THE AI TOOLCHAIN

NO. 045

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 3, 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 253 changes

Do you prefer **scroll**?

▸ // CONTENTS SHOW 55 of 57 tools

▾ // THE BRIEF HIDE SAME ISSUE, SAME PROMPT, TWO WRITERS: GPT-5.5 | Opus 5

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

Three unrelated things stand out today: Modal gives agents full Linux VMs when containers are too narrow, Docker Sandboxes moves isolated

vim ? help

agent runs onto managed cloud microVMs, and Braintrust makes coding-agent behavior easier to slice by project, span, and dashboard without custom reporting work.

DEPLOY

Run agent jobs in stronger isolation than a plain container

Modal's VM Sandboxes give agents full Linux VMs with Docker and kernel-level access; Docker's Cloud Sandboxes run agent tasks in isolated microVMs through `sbx`. Teams can hand off riskier code execution, tool use, and environment setup to disposable compute instead of maintaining bespoke runner fleets or over-permissioned containers.

Modal · Docker Sandboxes

OPERATE

See coding-agent behavior without building the reporting layer yourself

Braintrust added project-group placement, broader span customization, a Coding Agent Insights dashboard, and natural-language chart building; mlxtop now reads Prometheus metrics from eleven LLM runtimes with per-request speed and token views. That makes it cheaper to answer "which agent, model, or runtime got slower or more expensive?" without stitching traces, metrics, and dashboards by hand.

Braintrust · mlxtop

GOVERN

Put hard boundaries around what agents can reach and spend

Pipelock added operator containment controls, per-process receipt chains, receipt-gated credential allows, and token scanning; OpenBot now enforces network egress policy for Bot computers. The practical win is that agent permissions become explicit runtime policy instead of convention, so a bad tool call is blocked or evidenced before it leaks across a trust boundary.

Pipelock · OpenBot

Does **GPT-5.5** read better? 👍 👎

OFFENSIVE SECURITY

◆ Exploitation & C2

xalgorix

Feedback ⓘ v4.6.132 NOTES ↗ RANK

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

> xalgorix v4.6.132 adds resilience to long-running scans with a configurable non-progress timeout and a new event journal for tracking scan history, alongside checkpoint-based recovery from interruptions.

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

01 **Configurable non-progress limit for scans** NEW 81

A new `XALGORIX_MAX_NON_PROGRESS` environment variable controls the active non-progress limit for a scan, defaulting to one hour; setting it to `0` disables the limit entirely, letting scans run unattended against slow or rate-limited targets.

Prevent a long-running scan from stalling indefinitely by capping the non-progress window to 30 minutes, or disable the limit entirely for unattended overnight assessments.

```
$ XALGORIX_MAX_NON_PROGRESS=1800 xalgorix scan <target>
```

Disable the non-progress limit so a scan against a slow or rate-limited target is never halted automatically.

```
$ XALGORIX_MAX_NON_PROGRESS=0 xalgorix scan <target>
```

— Names exact env var, default, and disable value with runnable examples v4.6.132

THINNER COVERAGE BELOW

02 **Scan event journal with pagination** NEW 32

Scan events are now journaled with stable identities, bounded live history, and paginated access to retained events.

— Describes mechanism but no API, flag, or usage shown v4.6.132

03 **Interrupted-scan recovery via checkpoint restore** NEW

15

xalgorix v4.6.132 adds interrupted-scan recovery with checkpoint restore, allowing a scan to resume after being interrupted.

— Only a headline mention, no mechanism or usage detail given v4.6.132

v4.6.132 NOTES ↗ LATEST STABLE

xalgorix v4.6.132 adds interrupted-scan recovery with checkpoint restore and a configurable non-progress limit via `XALGORIX_MAX_NON_PROGRESS`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.6.132
https://github.com/xalgorix/xalgorix.git
# already have the repo? check out this version:
$ git checkout v4.6.132
```

└──▷ TRY IT

Prevent a long-running scan from stalling indefinitely by capping the non-progress window to 30 minutes, or disable the limit entirely for unattended overnight assessments.

```
$ XALGORIX_MAX_NON_PROGRESS=1800 xalgorix scan <target>
```

Disable the non-progress limit so a scan against a slow or rate-limited target is never halted automatically.

```
$ XALGORIX_MAX_NON_PROGRESS=0 xalgorix scan <target>
```

Adds `XALGORIX_MAX_NON_PROGRESS` environment variable to control the active non-progress limit; set to `0` to disable the limit (default: one hour).

Journals scan events with stable identities, bounded live history, and paginated access to retained events.

WAS THIS USEFUL? 👍 👎

BUILD

◆ AI Coding Agents

Warp

Feedback ⓘ 2026.09.23 (v0.2026.09.23.14.34) NOTES ↗ RANK

Warp is an AI terminal that runs commands and helps developers build software.

> Warp's latest releases let the agent run on a user's existing ChatGPT plan, added AWS ECR credentials to managed secrets, gave cloud agent runs crash-resilient handoff snapshots, and rounded out shell/terminal ergonomics with native fzf Alt-C support and easier team switching.

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

01 **ChatGPT sign-in for Warp Agent inference** NEW 70

Adds ChatGPT sign-in under `Settings > Agents > Warp Agent > Custom Inference`, letting users bring their existing ChatGPT subscription to power Warp's agent instead of paying separately.

Enable your existing ChatGPT plan as the inference backend for Warp's agent instead of the default.

> 📍 In Warp, go to Settings › Agents › Warp Agent › Custom Inference, then sign in with ChatGPT.

Connect your existing ChatGPT subscription to Warp's agent instead of paying separately.

> 📍 In Warp, go to Settings › Agents › Warp Agent › Custom Inference and sign in with ChatGPT.

— Exact settings path given, but no detail on billing or model behavior.

2026.09.23 (v0.2026.09.23.14.34)

02 **Native fzf Alt-C across bash, zsh, fish** NEW 70

Adds `fzf` `Alt-C` support across bash, zsh, and fish shell integrations, letting users press Alt-C inside a Warp terminal to open the fzf directory picker and cd into the selected path.

Use fzf Alt-C to interactively cd into a directory from bash, zsh, or fish — now natively wired into Warp shell integrations.

📍 Press Alt-C inside a Warp terminal running bash, zsh, or fish to open the fzf directory picker and cd into the selected path.

— Names the exact keybinding, tool, and shells, with a runnable action.

2026.09.30 (v0.2026.09.30.08.29)

THINNER COVERAGE BELOW

03 **Cloud agent handoff snapshot on interruption** NEW 55

Cloud agent runs now save a handoff snapshot on `SIGTERM` / `SIGINT` interruption so in-progress work can be resumed.

— Names the signals and resumability mechanism but no resume command.

2026.09.23 (v0.2026.09.23.14.34)

04 **Team switching and workspace discovery** IMPROVED 55

Adds the option to switch teams from `Settings > Teams`, moves the team switcher next to the profile avatar for easier access, and team discovery now shows native workspace options and workspace-scoped teams.

Switch between Warp team workspaces without leaving the app.

📍 Open Settings › Teams, then select the team you want to switch to.

— Exact settings path for switching teams; workspace-discovery mechanics stay vague.

2026.09.30 (v0.2026.09.30.08.29) 2026.09.23 (v0.2026.09.23.14.34)

05 **Conversation usage and cost popover** NEW 50

Adds a usage popover to the agent input footer displaying conversation usage and cost, so users can monitor token consumption mid-conversation before a long agent run gets expensive.

Monitor token consumption and cost mid-conversation to stay within budget before a long agent run gets expensive.

📍 Open the agent input footer in any active agent conversation and click the usage popover to view current conversation usage and cost.

— Describes what it shows and where to click, no numeric detail on cost calc.

2026.09.23 (v0.2026.09.23.14.34)

06 **Neovim 0.13 text objects in Vim mode** NEW 40

Adds Neovim 0.13 `al` and `il` text objects to Vim mode.

— Named exact text objects but no usage example provided.

2026.09.23 (v0.2026.09.23.14.34)

07 **AWS ECR credentials in managed secrets** NEW 30

Adds AWS ECR credential support in managed secrets.

— Names the integration but gives no setup steps or scope.

2026.09.23 (v0.2026.09.23.14.34)

08 **Immediate exit on zero idle timeout** IMPROVED 30

Agent runs with a zero post-completion idle timeout now exit immediately.

— States the behavior change with no further mechanism or config detail.

2026.09.23 (v0.2026.09.23.14.34)

09 **Tab group name highlighting in vertical tabs search** IMPROVED 25

Vertical tabs search now matches and highlights tab group names.

— One-line UI tweak with no further mechanism described. 2026.09.30 (v0.2026.09.30.08.29)

└──▷ ALSO FROM THESE RELEASES

2026.09.23 (v0.2026.09.23.14.34) NOTES ↗ STABLE

Warp adds ChatGPT sign-in, AWS ECR managed secrets, and a conversation usage/cost popover for the agent.

└──▷ HOW TO FIND IT

Enable your existing ChatGPT plan as the inference backend for Warp's agent instead of the default.

📍 In Warp, go to Settings › Agents › Warp Agent › Custom Inference, then sign in with ChatGPT.

Monitor token consumption and cost mid-conversation to stay within budget before a long agent run gets expensive.

📍 Open the agent input footer in any active agent conversation and click the usage popover to view current conversation usage and cost.

Adds ChatGPT sign-in under `Settings > Agents > Warp Agent > Custom Inference` so users can bring their existing ChatGPT plan to Warp's agent.

Adds AWS ECR credential support in managed secrets.

Adds a usage popover to the agent input footer displaying conversation usage and cost.

Cloud agent runs now save a handoff snapshot on `SIGTERM` / `SIGINT` interruption so in-progress work can be resumed.

Agent runs with a zero post-completion idle timeout now exit immediately.

+2 more

2026.09.30 (v0.2026.09.30.08.29) NOTES ↗ LATEST STABLE

Warp 2026.09.30 adds team-switching from Settings, tab-group search highlighting, and fzf Alt-C across bash, zsh, and fish.

└──▷ HOW TO FIND IT

Switch between Warp team workspaces without leaving the app.

📍 Open Settings › Teams, then select the team you want to switch to.

Use fzf Alt-C to interactively cd into a directory from bash, zsh, or fish — now natively wired into Warp shell integrations.

📍 Press Alt-C inside a Warp terminal running bash, zsh, or fish to open the fzf directory picker and cd into the selected path.

Adds the option to switch teams from Settings > Teams.

Vertical tabs search now matches and highlights tab group names.

Adds `fzf` `Alt-C` support across bash, zsh, and fish shell integrations.

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.761 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 v0.1.761 adds day-level drill-down in usage charts and counts OpenCode's own API calls.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.761 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.761
```

Adds day selection to usage charts: clicking a day filters totals, model/provider/agent breakdowns, request rows, and CSV exports to that day; clicking again restores the full period.

Counts API calls made by OpenCode on its own providers (e.g. OpenCode Zen) the same way Claude Code's and Codex's calls are counted, excluding calls already logged through magpie's gateway.

NEW INTEGRATION Droid (Factory CLI) session integration

▾ 24 more releases in this issue · 2026-10-03

v0.1.758 CODE ↗ STABLE

Magpie v0.1.758 adds adaptive Claude thinking support, routing group effort caps, cache-warm reasoning persistence, and keyboard shortcuts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.758 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.758
```

Adds `Cmd+,` (macOS) / `Ctrl+,` (Windows and Linux) keyboard shortcut to open Settings.

Sends `thinking.type.adaptive` instead of `thinking.type.enabled` for Claude models (e.g. claude-opus-5.5) that no longer accept the older thinking type.

Caps member model effort at the routing group's own reasoning levels, so a group configured up to `high` will never send `xhigh` or `max` even when an agent or suffix requests it.

Preserves a conversation's reasoning level across turns when a routing group's auto-effort cache is warm, going up on harder turns but not down until the cache goes cold.

v0.1.755 CODE ↗ STABLE

WorkBuddy subscription image models (text-to-image, image-to-image) are now available as magpie image models via Settings › Model.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.755 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.755
```

Adds WorkBuddy plan image model support: models tagged `text-to-image` and `image-to-image` in WorkBuddy's `/v3/config` — including `hunyuan-image-v3.0-art`, `hunyuan-image-alpha`, its `-edit` variant, and `gpt-image-2.5-sunburst` — are now surfaced as magpie image models under Settings › Model.

Requests are routed to WorkBuddy's `/v2/images/generations` and `/v2/images/edits` endpoints, with image edits sent as data URLs, and responses unwrapped from WorkBuddy's `{code, msg, data}` envelope into OpenAI's shape.

Works for both the built-in WorkBuddy integration and the `@magpie-community/opencode-workbuddy-auth` plugin — magpie shapes the request and the plugin's fetch signs and sends it.

Adds Provider.Origin() method to resolve the scheme-and-host a provider's requests target, enabling side-channel API calls (e.g. WorkBuddy's `/v3/config` alongside `/v2`) for both plugin and built-in providers.

v0.1.753 CODE ↗ STABLE

Magpie's web search now works across DeepSeek's Anthropic API, with automatic fallback when a searcher finds nothing

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.753 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.753
```

Adds DeepSeek's Anthropic API (`api.deepseek.com`) to the `Anthropic` protocol's `searchHosts` , enabling `web_search_20250305` and `web_search_tool_result` blocks through that endpoint.

Searchers that find nothing on their first API now fall through to the next available searcher automatically, rather than returning an empty result.

Populates `page_age` in `web_search_tool_result` blocks when the upstream hit carries an age, instead of always sending `null` .

v0.1.752 CODE ↗ STABLE

magpie v0.1.752 adds Goose as a custom provider, read-only Alma session browsing, and toggleable quota curves in the UI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.752 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.752
```

└──▷ USE IT

Register magpie as a Goose custom provider with per-model context windows and thinking-level support so Goose picks the right context limit and shows the Thinking Level picker for capable models.

custom_providers/magpie.json

```
{
  "id": "magpie",
  "dynamic_models": false,
  "skip_canonical_filtering": true,
  "models": [
    {
      "id": "claude-opus-4-5",
      "context_limit": 200000,
      "reasoning": true
    },
    {
      "id": "gpt-4o",
      "context_limit": 128000
    }
  ]
}
```

Runs Goose (Block's CLI and desktop app) as a custom provider via a JSON file in `custom_providers` beside `config.yaml`, passing each model's `context_limit` and `reasoning` fields so Goose respects per-model context windows and surfaces the Thinking Level picker.

Displays Alma's chat sessions (`chat_threads.db`) read-only on the Sessions page, reading from Alma's SQLite database in WAL mode at `~/Library/Application Support/alma` (macOS), `~/.config/alma` (Linux), or `%APPDATA%\alma` (Windows).

Adds a toggle to turn off quota curves in the allowances panel, reducing visual clutter.

v0.1.751 CODE ↗ STABLE

Magpie now loads pi coding agent packages as plugins, adding their providers, models, OAuth/key sign-in, and request handling.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.751 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.751
```

Adds support for `@earendil-works/pi-coding-agent` and `@earendil-works/pi-ai` packages (and legacy `@mariozechner/pi-` scope) as first-class magpie plugins, exposing each extension's providers, models, sign-in (OAuth or API key), token renewal, and request routing through magpie's existing plugin host.

Stores pi provider authentication in `plugin-auth.json`, with OAuth accounts as `{type: 'oauth', refresh, access, expires, …}` and API-key accounts as `{type: 'api', key}`, following the same structure as OpenCode plugin auth.

v0.1.749 CODE ↗ STABLE

Clicking a provider in the macOS menu-bar quota view now jumps directly to that provider's allowance card in the Quotas tab.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.749 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.749
```

v0.1.748 CODE ↗ STABLE

magpie now tells you when Antigravity's 429 is a system-prompt rejection, not a spent quota

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.748 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.748
```

Adds `antigravityTurnsAway` detection and `antigravityTurnedAwayHint` message to distinguish Antigravity's 429 rate-limit responses caused by Claude Code's or Claude Agent SDK's system prompt from genuine quota exhaustion, appended to the error when `accountAgent` is `antigravity` and the system prompt contains `x-anthropic-billing-header:` or `built on Anthropic's Claude Agent SDK`.

v0.1.745 CODE ↗ STABLE

Plugin cards on Plugins › Discover now have a functional `Update to vX.Y.Z` button that triggers in-place upgrades.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.745 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.745
```

v0.1.739 CODE ↗ STABLE

Multi-account subscriptions now show one account by default with a collapsible Show N more accounts button in the Usage and tray panel.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.739 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.739
```

v0.1.736 CODE ↗ STABLE

Magpie now routes GLM (Zhipu/Z.ai) Responses API requests with web_search natively, matching DeepSeek behaviour.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.736 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.736
```

Adds `open.bigmodel.cn` and `api.z.ai` to the `searchHosts[provider.Responses]` allowlist so Zhipu GLM and Z.ai Responses API requests carrying the `web_search` tool are passed through to `/v1/responses` unchanged, rather than being downgraded to Chat Completions.

v0.1.733 CODE ↗ STABLE

magpie v0.1.733 adds Codex Ultra support for GPT-6.1 Sol and related models via Copilot and other providers.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.733 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.733
```

Adds `OffersUltra` detection so Codex Ultra is offered on `gpt-6.1-sol`, `gpt-6-sol`, `gpt-6-astra`, `gpt-5.6-sol`, and `gpt-5.6-terra` wherever they are served (Copilot, API key, relay), while `gpt-6-luna` and `gpt-5.6-luna` stop at `max`.

Enables Codex multi-agent V2 task delegation — when a qualifying Ultra model is served at multi-agent V2, Codex hands parts of the task to its own agents automatically.

v0.1.732 CODE ↗ STABLE

Magpie now auto-describes images via Vision for text-only models whose capability lists make no mention of image support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.732 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.732
```

Automatically routes image inputs through the Vision describer when a model's provider list says nothing about image support, so text-only models receive a text description instead of a raw image payload.

v0.1.728 CODE ↗ STABLE

Magpie now mirrors Antigravity's model picker order and reads each model's context window, output limit, and image support from Antigravity directly.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.728 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.728
```

Reads `maxTokens`, `maxOutputTokens`, and `supportsImages` for each model from Antigravity's `fetchAvailableModels` response, so context windows and image capabilities reflect what Antigravity reports rather than defaults from another provider.

Preserves Antigravity's `agentModelSorts` picker order so newly added models (such as Claude Opus 5.5 and Sonnet 5.5) appear in the same position Antigravity places them, not appended by ID after older entries.

v0.1.727 CODE ↗ STABLE

Adds drag-to-reorder provider cards on the Usage page, persisted via new `UsageOrder` setting and `POST /api/usage/arrange` endpoint.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.727 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.727
```

└──▷ **TRY IT**

Programmatically set the Usage page provider card order — useful when scripting a fresh magpie setup to match a team standard.

```
$ curl -s -X POST http://localhost:<port>/api/usage/arrange \
  -H 'Content-Type: application/json' \
  -d '{"Order": ["kimi", "codex", "openai"]}'
```

Adds `POST /api/usage/arrange` endpoint accepting `{"Order": ["<provider-id>", ...]}` to persist the drag order of provider cards on the Usage page.

Adds `usageOrder` field (type `[]string`, JSON key `usageOrder`) to the settings store, recording the Usage page's provider card order by provider ID.

Enables drag-to-reorder for provider cards on the Usage page, matching the existing drag-to-reorder behaviour on the Agents page.

v0.1.724 CODE ↗ STABLE

Magpie v0.1.724 adds Alibaba Cloud Bailian's decision model as a `bailian-decision` routing preset and renames the decision API protocol to System One.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.1.724 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.724
```

Adds `bailian-decision` as a named routing preset, letting Bailian workspace-keyed providers (`.maas.aliyuncs.com` / `.maas.qianwenaiapi.com`) use the `decision-model-preview` model for routing decisions across `cn-beijing`, `ap-southeast-1`, and `token-plan` regions.

Renames the decision API protocol label from Jev to System One in the provider UI, now covering TypeSafe's Jev, compatible gateways, and Bailian's decision model under one protocol family.

Adds `decide` field support in the typed provider draft (`POST …/systemone`) so custom providers can specify a System One decision base URL alongside chat, responses, and

Anthropic endpoints.

v0.1.722 CODE ↗ STABLE

Adds quota history curves to the Usage page and tray card, backed by a new `usage/quotas/history` API endpoint and `quota-history.json` store.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.722 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.722
```

└──▷ TRY IT

Pull 35 days of quota history for all accounts to feed a custom dashboard or alerting script.

```
$ curl 'http://localhost:<port>/usage/quotas/history?days=35'
```

Exposes a `GET usage/quotas/history?days=35` API endpoint that returns each account's subscription-window quota readings over time, enabling the Usage page to draw a 'what was left over time' curve against an even burn rate.

Adds `quota-history.json` to persist per-provider, per-account quota window readings (percent left, window start, reset time) across runs, with cross-device merging via the existing usage sync (`davsync`).

Renders quota history curves on Usage page quota cards and a current-cycle line on the tray card, showing quota consumption against an even burn over up to 35 days.

v0.1.721 CODE ↗ STABLE

Adds a China mirror switch on the Plugins page to route plugin list, npm packages, and Bun downloads through mainland-friendly mirrors.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.721 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.721
```

└──▷ TRY IT

Enable the China mirror at runtime via the API so plugin and Bun downloads prefer mainland-accessible mirrors without touching the UI.

```
$ curl -X POST http://localhost:<port>/api/plugins/mirror -H
'Content-Type: application/json' -d '{"on": true}'
```

Adds a `ChinaMirror` setting (toggled via the Plugins page's '国内镜像' switch) that routes plugin registry, npm, and Bun downloads through `registry.npmmirror.com` and `cdn.jsdelivr.net` instead of GitHub/npm directly.

Exposes `POST /api/plugins/mirror` endpoint to programmatically toggle the China mirror switch on or off, and `GET /api/plugins` now returns a `mirror` field reflecting its current state.

v0.1.720 CODE ↗ STABLE

magpie v0.1.720 adds Droid (Factory CLI) session support: reads from `~/.factory/sessions` or `$FACTORY_HOME_OVERRIDE/.factory/sessions`, tracks spend, and surfaces `droid --resume <id>`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.720 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.720
```

Adds `droid` as a supported agent (`agent` field value) in the Sessions view, listing Droid (Factory CLI) sessions alongside Claude Code, Codex, Cline, and others.

Surfaces `droid --resume <id>` as the resume command for Droid sessions shown in the Sessions view.

Reads Droid session data from `~/.factory/sessions` (or `$FACTORY_HOME_OVERRIDE/.factory/sessions`), including token usage and cost from each session's `.settings.json` file.

v0.1.715 CODE ↗ STABLE

Adds a Settings switch (`CodexAgentsV1`) so OpenAI/GPT lead models handed to Codex advertise multi-agent V1, enabling subagents to run on magpie's other models.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.715 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.715
```

Adds `CodexAgentsV1` setting: when enabled, OpenAI models magpie surfaces to Codex declare `multi_agent_version` V1, allowing a GPT lead agent's subagents to run on magpie's non-OpenAI models.

v0.1.714 CODE ↗ STABLE

magpie v0.1.714 adds native Android/Termux support across builds, installation, browser integration, and autostart.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.714 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.714
```

Supports Android arm64/amd64 CLI builds and release targets via Termux, selecting `nogui` for Makefile builds.

Installs the Android CLI asset into `$PREFIX/bin` with SHA-256 verification, and explains source builds when no Android asset is present in a release.

Opens Web and OAuth links on Android using `termux-open-url` instead of a desktop browser.

Uses Termux:Boot (`~/.termux/boot/magpie`) to autostart the gateway after Android boots.

Upgrades the managed Bun runtime to 1.4.2 to gain official Android arm64/amd64 support (Bun 1.3 had no Android build).

v0.1.712 CODE ↗ STABLE

Adds native ModelScope image-generation support via async task API with per-user queue and retry handling.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.712 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.712
```

Adds `X-ModelScope-Async-Mode` header support and `GET /tasks/<id>` polling for ModelScope image generation, serialising requests one at a time per user and retrying on 'user queue is full' responses.

Adds `drawModelScope` gateway path for providers hosted at `api-inference.modelscope.cn`, handling ModelScope's non-OpenAI response shape (no `data` wrapper, uses `output_images` instead).

v0.1.710 CODE ↗ STABLE

Magpie v0.1.710 syncs gateway-key changes and distributes skills into Claude Desktop's Cowork skills-plugin

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.710 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.710
```

Propagates `caller-keys.json` gateway-key changes (rotate, disable, remove) through the existing settings-part hash and merge in WebDAV sync, so revoked credentials no longer survive a later settings-only upload from another computer.

Copies library skills into every Claude Desktop Cowork account's `skills-plugin` folder (`local-agent-mode-sessions/skills-plugin/<org>/<account>/`) and writes both a `manifest.json` entry and the skill files under `skills/<name>/SKILL.md`, so Desktop's Cowork lists them.

v0.1.709 CODE ↗ STABLE

magpie v0.1.709 adds oMLX local-server preset, Factory provider via Claude Code and Desktop, and a stale CLI detector with `cliBehind` API field

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.709 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.709
```

└──▷ TRY IT

Poll whether your local magpie CLI has fallen behind the running app — useful in a health-check script or CI step on macOS.

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

Adds `cliBehind` field to `/api/state` to expose whether the `magpie` CLI binary is a stale copy that has fallen behind the app.

Adds update.StaleCLI(), checking `MAGPIE_BIN_DIR` or `~/.local/bin/magpie`, to detect a copied (non-symlinked) CLI on macOS and notify the user.

Adds oMLX as a local-server preset alongside Ollama and LM Studio — its `:8000` server speaks Chat, Responses, and Anthropic Messages and requires no API key on a loopback bind.

Adds moved Factory provider support, reaching Factory again from Claude Code (via a global `CLAUDE.md` or a `SessionStart` hook) and from Claude Desktop, using plugin 0.1.12.

v0.1.708 CODE ↗ STABLE

Magpie v0.1.708 forwards `output_config.format` JSON schemas to Claude Code via `--json-schema` for structured output.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.1.708 https://github.com/yetone/magpie.git
# already have the repo? check out this version:
$ git checkout v0.1.708
```

Mac menu bar icon stays lit while the panel is open, matching the behaviour of other macOS status-bar apps.

WAS THIS USEFUL? 👍 👎

claude-mem

Feedback ⓘ v13.29.0 NOTES ↗ RANK

claude-mem gives coding agents persistent memory by capturing session activity, compressing it with a model, and injecting relevant context into later sessions.

> claude-mem v13.29.0 adds a persistent work-state memory tool, broad new model-provider options (OpenAI-compatible endpoints, Codex CLI, multi-key rotation, quota fallback), two new IDE integrations (Kimi Code, Oh My Pi), and several breaking changes to security defaults and timeout values.

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

01 **Codex CLI support with throughput controls** NEW 93

Adds `CLAUDE_MEM_PROVIDER=codex` (or `npx claude-mem install --provider codex [--model <id>]`) to run the observer through the Codex CLI app-server on a ChatGPT subscription, with `CLAUDE_MEM_CODEX_MODEL`, `CLAUDE_MEM_CODEX_REASONING_EFFORT` (default `low`), and `CLAUDE_MEM_CODEX_PATH` (default `codex`). Throughput is tunable via `CLAUDE_MEM_CODEX_MAX_CONCURRENT_AGENTS` (default `2`), `CLAUDE_MEM_CODEX_OBSERVATION_BATCH_SIZE` (default `8`), and `CLAUDE_MEM_CODEX_OBSERVATION_BATCH_MAX_CHARS` (default `32000`), settable in `settings.json`.

Install claude-mem to use your ChatGPT Codex subscription as the observer backend, with batching tuned for throughput.

```
$ npx claude-mem install --provider codex --model o4-mini
```

— Install command, defaults, and config file location all given. v13.29.0

02 **DNS-rebinding guard on the worker** BREAKING 90

The worker now returns 403 for requests whose `Host` is a DNS name other than `localhost`, `*.localhost`, `host.docker.internal`, `CLAUDE_MEM_WORKER_HOST`, the `CLAUDE_MEM_PUBLIC_URL` host, or an allowed origin. Adds `CLAUDE_MEM_ALLOWED_ORIGINS` (comma-separated) so requests through non-localhost host names (`.local` names, reverse proxies) can pass; add your hostname to it and restart the worker if you access the viewer via a custom host.

— Exact allowed hosts, env var, and remediation steps given. v13.29.0

03 **Kimi Code CLI support** NEW 90

Adds Kimi Code CLI support via `npx claude-mem install --ide kimi` (or `npx claude-mem kimi install|status|uninstall`), writing hooks to `$KIMI_CODE_HOME/config.toml` and the MCP server to `$KIMI_CODE_HOME/mcp.json` (default `~/.kimi-code`).

— Exact install commands and config file paths given. v13.29.0

04 **Oh My Pi (OMP) support** NEW 86

Adds Oh My Pi (OMP) support via `npx claude-mem install --ide omp`, installing a hook at `~/.omp/agent/hooks/pre/claude-mem.ts` (or `$PI_CODING_AGENT_DIR`) that captures sessions, prompts, tool results, and summaries.

— Exact install command and hook path given. v13.29.0

05 **OpenAI-compatible provider with presets** NEW 85

Adds `CLAUDE_MEM_PROVIDER=openai-compatible` to run the observer against any OpenAI `/chat/completions` endpoint, configured via `CLAUDE_MEM_OPENAI_COMPAT_PRESET`, `CLAUDE_MEM_OPENAI_COMPAT_BASE_URL`, `CLAUDE_MEM_OPENAI_COMPAT_MODEL`, and `CLAUDE_MEM_OPENAI_COMPAT_API_KEY`. Ships presets for `nvidia-nim`, `deepseek`, `groq`, `together`, `ollama`, `lmstudio`, `vllm`, `orcarouter`, `opencode-go`, `opencode-zen`, `minimax`, `minimax-cn`, `api-route`, and `opper`, plus a new OpenAI-compatible endpoint (BYOK) option in the viewer UI to edit preset, base URL and model interactively.

Switch the observer to a local Ollama instance using the new openai-compatible provider so no API key or external service is required.

```
$ CLAUDE_MEM_PROVIDER=openai-compatible
CLAUDE_MEM_OPENAI_COMPAT_PRESET=ollama
CLAUDE_MEM_OPENAI_COMPAT_BASE_URL=http://localhost:11434
CLAUDE_MEM_OPENAI_COMPAT_MODEL=llama3.2 npx claude-mem install
```

— Full env var set, 14 presets, and a runnable install example. v13.29.0

06 **Persistent work-state MCP tools** NEW 83

Adds `work_state_write` and `work_state_read` MCP tools giving agents a canonical to-do list that persists across sessions, backed by a new append-only `work_state_entries` table (schema v61) with fields capped at 2,000 characters as JSON.

— Names the MCP tools, table and schema version but no usage walkthrough. v13.29.0

07 **Breaking changes to timeout and model defaults** BREAKING 75

`CLAUDE_MEM_LLM_TIMEOUT_MS` default rises from `30000` to `180000` (a `settings.json` holding exactly `30000` is rewritten automatically; other explicit values are kept). The default OpenRouter model changes from `xiaomi/mimo-v2-flash:free` (retired) to `cohere/north-mini-code:free`, with a stored `CLAUDE_MEM_OPENROUTER_MODEL` matching the retired id rewritten once when the base URL is blank or openrouter.ai. The `UserPromptSubmit` hook timeout in Claude Code drops from 60 seconds to 15 seconds.

— Exact before/after values given, but migration is automatic with no reader action.

v13.29.0

08 **Worker timeout and failure-handling tuning** IMPROVED

70

Adds `CLAUDE_MEM_HOOK_FAIL_LOUD_THRESHOLD` (default `3`) to control how many consecutive unreachable-worker hooks trigger a per-session restart notice rather than blocking the session, `CLAUDE_MEM_SESSION_INIT_TIMEOUT_MS` (default 10 seconds; 7 seconds on Windows) to bound the worker round-trip budget during session init, and `CLAUDE_MEM_FIELD_OPTIMIZE_TIMEOUT_MS` (default `180000`) for oversized-field condensation timeout independent of the main LLM timeout.

— Three config keys with defaults, no setup example. v13.29.0

09 **Antigravity install fix requiring re-run** BREAKING

70

Antigravity users must re-run `npx claude-mem install --ide antigravity` because earlier installs wrote `~/.gemini/config/hooks.json` in a shape that `agy` ignores, so nothing was captured.

— Exact remediation command and broken file path given. v13.29.0

10 **Main-agent-only SessionStart context default** BREAKING

65

Adds `CLAUDE_MEM_CONTEXT_MAIN_AGENT_ONLY` (now defaulting to `true`) to exclude subagent observations from the SessionStart context window while keeping them available in search, timeline, and `get_observations`; set it to `false` to restore the previous behavior.

— Default change explained with exact rollback flag. v13.29.0

11 **Quota fallback provider routing** NEW

60

Adds `CLAUDE_MEM_QUOTA_FALLBACK_PROVIDER` (accepts `claude`, `gemini`, `openrouter`, or `openai-compatible`; empty/off by default) and `CLAUDE_MEM_QUOTA_FALLBACK_MODEL` to route observer work to a secondary provider during quota cooldowns.

— Env vars and accepted values named, no worked example. v13.29.0

12 **Multi-key API rotation across providers** NEW 60

Adds `CLAUDE_MEM_OPENAI_COMPAT_API_KEYS`, `CLAUDE_MEM_GEMINI_API_KEYS`, and `CLAUDE_MEM_OPENROUTER_API_KEYS` for multi-key pools (comma- or newline-separated) with automatic rotation on rate-limit, quota, or auth failure.

— Mechanism and three env vars named, no example shown. v13.29.0

13 **OpenRouter reasoning effort and request body control** NEW 60

Adds `CLAUDE_MEM_OPENROUTER_REASONING_EFFORT` (`none`, `minimal`, `low`, `medium`, or `high`) and `CLAUDE_MEM_OPENROUTER_EXTRA_BODY` (merges a JSON object into OpenRouter requests) for fine-grained OpenRouter control.

— Named vars with accepted values, no usage example. v13.29.0

14 **DeepSeek Harness transcript watching** NEW 60

Adds opt-in DeepSeek Harness transcript watching for zstd-compressed `*.jsonl.zstd` session logs via a `dsh` schema entry in `~/.claude-mem/transcript-watch.json`.

— Names the config file and format but no setup steps. v13.29.0

THINNER COVERAGE BELOW

15 **Observer output token cap** NEW 55

Adds `CLAUDE_MEM_OBSERVER_MAX_OUTPUT_TOKENS` (default `4096`) to set the output cap on OpenRouter, openai-compatible, cmem.ai gateway, and Gemini observer requests.

— One config key with default, scope named but thin. v13.29.0

┗──▷ BREAKING ON UPGRADE

! `CLAUDE_MEM_LLM_TIMEOUT_MS` default raised from `30000` to `180000`; a `settings.json` holding exactly `30000` is rewritten automatically, but any other explicit value is kept as-is.

! The default OpenRouter model changes from `xiaomi/mimo-v2-flash:free` (retired) to `cohere/north-mini-code:free`; a stored `CLAUDE_MEM_OPENROUTER_MODEL` equal to the retired id is rewritten once when the base URL is blank or openrouter.ai.

! `CLAUDE_MEM_CONTEXT_MAIN_AGENT_ONLY` defaults to `true`, meaning subagent observations are no longer included in the SessionStart context window; set it to `false` to restore the previous behavior.

! The worker now returns 403 for requests whose `Host` is a DNS name other than `localhost`, `*.localhost`, `host.docker.internal`, `CLAUDE_MEM_WORKER_HOST`, the `CLAUDE_MEM_PUBLIC_URL` host, or an allowed origin; add your hostname to `CLAUDE_MEM_ALLOWED_ORIGINS` and restart the worker if you access the viewer via a custom host.

! `UserPromptSubmit` hook timeout drops from 60 seconds to 15 seconds in Claude Code.

! Antigravity users must re-run `npx claude-mem install --ide antigravity` because earlier installs wrote `~/.gemini/config/hooks.json` in a shape that `agy` ignores, so nothing was captured.

v13.29.0 NOTES ↗ LATEST STABLE

claude-mem v13.29.0 adds persistent work-state MCP tools, an openai-compatible provider with presets, Codex and Kimi Code support, and multi-key rotation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v13.29.0 https://github.com/thedotmack/claude-mem.git
# already have the repo? check out this version:
$ git checkout v13.29.0
```

└──▷ TRY IT

Switch the observer to a local Ollama instance using the new openai-compatible provider so no API key or external service is required.

```
$ CLAUDE_MEM_PROVIDER=openai-compatible
CLAUDE_MEM_OPENAI_COMPAT_PRESET=ollama
CLAUDE_MEM_OPENAI_COMPAT_BASE_URL=http://localhost:11434
CLAUDE_MEM_OPENAI_COMPAT_MODEL=llama3.2 npx claude-mem install
```

Install claude-mem to use your ChatGPT Codex subscription as the observer backend, with batching tuned for throughput.

```
$ npx claude-mem install --provider codex --model o4-mini
```

Adds `work_state_write` and `work_state_read` MCP tools to give agents a canonical to-do list that persists across sessions, stored in a new append-only `work_state_entries` table (schema v61), with fields capped at 2,000 characters as JSON.

Adds `CLAUDE_MEM_PROVIDER=openai-compatible` to run the observer against any OpenAI `/chat/completions` endpoint, configured with `CLAUDE_MEM_OPENAI_COMPAT_PRESET`, `CLAUDE_MEM_OPENAI_COMPAT_BASE_URL`, `CLAUDE_MEM_OPENAI_COMPAT_MODEL`, and `CLAUDE_MEM_OPENAI_COMPAT_API_KEY`.

Adds `CLAUDE_MEM_PROVIDER=codex` (or `npx claude-mem install --provider codex [--model <id>]`) to run the observer through the Codex CLI app-server on a ChatGPT

subscription, with `CLAUDE_MEM_CODEX_MODEL`, `CLAUDE_MEM_CODEX_REASONING_EFFORT` (default `low`), and `CLAUDE_MEM_CODEX_PATH` (default `codex`) controls.

Adds Codex throughput controls `CLAUDE_MEM_CODEX_MAX_CONCURRENT_AGENTS` (default `2`), `CLAUDE_MEM_CODEX_OBSERVATION_BATCH_SIZE` (default `8`), and `CLAUDE_MEM_CODEX_OBSERVATION_BATCH_MAX_CHARS` (default `32000`) settable in `settings.json`.

Adds `CLAUDE_MEM_QUOTA_FALLBACK_PROVIDER` (accepts `claude`, `gemini`, `openrouter`, or `openai-compatible`; empty/off by default) and `CLAUDE_MEM_QUOTA_FALLBACK_MODEL` to route observer work to a secondary provider during quota cooldowns.

+13 more

└──▷ BREAKING ON UPGRADE

- ! `CLAUDE_MEM_LLM_TIMEOUT_MS` default raised from `30000` to `180000`; a `settings.json` holding exactly `30000` is rewritten automatically, but any other explicit value is kept as-is.
- ! The default OpenRouter model changes from `xiaomi/mimo-v2-flash:free` (retired) to `cohere/north-mini-code:free`; a stored `CLAUDE_MEM_OPENROUTER_MODEL` equal to the retired id is rewritten once when the base URL is blank or openrouter.ai.
- ! `CLAUDE_MEM_CONTEXT_MAIN_AGENT_ONLY` defaults to `true`, meaning subagent observations are no longer included in the SessionStart context window; set it to `false` to restore the previous behavior.
- ! The worker now returns 403 for requests whose `Host` is a DNS name other than `localhost`, `*.localhost`, `host.docker.internal`, `CLAUDE_MEM_WORKER_HOST`, the `CLAUDE_MEM_PUBLIC_URL` host, or an allowed origin; add your hostname to `CLAUDE_MEM_ALLOWED_ORIGINS` and restart the worker if you access the viewer via a custom host.
- ! `UserPromptSubmit` hook timeout drops from 60 seconds to 15 seconds in Claude Code.
- ! Antigravity users must re-run `npx claude-mem install --ide antigravity` because earlier installs wrote `~/.gemini/config/hooks.json` in a shape that `agy` ignores, so nothing was captured.

WAS THIS USEFUL?  

Superset

Feedback ⓘ cli-v1.35.0 NOTES ↗ STABLE RANK

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

Superset CLI v1.35.0 adds Linear-style per-team task numbers, a live Linear tab, and mobile cloud workspace archiving.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch cli-v1.35.0 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout cli-v1.35.0
```

Introduces Linear-style per-team task numbers (e.g. SUPER-123), assigned automatically via a database trigger.

Adds a live Linear tab (feature-flagged) for real-time Linear integration within tasks.

Adds a `task_imports` table to the database to support task import workflows.

Enables archiving of cloud workspaces on mobile, with a Cloud scope matching the desktop sidebar.

NEW INTEGRATION Granola plugin

NEW INTEGRATION YNAB plugin

NEW INTEGRATION Stripe plugin

UPDATED INTEGRATION Google connector OAuth client

NEW SUPPORTED PLATFORM iPad native support

▾ 1 more release in this issue · 2026-10-02

> desktop-v1.34.0 NOTES ↗ LATEST STABLE
>
> Adds CLI subcommands `superset schema`, `completion`, and `config check`; `window.superset.storage` for page persistence; GPT-6.1 Sol agent support; Linear-style task numbers (SUPER-123); cloud workspace archiving/unarchiving; Automations All Runs screen and agent-created event triggers; automations failure badge with realtime nudges; Code tab in workspace PR pane; local resource journal for host debugging; universal links for shared pages on mobile.
>
> └──▷ GET THIS VERSION ▾

```
$ git clone --branch desktop-v1.34.0 https://github.com/superset-sh/superset.git
# already have the repo? check out this version:
$ git checkout desktop-v1.34.0
```

Adds `superset schema`, `completion`, and `config check` subcommands to the CLI for schema inspection, shell completion, and config validation.

Adds `window.superset.storage` API for persistent storage in pages.

Adds GPT-6.1 Sol agent support.

Adds Linear-style task numbers per team (e.g. SUPER-123) with per-team sequences assigned by a database trigger.

Adds a Code tab to the workspace PR pane for reviewing pull request diffs.

+9 more

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ desktop-v0.0.43 NOTES ↗ CODE ↗ RANK

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

> Cline desktop v0.0.43 introduces a Connectors spotlight dialog that showcases app integrations like Gmail, Slack, and GitHub to existing users.

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

01 **Connectors spotlight dialog** NEW 62

A one-time What's new spotlight dialog introduces Connectors, showing connector brand logos (Gmail, Slack, Google Calendar, Linear, Sentry, Notion, GitHub), a clickable Organize your day example prompt, and an Open Connectors button that navigates to the Connectors tab in Customize. Users can replay this dialog anytime via Settings → About → Show what's new.

— Names exact UI path and connectors but no mechanism beyond display desktop-v0.0.43

desktop-v0.0.43 NOTES ↗ CODE ↗ LATEST STABLE

Cline desktop v0.0.43 adds a one-time Connectors spotlight dialog introducing app integrations to existing users.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch desktop-v0.0.43
https://github.com/cline/cline.git
# already have the repo? check out this version:
$ git checkout desktop-v0.0.43
```

Adds a one-time What's new spotlight dialog for Connectors, showing connector brand logos (Gmail, Slack, Google Calendar, Linear, Sentry, Notion, GitHub), a clickable Organize your day example prompt, and an Open Connectors button that navigates to the Connectors tab in Customize.

Exposes a Settings → About → Show what's new option to replay the Connectors spotlight dialog at any time.

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.288 NOTES ↗

CODE ↗ RANK

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

> Claude Code v2.1.288 adds a configurable finding cap for `/code-review`, a new $.ui.selection() mods API, a built-in `gh api` for cloud sessions lacking the GitHub CLI, and a `CLAUDE_CODE_DISABLE_STRUCTURED_OUTPUTS` env var, alongside agent-view navigation shortcuts, draft recovery, MCP re-auth prompts, a `claude purge` rename, per-model autocompact windows, and a narrower background command time limit.

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

01 **Configurable finding cap for /code-review** NEW 90

Adds `--max-findings <n>|all` to `/code-review` to report more or fewer findings than the default limit; the setting persists until `--max-findings default` is passed.

Cap a code review to the 5 most important findings to keep CI output concise, then restore default reporting.

```
$ /code-review --max-findings 5
```

— Named flag with persistence behaviour and runnable example. v2.1.288

02 **claude purge command rename** DEPRECATED 85

Renames `claude project purge` to `claude purge`; the old subcommand still works and prints a notice.

— Exact old and new commands with compatibility behaviour. v2.1.288

03 **Background command time limit scoped to unattended sessions**

IMPROVED 80

Changes the background command time limit to apply only in unattended sessions (`-p`, Agent SDK, CI, cloud); terminal, desktop app and VS Code sessions now have no limit.

— Names all affected session types with clear before/after scope. v2.1.288

04 **Env var to disable structured outputs** NEW 75

Adds `CLAUDE_CODE_DISABLE_STRUCTURED_OUTPUTS` environment variable to disable structured outputs for gateways or proxies that reject them.

Disable structured outputs when Claude Code is routed through a gateway that rejects them, restoring session title and hook functionality.

```
$ CLAUDE_CODE_DISABLE_STRUCTURED_OUTPUTS=1 claude
```

— Named env var with a runnable command example. v2.1.288

05 **Selection capture in mods API** NEW 65

Adds $.ui.selection() to the mods API, which returns the text last selected in fullscreen mode and, when the selection lies within one transcript row, that row.

— Named API with scoped return behaviour, no usage example. v2.1.288

06 **Rebindable agents view navigation** NEW 65

Adds Ctrl+F to find a session by name and Alt+\u2191/\u2193 to jump between groups in the agents view; both shortcuts, and rename, can be rebound in `keybindings.json`.

— Named shortcuts and config file, no deeper mechanism given. v2.1.288

07 **Per-model autocompact window** IMPROVED 65

Changes `/autocompact` to save the auto-compact window per model, so each model retains its own setting when you switch.

— Named command and before/after behaviour, no numbers given. v2.1.288

08 **Draft recovery after Ctrl+C** NEW 60

Adds recovery for a prompt cleared with Ctrl+C: pressing Up on the empty prompt restores the draft, including pasted text and images.

— Clear trigger and restored content, no config surface. v2.1.288

THINNER COVERAGE BELOW

09 **Built-in gh api for cloud sessions** NEW 45

Adds a built-in `gh api` to cloud sessions whose container image has no GitHub CLI installed.

— States the surface but no command or example. v2.1.288

10 **Re-authenticate prompt for MCP OAuth scope** NEW 40

Adds a re-authenticate prompt when an MCP server asks for more OAuth scope during a tool call.

— Describes trigger only, no flag or config named. v2.1.288

11 **Screen reader permission mode announcement** NEW 40

Adds a screen reader mode announcement of the new permission mode when you approve a plan, including with Shift+Tab.

— Thin description of an accessibility behaviour. v2.1.288

v2.1.288 NOTES ↗ CODE ↗ LATEST STABLE

Claude Code v2.1.288 adds $.ui.selection() for mods, `--max-findings` for `/code-review`, built-in `gh api` for cloud sessions, and draft-recovery with Up arrow.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.1.288 https://github.com/anthropics/claude-code.git
# already have the repo? check out this version:
$ git checkout v2.1.288
```

└──▷ TRY IT

Cap a code review to the 5 most important findings to keep CI output concise, then restore default reporting.

```
$ /code-review --max-findings 5
```

Disable structured outputs when Claude Code is routed through a gateway that rejects them, restoring session title and hook functionality.

```
$ CLAUDE_CODE_DISABLE_STRUCTURED_OUTPUTS=1 claude
```

Adds $.ui.selection() to the mods API: returns the text last selected in fullscreen mode and, when the selection lies within one transcript row, that row.

Adds `--max-findings <n>|all` to `/code-review` to report more or fewer findings than the default limit; persists until `--max-findings default` is passed.

Adds a built-in `gh api` to cloud sessions whose image has no GitHub CLI installed.

Adds `CLAUDE_CODE_DISABLE_STRUCTURED_OUTPUTS` environment variable to disable structured outputs for gateways or proxies that reject them.

Adds Ctrl+F to find a session by name and Alt+↑/↓ to jump between groups in the agents view; both shortcuts, and rename, can be rebound in `keybindings.json`.

+6 more

WAS THIS USEFUL? 👍 👎

Block Goose

Feedback ⓘ v1.53.0 NOTES ↗ CODE ↗ RANK

Goose is an open-source AI agent with a desktop app and CLI for coding and other automated tasks.

> Goose v1.53.0 adds a CLI session rename command, expands model support with live discovery from models.dev, and rounds out the desktop UI with a conversation-quoting action and a resizable Picture-in-Picture window for MCP Apps, alongside running the GDK agent loop on wasm32 targets.

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

01 Session rename CLI command NEW

70

Adds a `session rename` subcommand to the CLI for renaming existing sessions, letting users keep their session list organized after a conversation changes direction.

Rename a session by ID to keep your session list organized after a conversation changes direction.

```
$ goose session rename <session-id> <new-name>
```

— Names exact subcommand and includes runnable example v1.53.0

02 Model discovery and new model support NEW

60

Adds support for Opus 4.5, GPT-6-sol, and GPT-6-luna models, and now fetches model metadata from models.dev live, falling back to a bundled copy when unavailable.

— Names models and data source but no usage steps v1.53.0

THINNER COVERAGE BELOW

03 Working directory shown in session remove listing IMPROVED

45

The `session remove` command now shows the working directory path when listing sessions available for removal.

— Names command but no example or UI detail v1.53.0

04 Resizable Picture-in-Picture window for MCP Apps NEW

40

Adds a resizable Picture-in-Picture window for MCP Apps in the desktop UI.

— Names the feature but no mechanism or usage detail v1.53.0

05 **Quote conversation text into composer** NEW 35

The desktop UI adds a quoting action that appends selected conversation text to the composer.

— Describes action but no UI path or trigger given v1.53.0

06 **GDK agent loop on wasm32** NEW 35

Runs the GDK agent loop on wasm32 targets.

— Names target platform but no further detail v1.53.0

v1.53.0 NOTES ↗ CODE ↗ LATEST STABLE

Goose v1.53.0 adds a session rename CLI command, Meta Muse Code provider, lean ACP-only binary, and model discovery for GDK.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.53.0 https://github.com/aaif-goose/goose.git
# already have the repo? check out this version:
$ git checkout v1.53.0
```

└──▷ TRY IT

Rename a session by ID to keep your session list organized after a conversation changes direction.

```
$ goose session rename <session-id> <new-name>
```

Adds `session rename` subcommand to the CLI for renaming existing sessions.

Shows working directory path when listing sessions for removal via the `session remove` command.

Adds support for Opus 5.5, GPT-6-sol, and GPT-6-luna models.

Fetches model metadata from models.dev live with a bundled fallback.

Adds a quoting action in the desktop UI to append selected conversation text to the composer.

+2 more

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ v0.9.74 NOTES ↗

RANK

Graphify converts code, docs, SQL schemas, configs, and PDFs into a queryable knowledge graph with explainable relationships.

> Graphify v0.9.74 broadens its code-graph extraction engine with support for Verilog header files and sharper call/member resolution across six more languages.

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

01 **Expanded language graph coverage across seven languages**

IMPROVED 70

Graphify now parses `.vh` Verilog header files via the existing `tree-sitter-verilog` grammar, capturing macros, parameters, and modules alongside `.v` / `.sv` files. It also extracts Erlang local `fun Name/Arity` references as `indirect_call` edges to the arity-matched local function; Kotlin class-literal references in annotation arguments (e.g. @ManyToOne(targetEntity = Customer::class)) as attribute references; Ruby paren-less self-sends as calls, with local-variable/parameter/block-parameter scope correctly suppressing fabrication; TypeScript `abstract` method signatures in abstract classes as callable method members and call targets; Scala deferred method declarations in traits and abstract classes (`def foo: Int` , no body) as methods; and C++ member-function declarations inside a class — including pure-virtual, `const` , operator, and destructor forms — as methods rather than fields.

— Names exact grammars, syntax forms, and edge types but needs no user action. v0.9.74

v0.9.74 NOTES ↗ LATEST STABLE

Graphify v0.9.74 adds Verilog header extraction and expands language graph coverage for Ruby, TypeScript, Scala, C++, Kotlin, and Erlang.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.9.74 https://github.com/Graphify-Labs/graphify.git
# already have the repo? check out this version:
$ git checkout v0.9.74
```

Supports Verilog header files (`.vh`) through the existing `tree-sitter-verilog` grammar, capturing macros, parameters, and modules declared in headers alongside `.v` / `.sv` files.

> Extracts Erlang local fun references (`fun Name/Arity`) as `indirect_call` edges to the arity-matched local function.
>
> Extracts Kotlin class-literal references in annotation arguments (e.g. @ManyToOne(targetEntity = Customer::class)) as attribute references to the referenced type.
>
> Extracts Ruby paren-less self-sends as calls, with local-variable/parameter/block-parameter scope correctly suppressing fabrication.
>
> Extracts TypeScript `abstract` method signatures in abstract classes as callable method members and call targets.
>
> +2 more

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.2 NOTES ↗ RANK

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

> Letta Code v0.34.2 adds opt-in automatic reply relay for channels, live streaming of workflow progress, and an expanded subagent name pool.

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

01 **Live streaming of workflow progress** IMPROVED 42

Streams workflow completion status, token usage, and phase progress in real time during agent workflow execution.

— Explains what streams but not the mechanism or UI v0.34.2

02 **Automatic reply relay for channels** NEW 33

An opt-in feature that automatically forwards replies in channels without manual intervention, enabling messages to be relayed automatically.

— Describes behavior but no config key or flag to enable it v0.34.2

03 **Expanded subagent name pool** IMPROVED 25

Triples the generated subagent name pool, expanding the variety of names available for multi-agent setups.

— Bare numeric change, no further mechanism given v0.34.2

v0.34.2 NOTES ↗ LATEST STABLE

Letta Code v0.34.2 adds opt-in automatic reply relay for channels and live streaming of workflow completion, tokens, and phase progress.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.34.2 https://github.com/letta-ai/letta-code.git
# already have the repo? check out this version:
$ git checkout v0.34.2
```

Adds opt-in automatic reply relay for channels, enabling messages to be automatically forwarded without manual intervention.

Streams workflow completion status, token usage, and phase progress in real time during agent workflow execution.

> Triples the generated subagent name pool, expanding variety for multi-agent setups.

WAS THIS USEFUL?  

Amazon Kiro

Feedback ⓘ

Models: Claude Sonnet 5.5 Now Available (2026-10-02) BLOG ↗ RANK

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

> Kiro's biggest additions this window are Claude Sonnet 5.5 across every surface and the launch of Workflows, a multi-agent graph-based pipeline system, alongside a long list of CLI 2.27 refinements spanning Steering context, saved prompts, diff views, and settings changes.

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

01 **Kiro Workflows: multi-agent graph pipelines** NEW 100

Kiro Workflows introduces multi-agent, graph-based task pipelines across IDE, CLI, and Web under a shared runtime, enabled via the `kiroAgent.workflows.enabled` setting. Recipes are authored in JSON or YAML and composed of sequences, `repeat` loops with stop conditions, and `parallel` branches running independent agent sessions concurrently; built-in recipes include `investigate` (single-agent read-only background investigation) and `publish-pr` (opens a pull request and follows it to merge, retrying failed CI and addressing review feedback). Workflow steps and the main session exchange messages bidirectionally so steps can ask decisions and keep running autonomously, support per-step model and thinking-effort overrides (e.g. `claude-opus-4.5` at Extra high, `gpt-5.6-sol` at Extra high) with session-default inheritance, and retry connection failures in place for up to 24 hours before pausing, resuming progress and completion reporting after a restart. Preferred recipes can be stored in cloud configuration on Kiro Web for reuse across projects and devices, and workflows can run in cloud sessions from CLI and IDE via the workflows configuration page. A new `Workflows: sub-agent tool` setting (on by default) under `/settings features` controls whether main-chat delegates directly to sub-agents or only through Workflows when Workflows are enabled.

Enable Workflows for a project in Kiro IDE so you can start delegating multi-agent tasks.

📍 Open Workspace Configuration for the project › Choose Workflows › Enable Workflows › Start a new chat session (restart Kiro if needed to apply to existing sessions).

Enable workflows via the backing config key directly, for scripted or headless environments.

```yaml
kiroAgent.workflows.enabled
```

Restrict main-chat delegation to Workflows only, preventing direct sub-agent calls from the top-level chat.

> 📍 In the Kiro CLI chat, go to `/settings` › features and turn off `Workflows: sub-agent tool`.

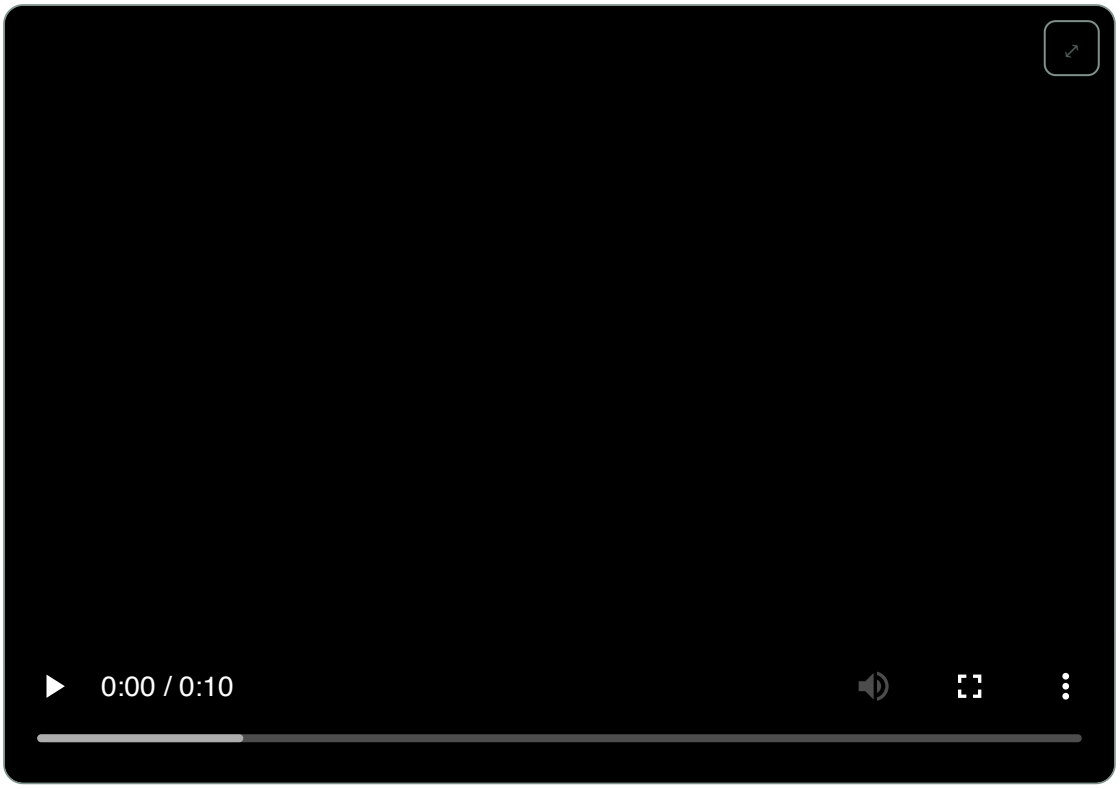
— Full mechanism, named config keys, enable steps, and runnable examples.

`CLI: Workflow Delegation, Live Steering…` `launch-20260930-93b04370`

---

02 **Claude Sonnet 5.5 available across Kiro** NEW 78

Claude Sonnet 5.5 is now available in Kiro IDE, CLI, Crew, and Web for Pro, Pro+, Pro Max, and Power customers, with a 1M token context window and up to 128K output tokens at no extra charge for long context. Adaptive Thinking is on by default with an `effort` setting across five levels (low to max) to steer reasoning depth and cost, and cyber safeguards/fallbacks apply because of the model's elevated cybersecurity capabilities, with higher-risk requests automatically redirectable to Sonnet 5. Run `kirocrew restart` in Crew, restart the IDE or CLI, or refresh Web to load the updated model list.



Reload the model list in Kiro Crew after Claude Sonnet 5.5 becomes available.

```
$ kirocrew restart
```

— Names context limits, effort levels, safeguards, and reload command.

`Models: Claude Sonnet 5.5 Now Available…` `launch-20261002-510ad9e3`

03 **Expanded diff view in Full-TUI write cards** IMPROVED

70

Full-TUI write cards now show changed lines with surrounding context (up to 40 diff rows by default), with Ctrl+O or auto-expand to reveal the complete live diff.

— Names default row limit and exact keybinding. CLI: Workflow Delegation, Live Steering…

04 **Live file/folder context in Steering documents** NEW

65

Steering documents now support `#[[folder:...]]` references for a one-level directory listing, alongside new one-line and line-range selectors for `#[[file:...]]` references.

Include a live folder listing in a Steering document so the model always sees the current top-level structure of a directory.

📍 In your Steering document, add a line such as: #[[folder:src/]] to include a one-level directory listing resolved at session load.

— Exact syntax named but limited explanation of resolution.

CLI: Workflow Delegation, Live Steering…

05 **`--effort` flag for non-interactive CLI runs** NEW

65

Adds `--effort` flag support for non-interactive runs to control reasoning effort, usable as in `kiro --effort high 'Refactor the auth module to use the new token interface'`.

Run a non-interactive Kiro CLI task with explicit reasoning effort to control token usage in CI pipelines.

```
$ kiro --effort high 'Refactor the auth module to use the new token interface'
```

— Exact flag with a runnable example command. CLI: Workflow Delegation, Live Steering…

06 **V3 settings changes: todo list removed, output style relocated**

65

BREAKING

The `chat.enableTodoList` config key has no effect in V3, and V3 no longer provides the `todo_list` tool. Output style has moved from the top-level `/settings` menu to `/settings display`.

— Names removed config key, tool, and new settings path.

CLI: Workflow Delegation, Live Steering…

07 **Saved prompts as slash commands in V3** NEW 60

Workspace, global, and MCP saved prompts are now exposed as slash commands in V3, with argument hint completion and file template or MCP content resolution before the expanded text is sent to the model.

– Mechanism described but no concrete command example.

CLI: Workflow Delegation, Live Steering…

THINNER COVERAGE BELOW

08 **`kiro-cli agent list` for V3 agents** NEW 45

Adds `kiro-cli agent list` support for V3-format agents.

– Named command but no further mechanism given. CLI: Workflow Delegation, Live Steering…

09 **`--resume` picks newest matching V3 session** IMPROVED 45

V3 `--resume` now selects the newest session matching the current local or cloud launch placement.

– States behaviour change with no usage example. CLI: Workflow Delegation, Live Steering…

10 **Registry mode local MCP entry override** IMPROVED 45

In registry mode, a matching local MCP entry now uses the registry server's command or URL while retaining supported local overrides.

– Describes resolution behaviour with no example.

CLI: Workflow Delegation, Live Steering…

11 **Faster V3 startup with large permissions files** IMPROVED 30

V3 startup can be up to approximately 0.5 seconds faster with large `permissions.yaml` files.

– Gives a number but no actionable step. CLI: Workflow Delegation, Live Steering…

12 **Owner-only permissions on Kiro home state** IMPROVED 30

Kiro home state and previous session data now use owner-only access permissions.

– Security hardening noted without further detail.

CLI: Workflow Delegation, Live Steering…

└──▷ BREAKING ON UPGRADE

! The `chat.enableTodoList` config key has no effect in V3, and V3 no longer provides the `todo_list` tool.

! Output style has moved from the top-level `/settings` menu to `/settings display`.

Models: Claude Sonnet 5.5 Now Available (2026-10-02) BLOG ↗ LATEST

STABLE

Claude Sonnet 5.5 arrives in Kiro IDE, CLI, Crew, and Web with 1M-token context and adaptive thinking on by default.

└──▷ TRY IT

Reload the model list in Kiro Crew after Claude Sonnet 5.5 becomes available.

```
$ kirocrew restart
```

Run `kirocrew restart` in Crew, restart the IDE or CLI, or refresh Web to load the updated model list.

announced · Oct 2, 2026 BLOG ↗

Kiro: Adds Claude Sonnet 5.5 as an available model across Kiro IDE, CLI, Crew, and Web for Kiro Pro, Pro+, Pro Max, and Power customers.

Supports an `effort` setting with five levels (low to max) via Adaptive Thinking to steer reasoning depth and cost on Sonnet 5.5.

Provides a 1M token context window and up to 128K output tokens on Sonnet 5.5 with no extra charge for long context.

Applies cyber safeguards and fallbacks to Sonnet 5.5 due to its elevated cybersecurity capabilities, with higher-risk requests automatically redirectable to Sonnet 5.

CLI: Workflow Delegation, Live Steering Context, and Saved Prompt Commands: Choose H

BLOG ↗ STABLE

Kiro CLI 2.27 adds Workflow delegation control, live Steering file/folder context, and saved prompts as slash commands.

└──▷ HOW TO FIND IT

Restrict main-chat delegation to Workflows only, preventing direct sub-agent calls from the top-level chat.

📍 In the Kiro CLI chat, go to `/settings` › features and turn off Workflows: sub-agent tool .

Include a live folder listing in a Steering document so the model always sees the current top-level structure of a directory.

📍 In your Steering document, add a line such as: #[[folder:src/]] to include a one-

level directory listing resolved at session load.

Run a non-interactive Kiro CLI task with explicit reasoning effort to control token usage in CI pipelines.

```
$ kiro --effort high 'Refactor the auth module to use the new token interface'
```

Adds `Workflows: sub-agent tool` setting (on by default) under `/settings features` to control whether main-chat delegates directly to sub-agents or only through Workflows when Workflows are enabled.

Adds `#[[folder:...]]` references in Steering documents for a one-level directory listing, alongside new one-line and line-range selectors for `#[[file:...]]` references.

Exposes workspace, global, and MCP saved prompts as slash commands in V3, with argument hint completion and file template or MCP content resolution before the expanded text is sent to the model.

Adds `--effort` flag support for non-interactive runs to control reasoning effort.

Adds `kiro-cli agent list` support for V3-format agents.

+6 more

└──▷ BREAKING ON UPGRADE

- ! The `chat.enableTodoList` config key has no effect in V3, and V3 no longer provides the `todo_list` tool.
- ! Output style has moved from the top-level `/settings` menu to `/settings display`.

announced · Sep 30, 2026 BLOG ↗

Kiro launches Workflows: multi-agent, graph-based task pipelines with sequences, loops, and parallel branches across IDE, CLI, and Web.

└──▷ HOW TO FIND IT

Enable Workflows for a project in Kiro IDE so you can start delegating multi-agent tasks.

📍 Open Workspace Configuration for the project › Choose Workflows › Enable Workflows › Start a new chat session (restart Kiro if needed to apply to existing sessions).

Enable workflows via the backing config key directly, for scripted or headless environments.

```yaml
kiroAgent.workflows.enabled
```

Enables Kiro Workflows via the `kiroAgent.workflows.enabled` setting, available in Kiro IDE, CLI, and Web under a single shared runtime.

Supports workflow recipes authored in JSON or YAML, with steps composed of sequences, `repeat` loops (with stop conditions), and `parallel` branches running independent agent sessions concurrently.

Adds built-in bundled recipes including `investigate` (single-agent read-only background investigation) and `publish-pr` (opens a pull request and follows it through to merge, retrying failed CI and addressing review feedback).

Lets workflow steps and the main session exchange messages bidirectionally — steps report progress, ask decisions, and the main session relays answers back to keep steps running autonomously.

Supports per-step model and thinking-effort overrides (e.g. `claude-opus-5.5` at `Extra high`, `gpt-5.6-sol` at `Extra high`) alongside session-default inheritance for omitted settings.

+1 more

WAS THIS USEFUL? 👍 👎

Daytona

Feedback ⓘ announced · Sep 30, 2026 BLOG ↗ RANK

Daytona provisions isolated development sandboxes for AI coding agents through an API and SDK.

New `@daytona/convex` component exposes Daytona sandboxes as reactive Convex tables with live queries, `runBackground`, and `daytona.listExecutions` for agent workloads.

NEW INTEGRATION Adds `@daytona/convex` npm package — a Convex component that integrates Daytona sandboxes into Convex backends, making sandbox and command-execution state available as reactive Convex tables with live queries, durable execution records, and multi-client consistency.

Exposes daytona.createSandbox(ctx, { autoStopInterval }) and daytona.runCode(ctx, { sandboxId, code, language }) APIs inside Convex actions for creating isolated sandboxes and running code (e.g. `language: 'python'`) without holding an action open past Convex's limits.

Provides `runBackground` for long-running command execution — returns an execution ID immediately and streams logs into a Convex row via a chain of scheduled functions, removing the 10-minute action ceiling.

Surfaces `daytona.listExecutions` as a subscribable live query so frontends can render real-time execution history and log streaming without polling or websocket plumbing.

Requires `DAYTONA_API_KEY` environment variable set via `npx convex env set DAYTONA_API_KEY <key>` to authenticate the component with Daytona.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

Feynman

Feedback ⓘ v0.5.24 NOTES ↗ RANK

Feynman is a personal AI research assistant that searches papers and the web, builds literature reviews, and produces cited research reports.

> Feynman's latest releases upgrade the agent runtime to Pi 1.0.0 with background subagents and hardened MCP OAuth, switch the terminal UI to fullscreen by default (a breaking change), and add CLI tooling for package removal and headless authentication.

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

01 Fullscreen terminal UI by default

BREAKING

88

The terminal UI is now fullscreen by default with its own independent scrollback buffer. Users relying on the terminal's native scrollback can revert with `feynman --tui-mode regular` or by persisting `{"tuiMode": "regular"}` in `~/.feynman/agent/settings.json`.

Run Feynman in the terminal's normal scrollback mode instead of the new fullscreen TUI — useful in environments where fullscreen breaks logging or session capture.

```
$ feynman --tui-mode regular
```

Persist the non-fullscreen TUI preference so every session starts in regular mode without passing the flag each time.

```json
{"tuiMode": "regular"}
```

Run the TUI in the terminal's normal scrollback mode instead of fullscreen, and persist the choice in config.

```
$ feynman --tui-mode regular
```

Preserve your terminal emulator's native scrollback when running Feynman interactively.

```
$ feynman --tui-mode regular
```

Permanently opt back into normal scrollback mode so every session uses the terminal's own buffer.

```json
{
  "tuiMode": "regular"
}
```

— Exact flag and config key with runnable examples, plus before/after. v0.5.24

THINNER COVERAGE BELOW

02 **Pi 1.0.0 runtime upgrade** IMPROVED 50

Upgrades to Pi 1.0.0, bundling `pi-subagents 0.75.0` and `pi-btw 0.7.1`, which enables background subagents to start, hardens MCP OAuth, and lowers codemode token use.

— Names versions and components but gives no usage mechanism. v0.5.24

03 **.docx reading without LibreOffice** NEW 50

Reads `.docx` paragraph text directly when LibreOffice is missing, so Word drafts can be reviewed without installing LibreOffice.

— Clear behavior change but no command or config to act on. product docs

04 **Copy-code Anthropic login for headless machines** NEW 40

Ships a copy-code Anthropic login flow for headless machines where browser-based authentication is unavailable.

— Names the flow but no steps or command shown. v0.5.24

└──▷ BREAKING ON UPGRADE

! The terminal UI is now fullscreen by default; users relying on the terminal's native scrollback must add `--tui-mode regular` or set `"tuiMode": "regular"` in `~/.feynman/agent/settings.json`.

v0.5.24 NOTES ↗ LATEST STABLE

Feynman v0.5.24 upgrades to Pi 1.0.0 with fullscreen TUI by default and new headless login support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.24 https://github.com/Companion-Inc/feynman.git
# already have the repo? check out this version:
$ git checkout v0.5.24
```

└──▷ **TRY IT**

Preserve your terminal emulator's native scrollback when running Feynman interactively.

```
$ feynman --tui-mode regular
```

Permanently opt back into normal scrollback mode so every session uses the terminal's own buffer.

~/.feynman/agent/settings.json

```
{
  "tuiMode": "regular"
}
```

Adds `--tui-mode regular` flag and `tuiMode` key in `~/.feynman/agent/settings.json` to revert the terminal UI to normal (non-fullscreen) scrollback mode.

Upgrades to Pi 1.0.0 runtime with pi-subagents 0.75.0, enabling background subagents to start on Pi 1.0.

Ships a copy-code Anthropic login flow for headless machines where browser-based auth is unavailable.

Makes the terminal UI fullscreen by default with its own independent scrollback buffer.

└──▷ **BREAKING ON UPGRADE**

! The terminal UI is now fullscreen by default; users relying on the terminal's native scrollback must add `--tui-mode regular` or set `"tuiMode": "regular"` in `~/.feynman/agent/settings.json`.

WAS THIS USEFUL? 👍 👎

OpenClaw

Feedback ⓘ v2026.9.8 NOTES ↗ CODE ↗ RANK

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

> OpenClaw v2026.9.8 adds a new voice provider option and improves visibility into plugin credentials.

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

01 **Gemini 3.8 Flash TTS voice support** NEW 40

Adds support for speaking with Gemini 3.8 Flash TTS as a voice provider.

— Names the provider but no setup or config details given v2026.9.8

02 **Plugin credential status view** NEW 40

Shows installed accounts and credential status for plugins in the UI.

— Describes UI location but no further mechanism detail v2026.9.8

v2026.9.8 NOTES ↗ CODE ↗ LATEST STABLE

OpenClaw v2026.9.8 adds Gemini 3.8 Flash TTS voice support and a plugin credential status view.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2026.9.8
https://github.com/openclaw/openclaw.git
# already have the repo? check out this version:
$ git checkout v2026.9.8
```

Adds support for speaking with Gemini 3.8 Flash TTS as a voice provider.

Shows installed accounts and credential status for plugins in the UI.

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.54.0 NOTES ↗ RANK

PydanticAI is a framework that builds type-safe Python agents with dependency injection, model integrations, tools, and structured outputs.

> PydanticAI v2.54.0 adds a public async HTTP client helper with OpenAI SDK pool limits, expands web search options across ExaSearch, YouSearch and OpenAI GPT-Live, adds session reconnect for OpenAILiveModel, and ships several clai2 CLI and reliability improvements alongside a breaking change to SubAgents folder loading.

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

01 Web search capability expansions across providers NEW

85

Adds a `native` option to `ExaSearch` and `YouSearch` to prefer the model's own built-in web search instead of the hosted API, introduces web_search_tool() to wire a WebSearch(local=...) backend directly as a tool, and extends `WebSearchTool` support to OpenAI GPT-Live through its delegated backend.

Prefer a model's native web search engine in ExaSearch instead of the Exa API, so the agent routes web lookups through the model's built-in search.

python

```python
from pydantic_ai.capabilities import ExaSearch

search = ExaSearch(native=True)
agent = Agent('openai:gpt-5.6-sol', capabilities=[search])
```

Use web_search_tool() to wire a local WebSearch backend directly as a tool, giving the agent on-demand web access without a hosted search integration.

python

```python
from pydantic_ai.capabilities import WebSearch
from pydantic_ai.capabilities.web_search import web_search_tool

tool = web_search_tool(WebSearch(local=True))
agent = Agent('openai:gpt-5.6-sol', tools=[tool])
```

— Names every surface and includes runnable code for two of the three v2.54.0

02 Public async HTTP client with OpenAI pool limits NEW

80

Exposes create_async_httpx2_client() as a public API for creating a shared async HTTP client, and HTTP clients created by Pydantic AI now default to the OpenAI SDK's connection pool limits.

Obtain a shared async HTTP client pre-configured with OpenAI SDK connection-pool limits for reuse across multiple model calls.

```python
from pydantic_ai.http_client import create_async_httpx2_client

client = create_async_httpx2_client()
agent = Agent('openai:gpt-5.6-sol', http_client=client)
```

— Named function with runnable example, limited explanation of pool defaults v2.54.0

03 **SubAgents agent_folders default changed** BREAKING 80

The `SubAgents` `agent_folders` default-change warning is dropped: since v2.52.0 the `agents` folder is no longer loaded by default — pass `agent_folders='agents'` explicitly to restore the old behaviour.

— States exact before/after and the config value to restore old behaviour v2.54.0

04 **clai2 CLI theming, update and model management** IMPROVED 75

clai2 file diffs are now tinted from the selected `/theme` palette, `/update` always installs the newest build and restarts into it while resuming the session, a double-`Esc` press opens a conversation rewind menu, and the `/model` menu now allows deleting unused models.

— Names every clai2 surface affected; UI paths given but no runnable commands v2.54.0

THINNER COVERAGE BELOW

05 **OpenAILiveModel session reconnect** IMPROVED 55

Dropped `OpenAILiveModel` sessions are now automatically reconnected by forking a stored session or replaying the conversation history.

— Explains mechanism but no example or config surface v2.54.0

06 **OpenAI realtime transcription model defaults** IMPROVED 50

OpenAI realtime now uses `gpt-live-transcribe` as the default model when `input_transcription_model='auto'`, and lists `gpt-transcribe` as an additional available transcription model.

— Names defaults and option but no usage example v2.54.0

07 **Replay-safe Logfire plugin setup** NEW 50

Adds a `replay_safe` flag to `LogfirePlugin` and makes custom `setup_logfire` callbacks replay-safe.

— Names the flag but no example of its use v2.54.0

08 **Dedicated CodeMode schema warning category** IMPROVED 45

The `CodeMode` missing-return-schema warning now raises under its own `CodeModeReturnSchemaWarning` category instead of a generic warning class.

— Names the class but no usage detail v2.54.0

09 **Cancel-and-resume streaming example** NEW 35

Adds a `cancel_and_resume` example demonstrating how to cancel and resume a streaming run.

— Points to an example but no detail on mechanism v2.54.0

10 **RepoContext inventory disabled by default** IMPROVED 35

Disables `RepoContext` inventory by default.

— Bare default change with no rationale or mechanism given v2.54.0

└──▷ BREAKING ON UPGRADE

! The `SubAgents` `agent_folders` default-change warning is dropped: since v2.52.0 the `agents` folder is no longer loaded by default — pass `agent_folders='agents'` explicitly to restore the old behaviour.

v2.54.0 NOTES ↗ LATEST STABLE

PydanticAI v2.54.0 adds session reconnect for OpenAILiveModel, native web search on GPT-Live, web_search_tool(), and a public create_async_httpx2_client().

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.54.0 https://github.com/pydantic/pydantic-ai.git
# already have the repo? check out this version:
$ git checkout v2.54.0
```

└──▷ USE IT

Prefer a model's native web search engine in ExaSearch instead of the Exa API, so the agent routes web lookups through the model's built-in search.

```python
from pydantic_ai.capabilities import ExaSearch

search = ExaSearch(native=True)
agent = Agent('openai:gpt-5.6-sol', capabilities=[search])
```

Use web_search_tool() to wire a local WebSearch backend directly as a tool, giving the agent on-demand web access without a hosted search integration.

```python
from pydantic_ai.capabilities import WebSearch
from pydantic_ai.capabilities.web_search import web_search_tool

tool = web_search_tool(WebSearch(local=True))
agent = Agent('openai:gpt-5.6-sol', tools=[tool])
```

Obtain a shared async HTTP client pre-configured with OpenAI SDK connection-pool limits for reuse across multiple model calls.

```python
from pydantic_ai.http_client import create_async_httpx2_client

client = create_async_httpx2_client()
agent = Agent('openai:gpt-5.6-sol', http_client=client)
```

Exposes create_async_httpx2_client() as a public API and defaults Pydantic AI-created HTTP clients to the OpenAI SDK's connection pool limits.

Adds `native` option to `ExaSearch` and `YouSearch` to prefer the model's native web search, and introduces web_search_tool() for WebSearch(local=...).

Reconnects dropped `OpenAILiveModel` sessions by forking a stored session or replaying the history.

Supports `WebSearchTool` on OpenAI GPT-Live through its delegated backend.

Uses `gpt-live-transcribe` as the default `input_transcription_model='auto'` for OpenAI realtime and lists `gpt-transcribe` as an available model.

+8 more

└──▷ BREAKING ON UPGRADE

! The `SubAgents` `agent_folders` default-change warning is dropped: since v2.52.0 the `agents` folder is no longer loaded by default — pass `agent_folders='agents'` explicitly to restore the old behaviour.

WAS THIS USEFUL? 👍 👎

CopilotKit OpenBot

Feedback ⓘ v0.1.0 NOTES ↗ RANK

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

> OpenBot's v0.1.0 release introduces enforced network egress policy for Bot computers and a round of Helm chart configuration for enterprise integrations, alongside new Automatic Learning skill sharing, a Parallel Search connector, and cross-Bot delegation for remote coworkers.

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

01 Egress policy enforcement for Bot computers BREAKING

85

A Bot's computer now blocks all network egress until the API server pushes its policy; deployments running a computer without an API server must set `EGRESS_POLICY_REQUIRED=0` to restore the previous behaviour. An egress rule with a malformed CIDR range (e.g. `10.0.0.5/`) is now refused, and any saved network policy containing one causes all Bots under it to fall back to an empty allowlist — reaching nothing — until corrected under Admin → Enterprise.

Run a Bot computer in an isolated environment without an API server, bypassing the new egress policy requirement.

```
$ EGRESS_POLICY_REQUIRED=0 ./start.sh
```

— Mechanism and before/after given plus exact bypass flag v0.1.0

02 Helm chart values for integrations and telemetry NEW

70

Adds `config.opentag`, `config.sms`, `config.push`, `config.deliveryPublicUrl`, `config.scim`, `config.inboundEmail`, and `config.otel` Helm chart values for Slack, Teams, SMS, push, SCIM, inbound email, and OpenTelemetry, with secrets (OpenTag secret, Twilio auth token, Expo access token, SCIM bearer tokens, OpenTelemetry headers) supplied under `secrets` or via an existing Secret/store by key.

— All config keys named but no worked example v0.1.0

03 Clearer error responses on memory and approvals APIs IMPROVED

70

`POST /api/memory/sources/:id/sync` now returns 400 with the policy refusal message when a connected app's policy refuses the memory source read, replacing the

previous opaque 503. `PATCH /api/approvals/preferences` and `POST /api/approvals/rules` now return 400 'Supply a valid request.' on non-JSON bodies instead of a raw 500.

— Endpoints and status codes named, no sample request v0.1.0

04 **Self-hosting help banner suppression** NEW 60

Adds `OPENBOT_SELF_HOST_BANNER=false` environment variable to suppress the self-hosting help bar for all users on a deployment.

Hide the self-hosting help bar for all users on a self-hosted deployment not on a paid Intelligence plan.

```
$ OPENBOT_SELF_HOST_BANNER=false ./start.sh
```

— Exact env var and runnable command provided v0.1.0

THINNER COVERAGE BELOW

05 **Package and skill slug validation at load time** IMPROVED 55

Validates `skills.yaml` slugs against the app's 2–40 character lowercase alphanumeric-with-hyphens rule at load time, refusing packages with out-of-spec slugs by name. Also validates tenant packages for duplicate skill names, agent IDs, channel IDs, and skill slugs at load time, refusing each with a sentence naming the file and the repeated ID.

— Exact validation rule given, no command to trigger or fix v0.1.0

06 **Automatic Learning skill sharing between bots** NEW 50

Adds Automatic Learning support, configuring which bots contribute conversations to a Learning container and receive its published skills. It is enabled by default on upgrade unless an administrator previously saved it off.

— Explains behaviour and default-on change but no config path v0.1.0

07 **Remote Bot-to-Bot delegation** IMPROVED 50

Remote Bots (coworkers at their own endpoint) can now hand work to other Bots via the same signed callback, grant check, caps, and audit rows as built-in Bots, including to other remote Bots.

— Mechanism named but no API surface given v0.1.0

08 **LF line endings on Windows clones** BREAKING 50

On an existing Windows clone, text files are now checked out with LF only after running `git rm -r --cached . && git reset --hard` on a clean tree.

— Exact remediation command given, narrow scope v0.1.0

09 **Parallel Search connector for web research** NEW 30

Adds a Parallel Search connector enabling public-web search and extraction for parallel research workflows.

— Named integration but no usage detail product docs

└──▷ BREAKING ON UPGRADE

! A Bot's computer blocks all network egress until the API server pushes its policy. Deployments running a computer without an API server must set `EGRESS_POLICY_REQUIRED=0` to restore the previous behaviour.

! An egress rule with a malformed CIDR range (e.g. `10.0.0.5/`) is now refused; any saved network policy containing one causes all Bots under it to fall back to an empty allowlist — reaching nothing — until the rule is corrected under Admin → Enterprise.

! Automatic Learning is enabled by default on upgrade unless an administrator previously saved it off.

! On an existing Windows clone, text files are checked out with LF only after running `git rm -r --cached . && git reset --hard` on a clean tree.

v0.1.0 NOTES ↗ LATEST STABLE

OpenBot v0.1.0 adds Helm chart values for Slack, Teams, SMS, push, SCIM, email, and OpenTelemetry, plus new egress policy enforcement.

└──▷ GET THIS VERSION ▾

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

└──▷ TRY IT

Run a Bot computer in an isolated environment without an API server, bypassing the new egress policy requirement.

```
$ EGRESS_POLICY_REQUIRED=0 ./start.sh
```

Hide the self-hosting help bar for all users on a self-hosted deployment not on a paid Intelligence plan.

```
$ OPENBOT_SELF_HOST_BANNER=false ./start.sh
```

Adds `config.opentag`, `config.sms`, `config.push`, `config.deliveryPublicUrl`, `config.scim`, `config.inboundEmail`, and `config.otel` Helm chart values for Slack, Teams, SMS, push, SCIM, inbound email, and OpenTelemetry — with secrets (OpenTag secret, Twilio auth token, Expo access token, SCIM bearer tokens, OpenTelemetry headers) under `secrets` or an existing Secret/store by key.

Adds `OPENBOT_SELF_HOST_BANNER=false` environment variable to suppress the self-hosting help bar for all users on a deployment.

Adds `POST /api/memory/sources/:id/sync` returning 400 with the policy refusal message when a connected app's policy refuses the memory source read, replacing the previous opaque 503.

Adds `PATCH /api/approvals/preferences` and `POST /api/approvals/rules` returning 400 'Supply a valid request.' on non-JSON bodies instead of a raw 500.

Remote Bots (coworkers at their own endpoint) can now hand work to other Bots via the same signed callback, grant check, caps, and audit rows as built-in Bots, including to other remote Bots.

+2 more

└──▷ BREAKING ON UPGRADE

- ! A Bot's computer blocks all network egress until the API server pushes its policy. Deployments running a computer without an API server must set `EGRESS_POLICY_REQUIRED=0` to restore the previous behaviour.
- ! An egress rule with a malformed CIDR range (e.g. `10.0.0.5/`) is now refused; any saved network policy containing one causes all Bots under it to fall back to an empty allowlist — reaching nothing — until the rule is corrected under Admin → Enterprise.
- ! Automatic Learning is enabled by default on upgrade unless an administrator previously saved it off.
- ! On an existing Windows clone, text files are checked out with LF only after running `git rm -r --cached . && git reset --hard` on a clean tree.

WAS THIS USEFUL? 👍 👎

CopilotKit

Feedback ⓘ v1.77.0 NOTES ↗ RANK

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

> CopilotKit v1.77.0 adds a persistent hide option for the web Inspector, makes the intelligence indicator's auto-mount behavior configurable, captures raw browser activity as AG-UI events for the learning pipeline, and consolidates Vue/Angular MCP-apps hosts onto a shared package.

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

01 **Configurable intelligence indicator auto-mount** IMPROVED 45

Makes the intelligence indicator auto-mount configurable in `react-core`, so developers can control whether it mounts automatically.

— Names the package but not the exact config option v1.77.0

02 **Persistent hide control for web Inspector** NEW 40

Adds an option to hide the web Inspector indefinitely, giving users a persistent dismiss control.

— Names the control but not the exact API or flag v1.77.0

03 **Raw browser activity as AG-UI events** NEW 40

Captures raw browser activity as AG-UI events in the learning subsystem, feeding real usage into the automatic-learning pipeline.

— Describes mechanism but no concrete API or event format v1.77.0

04 **Unified MCP-apps hosts for Vue and Angular** IMPROVED 40

Consolidates the Vue and Angular MCP-apps hosts onto the shared package, unifying framework support under one integration path.

— Names frameworks and consolidation but not package name v1.77.0

v1.77.0 NOTES ↗ LATEST STABLE

> CopilotKit v1.77.0 adds configurable intelligence-indicator mounting, raw browser activity capture as AG-UI events, and a persistent web-inspector hide option.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.77.0
https://github.com/CopilotKit/CopilotKit.git
# already have the repo? check out this version:
$ git checkout v1.77.0
```

Adds an option to hide the web Inspector indefinitely, giving users a persistent dismiss control.

Makes the intelligence indicator auto-mount configurable in `react-core`, so developers can control whether it mounts automatically.

Captures raw browser activity as AG-UI events in the learning subsystem, feeding real usage into the automatic-learning pipeline.

Consolidates the Vue and Angular MCP-apps hosts onto the shared package, unifying framework support under one integration path.

WAS THIS USEFUL? 👍 👎

Vercel AI SDK

Feedback ⓘ ai@7.0.127 NOTES ↗ RANK

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

> Vercel AI SDK 7.0.127 enhances tool search for agent tool selection with a custom search callback and a result cap.

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

01 **Custom callback and result cap for tool search** NEW

65

Tool search now supports a search() callback enabling custom selection and ranking of eligible deferred tools, plus a configurable `maxResults` option to cap the number of tools returned.

— Names both the callback and config option but no usage example. ai@7.0.127

ai@7.0.127 NOTES ↗ LATEST STABLE

Vercel AI SDK 7.0.127 adds a search() callback and `maxResults` cap to tool search for agent tool selection.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch ai@7.0.127 https://github.com/vercel/ai.git
# already have the repo? check out this version:
$ git checkout ai@7.0.127
```

Adds search() callback to tool search, enabling custom selection and ranking of eligible deferred tools.

Adds configurable `maxResults` to tool search to cap the number of tools returned.

WAS THIS USEFUL? 👍 👎

LlamaIndex

Feedback ⓘ announced · Oct 3, 2026 BLOG ↗ RANK

LlamaIndex is an open-source framework that connects language models to data sources.

> LlamaIndex shipped Extract v2.5, adding a new extraction engine, confidence scores and metadata retrieval, bounding-box Advanced Citations now available on the Agentic tier, native spreadsheet extraction, a terminal CLI, and an MCP server for connecting extraction agents to tools like Claude Code.

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

01 Advanced Citations extended to Agentic tier IMPROVED

83

The `"cite_sources": True` configuration key enables Advanced Citations, a bounding-box source-grounding feature, which is now available on the Agentic tier in addition to Agentic Plus. Grounding scores rose from 46.8 to 80.6 for Agentic and from 46.4 to 82.2 for Agentic Plus.

Run an extraction job with v2.5, citations, and confidence scores to enable human-in-the-loop review of low-confidence fields against source bounding boxes.

python

```
job = client.extract.run(
    file_input=file.id,
    configuration={
        "tier": "agentic",
        "version": "2.5",
        "extraction_target": "per_doc",
        "data_schema": Invoice.model_json_schema(),
        "cite_sources": True,
        "confidence_scores": True,
    },
)
result = client.extract.get(job.id, expand=["extract_metadata"])
print(result.extract_result)
print(result.extract_metadata.field_metadata.document_metadata)
```

— Clear mechanism, before/after scores, and named config key. launch-20261003-49d60145

02 v2.5 extraction engine and metadata options NEW

77

Adds the `"version": "2.5"` configuration key to client.extract.run() to select the new extraction engine, adds `"confidence_scores": True` to return per-field confidence scores alongside extracted values, and adds the `expand=["extract_metadata"]`

parameter to client.extract.get() to retrieve per-field metadata including grounding and confidence information.

Run an extraction job with v2.5, citations, and confidence scores to enable human-in-the-loop review of low-confidence fields against source bounding boxes.

```python
job = client.extract.run(
    file_input=file.id,
    configuration={
        "tier": "agentic",
        "version": "2.5",
        "extraction_target": "per_doc",
        "data_schema": Invoice.model_json_schema(),
        "cite_sources": True,
        "confidence_scores": True,
    },
)
result = client.extract.get(job.id, expand=["extract_metadata"])
print(result.extract_result)
print(result.extract_metadata.field_metadata.document_metadata)
```

— Names exact config keys and includes a runnable code example. launch-20261003-49d60145

03 **Terminal CLI for extraction (`llp`)** NEW 73

A new CLI tool, `llp`, installable via `go install github.com/run-llama/llama-parse-cli/cmd/llp@latest`, lets users extract documents from the terminal using a LlamaCloud API key.

Install the new terminal CLI to extract documents without writing Python, using your existing LlamaCloud API key.

```
$ go install github.com/run-llama/llama-parse-cli/cmd/llp@latest
export LLAMA_CLOUD_API_KEY=<your-key>
```

— Exact install command included, fully runnable. launch-20261003-49d60145

04 **MCP server for extraction agents** NEW 65

An MCP server at `https://mcp.llamaindex.ai/mcp` connects extraction agents to MCP clients such as Claude Code and Codex.

Connect the LlamaIndex MCP extraction server to Claude Code so an AI coding agent can extract structured data from documents on demand.

```
$ claude mcp add --transport http llamaparse
https://mcp.llamaindex.ai/mcp
```

— Exact endpoint and connect command provided. launch-20261003-49d60145

THINNER COVERAGE BELOW

05 **Native spreadsheet extraction mode** NEW 50

A new `spreadsheet mode` lets extraction agents work directly with workbook cells rather than a flattened document representation; it is enabled via the extraction configuration.

— Explains mechanism but no exact config key or example shown. launch-20261003-49d60145

announced · Oct 3, 2026 BLOG ↗

LlamaIndex Extract v2.5 adds native spreadsheet extraction, Advanced Citations for Agentic tier, and `version: '2.5'` config in the Python SDK.

└──▷ **USE IT**

Run an extraction job with v2.5, citations, and confidence scores to enable human-in-the-loop review of low-confidence fields against source bounding boxes.

python

```python
job = client.extract.run(
    file_input=file.id,
    configuration={
        "tier": "agentic",
        "version": "2.5",
        "extraction_target": "per_doc",
        "data_schema": Invoice.model_json_schema(),
        "cite_sources": True,
        "confidence_scores": True,
    },
)
result = client.extract.get(job.id, expand=["extract_metadata"])
print(result.extract_result)
print(result.extract_metadata.field_metadata.document_metadata)
```

Install the new terminal CLI to extract documents without writing Python, using your existing LlamaCloud API key.

```
$ go install github.com/run-llama/llama-parse-cli/cmd/llp@latest
export LLAMA_CLOUD_API_KEY=<your-key>
```

Connect the LlamaIndex MCP extraction server to Claude Code so an AI coding agent can extract structured data from documents on demand.

```
$ claude mcp add --transport http llamaparse
https://mcp.llamaindex.ai/mcp
```

Adds `"version": "2.5"` and `"cite_sources": True` configuration keys to client.extract.run() to select the new extraction engine and enable Advanced Citations.

Adds `"confidence_scores": True` configuration key to client.extract.run() to return per-field confidence scores alongside extracted values.

Extends Advanced Citations (bounding-box source grounding) from Agentic Plus to the Agentic tier, raising grounding scores from 46.8 to 80.6 for Agentic and from 46.4 to 82.2 for Agentic Plus.

Adds native spreadsheet extraction via a `spreadsheet mode` that lets agents work directly with workbook cells rather than a flattened document representation, enabled in the extraction configuration.

Adds `expand=["extract_metadata"]` parameter to client.extract.get() to retrieve per-field metadata including grounding and confidence information.

+2 more

WAS THIS USEFUL? 👍 👎

◆ MCP Servers & Tooling

MCP Python SDK

Feedback ⓘ v2.3.0 NOTES ↗ RANK

The MCP Python SDK provides clients and servers for exposing tools, resources, and prompts through Model Context Protocol.

> MCP Python SDK v2.3.0 lifts the SSE event size limit, adds subscription opt-out and low-level tool-schema injection, and ships several breaking changes to header validation, protocol serialization, and the httpx2 dependency requirement.

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

01 Configurable SSE event size limit NEW

85

Adds a `max_sse_event_size=` parameter to `streamable_http_client` and `StreamableHttpParameters` to raise or remove the previous 1 MiB-per-event SSE limit; pass an integer byte count or `None` for unlimited.

Stream large tool results (e.g. multi-MB file exports) that exceed the default 1 MiB SSE event cap.

python

```python
from mcp.client.streamable_http import streamable_http_client

async with streamable_http_client(
    "http://localhost:8000/mcp",
    max_sse_event_size=16 * 1024 * 1024,  # 16 MiB
) as (read, write):
    ...
```

— Runnable example with exact parameter and default limit named v2.3.0

02 Custom tool-schema injection replaces per-call tools/list validation

IMPROVED

80

Server(get_tool_input_schema=...) lets a low-level server supply a tool's schema for `Mcp-Param-*` header validation without invoking its `tools/list` handler. As a consequence, `MCPServer` no longer runs `tools/list` for every `tools/call` to validate `Mcp-Param-*` headers, so middleware that filters or rewrites `tools/list` results no longer affects header validation.

— Named config surface with clear before/after behaviour change v2.3.0

03 Eager validation of tool header annotations BREAKING

75

A tool decorated with @mcp.tool(), registered via `add_tool`, or built with `Tool.from_function` that carries an invalid `x-mcp-header` annotation now raises `InvalidSignature` at registration time, instead of starting the server and silently dropping the tool from client listings.

— Clear migration trigger and exact exception named v2.3.0

04 **Subscription opt-out for lightweight servers** NEW

70

MCPServer(subscriptions=False) stops a server from serving `subscriptions/listen` and makes it advertise `listChanged` and `subscribe` as false to clients.

Deploy a lightweight MCP server that does not implement resource subscriptions and advertises that clearly to clients.

```python
from mcp.server import MCPServer

mcp = MCPServer("MyServer", subscriptions=False)

@mcp.tool()
def ping() -> str:
    return "pong"
```

— Clear config flag with runnable example v2.3.0

05 **Empty optional protocol fields no longer sent** BREAKING

70

Outbound requests no longer include empty `_meta` and `params` fields; middleware and server code reading `ctx.meta` now receive `None` instead of `{}`, and `ctx.params` is `None` for requests without params. `initialize` also no longer sends `"experimental": {}`, so client code reading `capabilities.experimental` on a legacy connection must handle `None`.

— Names exact fields changed, needs code audit not a command v2.3.0

THINNER COVERAGE BELOW

06 **Automatic retry on header mismatch** IMPROVED

55

`Client.call_tool` now automatically re-lists tools and retries once after a `HeaderMismatch` (`-32020`) rejection.

— Automatic behaviour, no user-facing flag to act on v2.3.0

07 **httpx2 dependency requirement raised** BREAKING

40

`httpx2>=2.10.0` is now required (previously `>=2.5.0`); upgrades will fail if `httpx2` is pinned below 2.10.

— Simple version bump with exact constraint named `v2.3.0`

└──▷ BREAKING ON UPGRADE

! `httpx2>=2.10.0` is now required (previously `>=2.5.0`); upgrades will fail if `httpx2` is pinned below 2.10.

! A tool decorated with @mcp.tool(), registered via `add_tool`, or built with `Tool.from_function` that carries an invalid `x-mcp-header` annotation now raises `InvalidSignature` at registration time instead of starting the server and silently dropping the tool from client listings.

! Empty `_meta` and `params` fields are no longer sent on outbound requests; middleware and server code reading `ctx.meta` will now receive `None` instead of `{}`, and `ctx.params` will be `None` for requests without params.

! `initialize` no longer sends `"experimental": {}`; client code reading `capabilities.experimental` on a legacy connection must handle `None`.

! `MCPServer` no longer runs `tools/list` for every `tools/call` to validate `Mcp-Param-*` headers; middleware that filters or rewrites `tools/list` results no longer affects header validation.

v2.3.0 NOTES ↗ LATEST STABLE

MCP Python SDK v2.3.0 adds SSE event size control, subscription opt-out, and custom tool-schema injection for low-level servers.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.3.0
https://github.com/modelcontextprotocol/python-sdk.git
# already have the repo? check out this version:
$ git checkout v2.3.0
```

└──▷ USE IT

Stream large tool results (e.g. multi-MB file exports) that exceed the default 1 MiB SSE event cap.

```python
from mcp.client.streamable_http import streamable_http_client

async with streamable_http_client(
    "http://localhost:8000/mcp",
    max_sse_event_size=16 * 1024 * 1024,  # 16 MiB
) as (read, write):
    ...
```

Deploy a lightweight MCP server that does not implement resource subscriptions and advertises that clearly to clients.

```python
from mcp.server import MCPServer

mcp = MCPServer("MyServer", subscriptions=False)

@mcp.tool()
def ping() -> str:
    return "pong"
```

Adds `max_sse_event_size=` parameter to `streamable_http_client` and `StreamableHttpParameters` to raise or remove the 1 MiB-per-event SSE limit (pass an integer or `None`).

Adds MCPServer(subscriptions=False) to stop serving `subscriptions/listen` and advertise `listChanged` and `subscribe` as false.

Adds Server(get_tool_input_schema=...) so a low-level server can supply a tool's schema for `Mcp-Param-*` header validation without invoking its `tools/list` handler.

`Client.call_tool` now automatically re-lists tools and retries once after a `HeaderMismatch` (`-32020`) rejection.

└──▷ BREAKING ON UPGRADE

- ! `httpx2>=2.10.0` is now required (previously `>=2.5.0`); upgrades will fail if `httpx2` is pinned below 2.10.
- ! A tool decorated with @mcp.tool(), registered via `add_tool`, or built with `Tool.from_function` that carries an invalid `x-mcp-header` annotation now raises `InvalidSignature` at registration time instead of starting the server and silently dropping the tool from client listings.
- ! Empty `_meta` and `params` fields are no longer sent on outbound requests; middleware and server code reading `ctx.meta` will now receive `None` instead of `{}`, and `ctx.params` will be `None` for requests without params.

! `initialize` no longer sends `"experimental": {}`; client code reading `capabilities.experimental` on a legacy connection must handle `None`.

! `MCPServer` no longer runs `tools/list` for every `tools/call` to validate `Mcp-Param-*` headers; middleware that filters or rewrites `tools/list` results no longer affects header validation.

WAS THIS USEFUL? 👍 👎

MCP TypeScript SDK

Feedback ⓘ v2.3.0 NOTES ↗ RANK

The MCP TypeScript SDK provides TypeScript clients, servers, and transports for Model Context Protocol.

> This window's MCP TypeScript SDK release adds Tasks extension support (`tasks/get`, `tasks/cancel`) for the 2026-07-28 spec, wildcard origin handling so browser-extension clients can connect, and new `maxToolInputElements` / `expectedResource` guardrails, alongside a safer default for HTTP redirects and a ~50x SSE throughput gain for large messages; server and stateless transport reuse across requests is now rejected outright.

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

01 Same-origin redirect restriction with `redirectPolicy: 'follow'` opt-out

91

BREAKING

HTTP client transports (`StreamableHTTPClientTransport`, `SSEClientTransport`) no longer follow redirects by default unless they stay on the same origin (scheme, host, and port) and method; cross-origin or method-changing redirects now fail with a named error so the session is preserved. Set `redirectPolicy: 'follow'` on the transport to delegate redirect handling back to the fetch implementation and restore pre-2.3.0 behaviour; a `requestInit.redirect` of `'error'` or `'manual'` is honoured for the transport's own requests (POST, GET, DELETE of Streamable HTTP; POST of SSE) but not for OAuth requests. In browsers, a redirected request fails unless `redirectPolicy: 'follow'` is set.

Allow a transport whose endpoint permanently redirects to another origin — set `redirectPolicy: 'follow'` to restore pre-2.3.0 behaviour while you migrate.

typescript

```typescript
import { StreamableHTTPClientTransport } from
'@modelcontextprotocol/client/http';

const transport = new StreamableHTTPClientTransport(
  new URL('https://legacy-proxy.example.com/mcp'),
  { redirectPolicy: 'follow' }
);
```

— Exact flag, affected methods, and a migration example given v2.3.0

@modelcontextprotocol/client@2.3.0 1.32.0

02 **Single active connection per server/transport instance** BREAKING

91

Server.connect() now rejects if the instance is already connected, and a stateless `StreamableHTTPServerTransport` (with `sessionIdGenerator: undefined`) now allows only one request at a time. Reusing a single server or stateless transport across requests fails on the second call: connect() rejects with `SdkError` code `ALREADY_CONNECTED`, WebStandardStreamableHTTPServerTransport.handleRequest() rejects with "Stateless transport cannot be reused across requests. Create a new transport per request.", and NodeStreamableHTTPServerTransport.handleRequest() returns HTTP 500. Fix: create a new `McpServer` and transport per request, or per session when using a `sessionIdGenerator`.

— Named error codes/messages per transport plus an explicit fix v2.3.0

@modelcontextprotocol/server@2.3.0

---

03 **`expectedResource` audience restriction for bearer auth** NEW

86

Adds `expectedResource` option on `requireBearerAuth` / `verifyBearerToken` to restrict accepted tokens to those issued for this server's audience/resource URI, rejecting tokens minted for other services. Note: `@modelcontextprotocol/express` 2.0.1 does not forward `expectedResource` to the underlying `requireBearerAuth`, causing a TypeScript error unless upgraded to the release shipped alongside server@2.3.0.

Restrict bearer-token acceptance to tokens issued specifically for this server's audience.

typescript

```
import { requireBearerAuth } from
'@modelcontextprotocol/server/auth';

app.use(requireBearerAuth({
  verifyBearerToken,
  expectedResource: 'https://my-mcp-server.example.com/'
}));
```

Restrict bearer-token acceptance to tokens issued specifically for your MCP server's URL, so tokens minted for other services are rejected with 401.

```typescript
import { requireBearerAuth } from
'@modelcontextprotocol/server/auth';

app.use(
  requireBearerAuth({
    verifyAccessToken: async (token) => {
      // decode token and return AuthInfo with resource set from
the 'aud' claim
      const info = await myJwtVerifier(token);
      return { ...info, resource: info.aud };
    },
    expectedResource: 'https://mcp.example.com'
  })
);
```

Reject bearer tokens not explicitly issued for this server's resource URI, preventing token reuse across services.

```typescript
import { requireBearerAuth } from
'@modelcontextprotocol/server/auth';

app.use(requireBearerAuth({
  // ... existing verifier config ...
  expectedResource: 'https://my-mcp-server.example.com'
}));
```

— Named option, three runnable examples, and a compatibility caveat `v2.3.0`

`@modelcontextprotocol/server@2.3.0` `1.32.0`

04 **Wildcard origin support for browser-extension clients** NEW

80

Adds `<scheme>://*` wildcard entries (e.g. `moz-extension://*`, `chrome-extension://*`) to `validateOriginHeader` and the `allowedOrigins` option of the Express, Fastify, and Hono app helpers, letting MCP clients running as browser extensions connect even though their per-install extension IDs can't be enumerated at configuration time; `http://*` and `https://*` wildcards are explicitly not honoured.

Allow Firefox browser-extension MCP clients whose extension IDs vary per install, without whitelisting individual origins.

```typescript
import { requireBearerAuth } from
'@modelcontextprotocol/server/auth';
// or pass allowedOrigins to your transport / middleware:
const transport = new WebStandardStreamableHTTPServerTransport({
  allowedOrigins: ['moz-extension://*', 'chrome-extension://*']
});
```

— Named config option, framework helpers, and a working snippet v2.3.0

@modelcontextprotocol/server@2.3.0

05 **`maxToolInputElements` cap on tool call arguments** NEW

78

Adds `maxToolInputElements` option on `McpServer` to cap the number of array elements and object members in a tool call's arguments, rejecting oversized payloads before schema validation runs so the server stays alive and returns a structured error.

Defend against oversized tool-call payloads before schema validation runs, keeping the server alive while returning a structured error to the caller.

```typescript
import { McpServer } from '@modelcontextprotocol/server';

const server = new McpServer(
  { name: 'my-server', version: '1.0.0' },
  { maxToolInputElements: 1000 }
);
```

Protect a public-facing MCP server from oversized tool-call payloads by capping the total number of array elements and object members in any single argument.

```typescript
import { McpServer } from '@modelcontextprotocol/server';

const server = new McpServer(
  { name: 'my-server', version: '1.0.0' },
  { maxToolInputElements: 1000 }
);
```

— Exact config key with two runnable constructor examples v2.3.0

@modelcontextprotocol/server@2.3.0 1.32.0

06 **Faster SSE handling for large single-event messages** IMPROVED

68

Large messages received as a single SSE event over Streamable HTTP are now up to ~50x faster — a 50 MB result drops from ~13s to under 1s. Requires `eventsource-parser` 3.0.8 or later; earlier versions are rejected as incompatible.

— Concrete benchmark and dependency named, but no code to run v2.3.0

@modelcontextprotocol/client@2.3.0

07 Tasks extension endpoints (tasks/get, tasks/cancel) NEW

66

Adds `tasks/get` and `tasks/cancel` methods of the Tasks extension (SEP-2663), servable and callable on 2026-07-28 protocol connections. When one factory serves both protocol eras, handlers can be registered conditionally with `ctx.era === 'legacy'` to restrict behaviour to 2025-era clients only.

— Mechanism and spec version named, but no runnable example given v2.3.0

@modelcontextprotocol/client@2.3.0 @modelcontextprotocol/server@2.3.0

THINNER COVERAGE BELOW

08 `prompts/get` defaults missing arguments to empty object

42

BREAKING

`prompts/get` called without `arguments` is now validated as `{}`, matching `tools/call` behaviour; a top-level .optional() or .default(...) on a prompt's `argsSchema` no longer receives `undefined`.

— Thin breaking note with no example or migration command v2.3.0

└──▷ BREAKING ON UPGRADE

! Server.connect() now rejects if the instance is already connected — servers shared across requests will break; create `McpServer` and the transport inside the request handler instead.

! HTTP client transports now follow redirects only when they stay on the same origin (same scheme, host, and port); a deployment whose endpoint redirects to another host or port must configure the final URL or set `redirectPolicy: 'follow'` on the transport. In browsers, a redirected request fails unless that option is set.

! `prompts/get` called without `arguments` is now validated as `{}`, matching `tools/call` behaviour; a top-level .optional() or .default(...) on a prompt's `argsSchema` no longer receives `undefined`.

! The client now requires `eventsource-parser` 3.0.8 or later; earlier versions are incompatible.

! HTTP client transports no longer follow cross-origin or method-changing redirects by default; if your endpoint redirects to another origin, configure the transport with the final destination URL, or set `redirectPolicy: 'follow'` to restore the previous behaviour.

! A `requestInit.redirect` of `'error'` or `'manual'` is now honoured for the transport's own requests (POST, GET, DELETE of Streamable HTTP; POST of SSE); for OAuth requests and

any other value it is not consulted.

! `SSEClientTransport` and `eventsource-parser -based` paths now require `eventsource-parser` 3.0.8 or later.

! A `Server` or `McpServer` now allows only one active connection at a time; a stateless `StreamableHTTPServerTransport` (with `sessionIdGenerator: undefined`) now allows only one request at a time. Any setup that reuses a single server or stateless transport across multiple requests will fail on the second request: connect() rejects with `SdkError` code `ALREADY_CONNECTED`, WebStandardStreamableHTTPServerTransport.handleRequest() rejects with 'Stateless transport cannot be reused across requests. Create a new transport per request.', and NodeStreamableHTTPServerTransport.handleRequest() returns HTTP 500. Fix: build a new `McpServer` and transport per request (or per session when using a `sessionIdGenerator`).

! `@modelcontextprotocol/express` 2.0.1 does not forward `expectedResource` to the underlying `requireBearerAuth`; TypeScript reports an error if `expectedResource` is passed to the 2.0.1 version. Upgrade `@modelcontextprotocol/express` to the release that ships alongside server@2.3.0.

! HTTP client transports (`StreamableHTTPClientTransport`, `SSEClientTransport`) no longer follow redirects to a different host or port by default; any deployment whose endpoint redirects cross-origin must set `redirectPolicy: 'follow'` or update to the final URL. In browsers, a cross-origin redirect fails unless `redirectPolicy: 'follow'` is set.

v2.3.0 NOTES ↗ LATEST STABLE

MCP TypeScript SDK v2.3.0 adds browser-extension origin support and Tasks extension endpoints for the 2026-07-28 spec.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.3.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout v2.3.0
```

└──▷ USE IT

Restrict bearer-token acceptance to tokens issued specifically for this server's audience.

typescript

```
import { requireBearerAuth } from
'@modelcontextprotocol/server/auth';

app.use(requireBearerAuth({
  verifyBearerToken,
  expectedResource: 'https://my-mcp-server.example.com/'
}));
```

Adds `tasks/get` and `tasks/cancel` Tasks extension methods, servable and callable on a 2026-07-28 connection.

Adds `<scheme>://*` wildcard entries (e.g. `moz-extension://*`) to `validateOriginHeader` and the `allowedOrigins` option of the Express, Fastify, and Hono app helpers, enabling MCP clients running as browser extensions to connect.

Adds `maxToolInputElements` option on `McpServer` to cap the number of array elements and object members in a tool call's arguments.

Adds `expectedResource` option on `requireBearerAuth` / `verifyBearerToken` to restrict accepted tokens to those issued for this server's audience.

Adds `redirectPolicy: 'follow'` option on HTTP client transports to allow cross-origin or cross-host redirects explicitly.

└──▷ BREAKING ON UPGRADE

- **!** Server.connect() now rejects if the instance is already connected — servers shared across requests will break; create `McpServer` and the transport inside the request handler instead.
- **!** HTTP client transports now follow redirects only when they stay on the same origin (same scheme, host, and port); a deployment whose endpoint redirects to another host or port must configure the final URL or set `redirectPolicy: 'follow'` on the transport. In browsers, a redirected request fails unless that option is set.
- **!** `prompts/get` called without `arguments` is now validated as `{}`, matching `tools/call` behaviour; a top-level .optional() or .default(...) on a prompt's `argsSchema` no longer receives `undefined`.
- **!** The client now requires `eventsource-parser` 3.0.8 or later; earlier versions are incompatible.

`@modelcontextprotocol/client@2.3.0` NOTES ↗ STABLE

MCP TypeScript SDK client adds a redirect-safety policy, Tasks-extension support, and a 50x SSE throughput leap for large messages.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @modelcontextprotocol/client@2.3.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout @modelcontextprotocol/client@2.3.0
```

└──▷ USE IT

Allow a transport whose endpoint permanently redirects to another origin — set `redirectPolicy: 'follow'` to restore pre-2.3.0 behaviour while you migrate.

```typescript
import { StreamableHTTPClientTransport } from
'@modelcontextprotocol/client/http';

const transport = new StreamableHTTPClientTransport(
  new URL('https://legacy-proxy.example.com/mcp'),
  { redirectPolicy: 'follow' }
);
```

Adds `redirectPolicy: 'follow'` on HTTP client transports to delegate redirect handling back to the fetch implementation; all other redirects are now restricted to same-origin, same-method only, and cross-origin redirects fail with a named error so the session is preserved.

Adds support for `tasks/get` and `tasks/cancel` of the Tasks extension (SEP-2663) on 2026-07-28 connections; register handlers per-era with `ctx.era === 'legacy'` when one factory serves both protocol eras.

Large messages received as a single SSE event over Streamable HTTP are now up to ~50x faster — a 50 MB result drops from ~13 s to under 1 s; requires `eventsource-parser` 3.0.8 or later.

└──▷ BREAKING ON UPGRADE

- **!** HTTP client transports no longer follow cross-origin or method-changing redirects by default; if your endpoint redirects to another origin, configure the transport with the final destination URL, or set `redirectPolicy: 'follow'` to restore the previous behaviour.
- **!** A `requestInit.redirect` of `'error'` or `'manual'` is now honoured for the transport's own requests (POST, GET, DELETE of Streamable HTTP; POST of SSE); for OAuth requests and any other value it is not consulted.
- **!** `SSEClientTransport` and `eventsource-parser -based` paths now require `eventsource-parser` 3.0.8 or later.

`@modelcontextprotocol/server@2.3.0` NOTES ↗ STABLE

MCP TypeScript SDK: Adds wildcard scheme entries (e.g., `moz-extension://*`, `chrome-extension://*`) to `allowedOrigins` and `validateOriginHeader`, allowing MCP servers to admit browser-extension clients whose IDs cannot be enumerated at configuration time; `http://*` and `https://*` are explicitly not honoured.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch @modelcontextprotocol/server@2.3.0
https://github.com/modelcontextprotocol/typescript-sdk.git
# already have the repo? check out this version:
$ git checkout @modelcontextprotocol/server@2.3.0
```

└──▷ USE IT

Restrict bearer-token acceptance to tokens issued specifically for your MCP server's URL, so tokens minted for other services are rejected with 401.

```typescript
import { requireBearerAuth } from
'@modelcontextprotocol/server/auth';

app.use(
  requireBearerAuth({
    verifyAccessToken: async (token) => {
      // decode token and return AuthInfo with resource set from
the 'aud' claim
      const info = await myJwtVerifier(token);
      return { ...info, resource: info.aud };
    },
    expectedResource: 'https://mcp.example.com'
  })
);
```

Defend against oversized tool-call payloads before schema validation runs, keeping the server alive while returning a structured error to the caller.

```typescript
import { McpServer } from '@modelcontextprotocol/server';

const server = new McpServer(
  { name: 'my-server', version: '1.0.0' },
  { maxToolInputElements: 1000 }
);
```

Allow Firefox browser-extension MCP clients whose extension IDs vary per install, without whitelisting individual origins.

```typescript
import { requireBearerAuth } from
'@modelcontextprotocol/server/auth';
// or pass allowedOrigins to your transport / middleware:
const transport = new WebStandardStreamableHTTPServerTransport({
  allowedOrigins: ['moz-extension://*', 'chrome-extension://*']
});
```

Supports `tasks/get` and `tasks/cancel` of the Tasks extension (SEP-2663) on 2026-07-28 connections when the handler is registered and the request carries an explicit schema; register conditionally with `ctx.era === 'legacy'` to restrict to 2025-era clients only.

└──▷ BREAKING ON UPGRADE

- **!** A `Server` or `McpServer` now allows only one active connection at a time; a stateless `StreamableHTTPServerTransport` (with `sessionIdGenerator: undefined`) now allows only one request at a time. Any setup that reuses a single server or stateless transport across multiple requests will fail on the second request: connect() rejects with `SdkError` code `ALREADY_CONNECTED`, WebStandardStreamableHTTPServerTransport.handleRequest() rejects with 'Stateless transport cannot be reused across requests. Create a new transport per request.', and NodeStreamableHTTPServerTransport.handleRequest() returns HTTP 500. Fix: build a new `McpServer` and transport per request (or per session when using a `sessionIdGenerator`).
- **!** `@modelcontextprotocol/express` 2.0.1 does not forward `expectedResource` to the underlying `requireBearerAuth`; TypeScript reports an error if `expectedResource` is passed to the 2.0.1 version. Upgrade `@modelcontextprotocol/express` to the release that ships alongside server@2.3.0.

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Modal

Feedback ⓘ changelog · Oct 3, 2026 CHANGELOG ↗ RANK

Bring your own code, and run CPU, GPU, and data-intensive compute at scale. The serverless platform for AI and data teams.

> Modal shipped VM Sandboxes to general availability, giving agents full Linux VMs with Docker and kernel-level access, and added isolated Sidecar containers for trust-boundary separation, alongside several breaking changes to Sandbox filesystem APIs and CLI JSON output formats.

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

01 **VM Sandboxes with full Linux runtime** NEW 95

Passing `runtime="vm"` to modal.Sandbox.create() (or `modal sandbox create --runtime vm`) launches a full Linux VM instead of the default gVisor container, enabling Docker-in-Docker, FUSE filesystems, cgroups, network namespaces, and arbitrary kernel features. VM Sandboxes share the same `modal.Image`, exec/FS APIs, sub-second cold-starts, CPU/memory bursting, and concurrent-scale guarantees as existing gVisor-based Sandboxes, and modal.Probe.with_exec() can be passed as `readiness_probe` to Sandbox.create() so callers wait until a specific process (e.g. `docker info`) succeeds before the Sandbox is declared ready.

Spin up a Docker-in-Docker VM Sandbox and wait until Docker is ready before running a container — useful for agent evals or dev-environment clones that require a real Docker daemon.

python

```
import modal
app = modal.App.lookup("sandbox-manager",
create_if_missing=True)
probe = modal.Probe.with_exec("docker", "info")
sb = modal.Sandbox.create(
    app=app,
    image=modal.Image.from_registry("docker:dind"),
    readiness_probe=probe,
    runtime="vm",
)
sb.wait_until_ready()
p = sb.exec("docker", "run", "hello-world")
print(p.stdout.read())
```

Verify you are inside a real Linux VM (not a gVisor container) by checking the kernel identity after creating the Sandbox with `runtime="vm"`.

```python
import modal
app = modal.App.lookup("sandbox-manager",
create_if_missing=True)
sb = modal.Sandbox.create(app=app, runtime="vm")
print(sb.exec("uname", "-srn").stdout.read())  # Linux modal
7.2.6
```

Launch a Sandbox using the full VM runtime to enable Docker-in-Sandbox workflows.

```
$ modal sandbox create --runtime vm my-image
```

— Full mechanism, named flag/params, and runnable code and CLI examples

launch-20261001-3a38039e changelog-20261003-325f79ab

02 **Sandbox Sidecars for trust-boundary isolation** NEW 95

sb._experimental_sidecars.create() dynamically spins up isolated Sidecar containers alongside a running Sandbox at runtime, sharing CPU and memory with their parent (up to 250 Sidecars per Sandbox) for cheaper trust-boundary separation than separate Sandboxes. Each Sidecar gets its own outbound network policy independent of the main Sandbox, traffic can be forced through a proxy Sidecar, and isolation uses the same gVisor or VM boundary determined by the runtime passed to Sandbox.create().

Run a trusted HTTP proxy Sidecar next to an agent Sandbox so credentials and egress control never touch untrusted code.

```python
import modal

app = modal.App.lookup("sidecar-example",
create_if_missing=True)
image = modal.Image.debian_slim().build(app)

sb = modal.Sandbox.create("sleep", "600", app=app, image=image,
timeout=300)

sidecar = sb._experimental_sidecars.create(
    "python", "-m", "http.server", "8080",
    name="web",
    image=image,
)
```

— Named API, concrete limit, and a runnable proxy example launch-20261001-8c394fb7

03 **Short-lived API tokens via CLI** NEW 70

`modal token new --expires-in 8h` issues a short-lived API token (expiring after a set duration like 8 hours) for use in CI pipelines.

Issue a short-lived API token that expires after 8 hours for use in CI pipelines.

```
$ modal token new --expires-in 8h
```

— Exact runnable CLI command with flag and example duration changelog-20261003-325f79ab

THINNER COVERAGE BELOW

04 **Automatic Sandbox connection lifecycle** IMPROVED 50

Modal now drops the client connection to a Sandbox automatically after inactivity and reconnects on next use, making Sandbox.detach() optional rather than required.

— Describes behavior change but no runnable example changelog-20261003-325f79ab

05 **Larger call graph output from FunctionCall** IMPROVED 45

FunctionCall.get_call_graph() now returns up to 5,000 nodes, up from a previous cap of 100.

— Clear before/after numbers but no usage example changelog-20261003-325f79ab

└──▷ BREAKING ON UPGRADE

! The legacy Sandbox filesystem APIs Sandbox.open(), Sandbox.ls(), Sandbox.mkdir(), Sandbox.rm(), and Sandbox.watch(), the associated file-handle type, and `modal.exception.FilesystemExecutionError` have been removed — use the `Sandbox.filesystem` APIs instead.

! An unspecified experimental CLI has been removed — use the recommended Server/endpoint tooling instead.

! `modal volume ls` JSON output now shows `created_modified` as an ISO 8601 timestamp and `size` as an integer number of bytes (previously different types/formats).

! `modal queue list` JSON output now shows `partitions` and `total_size` as integers and adds a boolean `total_size_truncated` field.

! Several undocumented properties and methods have been removed from the SDK.

> changelog · Oct 3, 2026 CHANGELOG ↗ LATEST STABLE
>
> Modal: Expands FunctionCall.get_call_graph() to return up to 5,000 nodes (previously 100).
>
> └──▷ TRY IT
>
> Launch a Sandbox using the full VM runtime to enable Docker-in-Sandbox workflows.

```
$ modal sandbox create --runtime vm my-image
```

Issue a short-lived API token that expires after 8 hours for use in CI pipelines.

```
$ modal token new --expires-in 8h
```

Drops client connection to a Sandbox automatically after inactivity and reconnects on next use, making Sandbox.detach() optional.

└──▷ **BREAKING ON UPGRADE**

- ! The legacy Sandbox filesystem APIs Sandbox.open(), Sandbox.ls(), Sandbox.mkdir(), Sandbox.rm(), and Sandbox.watch(), the associated file-handle type, and `modal.exception.FilesystemExecutionError` have been removed — use the `Sandbox.filesystem` APIs instead.
- ! An unspecified experimental CLI has been removed — use the recommended Server/endpoint tooling instead.
- ! `modal volume ls` JSON output now shows `created_modified` as an ISO 8601 timestamp and `size` as an integer number of bytes (previously different types/formats).
- ! `modal queue list` JSON output now shows `partitions` and `total_size` as integers and adds a boolean `total_size_truncated` field.
- ! Several undocumented properties and methods have been removed from the SDK.

announced · Oct 1, 2026 **BLOG** ↗

Modal VM Sandboxes now GA: pass `runtime="vm"` to get a full Linux VM with Docker, FUSE, and kernel-level access for agents.

└──▷ **USE IT**

Spin up a Docker-in-Docker VM Sandbox and wait until Docker is ready before running a container — useful for agent evals or dev-environment clones that require a real Docker daemon.

```python
import modal
app = modal.App.lookup("sandbox-manager", create_if_missing=True)
probe = modal.Probe.with_exec("docker", "info")
sb = modal.Sandbox.create(
    app=app,
    image=modal.Image.from_registry("docker:dind"),
    readiness_probe=probe,
    runtime="vm",
)
sb.wait_until_ready()
p = sb.exec("docker", "run", "hello-world")
print(p.stdout.read())
```

Verify you are inside a real Linux VM (not a gVisor container) by checking the kernel identity after creating the Sandbox with `runtime="vm"`.

```python
import modal
app = modal.App.lookup("sandbox-manager", create_if_missing=True)
sb = modal.Sandbox.create(app=app, runtime="vm")
print(sb.exec("uname", "-srn").stdout.read())  # Linux modal 7.2.6
```

Adds `runtime="vm"` parameter to modal.Sandbox.create() to launch full Linux VMs instead of the default gVisor containers, enabling Docker-in-Docker, FUSE filesystems, cgroups, network namespaces, and arbitrary kernel features.

Adds modal.Probe.with_exec() for `readiness_probe` on modal.Sandbox.create(), allowing callers to wait until a specific process (e.g. `docker info`) succeeds before the Sandbox is declared ready.

VM Sandboxes share the same `modal.Image`, exec/FS APIs, sub-second cold-starts, CPU/memory bursting, and concurrent-scale guarantees as the existing gVisor-based Sandboxes.

announced · Oct 1, 2026 BLOG ↗

Modal Sandboxes gain Sidecars: isolated co-located containers with 3x lower latency than separate Sandboxes for trusted/untrusted code separation.

└──▷ USE IT

Run a trusted HTTP proxy Sidecar next to an agent Sandbox so credentials and egress control never touch untrusted code.

```python
import modal

app = modal.App.lookup("sidecar-example", create_if_missing=True)
image = modal.Image.debian_slim().build(app)

sb = modal.Sandbox.create("sleep", "600", app=app, image=image,
timeout=300)

sidecar = sb._experimental_sidecars.create(
    "python", "-m", "http.server", "8080",
    name="web",
    image=image,
)
```

Adds sb._experimental_sidecars.create() to the Modal SDK to dynamically spin up isolated Sidecar containers alongside a running Sandbox at runtime.

Sidecar containers share CPU and memory with their parent Sandbox (up to 250 Sidecars per Sandbox), making co-located trust-boundary execution more cost-efficient than separate Sandboxes.

Each Sidecar gets its own outbound network policy independent of the main Sandbox, and the Sandbox's outbound traffic can be forced through a proxy Sidecar.

Sidecar isolation uses the same gVisor or VM boundary as separate Sandboxes, controlled by the runtime passed to Sandbox.create().

WAS THIS USEFUL? 👍 👎

AWS Amazon Bedrock AgentCore

Feedback

Gateway: Private certificate authority support for targets (2026-10-02)

BLOG ↗ RANK

Amazon Bedrock AgentCore provides managed agent runtimes, memory, identity, gateways, observability, and evaluation services.

> Bedrock AgentCore Gateway now supports private certificate authorities for TLS, letting targets connect directly to private VPC endpoints.

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

01 **Private CA support for Gateway targets** NEW 62

Bedrock AgentCore Gateway now supports TLS certificates issued by a private certificate authority (CA) for MCP server, OpenAPI, and HTTP proxy targets, enabling direct connections to private VPC endpoints without needing an intermediate Application Load Balancer.

— Names mechanism and target types but no config steps shown

Gateway: Private certificate authority …

> Gateway: Private certificate authority support for targets (2026-10-02)
>
> BLOG ↗ LATEST STABLE
>
> Bedrock AgentCore Gateway supports private CA TLS certificates for MCP, OpenAPI, and HTTP proxy targets, enabling direct VPC endpoint connections.

WAS THIS USEFUL? 👍 👎

Google Gen AI JavaScript SDK

Feedback ⓘ v2.27.0

NOTES ↗ RANK

The Google Gen AI JavaScript SDK provides TypeScript and JavaScript interfaces to Gemini APIs and live model sessions.

> The Google Gen AI JavaScript SDK added resumable generation requests via a continuation token and expanded its model catalog with three new identifiers.

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

01 Continuation token for resuming generation requests NEW

70

The SDK now supports `continuation_token` in `GenerateContent` to resume or continue a generation request, and the same support was noted for Interactions. A response's `continuation_token` can be passed back into a subsequent `generateContent` call to continue a prior generation.

Resume a previously started generation by passing a continuation token back to `generateContent`.

typescript

```
const response = await ai.models.generateContent({
  model: 'gemini-flash-latest',
  contents: 'Continue the story.',
  continuation_token: previousResponse.continuation_token,
});
```

— Named field with a runnable code example v2.26.0 v2.27.0

THINNER COVERAGE BELOW

02 New models added to Model enum NEW

40

Adds `lyria-3.5`, `gemini-omni-1.1-flash`, and `gemini-omni-flash-preview` to the Interactions `Model` enum.

— Named models but no usage detail v2.27.0

v2.27.0 NOTES ↗ LATEST STABLE

Adds `continuation_token` support in Interactions and three new model identifiers to the `Model` enum.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.27.0 https://github.com/googleapis/js-genai.git
# already have the repo? check out this version:
$ git checkout v2.27.0
```

Adds `lyria-3.5`, `gemini-omni-1.1-flash`, and `gemini-omni-flash-preview` to the Interactions `Model` enum.

v2.26.0 NOTES ↗ STABLE

Adds `continuation_token` support to `GenerateContent` in the Google Gen AI JS SDK v2.26.0.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v2.26.0 https://github.com/googleapis/js-genai.git
# already have the repo? check out this version:
$ git checkout v2.26.0
```

└──▷ **USE IT**

Resume a previously started generation by passing a continuation token back to `generateContent`.

typescript

```
const response = await ai.models.generateContent({
  model: 'gemini-flash-latest',
  contents: 'Continue the story.',
  continuation_token: previousResponse.continuation_token,
});
```

Supports `continuation_token` in `GenerateContent` to resume or continue a generation request.

WAS THIS USEFUL? 👍 👎

Google Gen AI Python SDK

Feedback ⓘ v2.28.0 NOTES ↗

RANK

The Google Gen AI Python SDK provides Python interfaces to Gemini model generation, streaming, tools, and embeddings.

> The Google Gen AI Python SDK added resumable generation via continuation tokens across both GenerateContent and Interactions, per-part speech style control, and three new model enum entries.

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

01 **SpeechMetadata for per-part voice and style** NEW 85

`Part` gains a `SpeechMetadata` field with `voice` (speaker identifier) and `style` (e.g. `excited, fast-paced`) sub-fields, enabling per-part speaker and style control for multi-speaker speech synthesis.

Attach per-part speaker and style metadata to drive multi-speaker synthesis with distinct voices and delivery styles.

python

```
from google.genai import types

part = types.Part(
    text='Welcome to the show!',
    speech_metadata=types.SpeechMetadata(
        voice='Alice',
        style='excited, fast-paced'
    )
)
```

— Named field, sub-fields, and runnable code example product docs

02 **Continuation tokens for resumable generation** NEW 75

A new `continuation_token` field appears on generation responses and requests: when `finish_reason` is `CONTINUATION`, the returned token can be passed back in a subsequent request to resume generation from where it stopped. Support was added to `GenerateContent` and then to `Interactions`.

— Names field, finish_reason value, and both APIs but no code sample v2.28.0 v2.27.0

THINNER COVERAGE BELOW

03 **New model enum values for Interactions** NEW 45

The Interactions `Model` enum now includes `lyria-3.5`, `gemini-omni-1.1-flash`, and `gemini-omni-flash-preview`.

— Names exact enum values but no usage detail `v2.28.0`

`v2.28.0` **NOTES ↗** **LATEST** **STABLE**

Google Gen AI Python SDK v2.28.0 adds three new model enum values and continuation_token support in Interactions.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.28.0 https://github.com/googleapis/python-genai.git
# already have the repo? check out this version:
$ git checkout v2.28.0
```

Adds `lyria-3.5`, `gemini-omni-1.1-flash`, and `gemini-omni-flash-preview` to the Interactions `Model` enum.

`v2.27.0` **NOTES ↗** **STABLE**

Google Gen AI Python SDK v2.27.0 adds continuation_token support in GenerateContent for resumable generation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.27.0 https://github.com/googleapis/python-genai.git
# already have the repo? check out this version:
$ git checkout v2.27.0
```

WAS THIS USEFUL? 👍 👎

Google MaxText

Feedback ⓘ maxtext-v0.2.5 NOTES ↗ RANK

MaxText provides JAX implementations and distributed-training workflows for language models on TPUs and GPUs.

> MaxText's v0.2.5 release lands DeepSeek-V4 and Qwen3.5 model support, GRPO reinforcement-learning infrastructure, native FP8 inference, and a migration to the Orbax v1 checkpointing API, alongside new documentation for custom mesh rules, multi-host knowledge distillation and SFT workflows on GKE.

└──▷ **WHAT SHIPPED · 23 FEATURES** most completely described first ? what's the number?

01 Multi-host SFT workload submission via gcluster job submit NEW

95

Adds `gcluster job submit` with `--compute-type` and `--topology` flags to launch supervised fine-tuning (SFT) workloads across multi-host TPU slices on GKE via Cluster Toolkit. Adds `--pathways` for running SFT with Pathways, with `--pathways-gcs-location` for checkpoint storage and `ENABLE_PATHWAYS_PERSISTENCE=1` for persistence; `gcluster job logs --main-only false` retrieves logs from all pods when a Pathways workload spans more than 5 pods. Adds `gcluster job config set` to configure the target cluster/project/location, and `gcluster job list` / `gcluster job cancel` for monitoring and cancelling workloads. Workload names must not exceed 28 characters (22 with Pathways) and must be valid DNS labels.

Submit a multi-host SFT training job on a v6e-256 TPU slice using Cluster Toolkit after configuring cluster access.

```
$ gcluster job submit --image <IMAGE> --name <WORKLOAD_NAME> --compute-type ct6e-hightpu-4t --topology 16x16 "python3 -m maxtext.trainers.post_train.sft.train_sft run_name=<RUN_NAME> per_device_batch_size=1 steps=<STEPS> profiler=xplane hf_path=<HF_PATH>"
```

Submit an SFT workload with Pathways on GKE, enabling persistence and specifying a GCS checkpoint location.

```
$ gcluster job submit --image <IMAGE> --name <WORKLOAD_NAME> --pathways --compute-type ct5p-hightpu-4t --topology 4x4x4 --pathways-gcs-location <GCS_PATH> "ENABLE_PATHWAYS_PERSISTENCE=1 python3 -m maxtext.trainers.post_train.sft.train_sft run_name=<RUN_NAME> per_device_batch_size=1 steps=<STEPS> profiler=xplane checkpoint_storage_use_zarr3=<VALUE>"
```

Retrieve logs from all pods in a Pathways SFT workload (required when the workload spans more than 5 pods).

```
$ gcluster job logs --main-only false
```

— Full runnable CLI flags, commands and naming constraints `product docs`

02 **Multi-host knowledge distillation on GKE via Cluster Toolkit**

NEW 91

Adds `src/maxtext/trainers/post_train/distillation/scripts/run_distill_ctk.sh` for multi-host knowledge distillation on GKE via Cluster Toolkit, covering runner image build/push, workload submission, log streaming, workload cancellation (`cleanup`), and an auto-resume loop for long-running jobs. Adds an online distillation trainer backed by Tunix, installed via `install_tpu_post_train_extra_deps` with the tested Tunix version pinned in `src/dependencies/extra_deps/post_train_github_deps.txt` , plus launch scripts/configs for distillation training and integration of MaxText's trainer backend into a distributed GSM8K example.

Submit a multi-host distillation workload to a GKE cluster and stream its logs using the Cluster Toolkit runner script.

```
$ PROJECT_ID=<PROJECT_ID> GKE_CLUSTER=<CLUSTER_NAME> LOCATION=
<LOCATION> COMPUTE_TYPE=tpu7x-standard-4t TOPOLOGY=4x4x4
DOCKER_IMAGE=gcr.io/${PROJECT_ID}/maxtext_base_image:${USER}-
distill bash
src/maxtext/trainers/post_train/distillation/scripts/run_distill
_ctk.sh
```

Cancel and clean up a running distillation workload on GKE after it is no longer needed.

```
$ bash
src/maxtext/trainers/post_train/distillation/scripts/run_distill
_ctk.sh cleanup
```

— Runnable script path and commands for submit and cleanup `maxtext-v0.2.5`

03 **Custom mesh rule for latency-bound FSDP attention sharding** NEW

88

Adds the `fsdp-as-dp-for-attn.yml` custom mesh rule for large-scale training where FSDP rank is large enough that small-weight all-gathers become latency bound: it shards

attention weights (`kv_lora` , `embed_attn`) on the smaller axis while replicating over `fsdp_transpose` , fully replicates the MoE router weight (`embed_router`) with `[]` , and keeps larger routed and shared expert weights sharded on `fsdp_transpose` .

Apply the attention-as-data-parallel mesh rule when FSDP rank is large and attention all-gathers are latency bound in MoE training.

```text
fsdp-as-dp-for-attn.yml
```

— Names exact config file and sharding mechanism per axis `product docs`

04 **Native FP8 quantization and inference engine** NEW 78

Adds `drhs_grad_quantization_calibration_method` for weight-gradient-arm cotangent calibration in a new FP8 weight-only dynamic dequantization engine, and `serve_fp8_weight` , a native FP8 inference path for DenseGeneral and GMM v2 with pre-quantized weights for TPU inference fused MoE kernels.

Enable native FP8 inference with pre-quantized weights for reduced memory and faster decode on TPUs.

```yaml
serve_fp8_weight: true
```

— Named config keys with a runnable YAML example `maxtext-v0.2.5`

05 **MoE training configuration additions** NEW 75

Adds `moe_dropless_fallback` config key (values `None` , `step` , `layer`) with `eval_ragged_buffer_factor` for evaluation-specific buffer scaling and host-driven step replay retry on buffer overflow. Adds `moe_pin_sparse_core_all_gathers` to offload FSDP and Expert Parallel all-gathers to SparseCores, `moe_x_sorted` to save routed expert inputs across remat boundaries, and `shard_embed_moe_on_fsdp` as an explicit tracking flag for MoE embedding sharding.

Enable MoE dropless fallback at the step level to handle buffer overflows without memory overhead during MoE training runs.

```yaml
moe_dropless_fallback: step
eval_ragged_buffer_factor: 2.0
```

Offload FSDP and Expert Parallel all-gathers to SparseCores to reduce host-side communication overhead in large MoE models.

```yaml
moe_pin_sparse_core_all_gathers: true
```

— Config keys with example YAML snippets for two of them `maxtext-v0.2.5`

06 **DeepSeek-V4 and DeepSeek-V3 Lineage model support** NEW

72

Adds DeepSeek-V4 checkpoint conversion and parameter mapping with 2D transposed shape alignment, DeepSeek-V4 indexer loss, and separate final normalization for Multi-Token Prediction (MTP). Integrates DeepSeek-V3 Lineage (`dsv3.py`) with routed expert output weights stored in Lineage's layout and router bias updates sown along the `use_lineage` path.

— Detailed mechanism and named file but no runnable example `maxtext-v0.2.5`

07 **GRPO RL training and vLLM rollout infrastructure** NEW

70

Adds `free_kv_cache_during_weight_sync` , `reward_num_workers` , and `gc_collect_after_weight_sync` settings to the GRPO (`AgenticGRPOLearner`) post-training RL infrastructure, plus RL sequence packing and vLLM block-size configuration knobs. Adds `layer_name_to_kvcache_index` for indexed KV caches in the vLLM adapter, and Mamba prefix caching for Qwen3 GDN layers in vLLM rollout. Adds a Gemma 4 26B GRPO tutorial and a Gemma 4 E4B RL tutorial for post-training RL workflows.

— Many named config keys but no mechanism detail or example `maxtext-v0.2.5`

08 **Attention kernel and context-parallelism performance improvements**

IMPROVED

66

Enhances the JAX Flash Attention kernel with a specialized high-performance execution path for long sequences, splitting at a 4,096 sequence length threshold. Adds a fused Pallas TPU StreamIndex score kernel, QK attention head chunking for Compressed Self-Attention (CSA) memory reduction, and HCA static compilation for Splash Attention on DeepSeek-V4. Shards GatedDeltaNet sequences under `ici_context_parallelism` and enables native block-sparse execution for indexer masks under Context Parallelism (CP).

— Concrete threshold and named kernels, no direct entry point `maxtext-v0.2.5`

09 **Checkpointing migrated to Orbax v1 API** BREAKING

65

Refactors MaxText checkpointing to natively use the Orbax v1 API, retiring the legacy v0 implementation; adds dequantize-on-load parameter restoration and support for

restoring checkpoint weights stored outside the standard `'params'` collection. Existing checkpoint restore code using v0 paths requires updating.

— Clear before/after but no migration command shown `maxtext-v0.2.5`

THINNER COVERAGE BELOW

10 **MLPerf training support** NEW 59

Adds the `dsv3-mlperf-4k` custom mesh and sharding rule, full MLPerf training logging compliance, and per-file `c4_mlperf` train sharding for accelerated step-0 initialization.

— Named mesh and sharding config, no walkthrough `maxtext-v0.2.5`

11 **MoE kernel and layout optimizations** NEW 55

Adds MoE 128-lane chunking and padding for GMM layout, token-quantized all-gather pipelines with Ring of Experts, and quantized all-gather for the Ring of Experts combine backward. Integrates TransformerEngine (TE) MoEBlock and Tokamax GMM v2 heuristic tiling functions.

— Internal kernel work named but no usage surface given `maxtext-v0.2.5`

12 **Multimodal SFT and processing pipelines** NEW 55

Adds a Grain-based multimodal Supervised Fine-Tuning (SFT) data processing pipeline with end-to-end video SFT supporting variable frame sizes and durations, and a ChartNet SFT configuration. Adds a custom multimodal processor and end-to-end pipeline for Omni multimodal models, and enables `vllm_decode` inference support for Qwen3-VL.

— Several named pipelines/models but no entry points given `maxtext-v0.2.5`

13 **Sharding-aware distributed Muon optimizer** NEW 54

Integrates a sharding-aware distributed Muon optimizer with support for Qwen3 and GPT-OSS MoE blocks and a configuration flag to govern applying Muon to MoE routers.

— Mentions a config flag but doesn't name it `maxtext-v0.2.5`

14 **Qwen3.5 model configs** NEW 52

Adds `qwen3.5-35b-a3b-fp8` and `qwen3.5-397b-a17b-fp8` model configs with HuggingFace parameter mapping and weight conversion.

— Named configs but thin on mechanism `maxtext-v0.2.5`

15 **New model architectures and blocks** NEW 48

Adds model definitions and bring-up support for Cosmos 3 Nano and Cosmos 3 Super Reasoner, a Weaver Mixture-of-Transformers block with standardized Weaver model naming across configs, RMSNorm normalization support for Apple Envy decoder blocks

and OLMo 3 MLP blocks (`DecoderBlockType.OLMO3`), and attention-sink support for GPT-OSS models.

– List of named architectures with little mechanism `maxtext-v0.2.5`

16 **Explicit sharding and ZeRO-1 across model families** IMPROVED 48

Onboards explicit sharding and ZeRO-1 gradient accumulation across Qwen3.5, Qwen3-Next, Qwen3, Qwen2, Kimi-K2, Mistral, Mixtral, and Gemma model families.

– Names all families but no config or flag given `maxtext-v0.2.5`

17 **Observability configuration additions** NEW 45

Adds `goodput_job_name` configuration option for Goodput monitoring, and custom GCS path support for ML Diagnostics telemetry.

– Named config keys, no mechanism beyond naming `maxtext-v0.2.5`

18 **m3 package scaffolding** NEW 44

Adds m3 package scaffolding with architecture rules, layout verification tests, an m3 core RoPE implementation, mesh creation, and sharding rules.

– Named components but no usage path `maxtext-v0.2.5`

19 **Default dataset_type changed to synthetic** BREAKING 40

Changes the default `dataset_type` to `synthetic` in data pipeline configuration; workloads relying on the previous default must now explicitly set `dataset_type` in their config.

– Named config key and clear remediation step `maxtext-v0.2.5`

20 **Megatron MMap data format support for Grain** NEW 40

Adds Megatron MMap data format support for Grain data loaders with an integrated mmap index builder.

– Names format and loader but no usage detail `maxtext-v0.2.5`

21 **DiLoCo pre-training documentation and tutorials** NEW 40

Adds DiLoCo pre-training documentation and tutorials with launch scripts migrated to Cluster Toolkit.

– Thin description of docs and script migration `maxtext-v0.2.5`

22 **Dependency upgrades** IMPROVED 40

Upgrades JAX dependencies to 0.11.1, Tokamax to 0.0.14, and DrJAX to 0.2.1.

— Exact versions named, no behavior change described maxtext-v0.2.5

23 **Quantized Multi-Token Prediction for pre-training** IMPROVED

25

Enables quantized Multi-Token Prediction (MTP) for accelerated pre-training.

— Single sentence with no mechanism or config named maxtext-v0.2.5

└──▷ BREAKING ON UPGRADE

! The default `dataset_type` is now `synthetic`; workloads relying on the previous default must explicitly set `dataset_type` in their config.

! MaxText checkpointing has migrated to the Orbax v1 API, retiring legacy v0 implementations; existing checkpoint restore code using v0 paths requires updating.

maxtext-v0.2.5 NOTES ↗ LATEST STABLE

MaxText v0.2.5 adds DeepSeek-V4, Qwen3.5, GRPO RL infrastructure, FP8 inference, Orbax v1 checkpointing, and Megatron MMap data support.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch maxtext-v0.2.5 https://github.com/AI-Hypercomputer/maxtext.git
# already have the repo? check out this version:
$ git checkout maxtext-v0.2.5
```

└──▷ USE IT

Enable MoE dropless fallback at the step level to handle buffer overflows without memory overhead during MoE training runs.

```yaml
moe_dropless_fallback: step
eval_ragged_buffer_factor: 2.0
```

Offload FSDP and Expert Parallel all-gathers to SparseCores to reduce host-side communication overhead in large MoE models.

```yaml
moe_pin_sparse_core_all_gathers: true
```

Enable native FP8 inference with pre-quantized weights for reduced memory and faster decode on TPUs.

```yaml
serve_fp8_weight: true
```

Adds `free_kv_cache_during_weight_sync`, `reward_num_workers`, and `gc_collect_after_weight_sync` settings to the GRPO (`AgenticGRPOLearner`) post-training RL infrastructure, plus RL sequence packing and vLLM block-size configuration knobs.

Adds `moe_dropless_fallback` config key (values: `None`, `step`, `layer`) with `eval_ragged_buffer_factor` for evaluation-specific buffer scaling and host-driven step replay retry on buffer overflow.

Adds `moe_pin_sparse_core_all_gathers` config option to offload FSDP and Expert Parallel all-gathers to SparseCores.

Adds `goodput_job_name` configuration option for Goodput monitoring.

Adds `drhs_grad_quantization_calibration_method` for weight-gradient-arm cotangent calibration in the new FP8 weight-only dynamic dequantization engine.

+33 more

└──▷ BREAKING ON UPGRADE

- ! The default `dataset_type` is now `synthetic`; workloads relying on the previous default must explicitly set `dataset_type` in their config.
- ! MaxText checkpointing has migrated to the Orbax v1 API, retiring legacy v0 implementations; existing checkpoint restore code using v0 paths requires updating.

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.24.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> The OpenAI Python SDK's latest releases focus on the Agents API, adding typed stream returns, file staging, agent-tool actions, and final-output collection, alongside new custom voice creation and agent session events.

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

01 **Typed stream returns and final-output collection in Agents API** NEW 50

Session streams in the agents API now return typed, structured objects instead of raw events, and streams support final-output collection for beta Agents.

– Describes behavior change but no concrete type names or code sample v3.23.0

02 **File staging and turn-artifact download in Agents API** NEW 40

The agents API gains file staging and the ability to download turn artifacts.

– Names the capability but no API signature or usage shown v3.23.0

03 **Typed application actions as agent tools** NEW 40

The beta agents API now exposes typed application actions as agent tools.

– States capability without mechanism or example v3.23.0

04 **Custom voice creation** NEW 25

Adds custom voice creation to the API surface.

– Bare mention with no detail on how voices are created v3.24.0

05 **Agent session events** NEW 25

Adds agent session events to the API surface.

– Bare mention with no schema or usage detail v3.24.0

v3.24.0 NOTES ↗ LATEST STABLE

OpenAI Python SDK v3.24.0 adds custom voice creation and agent session events to the API.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.24.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.24.0
```

v3.23.0 NOTES ↗ STABLE

OpenAI Python SDK v3.23.0 adds typed agent stream returns, file staging/artifact download, session traces, Realtime translations, and typed application actions as agent tools

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.23.0 https://github.com/openai/openai-python.git
# already have the repo? check out this version:
$ git checkout v3.23.0
```

Adds typed return values from session streams in the agents API, so callers receive structured objects instead of raw events.

Adds file staging and turn-artifact download to the agents API.

Exposes typed application actions as agent tools in the beta agents API.

Adds final-output collection from beta Agents streams.

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ v0.8.112 NOTES ↗ RANK

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

> HyperFrames shipped a new motion-blur-streak animation skill and a set of Studio timeline refinements covering clip insertion previews, export metadata, and clip labeling.

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

01 **Motion-blur-streak skill for tracked elements** NEW 50

Adds a motion-blur-streak skill with a per-frame-driven form for elements riding a baked track.

— Names mechanism but no flag or config to invoke it v0.8.112

02 **Studio timeline UI improvements** IMPROVED 41

Studio previews timeline track insertion so you can see where a dragged clip will land before dropping it, exports timeline row order and fade handle boxes with a readable fx badge, and adds a Filmstrip name band to timeline clips.

— Describes UI changes without exact navigation or config v0.8.112

v0.8.112 NOTES ↗ STABLE

HyperFrames v0.8.112 adds motion-blur-streak skill, timeline track insertion preview, and filmstrip name bands for clips.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.112 https://github.com/heygen-com/hyperframes.git
# already have the repo? check out this version:
$ git checkout v0.8.112
```

Adds motion-blur-streak skill with a per-frame-driven form for elements riding a baked track.

Previews timeline track insertion in Studio so you can see where a dragged clip will land before dropping it.

Exports timeline row order and fade handle boxes with a readable fx badge in Studio.

Adds a Filmstrip name band to timeline clips in Studio.

WAS THIS USEFUL? 👍 👎

ArcBox

Feedback ⓘ v0.8.1 NOTES ↗ RANK

ArcBox runs containers, microVM agent sandboxes, and full Linux or macOS virtual machines on a Mac from one Rust daemon and CLI.

> ArcBox added per-machine DNS names and network provisioning, rebuilt its uninstall command around a full file inventory, and extended the Python SDK's SandboxEvent with storage metering and ordered sequencing.

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

01 DNS names and network provisioning for machines NEW

83

Each distro machine is reachable at `<machine>.arcbox.local` and is assigned its hostname, bridge NIC, and a default route at provisioning time. ArcBox refuses to create a machine whose derived hostname conflicts with an existing one, and the development profile now gets its own DNS port so it survives a conflict with the default DNS port already in use.

Reach a Linux VM by its DNS name over the local network without looking up its IP — useful when scripting against a machine or opening a browser to a service running inside it.

```
$ abctl machine start dev
curl http://dev.arcbox.local:8080
```

— Names hostname, bridge NIC, route, DNS ports; runnable curl example v0.8.1

02 Rebuilt abctl uninstall command IMPROVED

65

`abctl uninstall` is rebuilt on an inventory of everything ArcBox writes to disk, reporting each removal step as it runs so cleanup is transparent.

Cleanly remove ArcBox and all its written files, seeing exactly what each step does as it runs.

```
$ abctl uninstall
```

— Names mechanism and exact command, no numbers on scope v0.8.1

THINNER COVERAGE BELOW

03 SandboxEvent storage metering and ordering NEW

58

The Python SDK adds a `storage_bytes` field to every sandbox state to meter the retained storage footprint of each state, and adds a monotonic sequence number to `SandboxEvent` enabling reliable ordering and gap detection of sandbox lifecycle events.

— Names exact fields but gives no usage example `sdk-python-v0.1.3`

`v0.8.1` NOTES ↗ **LATEST** STABLE

ArcBox v0.8.1 adds per-machine DNS names at <machine>.arcbox.local, a rebuilt `abctl uninstall`, and bridge address visibility in the CLI.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.8.1
https://github.com/arcboxlabs/arcbox.git
# already have the repo? check out this version:
$ git checkout v0.8.1
```

└──▷ TRY IT

Reach a Linux VM by its DNS name over the local network without looking up its IP — useful when scripting against a machine or opening a browser to a service running inside it.

```
$ abctl machine start dev
curl http://dev.arcbox.local:8080
```

Cleanly remove ArcBox and all its written files, seeing exactly what each step does as it runs.

```
$ abctl uninstall
```

Adds `abctl uninstall` rebuilt on an inventory of everything ArcBox writes, reporting each removal step as it runs.

Assigns distro machines their hostname, bridge NIC, and a default route at provisioning time.

Enforces uniqueness: refuses to create a machine whose derived hostname conflicts with an existing machine's hostname.

Gives the development profile its own DNS port so it survives a conflict with the default DNS port already in use.

`sdk-python-v0.1.3` NOTES ↗ STABLE

ArcBox Python SDK v0.1.3 adds monotonic sequence numbers and storage_bytes metering to SandboxEvent.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch sdk-python-v0.1.3
https://github.com/arcboxlabs/arcbox.git
# already have the repo? check out this version:
$ git checkout sdk-python-v0.1.3
```

Adds `storage_bytes` field to every sandbox state, metering the retained storage footprint of each state.

Adds a monotonic sequence number to `SandboxEvent`, enabling reliable ordering and gap detection of sandbox lifecycle events.

WAS THIS USEFUL? 👍 👎

Baseten

Feedback ⓘ Baseten CLI 1.1.0 NOTES ↗ RANK

Baseten deploys and serves machine learning models as autoscaling inference endpoints, with a model API catalog and dedicated GPU deployments.

> Baseten expanded its Model API catalog with new coding-agent, vision, reasoning, and hosted LLM models (GLM-5.3, GLM-5.2, DeepSeek V4 Flash 0731, Kimi K3), added a reasoning token budget cap, and shipped a CLI command to rename models in place.

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

01 Reasoning token budget control plus new reasoning models NEW

90

Added the `reasoning.max_tokens` field to cap reasoning tokens independently from the overall `max_tokens` completion limit; `deepseek-ai/DeepSeek-V4-Flash-0731` enforces it with a default of 4,096 tokens while other reasoning models ignore it. Also added `zai-org/GLM-5.2` as a new supported reasoning model accessible via the `model` field.

Raise the reasoning token budget for DeepSeek V4 Flash 0731 when a hard problem needs more thinking room than the 4,096-token default.

```json
{
  "model": "deepseek-ai/DeepSeek-V4-Flash-0731",
  "reasoning": {"max_tokens": 20000},
  "max_tokens": 24000
}
```

— Names field, default value, enforcement scope and an example product docs

02 Kimi K3 support in Vision API NEW

85

Added `moonshotai/Kimi-K3` as a supported model on the Vision API, using 28×28-pixel image tiles where each tile becomes one input token, priced at $3.00 per million uncached input tokens.

Send an image to the Vision API using Kimi K3 for multimodal inference.

```
$ curl -X POST https://bridge.baseten.co/v1/chat/completions \
  -H 'Authorization: Api-Key <your-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "moonshotai/Kimi-K3",
    "messages": [
      {"role": "user", "content": [
        {"type": "image_url", "image_url": {"url": "<image-
url>"}},
        {"type": "text", "text": "Describe this image."}
      ]}
    ]
  }'
```

— Mechanism, pricing and a runnable example all included `product docs`

---

03 **`baseten model rename` CLI command** NEW 75

Baseten CLI 1.1.0 adds `baseten model rename` to rename a model from the terminal while preserving its ID, endpoints, and deployments — only the name in `config.yaml` needs updating.

— Exact command and config key named, no usage example shown `Baseten CLI 1.1.0`

---

04 **Model catalog growth across coding-agent and hosted LLM APIs**

NEW 65

Added `zai-org/GLM-5.3` as a supported model on the coding-agents model API, and separately made it callable through the general hosted LLM API alongside `deepseek-ai/DeepSeek-V4-Flash-0731` and `moonshotai/Kimi-K3`, all usable via OpenAI- and Anthropic-compatible SDKs without deploying infrastructure.

Call GLM-5.3 through the OpenAI-compatible API without spinning up any infrastructure.

```
$ curl https://api.baseten.co/v1/chat/completions \
  -H 'Authorization: Bearer $BASETEN_API_KEY' \
  -H 'Content-Type: application/json' \
  -d '{"model": "zai-org/GLM-5.3", "messages": [{"role": "user",
"content": "Explain zero-trust networking."}]}'
```

— Names models and API surfaces but no pricing or limits given `product docs`

Baseten CLI 1.1.0 NOTES ↗ LATEST STABLE

> Baseten CLI 1.1.0 adds `baseten model rename` to rename models from the terminal without changing their ID or endpoints.

WAS THIS USEFUL? 👍 👎

Fireworks AI Feedback ⓘ

Higher serverless rate limits for Small models NOTES ↗ RANK

Fireworks AI provides hosted inference, model fine-tuning, and deployment APIs for open-weight models.

> Fireworks AI expanded its Small serverless tier to cover more models and raised the tier's rate-limit ceilings.

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

01 **Expanded Small serverless tier limits and coverage** IMPROVED 63

The Small serverless tier now covers all models with fewer than 600B total parameters, including GLM 5.3 Flash, DeepSeek V4.1 Flash, and OpenAI GPT OSS 120B. Adaptive rate-limit ceilings for the tier were raised to 108M Total Prompt TPM, 27M Uncached Prompt TPM, and 1.08M Generated TPM.

— Names exact models and numeric limits but no API/config usage shown.

Higher serverless rate limits for Small…

> Higher serverless rate limits for Small models NOTES ↗ LATEST STABLE
>
> Fireworks AI raises Small-tier serverless rate limits to 108M Total Prompt TPM and expands the tier to models under 600B parameters.
>
> Expands the Small serverless tier to cover all models with fewer than 600B total parameters, including GLM 5.3 Flash, DeepSeek V4.1 Flash, and OpenAI GPT OSS 120B.
>
> Raises Small-tier adaptive rate-limit ceilings to 108M Total Prompt TPM, 27M Uncached Prompt TPM, and 1.08M Generated TPM.

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.27.0 NOTES ↗ RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> The OpenAI TypeScript SDK expanded its beta Agents API with hosted file handling, structured streaming output, session trace retrieval, and typed tool bindings.

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

01 **Agent session trace listing** NEW 45

Adds agent session trace listing via the API (#2845), enabling retrieval of per-session trace records for beta Agents.

– References PR but not an endpoint or method v7.26.0

02 **Typed output parsing and final result collection for Agent streams**

NEW 40

Beta Agents streams now collect the final output from a run, making the end result directly accessible, and parse typed output from agent streams (part 1 of n), surfacing strongly-typed structured results from streaming agent responses.

– Describes mechanism but no concrete API names v7.26.0

03 **Hosted file prep and artifact downloads for Agents** NEW 35

The agents API now supports preparing hosted files and downloading result artifacts, letting agent workflows stage input files and retrieve generated outputs.

– Names capability but no API surface or flag given v7.27.0

04 **Typed function binding to Agent tools** NEW 35

Binds typed application functions to Agents tools in the beta SDK, allowing TypeScript-typed functions to be registered as agent tool implementations.

– Explains purpose but no flag or signature given v7.26.0

v7.27.0 NOTES ↗ LATEST STABLE

Agents in the OpenAI TypeScript SDK can now prepare hosted files and download result artifacts.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v7.27.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.27.0
```

Adds support for preparing hosted files and downloading result artifacts in the agents API.

v7.26.0 NOTES ↗ STABLE

OpenAI Node SDK v7.26.0 adds typed output and tool binding for beta Agents streams, plus agent session trace listing.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v7.26.0 https://github.com/openai/openai-node.git
# already have the repo? check out this version:
$ git checkout v7.26.0
```

Adds agent session trace listing via the API (#2845), enabling retrieval of per-session trace records for beta Agents.

Collects final output from beta Agents streams, making the end result of a streamed agent run directly accessible.

Parses typed output from agent streams (part 1 of n), surfacing strongly-typed structured results from streaming agent responses.

Binds typed application functions to Agents tools in the beta SDK, allowing TypeScript-typed functions to be registered as agent tool implementations.

WAS THIS USEFUL? 👍 👎

Cloudflare Workers AI

Feedback

Workers AI - Introducing Clef: Cloudflare's first open-source decision models, now on

BLOG ↗ RANK

Cloudflare Workers AI runs open-weight models, including the Clef decision models, as serverless inference on Cloudflare's global network.

> Cloudflare launched Clef, its first open-source decision models on Workers AI, adding structured probability outputs, multimodal input, and a drop-in path for existing text-classification integrations.

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

01 **Structured question types: noul, choice, score** NEW

90

Clef introduces three structured question types — `noul` (yes/no probability), `choice` (pick one option from a defined set with per-option probabilities), and `score` (probability-weighted rubric rating) — supporting up to 64 questions per request, letting callers get structured probabilities instead of parsing free-form text.

Route a support ticket in the hot path — determine urgency and owning team in one structured call, with no free-form output to parse.

```
$ curl
https://api.cloudflare.com/client/v4/accounts/{account_id}/ai/ru
n/@cf/cloudflare/clef-flash \
  -H 'Authorization: Bearer <API_TOKEN>' \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "clef-flash",
    "state": "Checkout has been failing for every customer for
the last hour.",
    "questions": {
      "urgent": {
        "type": "noul",
        "instructions": "Is this support request urgent?"
      },
      "team": {
        "type": "choice",
        "instructions": "Which team should handle this
request?",
        "criteria": {
          "billing": "Payments, invoices, and refunds",
          "technical": "Outages, errors, and configuration",
          "sales": "Plans and upgrades"
        }
      }
    }
  }'
```

Score a user submission against a trust-and-safety policy rubric before taking automated action — using the `score` question type inside a Worker.

```javascript
const response = await env.AI.run('@cf/cloudflare/clef', {
  model: 'clef',
  state: userSubmissionText,
  questions: {
    policy_compliance: {
      type: 'score',
      instructions: 'How well does this submission comply with
our content policy?',
      criteria: {
        1: 'Clear violation',
        2: 'Likely violation',
        3: 'Borderline',
        4: 'Compliant',
        5: 'Fully compliant'
      }
    }
  }
});
// response.answers.policy_compliance.score -> probability-
weighted score
// Act on score threshold without parsing free-form text
```

— Names all three question types with runnable curl and Worker code

Workers AI - Introducing Clef: Cloudfla…

02 **Clef and Clef Flash decision models launch** NEW 70

Cloudflare ships `@cf/cloudflare/clef` (27B parameters, 64K-token context) and `@cf/cloudflare/clef-flash` (9B parameters, 64K-token context), its first self-trained models, returning structured probabilities instead of free-form text. Both are available via env.AI.run() and the REST API at `/ai/run`, and are compatible with AI Gateway through those same env.AI.run() and `/ai/run` endpoints.

— Names both models, sizes, context length, and access endpoints

Workers AI - Introducing Clef: Cloudfla…

THINNER COVERAGE BELOW

03 **System One API for drop-in Jev replacement** NEW 55

Clef implements the System One API, making it a drop-in replacement for Jev integrations by changing only the endpoint and model identifier.

— Names the migration mechanism but no worked example

Workers AI - Introducing Clef: Cloudfla…

04 **Visual classification support in Clef** NEW 50

Clef accepts up to four images alongside the text state for visual classification, a capability text-only decision models lack.

— States the capability and limit but no example shown

Workers AI - Introducing Clef: Cloudfla…

05 **Open-source Clef model weights on Hugging Face** NEW

50

Cloudflare releases model weights for both `clef` and `clef-flash` under the Apache 2.0 license on Hugging Face.

— Names license, platform, and both models Workers AI - Introducing Clef: Cloudfla…

06 **Reinforcement learning fine-tuning service for Clef** NEW

40

Cloudflare launches a reinforcement learning fine-tuning service to tune Clef for custom workloads, currently requiring design-partner sign-up.

— Describes offering but access is gated and unspecified

Workers AI - Introducing Clef: Cloudfla…

Workers AI - Introducing Clef: Cloudflare's first open-source decision models, now c

BLOG ↗ LATEST STABLE

Cloudflare launches `@cf/cloudflare/clef` and `@cf/cloudflare/clef-flash`, open-source decision models on Workers AI returning structured probabilities instead of free-form t

└──▷ TRY IT

Route a support ticket in the hot path — determine urgency and owning team in one structured call, with no free-form output to parse.

```
$ curl
https://api.cloudflare.com/client/v4/accounts/{account_id}/ai/run/@
cf/cloudflare/clef-flash \
  -H 'Authorization: Bearer <API_TOKEN>' \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "clef-flash",
    "state": "Checkout has been failing for every customer for the
last hour.",
    "questions": {
      "urgent": {
        "type": "noul",
        "instructions": "Is this support request urgent?"
      },
      "team": {
        "type": "choice",
        "instructions": "Which team should handle this request?",
        "criteria": {
          "billing": "Payments, invoices, and refunds",
          "technical": "Outages, errors, and configuration",
          "sales": "Plans and upgrades"
        }
      }
    }
  }'
```

Score a user submission against a trust-and-safety policy rubric before taking automated action — using the `score` question type inside a Worker.

```javascript
const response = await env.AI.run('@cf/cloudflare/clef', {
  model: 'clef',
  state: userSubmissionText,
  questions: {
    policy_compliance: {
      type: 'score',
      instructions: 'How well does this submission comply with our
content policy?',
      criteria: {
        1: 'Clear violation',
        2: 'Likely violation',
        3: 'Borderline',
        4: 'Compliant',
        5: 'Fully compliant'
      }
    }
  }
});
// response.answers.policy_compliance.score -> probability-weighted
score
// Act on score threshold without parsing free-form text
```

Adds `@cf/cloudflare/clef` (27B, 64K-token context) and `@cf/cloudflare/clef-flash` (9B, 64K-token context) — Cloudflare's first trained models — available via env.AI.run() and the REST API at `/ai/run`.

Introduces three structured question types — `noul` (yes/no probability), `choice` (pick one from a defined set with per-option probabilities), and `score` (probability-weighted rubric rating) — with up to 64 questions per request.

Supports visual classification: Clef accepts up to four images alongside the text state, unlike text-only decision models.

Implements the System One API, making Clef a drop-in replacement for Jev integrations by changing only the endpoint and model identifier.

Releases model weights for both `clef` and `clef-flash` under the Apache 2.0 license on Hugging Face.

+2 more

WAS THIS USEFUL? 👍 👎

Ollama

Feedback ⓘ v0.35.1 NOTES ↗ CODE ↗ RANK

Local large language model runner that downloads, serves, and chats with models through a CLI and REST API.

> Ollama v0.35.1 introduces a new System One API for running Clef decision models with multimodal input, adds Modelfile capability declarations for decision models, and raises the per-response web search limit.

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

01 System One API for Clef decision models NEW 88

Adds `POST /v1/systemone` supporting Clef (27B) and Clef Flash (9B) decision models, accepting `model`, `state`, `images` (base64), and `questions` fields and returning scored `answers` with `usage` token counts. Multimodal support lets `images` be included alongside `state` text, shared across all questions and scored jointly.

Run a multimodal yes/no decision query against a screenshot using Clef Flash via the System One API.

```
$ curl http://localhost:11434/v1/systemone -d '{
  "model": "clef-flash",
  "state": "The user took this screenshot.",
  "images": ["<base64-encoded image>"],
  "questions": {
    "has_ollama": {"type": "noul", "instructions": "Does this
image contain Ollama?"}
  }
}'
```

— Full endpoint, fields, and runnable curl example given v0.35.1

02 Decision model capability declarations NEW 74

Adds `CAPABILITY` declarations to Modelfiles, preserved through GGUF and safetensors creation, model inheritance, and Modelfile export. `ollama show` and the model list now report only `decision` as the capability for decision models, preventing clients from offering them for general chat, tools, or thinking.

Declare a model as a decision model in a Modelfile so it is routed to System One and not offered for general chat.

```ini
CAPABILITY decision
```

— Config key and behavior named, but no CLI output shown v0.35.1

THINNER COVERAGE BELOW

03 **Higher per-response web search limit** IMPROVED 35

Raises the per-response web search limit from 3 to 10 for models using web search.

— States the before/after numbers but no mechanism or usage v0.35.1

v0.35.1 NOTES ↗ CODE ↗ LATEST STABLE

Ollama v0.35.1 adds Clef decision models via `/v1/systemone`, multimodal scoring, `CAPABILITY` Modelfile declarations, and 10x web search per response.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.35.1 https://github.com/ollama/ollama.git
# already have the repo? check out this version:
$ git checkout v0.35.1
```

└──▷ TRY IT

Run a multimodal yes/no decision query against a screenshot using Clef Flash via the System One API.

```
$ curl http://localhost:11434/v1/systemone -d '{
  "model": "clef-flash",
  "state": "The user took this screenshot.",
  "images": ["<base64-encoded image>"],
  "questions": {
    "has_ollama": {"type": "noul", "instructions": "Does this image
contain Ollama?"}
  }
}'
```

Declare a model as a decision model in a Modelfile so it is routed to System One and not offered for general chat.

Modelfile

```
CAPABILITY decision
```

Adds `POST /v1/systemone` endpoint supporting Clef (27B) and Clef Flash (9B) decision models, accepting `model`, `state`, `images` (base64), and `questions` fields and returning scored `answers` with `usage` token counts.

> Adds multimodal support to `/v1/systemone` requests: `images` can now be included alongside `state` text, shared across all questions and scored jointly.
>
> Adds `CAPABILITY` declarations to Modelfiles, preserved through GGUF and safetensors creation, model inheritance, and Modelfile export.
>
> `ollama show` and the model list now report only `decision` as the capability for decision models, preventing clients from offering them for general chat, tools, or thinking.
>
> Raises the per-response web search limit from 3 to 10 for models using web search.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

mlxtop

Feedback ⓘ v2.0.0 NOTES ↗ RANK

The mlxtop terminal UI monitors local LLMs on Apple Silicon, showing loaded models, memory use, GPU activity, and generation speed.

> mlxtop expanded from an Apple Silicon-only monitor to support eleven LLM runtimes — including vLLM, SGLang, and LocalAI — via Prometheus metrics and authenticated endpoints, while gaining a full Overview dashboard with a Journal, synchronized zoom, per-request speed/token metrics, and simulated NVIDIA multi-GPU display.

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

01 **Overview dashboard with Journal and zoom** NEW 90

Adds an Overview dashboard with a `SYSINFO` strip and `memory` / `paging` / `GPU` / `cache` / `queue` time-series panels, a bottom `Journal` (press `3`) showing timestamped changes newest-first and the strongest correlated signal with affected throughput, and a compact row holding prompt load, `Cache`, and `Queue` side by side with `Enter` expanding each panel to full history. The Paging/I/O panel gained a `SWAP` capacity bar showing used/total with a separate B/s history trace, and the dashboard now shows cyan resident-memory history with separate pressure severity (process-memory readings remain available in Top and text reports). Time-series panels share zoom controls (`+` / `-`, mouse wheel, `0` to reset, with independent zoom for prompt and latency bars) plus synchronized rolling chart windows with `window 34s` labels, and `Tab` / `Shift-Tab` now navigate across views including while filtering processes.

— Names every key panel, hotkey and zoom control across both releases. v2.0.0

02 **Per-request speed and token metrics** NEW 90

The prompt panel shows per-request output speed (`LIVE`, `AVG`, `LAST`) and output token count, e.g. `OUT 346 · LAST 40.0 tok/s`, and displays a first-token latency chart beside the Journal when `timings.time_to_first_token_ms` is present and the terminal is at least 120 columns wide. These rely on `output_tokens_per_second` and `time_to_first_token_ms` fields in completed-request timings JSON, and a counters-only usage recorder now also records native completion speeds, cached tokens, and explicit first-token timings.

Pass completed-request timing so mlxtop can display per-request output speed and first-token latency in the prompt panel.

```
$ {"timings": {"time_to_first_token_ms": 1250,
"output_tokens_per_second": 40.0}}
```

— Gives exact JSON fields and display format for the metrics. v2.0.0

03 **Provider and version-pinning environment variables** NEW

85

Adds `MLXTOP_PROVIDER` to explicitly select a provider endpoint, including remote servers, and `MLXTOP_VERSION` to pin a specific release version when using the shell installer.

Pin mlxtop to a remote oMLX server instead of relying on auto-detection, useful when the model server runs on a different host.

```
$ MLXTOP_PROVIDER=omlx mlxtop
```

Pin the installer to v2.0.0 exactly rather than pulling the latest stable release — useful in reproducible CI environments.

```
$ MLXTOP_VERSION=2.0.0 curl -fsSL
https://raw.githubusercontent.com/maximpri/mlxtop/main/scripts/i
nstall.sh | sh
```

— Both env vars are given with runnable example commands. v2.0.0

04 **Linux memory-pressure threshold config** NEW

80

Adds `memory_warn_load` and `memory_critical_load` config keys to set Linux memory-pressure band thresholds, letting the pressure gauge turn yellow and red at custom load levels.

Tune Linux memory-pressure warning and critical thresholds so the pressure gauge turns yellow and red at custom load levels.

```yaml
memory_warn_load: 0.70
memory_critical_load: 0.90
```

— Config keys and example values make it directly usable. product docs

05 **Support for eleven LLM runtimes** NEW

72

Adds support for eleven LLM runtimes — oMLX, Ollama, LM Studio/llmster, llama.cpp, KoboldCpp, MLX-LM, LocalAI, vLLM, SGLang, Jan, and GPT4All — with live serving metrics, loaded-model information, or model catalogues plus client-reported usage depending on each engine's API. vLLM and SGLang are supported via Prometheus metrics endpoints, with native Ollama and LM Studio model inventories.

— Lists every runtime but no concrete selection command. v2.0.0

06 **Linux static binary release packaging** NEW 60

Ships Linux x86_64 and ARM64 static musl tar.gz archives without runtime library dependencies, with `SHA256SUMS` covering all four release download packages.

— Names exact archive format and checksum file but no install steps. v2.0.0

THINNER COVERAGE BELOW

07 **Authenticated HTTP/HTTPS endpoint support** NEW 45

Adds authenticated HTTP/HTTPS endpoint support with verified certificates and explicit opt-in required before sending credentials to remote servers.

— Describes the capability but no config key or flag given. v2.0.0

08 **NVIDIA multi-GPU display** NEW 35

Adds NVIDIA multi-GPU support to the Overview dashboard, shown with per-card utilization and VRAM across multiple simulated NVIDIA cards.

.

— Only described via a simulated screenshot, no mechanism. product docs

v2.0.0 NOTES ↗ LATEST STABLE

mlxtop v2.0.0 expands to eleven LLM runtimes including vLLM, SGLang, and LocalAI, with Prometheus metrics, authenticated endpoints, and richer memory and usage counters.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.0.0 https://github.com/maximpri/mlxtop.git
# already have the repo? check out this version:
$ git checkout v2.0.0
```

└──▷ TRY IT

Pin the installer to v2.0.0 exactly rather than pulling the latest stable release — useful in reproducible CI environments.

```
$ MLXTOP_VERSION=2.0.0 curl -fsSL
https://raw.githubusercontent.com/maximpri/mlxtop/main/scripts/inst
all.sh | sh
```

Adds support for eleven LLM runtimes: oMLX, Ollama, LM Studio/llmster, llama.cpp, KoboldCpp, MLX-LM, LocalAI, vLLM, SGLang, Jan, and GPT4All — with live serving metrics, loaded-model information, or model catalogues plus client-reported usage depending on each engine's API.

Supports vLLM and SGLang via Prometheus metrics endpoints and native Ollama/LM Studio model inventories.

Adds authenticated HTTP/HTTPS endpoint support with verified certificates and explicit opt-in for sending credentials to remote servers.

Adds `MLXTOP_VERSION` environment variable to pin a specific release version when using the shell installer.

Displays cyan resident-memory history with separate pressure severity in the dashboard; process-memory readings remain available in Top and text reports.

+4 more

WAS THIS USEFUL? 👍 👎

LocalