and bytes in array interface** NEW

38

Adds explicit `index` and `bytes` support in the Python array interface.

– Thin one-line addition with no usage example or context. v0.32.3

09 **Shapeless compilation of scan operations** NEW

38

Supports shapeless compilation of scan operations.

– States the capability but no usage example or scope detail. v0.32.3

10 **Windows platform I/O fixes** IMPROVED

38

Uses `DXGI` for accurate WDDM VRAM memory budget on Windows and switches to 64-bit file seeks on Windows for large-file correctness.

— Names the Windows APIs involved but no user-facing action described. v0.32.3

11 **Adaptive concurrency cap on load** IMPROVED 35

Makes the concurrency cap on load adaptive so that I/O scales with the machine it runs on.

— Explains the behavior change but no config surface or numbers. v0.32.3

12 **M5 Ultra matmul tunings** IMPROVED 30

Adds M5 Ultra tunings for non-quantized matmuls.

— Names the hardware target but gives no benchmark or mechanism. v0.32.3

└──▷ BREAKING ON UPGRADE

! `unstack` now returns a tuple instead of a list in Python — code that relies on list-specific methods (e.g. `.append`) on the return value will break.

v0.32.3 NOTES ↗ LATEST STABLE

MLX v0.32.3 adds a correction parameter to std/var, get_array_buffer_size, matrix transpose, fused cross-entropy kernels, and D512 Metal vector attention.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.32.3 https://github.com/ml-explore/mlx.git
# already have the repo? check out this version:
$ git checkout v0.32.3
```

└──▷ USE IT

Compute the sample standard deviation (Bessel's correction, `ddof=1`) the way NumPy does by default.

```python
import mlx.core as mx

x = mx.array([2.0, 4.0, 4.0, 4.0, 5.0, 5.0, 7.0, 9.0])
print(mx.std(x, correction=1))  # sample std, not population std
```

Query how many bytes a large array's Metal/CUDA buffer actually occupies (useful for memory-budget accounting).

```python
import mlx.core as mx

weights = mx.ones([4096, 4096], dtype=mx.float16)
buf_bytes = mx.get_array_buffer_size(weights)
print(f'Buffer size: {buf_bytes / 1024**2:.1f} MB')
```

Adds a `correction` parameter to `std` and `var` for Bessel's correction control.

Adds `get_array_buffer_size` to query the underlying buffer size for arrays.

Adds fused Metal kernels for `fast.cross_entropy` to accelerate loss computation.

Adds Metal kernels for gated delta nets.

Adds Metal SDPA support for `D96/V64` head dimensions.

+11 more

└─▷ BREAKING ON UPGRADE

! `unstack` now returns a tuple instead of a list in Python — code that relies on list-specific methods (e.g. `.append`) on the return value will break.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Docling

Feedback ⓘ v2.131.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling's window centers on new remote-inference options (KServe v2 for OCR/layout/classification, OpenAI-compatible APIs for VLM stages), full Apple Keynote parsing with chart data extraction and optional LibreOffice-based chart rendering, and round-trippable Docling JSON input alongside PaddleOCR-VL result import.

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

01 **KServe v2 remote inference for OCR, layout, and classification**

NEW 90

Docling adds `KserveV2OcrOptions` to send OCR requests to a KServe v2-compatible remote inference server (e.g. Triton Inference Server) over gRPC or HTTP, configured with `url`, `transport`, and `model_name` fields; remote OCR requires `enable_remote_services=True` in `PdfPipelineOptions`, otherwise pipeline build raises `OperationNotAllowed`. It also adds `ApiKserveV2ObjectDetectionEngineOptions` to serve object-detection layout models and `ApiKserveV2ImageClassificationEngineOptions` to serve picture-classification models from a KServe v2 inference server, both configured as engine `engine_options`.

Run PDF OCR against a Triton Inference Server over gRPC instead of local OCR — useful when GPU inference is centralised on a remote host.

```python
from docling.datamodel.base_models import InputFormat
from docling.datamodel.pipeline_options import
PdfPipelineOptions, KserveV2OcrOptions
from docling.document_converter import DocumentConverter,
PdfFormatOption

pipeline_options = PdfPipelineOptions(
    do_ocr=True,
    enable_remote_services=True,
    ocr_options=KserveV2OcrOptions(
        url="localhost:8001",
        transport="grpc",
        model_name="rapidocr",
    ),
)

converter = DocumentConverter(
    format_options={
        InputFormat.PDF:
PdfFormatOption(pipeline_options=pipeline_options)
    }
)
```

— Names every config class/field with a runnable example. product docs

02 **Docling JSON round-tripping with local image re-fetch** NEW

90

Docling JSON is now a re-entrant input format (`JSON_DOCLING`), letting a saved `.json` document be reloaded and re-exported to another format, with embedded `data:` images and `http(s)` URLs always retained. `DeclarativeBackendOptions.enable_local_fetch` on `DoclingJSONFormatOption` allows re-loading locally referenced images (bare paths, relative paths, or `file:` URIs) from a trusted Docling JSON file, including documents saved with `ImageRefMode.REFERENCED` .

Re-export a saved Docling JSON document (with separately stored images) to another format without losing the referenced local images.

```python
from docling.datamodel.base_models import InputFormat
from docling.datamodel.pipeline_options import
DeclarativeBackendOptions
from docling.document_converter import DocumentConverter,
FormatOption
from docling.backend.docling_parse_backend import
DoclingJSONFormatOption

converter = DocumentConverter(
    format_options={
        InputFormat.JSON_DOCLING: DoclingJSONFormatOption(

backend_options=DeclarativeBackendOptions(enable_local_fetch=Tru
e)
        )
    }
)
result = converter.convert('saved_document.json')
print(result.document.export_to_markdown())
```

— Runnable example plus exact config key and format name. product docs

03 **Apple Keynote (.key) file parsing with chart data** NEW

80

Docling now parses Apple Keynote (`.key`) presentations as structured documents, extracting charts as classified pictures with their underlying data. Chart data is stored in `meta.tabular_chart` and the chart title becomes the picture caption; charts in iWork '09 presentations are not read.

Convert an Apple Keynote presentation to Markdown, extracting text and chart data from slides.

```python
from docling.document_converter import DocumentConverter

converter = DocumentConverter()
result = converter.convert("slides.key")
print(result.document.export_to_markdown())
```

— Runnable conversion example; storage field and version limits named. v2.131.0

04 **Keynote chart image rendering via LibreOffice** NEW

60

Adds an opt-in option (default `False`) to render an image for each chart in a Keynote presentation and attach it to the chart's `PictureItem`, by rebuilding the chart as a single-chart Office document and rasterizing it with LibreOffice. Charts with no Office equivalent — mixed, two-axis, bubble, or interactive charts — receive no image.

— Mechanism explained but no exact option name or example. product docs

THINNER COVERAGE BELOW

05 **OpenAI-compatible API support for VLM pipelines** NEW

55

Adds `ApiVlmEngineOptions` to route VLM stages (VLM conversion, code/formula enrichment, picture description) through an OpenAI-compatible API as the stage `engine_options`, and `ApiVlmOptions` to configure VLM pipeline models that call an OpenAI-compatible API.

— Named config classes but no example or defaults given. product docs

06 **Import saved PaddleOCR-VL results** NEW

35

Docling can import a saved PaddleOCR-VL structured result into a `DoclingDocument` without running Paddle in-process within Docling.

— Only a one-line description, no API or example given. product docs

v2.131.0 NOTES ↗ LATEST STABLE

Docling v2.131.0 adds Apple Keynote (.key) parsing with chart-to-picture extraction including embedded chart data.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.131.0 https://github.com/docling-project/docling.git
# already have the repo? check out this version:
$ git checkout v2.131.0
```

└──▷ USE IT

Convert an Apple Keynote presentation to Markdown, extracting text and chart data from slides.

```python
from docling.document_converter import DocumentConverter

converter = DocumentConverter()
result = converter.convert("slides.key")
print(result.document.export_to_markdown())
```

> Adds Apple Keynote (`.key`) file support, enabling Docling to parse `.key` presentations as structured documents.
>
> Parses Keynote charts as classified pictures with their underlying data, making chart content extractable from `.key` files.

WAS THIS USEFUL? 👍 👎

Volcengine OpenViking

Feedback ⓘ sdk/go/v0.0.4 NOTES ↗

RANK

OpenViking is a context database that manages agent memory, knowledge retrieval, and skills.

> OpenViking's Go SDK adds pagination support to its directory and tree listing operations, along with new summary controls for listings.

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

01 **Paginated ls and tree listing** NEW 75

The Go SDK adds `ListPage` and `TreePage` interfaces that return `has_more` for paginated directory and tree listing. It also adds directory summary and overview control parameters to the `ls` command, and aligns query parameter capabilities between `ls` and `tree`.

— Names exact interfaces and fields but no usage example sdk/go/v0.0.4

sdk/go/v0.0.4 NOTES ↗ LATEST STABLE

OpenViking Go SDK v0.0.4 adds ListPage and TreePage with has_more pagination support for ls and tree operations.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk/go/v0.0.4
https://github.com/volcengine/OpenViking.git
# already have the repo? check out this version:
$ git checkout sdk/go/v0.0.4
```

Adds `ListPage` and `TreePage` interfaces that return `has_more` for paginated directory and tree listing.

Adds directory summary and overview control parameters to the `ls` command.

Aligns query parameter capabilities between `ls` and `tree`.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20260929 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's latest snapshot adds a project-level thread/trace statistics endpoint, standardizes Gateway-backed prompts on the ChatOpenAI manifest, and lets teams tag custom Gateway models with provider metadata, alongside a tightened permission requirement on dataset exports.

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

01 **Thread and trace statistics endpoint** NEW 95

New `GET /v2/threads/stats` endpoint retrieves thread and trace counts, latency, token, cost, and feedback statistics for a tracing project. It is supported on self-hosted deployments backed by ClickHouse, including when SmithDB queries are disabled, though scoped trace and thread filters still require SmithDB.

Pull per-project thread counts, latency, and cost stats in one API call — useful in dashboards or cost-monitoring scripts.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

— Full endpoint, method, and self-hosted caveat given snapshot-20260929

02 **Standard ChatOpenAI manifest for Gateway prompts** IMPROVED 75

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference, enabling pull-and-run with standard LangChain SDKs.

— Names manifest and secret key but no runnable example snapshot-20260929

THINNER COVERAGE BELOW

03 **Application resource tag value** NEW 45

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

— UI path given but no further mechanism snapshot-20260929

└──▷ **BREAKING ON UPGRADE**

! Dataset CSV and JSONL exports now require both dataset read and download permissions; direct API requests without download permission return a forbidden 403 response instead of succeeding.

snapshot-20260929 NOTES ↗ **LATEST** STABLE

LangSmith adds `/v2` /threads/stats, IAM role ECR access, ls_provider trace metadata, and Gateway fallback timeouts in this snapshot.

└──▷ **TRY IT**

Pull per-project thread counts, latency, and cost stats in one API call — useful in dashboards or cost-monitoring scripts.

```
$ curl -X GET 'https://api.smith.langchain.com/v2/threads/stats?
project_id=<project_id>' \
  -H 'x-api-key: <LANGSMITH_API_KEY>'
```

Tag a custom OpenAI-compatible model in the LLM Gateway so traces carry provider metadata and map to the correct pricing tier.

```json
{
  "model": "my-org/custom-llama-3",
  "ls_provider": "openai",
  "base_url": "https://my-gateway.example.com/v1"
}
```

Set the license-expiry warning window to 14 days on a self-hosted deployment so operators get earlier notice.

```.env
LICENSE_EXPIRATION_WARNING_DAYS=14
```

Adds `GET /v2/threads/stats` endpoint to retrieve thread and trace counts, latency, token, cost, and feedback statistics for a tracing project.

Supports `GET /v2/threads/stats` on self-hosted deployments backed by ClickHouse, including when SmithDB queries is disabled; scoped trace and thread filters still require SmithDB.

Prompts saved with a LangSmith Gateway model now use a standard `ChatOpenAI` manifest with the unified Gateway URL and a `LANGSMITH_GATEWAY_API_KEY` secret reference, enabling pull-and-run with standard LangChain SDKs.

Workspace editors can now assign resources to an Application tag value from Settings > Resource Tags.

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; direct API requests without download permission return a forbidden 403 response instead of succeeding.

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.47.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.47.0 improves AI gateway routing and input capture, adds a ClickHouse settings passthrough for self-hosted deployments, and introduces dedicated trace transcripts for topics.

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

01 ClickHouse extra settings passthrough NEW

70

Adds the `CLICKHOUSE_EXTRA_SETTINGS` environment variable as a passthrough for arbitrary ClickHouse query settings, allowing self-hosted deployments to configure options such as `max_result_rows` at startup.

Allow self-hosted ClickHouse to accept larger result sets by passing through custom query settings at startup.

```
$ CLICKHOUSE_EXTRA_SETTINGS='{"max_result_rows": 1000000}'
docker compose up
```

— Named env var with runnable example command v4.47.0

02 AI gateway environment routing and input capture NEW

65

The AI gateway now reads the `langfuse-environment` HTTP header to set the trace environment on incoming requests, and captures full-mode inputs up to 5 MiB, marking inputs as omitted when that limit is exceeded.

Route AI gateway traffic into a specific Langfuse environment (e.g. `production` vs `staging`) without changing the project key.

```
$ curl https://<your-langfuse-host>/api/ai-gateway/... \
  -H 'langfuse-environment: production' \
  -H 'Authorization: Bearer <key>' \
  -d '{...}'
```

— Names header and size limit but not full mechanism v4.47.0

THINNER COVERAGE BELOW

03 Trace transcripts for topics NEW

30

Renders and measures dedicated trace transcripts for topics, with nesting level exposed per turn.

— Described only in brief prose, no surface named v4.47.0

v4.47.0 NOTES ↗ LATEST STABLE

Langfuse v4.47.0 adds AI gateway environment headers, 5 MiB full-mode input capture, ClickHouse Native write strategy, and topic transcript rendering.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.47.0
https://github.com/langfuse/langfuse.git
# already have the repo? check out this version:
$ git checkout v4.47.0
```

└──▷ TRY IT

Route AI gateway traffic into a specific Langfuse environment (e.g. production vs staging) without changing the project key.

```
$ curl https://<your-langfuse-host>/api/ai-gateway/... \
  -H 'langfuse-environment: production' \
  -H 'Authorization: Bearer <key>' \
  -d '{...}'
```

Allow self-hosted ClickHouse to accept larger result sets by passing through custom query settings at startup.

```
$ CLICKHOUSE_EXTRA_SETTINGS='{"max_result_rows": 1000000}' docker
compose up
```

AI gateway reads the `langfuse-environment` HTTP header to set the trace environment on incoming requests.

AI gateway captures full-mode inputs up to 5 MiB and marks omitted inputs when the limit is exceeded.

Adds `CLICKHOUSE_EXTRA_SETTINGS` environment variable as a passthrough for arbitrary ClickHouse query settings.

Renders and measures dedicated trace transcripts for topics, with nesting level exposed per turn.

WAS THIS USEFUL? 👍 👎

UK AI Security Institute Inspect AI Feedback ⓘ

0.3.272 CODE ↗ RANK

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

Adds `inspect view --show-shards` for shard log visibility; improves computer tool click-without-coordinate handling and fixes `back_click` / `forward_click`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.3.272
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.272
```

Adds `inspect view --show-shards` flag to list shard logs that are otherwise hidden once a successful merged log covers them; bundles also omit covered shards.

Improves the computer tool so click actions called without a `coordinate` now click at the current cursor position instead of raising a tool error, and the tool description is updated to state which actions require `coordinate`.

Fixes `back_click` and `forward_click` in the computer tool (requires rebuilt `aisiuk/inspect-computer-tool` image); previously both actions failed inside the container regardless of arguments.

NEW SUPPORTED PLATFORM Claude Sonnet 5.5 model support

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.84 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik's latest releases add OpenRouter-based Jev support to online evaluation rules, expand annotation queue automation with a service-level toggle and provenance tracking, introduce an Enterprise-preview data anonymization feature for trace data, and enable ClickHouse-backed Custom Charts per workspace.

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

01 **Data Anonymization for trace data (Enterprise preview)** NEW

60

An Enterprise preview feature automatically masks emails, phone numbers, and other sensitive values in trace data for users who lack the Original data view permission.

– Names permission and masked data types, mechanism partly explained product docs

THINNER COVERAGE BELOW

02 **Jev via OpenRouter in Decisions API** NEW 45

The Decisions API used for online evaluation rules now supports Jev via OpenRouter, adding a new model provider option for automated evaluation rules.

– Names API and provider but no config detail 2.2.84

03 **Annotation queue automation controls** NEW 45

Opik adds queue automation configuration and item provenance display in the UI, and introduces a service-level toggle that lets operators enable or disable annotation queue automation entirely.

– Describes what changed but not exact settings location 2.2.84 2.2.83

04 **ClickHouse account support for Custom Charts** NEW 40

Custom Charts now supports ClickHouse accounts, with a per-workspace toggle to enable the capability.

– Names feature and toggle but no config path given 2.2.83

2.2.84 NOTES ↗ LATEST STABLE

Opik 2.2.84 adds Jev via OpenRouter support in online evaluation rules and queue automation with item provenance in the UI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2.2.84 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.84
```

Supports Jev via OpenRouter in the Decisions API for online evaluation rules.

Adds queue automation configuration and item provenance display in the UI.

2.2.83 NOTES ↗ STABLE

Opik 2.2.83 adds a service toggle to gate annotation queue automation and introduces Custom Charts workspace support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2.2.83 https://github.com/comet-ml/opik.git
# already have the repo? check out this version:
$ git checkout 2.2.83
```

Adds a service toggle to gate annotation queue automation, allowing operators to enable or disable the feature at the service level.

Adds Custom Charts ClickHouse account support with a per-workspace toggle.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20260929 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust shipped a wave of tracing and evaluation upgrades this window: a new `bt trace` CLI for local redaction of coding-agent spans, ingestion redaction for project logs, batch-API workflow evals, application log ingestion via SDK and OpenTelemetry, OpenInference trace mapping, scheduled Pattern analysis via Loop, a new Jev built-in judge, and reworked model access controls that include a breaking change to provider precedence.

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

01 **bt trace CLI for local span redaction plugins** NEW 85

The `bt` CLI (v0.21.0+) adds `bt trace enable`, `bt trace run`, and `bt trace import` subcommands to register JavaScript span plugins that redact secrets, remove sensitive fields, or add tags and metadata to coding-agent traces before they leave the machine. If a plugin fails, affected spans are withheld rather than sent untransformed.

Redact secrets from coding-agent spans locally before they are sent to Braintrust, using a registered JavaScript plugin.

```
$ bt trace enable my-redact-plugin.js
bt trace run my-agent-script.js
```

— Exact CLI subcommands, version, and a runnable example. snapshot-20260929

02 **Application log ingestion via SDK and OpenTelemetry** NEW

80

Supports ingesting application logs via the Python SDK (v0.42.0+) and OpenTelemetry log export from any language; logs appear alongside traces, are browsable and filterable by severity with a new Logs row type, and are queryable with SQL. Requires data plane v2.14.0+.

Deduplicate SQL query results to see one row per unique span type and model combination in the Logs SQL editor.

```
$ SELECT DISTINCT span_type, model FROM traces WHERE project_id
= '<project_id>'
```

— Names SDK version and data plane requirement, includes runnable SQL example.

snapshot-20260929

03 **Model access controls and Models tab** BREAKING

80

Adds new model access controls — model restrictions on AI providers — and a new Models tab with a Gateway command for testing individual models; organization providers now support model selection on both hosted and self-hosted deployments, and Selected models is available for project providers on hosted deployments. On hosted deployments, project providers now override organization providers only for models they serve, while organization providers can serve other models using their own credentials — if you rely on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

— Names UI surfaces and migration note, no exact config key. snapshot-20260929

04 **OpenInference trace mapping for hosted deployments** NEW

70

Maps OpenInference traces to structured messages, tool calls, span types, token metrics, tags, and scores on Braintrust-hosted deployments; supports OpenInference instrumentors for LangChain, LlamaIndex, and DSPy with an OpenTelemetry pipeline. Requires data plane v2.16.0+.

— Names instrumentors and version requirement but no direct command. snapshot-20260929

05 **Patterns scheduled analysis via Loop** NEW

70

Patterns runs Loop against a project on a schedule to surface recurring problems and trends (e.g. failure modes by user cohort, rising LLM call cost) not covered by existing scorers; each finding is saved with supporting traces and a suggested fix, and can be converted to a scorer, classifier, or Loop investigation. Requires data plane v2.15.0+.

— Explains mechanism and outputs but no exact command or UI path. snapshot-20260929

06 **Jev built-in judge for scorers** NEW 70

Adds Jev (by TypeSafe) as a built-in judge for scorers and classifiers — returns a specific choice or numeric result instead of free text, with confidence and per-choice probabilities recorded; traceable via Python SDK v0.41.0+ or TypeScript SDK v3.34.0+. Requires data plane v2.15.0+.

— Names SDK versions and data plane requirement but no direct usage snippet.

snapshot-20260929

07 **Batch-API evaluation runs** NEW 65

Python evaluation tasks and scorers can now run through provider batch APIs (SDK v0.39.0+), allowing a run to be started, paused, and resumed via polling or webhook — including by a different process — with results recorded in a single experiment.

— Describes mechanism and SDK version but no exact API call. snapshot-20260929

THINNER COVERAGE BELOW

08 **Ingestion redaction for project logs** NEW 45

Braintrust adds ingestion redaction to apply a shared policy that detects sensitive text in incoming project logs, configurable per project or organization-wide.

— No config key or command shown, just a policy description. product docs

└──▷ BREAKING ON UPGRADE

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials. If you rely on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

snapshot-20260929 NOTES ↗ LATEST STABLE

Braintrust adds batch-API workflow evals, application log ingestion, OpenInference tracing, local span transforms, and columnar metric storage.

└──▷ TRY IT

Redact secrets from coding-agent spans locally before they are sent to Braintrust, using a registered JavaScript plugin.

```
$ bt trace enable my-redact-plugin.js
bt trace run my-agent-script.js
```

Deduplicate SQL query results to see one row per unique span type and model combination in the Logs SQL editor.

```
$ SELECT DISTINCT span_type, model FROM traces WHERE project_id =
'<project_id>'
```

Adds `bt trace enable`, `bt trace run`, and `bt trace import` subcommands to the `bt` CLI (v0.21.0+) to register JavaScript span plugins that redact secrets, remove sensitive fields, or add tags and metadata to coding-agent traces before they leave the machine; if a plugin fails, affected spans are withheld rather than sent untransformed.

Supports running Python evaluation tasks and scorers through provider batch APIs (SDK v0.39.0+), allowing a run to be started, paused, and resumed via polling or webhook — including by a different process — with results recorded in a single experiment.

Supports ingesting application logs via the Python SDK (v0.42.0+) and OpenTelemetry log export from any language; logs appear alongside traces, are browsable and filterable by severity with a new Logs row type, and are queryable with SQL. Requires data plane v2.14.0+.

Maps OpenInference traces to structured messages, tool calls, span types, token metrics, tags, and scores on Braintrust-hosted deployments; supports OpenInference instrumentors for LangChain, LlamaIndex, and DSPy with an OpenTelemetry pipeline. Requires data plane v2.16.0+.

Adds Patterns, which runs Loop against a project on a schedule to surface recurring problems and trends (e.g. failure modes by user cohort, rising LLM call cost) not covered by existing scorers; each finding is saved with supporting traces and a suggested fix, and can be converted to a scorer, classifier, or Loop investigation. Requires data plane v2.15.0+.

+2 more

└─▷ BREAKING ON UPGRADE

! On Braintrust-hosted deployments, project providers now override organization providers only for models they serve; organization providers can serve other models using their own credentials. If you rely on project providers to restrict model access or spending across all models, review organization provider model selections and permissions.

WAS THIS USEFUL? 👍 👎

agentacct

Feedback ⓘ v0.12.8 NOTES ↗ RANK

The agentacct dashboard records coding-agent work steps, tools, file changes, tests, time, and token costs locally.

> agentacct v0.12.8 focuses entirely on recorder recovery, letting the macOS app restart a dead recorder after reboot — even one it doesn't own, such as a pipx/uv-installed CLI.

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

01 **Automatic recorder recovery in macOS app** NEW 95

The macOS app gains a one-click Start recorder action surfaced in the recovery gate, health popover, notice stack, menu-bar dropdown, and setup sheet reconnect button; each runs `agentacct start --no-sync-clients --store-dir <store> --json` and confirms success with `status --json`. When the app owns no verified recorder, it also discovers and launches an externally installed `agentacct` binary by resolving `~/.local/bin/agentacct` and then `PATH`, accepting the candidate only if its `--version` banner matches the expected format.

Restart a dead recorder from the terminal after a reboot, mirroring what the macOS app's one-click button now does — useful on a machine where agentacct was installed via pipx or uv rather than the app.

```
$ agentacct start --no-sync-clients --store-dir
~/.local/share/agentacct --json && agentacct status --json
```

— Names exact commands, flags, and discovery path used for recovery. v0.12.8

v0.12.8 NOTES ↗ LATEST STABLE

agentacct v0.12.8 lets the macOS app restart a dead recorder after reboot, including pipx/uv-installed CLIs it does not own.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.12.8
https://github.com/mikehasa/agentacct.git
# already have the repo? check out this version:
$ git checkout v0.12.8
```

└──▷ TRY IT

Restart a dead recorder from the terminal after a reboot, mirroring what the macOS app's one-click button now does — useful on a machine where agentacct was installed via pipx or uv rather than the app.

```
$ agentacct start --no-sync-clients --store-dir
~/.local/share/agentacct --json && agentacct status --json
```

Adds one-click `Start recorder` to the macOS app's recovery gate, health popover, notice stack, menu-bar dropdown, and setup sheet reconnect button — each runs `agentacct start --no-sync-clients --store-dir <store> --json` and confirms success with `status --json`.

Enables the macOS app to discover and launch an externally installed `agentacct` binary (resolving `~/.local/bin/agentacct` then `PATH`) when the app owns no verified recorder, accepting the candidate only if its `--version` banner matches the expected format.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

Docker Sandboxes

Feedback ⓘ v0.46.0 NOTES ↗ RANK

Docker Sandboxes runs AI agents in isolated Docker environments with controlled filesystem, network, and tool access.

> Docker Sandboxes v0.46.0 and accompanying docs updates add a persistent, daemon-managed model service with a new `sbx daemon restart` command, cloud sandbox kit argument passthrough and auto-restart-on-timeout, plus a raft of smaller CLI, kit, and diagnostics improvements — alongside a breaking change to how secret commands resolve relative paths.

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

01 Model service management and `sbx daemon restart` NEW

93

The Docker Sandboxes daemon now runs a persistent model service that starts at daemon launch, downloads the `llama.cpp` runtime on first use, and is shared across all sandboxes for the model store and loaded models. On Linux the inference server resolution order is: Docker container, then Podman container, then a downloaded binary, then `llama-server` on the host `PATH`. The new `sbx daemon restart` command reloads environment variables such as provider API keys and picks up a newly installed `llama-server` on the host `PATH` without a full host restart; note that setting a provider API key only in the shell where `sbx` is run does not update an already-running model service — it must be set in the daemon's environment and `sbx daemon restart` run for it to take effect.

After adding or rotating a model provider API key in the daemon's shell environment, reload the running service so sandboxes pick up the new credential.

```
$ export OPENAI_API_KEY=sk-... && sbx daemon restart
```

— Explains mechanism, resolution order, and gives exact restart command. product docs

02 Secret command execution directory change BREAKING

90

Secret commands configured with `sbx secret set --command`, `sbx secret set-custom --command`, or `secrets.<name>.command` in an environment file now execute from a fresh temporary directory on the host, so relative paths such as `./credential-helper` no longer resolve from the project directory or the `sbx` invocation directory. This execution change takes effect after upgrading and restarting the daemon, even

before approving the one-time plan change that `sbx env run` prompts for on existing environments.

— Detailed before/after with named config keys and migration timing. v0.46.0

03 **Cloud sandbox kit argument passthrough** NEW 85

`sbx --cloud create` gains `--kit-arg <name>=<value>` to pass named arguments directly to v2 kit mixins on the command line, and `--kit-args-file <FILE>` to load kit arguments from a file with one `name=value` entry per line, with `--kit-arg` values overriding file values.

Provision a cloud sandbox with a versioned company CLI kit, pinning the kit version via `--kit-arg` without editing any file.

```
$ sbx --cloud create --name cloud-tools claude --kit
docker.io/<NAMESPACE>/company-cli:1.0.0 --kit-arg company-
cli.version=1.2
```

Load multiple kit arguments from a file when you have many parameters to set, keeping the command line clean.

```
$ sbx --cloud create --name cloud-tools claude --kit
docker.io/<NAMESPACE>/company-cli:1.0.0 --kit-args-file kit-
args.txt
```

Pass kit-specific arguments when creating a cloud sandbox, so the kit can tailor its setup without hardcoding values.

```
$ sbx --cloud create --kit my-agent-kit --kit-arg MODEL=gpt-4o -
-kit-args-file ./kit-params.env
```

— Named flags with runnable example commands. v0.46.0

04 **`sbx diagnose` diagnostics improvements** IMPROVED 70

`sbx diagnose --upload` now returns a non-zero exit status when the diagnostics upload fails, enabling script-level failure detection, and `sbx diagnose` now reports `runtime_socket_limit_bytes` for the socket-path length limit used by the container runtime.

Detect a failed diagnostics upload in CI so the pipeline can alert or retry rather than silently continuing.

```
$ sbx diagnose --upload || echo 'Diagnostics upload failed,
check connectivity'
```

— Two named fields with one runnable example. v0.46.0

05 **Kit management reliability in `sbx kit add`** IMPROVED 65

Kits can now install files in the agent's skills directory even when the shared skills store is read-only. Kits added through the runtime API now retain their network rules and applicable agent instructions when another kit addition recreates the sandbox container. `sbx kit add` warns when it cannot save the updated sandbox record, naming which kit settings could be lost and under what conditions.

— Three named kit behaviors with mechanism, no examples. v0.46.0

06 **Windows MSI bundles local model serving support** IMPROVED 65

Windows MSI installations now include `llmman` and the guest kernel, enabling local model serving with `sbx run --model` and fixing sandbox launches that previously failed with `No kernel specified`.

— Names installer components and the enabled flag. v0.46.0

07 **Auto-restart on timeout for cloud sandboxes** NEW 60

Adds `--on-timeout restart` to `sbx --cloud create` so a sandbox automatically stops and restarts when it reaches its time limit instead of remaining stopped.

Keep a long-running cloud sandbox alive by restarting it automatically when its time limit expires, rather than leaving it stopped.

```
$ sbx --cloud create --on-timeout restart --kit my-agent-kit
```

— Names the flag and a working command example. v0.46.0

THINNER COVERAGE BELOW

08 **`sbx umount` for removing saved mounts** NEW 55

Adds `sbx umount` support for removing a saved mount from a stopped sandbox using the original host path, even after that directory has been deleted.

— Names the command and behavior but gives no example. v0.46.0

09 **`sbx env` unrecognized key reporting** IMPROVED 55

`sbx env` now reports unrecognized environment-file keys with the file, line, and column where they were declared, including across merged files.

— Command named and mechanism described, no example. v0.46.0

10 **WSL2 image paste clipboard fallback** IMPROVED 55

Image paste in WSL2 falls back to the Windows clipboard when Linux clipboard tools return no image, requiring `clipboard.imagePaste` to be enabled.

— Names the config key but gives no usage example. v0.46.0

11 **Balanced network policy allows Playwright CDN access** IMPROVED 50

Sandboxes created with the `balanced` network policy preset can now reach `cdn.playwright.dev` over HTTPS to download Playwright browser binaries.

— Names preset and domain but no command shown. v0.46.0

└──▷ BREAKING ON UPGRADE

! Setting a provider API key only in the shell where `sbx` is run does not update an already-running model service — the key must be set in the daemon's environment and `sbx daemon restart` must be run to take effect.

! Secret commands configured with `sbx secret set --command`, `sbx secret set-custom --command`, or `secrets.<name>.command` in an environment file now execute from a fresh temporary directory on the host. Relative paths such as `./credential-helper` no longer resolve from the project directory or the `sbx` invocation directory. The execution change takes effect after upgrading and restarting the daemon — even before approving the one-time plan change that `sbx env run` prompts for on existing environments.

v0.46.0 NOTES ↗ LATEST STABLE

Docker Sandboxes v0.46.0 adds restart-on-timeout for cloud sandboxes, kit-arg passthrough, and richer CLI diagnostics.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.46.0 https://github.com/docker/sbx-releases.git
# already have the repo? check out this version:
$ git checkout v0.46.0
```

└──▷ TRY IT

Keep a long-running cloud sandbox alive by restarting it automatically when its time limit expires, rather than leaving it stopped.

```
$ sbx --cloud create --on-timeout restart --kit my-agent-kit
```

Pass kit-specific arguments when creating a cloud sandbox, so the kit can tailor its setup without hardcoding values.

```
$ sbx --cloud create --kit my-agent-kit --kit-arg MODEL=gpt-4o --kit-args-file ./kit-params.env
```

Detect a failed diagnostics upload in CI so the pipeline can alert or retry rather than silently continuing.

```
$ sbx diagnose --upload || echo 'Diagnostics upload failed, check connectivity'
```

Adds `--on-timeout restart` to `sbx --cloud create` so a sandbox automatically stops and restarts when it reaches its time limit instead of remaining stopped.

Adds `--kit-arg` and `--kit-args-file` flag support to `sbx --cloud create` when supplying a `--kit`.

Adds `sbx umount` support for removing a saved mount from a stopped sandbox using the original host path even after that directory has been deleted.

`sbx diagnose --upload` now returns a non-zero exit status when the diagnostics upload fails, enabling script-level failure detection.

`sbx diagnose` now reports `runtime_socket_limit_bytes` for the socket-path length limit used by the container runtime.

+7 more

└──▷ BREAKING ON UPGRADE

! Secret commands configured with `sbx secret set --command`, `sbx secret set-custom --command`, or `secrets.<name>.command` in an environment file now execute from a fresh temporary directory on the host. Relative paths such as `./credential-helper` no longer resolve from the project directory or the `sbx` invocation directory. The execution change takes effect after upgrading and restarting the daemon — even before approving the one-time plan change that `sbx env run` prompts for on existing environments.

WAS THIS USEFUL? 👍 👎

FailproofAI

Feedback ⓘ 1.0.9 NOTES ↗ RANK

Observability and enforcement for AI agent harnesses. Capture every run and runtime reliability with policy enforcement.

> FailproofAI 1.0.9 reworks how Jev's enforcement is set up and verified, requiring explicit setup and policy installation commands, adds atomic writes for agent config files, and ships a Hermes health-check flag plus several smaller CLI fixes.

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

01 Jev setup and status commands

BREAKING

90

`failproofai jev setup --provider failproofai` and `failproofai jev setup --mode observe` configure Jev's new `observe` mode (formerly `shadow`); `failproofai jev status` reports whether Jev is active, which mode it is in, and whether policies are installed. Jev's `mode: "shadow"` in `jev.json` is no longer accepted — Jev stays off until one of the setup commands is run.

Migrate a Jev-configured machine to the new `observe` mode and install policy checks so enforcement is active.

```
$ failproofai jev setup --mode observe
failproofai policies add FailproofAI/jev-policies
failproofai jev status
```

— Exact commands, config keys and migration path given 1.0.9

02 Jev policy checks require explicit install

BREAKING

90

Jev's policy checks are no longer bundled in the package; a Jev-configured machine enforces nothing from Jev until you run `failproofai policies add FailproofAI/jev-policies`, now the required step. A fix also ensures adding `FailproofAI/jev-policies` no longer disables built-in policies enforced from your config.

Migrate a Jev-configured machine to the new `observe` mode and install policy checks so enforcement is active.


```
$ failproofai jev setup --mode observe
failproofai policies add FailproofAI/jev-policies
failproofai jev status
```

— Exact required command and behavior fix documented 1.0.9

03 **Downgrade to 1.0.8 breaks Hermes plugin management** BREAKING

90

Downgrading to 1.0.8 or older breaks Hermes: `failproofai config` stops with "Refusing to overwrite an unmanaged Hermes plugin." Remove the linked plugin first with `rm ~/.hermes/plugins/failproofai ~/.hermes/profiles/*/plugins/failproofai 2>/dev/null`.

— Exact error message and remediation command provided 1.0.9

04 **Hermes profile health check flag** NEW

80

`failproofai config --status` surfaces Hermes profiles still on legacy shell hooks, disabled plugins, unparseable configs, and plugin-name conflicts.

Audit every Hermes profile after upgrading to confirm shell hooks are gone, plugins are active, and configs parse cleanly.

```
$ failproofai config --status
```

— Exact flag and diagnostic conditions named 1.0.9

05 **Atomic writes for agent config files** IMPROVED

75

Agent config files (`~/.hermes/config.yaml`, `~/.claude/settings.json`, `~/.openclaw/openclaw.json`) are now written atomically via a temp-file swap, preserving permissions and leaving a `<name>.failproofai-backup` beside each edited file.

— Mechanism and exact file paths given, no command needed 1.0.9

THINNER COVERAGE BELOW

06 **Harness add-path preserved across update and config** IMPROVED

55

`failproofai update` and `failproofai config` now preserve extra paths added with `failproofai harness add-path`.

— Names exact commands but no further detail given 1.0.9

07 **Guardrails summary breakdown by Jev mode** IMPROVED

50

`fp guardrails summary` now shows Jev machines broken down by mode.

— Single-line description, no example or mechanism 1.0.9

└──▷ BREAKING ON UPGRADE

! Jev's `mode: "shadow"` in `jev.json` is no longer accepted; Jev stays off until you run `failproofai jev setup --provider failproofai` (FailproofAI Cloud) or `failproofai jev setup --mode observe` (custom endpoint).

! Jev's policy checks are no longer bundled in the package; a Jev-configured machine enforces nothing from Jev until you run `failproofai policies add FailproofAI/jev-policies`.

! Downgrading to 1.0.8 or older breaks Hermes: `failproofai config` stops with 'Refusing to overwrite an unmanaged Hermes plugin'. Remove the linked plugin first with `rm ~/.hermes/plugins/failproofai ~/.hermes/profiles/*/plugins/failproofai 2>/dev/null`.

1.0.9 NOTES ↗ LATEST STABLE

FailproofAI 1.0.9 restores Hermes cron-job coverage, ships `jev setup` and `jev status`, and adds atomic agent-config writes.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 1.0.9
https://github.com/FailproofAI/failproofai.git
# already have the repo? check out this version:
$ git checkout 1.0.9
```

└──▷ TRY IT

Migrate a Jev-configured machine to the new `observe` mode and install policy checks so enforcement is active.

```
$ failproofai jev setup --mode observe
failproofai policies add FailproofAI/jev-policies
failproofai jev status
```

Audit every Hermes profile after upgrading to confirm shell hooks are gone, plugins are active, and configs parse cleanly.

```
$ failproofai config --status
```

Adds `failproofai jev setup --provider failproofai` and `failproofai jev setup --mode observe` to configure Jev's new `observe` mode (formerly `shadow`).

Adds `failproofai jev status` to report whether Jev is active, which mode it is in, and whether policies are installed.

Adds `failproofai policies add FailproofAI/jev-policies` as the new required step to install Jev's policy checks, which are no longer bundled in the package.

Adds `failproofai config --status` flag to surface Hermes profiles still on legacy shell hooks, disabled plugins, unparseable configs, and plugin-name conflicts.

Agent config files (`~/.hermes/config.yaml` , `~/.claude/settings.json` , `~/.openclaw/openclaw.json`) are now written atomically via a temp-file swap, preserving permissions and leaving a `<name>.failproofai-backup` beside each edited file.

+3 more

└─▷ BREAKING ON UPGRADE

- ! Jev's `mode: "shadow"` in `jev.json` is no longer accepted; Jev stays off until you run `failproofai jev setup --provider failproofai` (FailproofAI Cloud) or `failproofai jev setup --mode observe` (custom endpoint).
- ! Jev's policy checks are no longer bundled in the package; a Jev-configured machine enforces nothing from Jev until you run `failproofai policies add FailproofAI/jev-policies` .
- ! Downgrading to 1.0.8 or older breaks Hermes: `failproofai config` stops with 'Refusing to overwrite an unmanaged Hermes plugin'. Remove the linked plugin first with `rm ~/.hermes/plugins/failproofai ~/.hermes/profiles/*/plugins/failproofai 2>/dev/null` .

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ snapshot-20260929 NOTES ↗ RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic launched Claude Opus 5.5 and Claude Sonnet 5.5 with 1M-token context windows and always-on adaptive thinking, alongside a cluster of breaking changes to sampling and tool-use parameters, expanded prompt caching with inline tool definitions, GA cache diagnostics, and Compliance API updates.

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

01 Prompt caching support for new models and inline tool definitions

NEW 96

Prompt caching now supports Claude Opus 5.5, Claude Sonnet 5.5, Claude Mythos 5.1, and Claude Opus 4.1, with a minimum cache breakpoint threshold of 512 tokens for Claude Fable 5.1, Claude Mythos 5.1, `claude-opus-5-5`, Claude Opus 5, Claude Sonnet 5.5, and Claude Fable 5. Cache-hit pricing is differentiated: Claude Fable 5.1 and Claude Mythos 5.1 hits cost 0.025x base input price, `claude-opus-5-5` hits cost 0.05x, and all other models use the standard 0.1x multiplier. A new `inline-tools-2026-09-15` beta header enables mid-conversation tool additions or definition changes via a `tool_addition` block (`tool: {"type": "tool_definition", "definition": {...}}`) in a mid-conversation system message, preserving the cached prefix so only the appended message is processed as new input and versioning tools without invalidating the cache. If the last block is not a valid automatic cache breakpoint target, caching silently walks backward to the nearest eligible block, skipping caching only if none is found.

— Names header, pricing multipliers and mechanism in full snapshot-20260929

02 Adaptive thinking behavior changes on Sonnet 5.5 and Opus 5.5

93

BREAKING

Adaptive thinking is now always on for both models and cannot be turned off via `thinking: {"type": "disabled"}`; on Claude Opus 5.5, thinking depth is controlled solely via the `effort` parameter (default `medium`) and both `thinking: {"type": "disabled"}` and `thinking: {"type": "enabled", ...}` return a 400 error, while on Claude Sonnet 5.5 the new `between_tools` thinking setting disables up-front thinking while keeping adaptive thinking active at `high` effort or below. Text appearing between tool calls is now returned inside `thinking` blocks rather than as plain text, so applications must set a `display` value to surface it or use `between_tools` to

suppress it. Thinking blocks are tied to the producing model and conversation, and on Claude Sonnet 5.5 blocks are scoped to the producing account or linked accounts — blocks sent from an unlinked account are silently dropped before the model processes them.

— Exact flags and behavior named across both models `snapshot-20260929`

03 **Claude Opus 5.5 model launch** NEW 90

Launches `claude-opus-5-5` with a 1M token context window, 128K max output tokens, always-on adaptive thinking controlled via the `effort` parameter (default `medium`), and pricing at $4/$20 USD per MTok, available on the Claude API, Amazon Bedrock, AWS, Google Cloud, and Microsoft Foundry. Supports prompt caching with a minimum cacheable prompt length of 512 tokens, priced at $5/MTok for 5-minute cache writes, $8/MTok for 1-hour cache writes and $0.20/MTok for cache reads, plus a 50% Batch API discount on input/output tokens with a beta max output of 300K tokens via the `output-300k-2026-03-24` beta header.

— Full specs and pricing named, but no runnable example `snapshot-20260929`

04 **Sampling parameter and tool-use restrictions on new models** 84

BREAKING

On Claude Sonnet 5.5, setting `temperature`, `top_p`, or `top_k` to a non-default value now returns a 400 error, and `tool_choice` types `any` and `tool` return a 400 error (use `auto` with strict tool use instead); the same `tool_choice` restriction applies to Claude Opus 5.5. Forced tool use returns an error on both Claude Sonnet 5.5 and Claude Opus 5.5. On the Claude API and Google Cloud, both models reject the `computer_20251124` computer use tool and require the new `computer_toolset_20260801` toolset instead. The advisor tool rejects `claude-opus-4-8`, `claude-opus-4-7`, and `claude-sonnet-5` as advisors when using either new model.

— Names exact parameters, endpoints and rejected models `snapshot-20260929`

05 **Cache diagnostics general availability** IMPROVED 81

Cache diagnostics are now GA on the Claude API — include the `diagnostics` object on a `POST /v1/messages` request to opt in. The `diagnostics` field is now always present in responses (null when not opted in), and the `cache-diagnosis-2026-04-07` beta header is no longer required; clients that previously checked for field absence must be updated.

— Exact endpoint and field named, runnable today `snapshot-20260929`

06 **Claude Sonnet 5.5 model launch** NEW 78

Launches `claude-sonnet-5-5` with a 1M token context window, 128K max output tokens, adaptive thinking on by default, and pricing at $4/input MTok and $20/output

MTok with a July 2025 training data cutoff. Supports up to 300K output tokens on the Message Batches API via the `output-300k-2026-03-24` beta header.

— Specs and pricing named, no usage example `snapshot-20260929`

07 **Compliance API local session endpoints and user-data scope**

IMPROVED

74

Compliance API local session endpoints for Claude for Microsoft 365 sessions graduate out of beta, covering `product_surface` values beginning with `office_agents` in Excel, PowerPoint, Word, and Outlook. A new `read:compliance_user_data` scope on Compliance Access Keys allows looking up file names, project document names, or artifact titles by activity ID via the Compliance API.

— Names scope and product_surface values precisely `snapshot-20260929`

08 **Microsoft Foundry model availability expansion** NEW

68

Claude in Microsoft Foundry adds support for Claude Opus 5.5 and Claude Sonnet 5.5 (Sonnet 5.5 limited to Global Standard deployments only), plus Claude Fable 5.1, Claude Opus 5, Claude Opus 4.8, and Claude Sonnet 5 billed through the Azure Marketplace, and Claude Mythos 5.1 and Claude Mythos 5 as limited-availability models.

— Names every model and deployment tier, no exact config `product docs`

09 **Compliance API Activity Feed drops file and title fields**

65

BREAKING

The Compliance API Activity Feed no longer returns file names, project document names, or artifact titles; the `filename` and `title` fields on file, project document, and artifact activities are now always empty or omitted, including for activities recorded before this change.

— Exact fields named, retroactive scope described `snapshot-20260929`

THINNER COVERAGE BELOW

10 **Billing for pre-output refusals** BREAKING

54

Refusals that occur before any output is generated with `stop_details.category` of `"bio"`, `"frontier_llm"`, or `"reasoning_extraction"` are now billed at the standard rates of the model that ran the request, where previously they were not billed.

— Names exact stop categories but no mitigation step `snapshot-20260929`

11 **Code execution tool model support expansion** IMPROVED

52

The code execution tool adds `claude-opus-5-5` support and expands its supported model version lists to include 4.5, 4.6, 4.7, 4.8, 5, and 5.5 (with a subset list of 4.5, 4.6, 5, and 5.5).

— Names the tool and versions but little mechanism `product docs`

---

12 **Fast mode research preview for Opus 5.5** NEW 27

Introduces Fast mode (research preview) for `claude-opus-5-5` on the Claude API.

— Bare mention with no mechanism or access details `snapshot-20260929`

╰──▷ BREAKING ON UPGRADE

! Setting `temperature`, `top_p`, or `top_k` to a non-default value returns a 400 error on Claude Sonnet 5.5 — code that set these parameters on Claude Sonnet 5 will fail.

! Forced tool use returns an error on Claude Sonnet 5.5, breaking workflows that relied on it.

! The `computer_20251124` computer use tool is not accepted on the Claude API or Google Cloud, breaking integrations that specified that tool version.

! The advisor tool rejects Claude Opus 4.8, Claude Opus 4.7, and Claude Sonnet 5 as advisors — pipelines using those models as advisors will fail.

! Text between tool calls is now returned in `thinking` blocks instead of as plain text — applications that stream that text to users will go silent between tool calls until they set a `display` value or use `between_tools`.

! Adaptive thinking (`thinking`) cannot be disabled on Claude Opus 5.5; code that sets `thinking` to off will break.

! Forced tool use now returns an error on Claude Opus 5.5.

! Thinking blocks are tied to the model and the conversation; code that moves thinking blocks across models or conversation contexts will break.

! On the Claude API and Google Cloud, the `computer_20251124` computer use tool is not accepted by Claude Opus 5.5.

! On `claude-sonnet-5-5`, sending `thinking: {"type": "disabled"}` to turn off up-front thinking no longer works at high effort or below; send `thinking: {"type": "between_tools"}` instead.

! On `claude-sonnet-5-5`, `tool_choice` types `any` and `tool` return a 400 error.

! On the Claude API and Google Cloud, `claude-sonnet-5-5` does not accept the `computer_20251124` computer use tool; migrate to the newer tool.

! The advisor tool rejects `claude-opus-4-8`, `claude-opus-4-7`, and `claude-sonnet-5` as advisors when using `claude-sonnet-5-5`.

! On `claude-opus-5-5`, `thinking: {"type": "disabled"}` and `thinking: {"type": "enabled", ...}` both return a 400 error; omit the `thinking` field and use the `effort` parameter instead.

! On `claude-opus-5-5`, `tool_choice` types `any` and `tool` return a 400 error; use `auto` with strict tool use instead.

! On the Claude API and Google Cloud, `claude-opus-5-5` requires `computer_toolset_20260801`; `computer_20251124` returns a 400 error on those platforms.

! The Compliance API Activity Feed no longer returns file names, project document names, or artifact titles; the `filename` and `title` fields on file, project document, and artifact activities are now always empty or omitted, including for activities recorded before this change.

! Cache diagnostics responses from `POST /v1/messages` now always include the `diagnostics` field (null when not opted in), where previously the field was omitted; clients that check for field absence must be updated.

! Refusals before any output with `stop_details.category` of `"bio"`, `"frontier_llm"`, or `"reasoning_extraction"` are now billed at the model's standard rates, where previously they were not billed.

`snapshot-20260929` NOTES ↗ LATEST STABLE

Claude Opus 5.5 and Sonnet 5.5 launch with 1M-token context, adaptive thinking, inline tool definitions, and GA cache diagnostics.

Launches `claude-opus-5-5` with a 1M token context window, 128k max output tokens, always-on adaptive thinking, and pricing at $4/$20 USD per MTok — available on the Claude API, Amazon Bedrock, AWS, Google Cloud, and Microsoft Foundry.

Supports `computer_toolset_20260801` toolset for computer use on `claude-opus-5-5` on the Claude API and Google Cloud; `computer_20251124` returns a 400 error on those platforms.

Enables inline tool definitions inside mid-conversation system messages in beta via the `inline-tools-2026-09-15` beta header, using a `tool_addition` block with `tool: {"type": "tool_definition", "definition": {...}}` to add, update, or version tools without invalidating the prompt cache.

Makes cache diagnostics GA on the Claude API — include the `diagnostics` object on a `POST /v1/messages` request to opt in; the `diagnostics` field is now always present in responses (null when not opted in), and the `cache-diagnosis-2026-04-07` beta header is no longer required.

Launches `claude-sonnet-5-5` with updated thinking controls: use `thinking: {"type": "between_tools"}` instead of `"disabled"` at high effort or below to turn off up-front thinking.

+5 more

└──▷ **BREAKING ON UPGRADE**

! On `claude-sonnet-5-5`, sending `thinking: {"type": "disabled"}` to turn off up-front thinking no longer works at high effort or below; send `thinking: {"type": "between_tools"}` instead.

! On `claude-sonnet-5-5`, `tool_choice` types `any` and `tool` return a 400 error.

! On the Claude API and Google Cloud, `claude-sonnet-5-5` does not accept the `computer_20251124` computer use tool; migrate to the newer tool.

- ! The advisor tool rejects `claude-opus-4-8`, `claude-opus-4-7`, and `claude-sonnet-5` as advisors when using `claude-sonnet-5-5`.
- ! On `claude-opus-5-5`, `thinking: {"type": "disabled"}` and `thinking: {"type": "enabled", ...}` both return a 400 error; omit the `thinking` field and use the `effort` parameter instead.
- ! On `claude-opus-5-5`, `tool_choice` types `any` and `tool` return a 400 error; use `auto` with strict tool use instead.
- ! On the Claude API and Google Cloud, `claude-opus-5-5` requires `computer_toolset_20260801`; `computer_20251124` returns a 400 error on those platforms.
- ! The Compliance API Activity Feed no longer returns file names, project document names, or artifact titles; the `filename` and `title` fields on file, project document, and artifact activities are now always empty or omitted, including for activities recorded before this change.
- ! Cache diagnostics responses from `POST /v1/messages` now always include the `diagnostics` field (null when not opted in), where previously the field was omitted; clients that check for field absence must be updated.
- ! Refusals before any output with `stop_details.category` of `"bio"`, `"frontier_llm"`, or `"reasoning_extraction"` are now billed at the model's standard rates, where previously they were not billed.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Sonnet 5.5 Feedback ⓘ

released · Sep 29, 2026 MODEL ↗

Claude Sonnet 5.5 from Anthropic was released on 28 Sep 2026. 1M-token context; $2.00 in / $10.00 out per million tokens (OpenRouter).

WAS THIS USEFUL? 👍 👎

Google Gemini API

Feedback ⓘ changelog · Sep 29, 2026

CHANGELOG ↗ RANK

Gemini API provides Gemini models for generating text, images, and other multimodal content in applications.

> Gemini API introduced next-generation text-to-speech capabilities, including two new TTS models, a voice discovery endpoint, and tools for designing and replicating custom voices.

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

01 **Flagship creative TTS model** NEW 85

Releases `gemini-3.8-flash-tts`, a flagship creative TTS model engineered for studio-grade voice fidelity, nuanced acting, regional dialects, and long-form multi-turn stability.

Generate studio-grade narration for long-form content using the flagship TTS model.

```
$ curl
https://generativelanguage.googleapis.com/v1beta/models/gemini-
3.8-flash-tts:generateContent \
  -H 'Authorization: Bearer $API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"contents":[{"parts":[{"text":"Welcome to the quarterly
security briefing."}]}]}'
```

— Named model with attributes and a runnable generateContent example

changelog-20260929-c08e2a13

02 **Fast, cost-efficient lite TTS model** NEW 80

Releases `gemini-3.8-flash-lite-tts`, a fast, cost-efficient TTS model built for high-throughput production and real-time voice agent cascades.

Drive high-throughput voice agent pipelines at lower cost using the lite TTS model.

```
$ curl
https://generativelanguage.googleapis.com/v1beta/models/gemini-
3.8-flash-lite-tts:generateContent \
  -H 'Authorization: Bearer $API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"contents":[{"parts":[{"text":"Your verification code is
482913."}]}]}'
```

– Named model with use-case scope and a runnable generateContent example

changelog-20260929-c08e2a13

03 **Extended Voice Library discovery endpoint** NEW 75

Adds the `GET /v1beta/voices` endpoint to query 150+ prebuilt and custom voices in the Extended Voice Library.

Discover all available prebuilt and custom voices before choosing one for a real-time voice agent.

```
$ curl https://generativelanguage.googleapis.com/v1beta/voices -
H 'Authorization: Bearer $API_KEY'
```

– Endpoint and voice count given, but no further query options detailed

changelog-20260929-c08e2a13

THINNER COVERAGE BELOW

04 **Voice Design for custom personas** NEW 30

Adds Voice Design, enabling creation of persistent custom vocal personas from text prompts.

– Describes purpose only, no endpoint or example given changelog-20260929-c08e2a13

05 **Voice Replication with consent verification** NEW 30

Adds Voice Replication with consent verification, allowing voices to be replicated via the Gemini API.

– Mentions consent verification but no mechanism or example changelog-20260929-c08e2a13

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

Gemini API gains next-gen TTS models, a `/v1beta` /voices endpoint, voice design, voice replication, and a 150+ voice library.

└──▷ TRY IT

Discover all available prebuilt and custom voices before choosing one for a real-time voice agent.

```
$ curl https://generativelanguage.googleapis.com/v1beta/voices -H
'Authorization: Bearer $API_KEY'
```

Generate studio-grade narration for long-form content using the flagship TTS model.

```
$ curl
https://generativelanguage.googleapis.com/v1beta/models/gemini-3.8-
flash-tts:generateContent \
  -H 'Authorization: Bearer $API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"contents":[{"parts":[{"text":"Welcome to the quarterly
security briefing."}]}]}'
```

Drive high-throughput voice agent pipelines at lower cost using the lite TTS model.

```
$ curl
https://generativelanguage.googleapis.com/v1beta/models/gemini-3.8-
flash-lite-tts:generateContent \
  -H 'Authorization: Bearer $API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"contents":[{"parts":[{"text":"Your verification code is
482913."}]}]}'
```

Adds the `GET /v1beta/voices` endpoint to query 150+ prebuilt and custom voices in the Extended Voice Library.

Releases `gemini-3.8-flash-tts`, a flagship creative TTS model engineered for studio-grade voice fidelity, nuanced acting, regional dialects, and long-form multi-turn stability.

Releases `gemini-3.8-flash-lite-tts`, a fast, cost-efficient TTS model built for high-throughput production and real-time voice agent cascades.

Adds Voice Design, enabling creation of persistent custom vocal personas from text prompts.

Adds Voice Replication with consent verification, allowing voices to be replicated via the Gemini API.

WAS THIS USEFUL? 👍 👎

OpenAI

Feedback ⓘ changelog · Sep 29, 2026 CHANGELOG ↗ RANK

OpenAI provides AI models and APIs for building applications that generate and reason with text, images, and audio.

> OpenAI shipped two new reasoning models, GPT-6 Sol and GPT-6 Luna, both supporting text and image inputs through the Responses and Chat Completions APIs.

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

01 **GPT-6 Sol and GPT-6 Luna reasoning models** NEW 78

OpenAI added two new reasoning models accepting text and image inputs via the Responses and Chat Completions APIs: `gpt-6-sol` (GPT-6 Sol) priced at $2/$0.20 cached/$10 output per 1M tokens, and `gpt-6-luna` (GPT-6 Luna), a lower-cost option priced at $0.10/$0.01 cached/$0.50 output per 1M tokens.

Invoke the new GPT-6 Sol reasoning model with an image input via the Chat Completions API to run a visual analysis task.

```
$ curl https://api.openai.com/v1/chat/completions \
  -H 'Authorization: Bearer $OPENAI_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gpt-6-sol", "messages": [{"role": "user",
"content": [{"type": "text", "text": "Describe any anomalies in
this screenshot."}, {"type": "image_url", "image_url": {"url":
"https://example.com/screenshot.png"}}]}]}'
```

Use the lower-cost GPT-6 Luna model for high-volume image classification workflows where budget efficiency matters.

```
$ curl https://api.openai.com/v1/chat/completions \
  -H 'Authorization: Bearer $OPENAI_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gpt-6-luna", "messages": [{"role": "user",
"content": [{"type": "text", "text": "Classify the content of
this image."}, {"type": "image_url", "image_url": {"url":
"https://example.com/image.png"}}]}]}'
```

— Names both models, pricing and APIs; runnable curl examples provided.

changelog-20260929-64b08dd0

changelog · Sep 29, 2026 CHANGELOG ↗ LATEST STABLE

OpenAI adds GPT-6 Sol and GPT-6 Luna reasoning models with text and image input via Responses and Chat Completions APIs.

└──▷ TRY IT

Invoke the new GPT-6 Sol reasoning model with an image input via the Chat Completions API to run a visual analysis task.

```
$ curl https://api.openai.com/v1/chat/completions \
  -H 'Authorization: Bearer $OPENAI_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gpt-6-sol", "messages": [{"role": "user",
"content": [{"type": "text", "text": "Describe any anomalies in
this screenshot."}, {"type": "image_url", "image_url": {"url":
"https://example.com/screenshot.png"}}]}]}'
```

Use the lower-cost GPT-6 Luna model for high-volume image classification workflows where budget efficiency matters.

```
$ curl https://api.openai.com/v1/chat/completions \
  -H 'Authorization: Bearer $OPENAI_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "gpt-6-luna", "messages": [{"role": "user",
"content": [{"type": "text", "text": "Classify the content of this
image."}, {"type": "image_url", "image_url": {"url":
"https://example.com/image.png"}}]}]}'
```

Adds `gpt-6-sol` (GPT-6 Sol), a reasoning model accepting text and image inputs at $2/$0.20 cached/$10 output per 1M tokens, available via the Responses and Chat Completions APIs.

Adds `gpt-6-luna` (GPT-6 Luna), a lower-cost reasoning model accepting text and image inputs at $0.10/$0.01 cached/$0.50 output per 1M tokens, available via the Responses and Chat Completions APIs.

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

DeepSeek V3 0324

Feedback ⓘ price increase · Sep 28, 2026

MODEL ↗

DeepSeek V3 0324 is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-chat-v3-0324): $0.25 → $0.29 in, $1.00 → $1.14 out per million tokens (+16%), first seen 28 Sep 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V3.2 Exp

Feedback ⓘ price increase · Sep 28, 2026

MODEL ↗

DeepSeek V3.2 Exp is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v3.2): $0.269 → $0.28 in, $0.40 → $0.42 out per million tokens (+5%), first seen 28 Sep 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4.1 Flash

Feedback ⓘ

price increase · Sep 28, 2026 MODEL ↗

DeepSeek V4.1 Flash is a multimodal model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v4.1-flash): $0.099 → $0.30 in, $0.60 → $1.20 out per million tokens (+203%), first seen 28 Sep 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.3 Flash

Feedback ⓘ price increase · Sep 28, 2026

MODEL ↗

GLM 5.3 Flash is a reasoning model from Z.ai.

Price increase on OpenRouter (z-ai/glm-5.3-flash): $0.045 → $0.15 in, $0.14 → $0.50 out per million tokens (+257%), first seen 28 Sep 2026.

WAS THIS USEFUL? 👍 👎

Google Gemma 4 26B A4B

Feedback ⓘ

price increase · Sep 28, 2026 MODEL ↗

Gemma 4 26B A4B is a multimodal model from Google.

Price increase on OpenRouter (google/gemma-4-26b-a4b-it): $0.0675 → $0.09 in, $0.225 → $0.30 out per million tokens (+33%), first seen 28 Sep 2026.

WAS THIS USEFUL? 👍 👎

Moonshot AI Kimi K2.6

Feedback ⓘ price cut · Sep 28, 2026

MODEL ↗

Kimi K2.6 is a multimodal model from Moonshot AI.

Price cut on OpenRouter (moonshotai/kimi-k2.6): $0.95 → $0.65 in, $4.00 → $3.41 out per million tokens (-32%), first seen 28 Sep 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-35B-A3B

Feedback ⓘ

price cut · Sep 28, 2026 MODEL ↗

Qwen3.5-35B-A3B is a multimodal model from Alibaba.

Price cut on OpenRouter (qwen/qwen3.5-35b-a3b): $0.3125 → $0.1625 in, $1.25 → $1.30 out per million tokens (-48%), first seen 28 Sep 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 29, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue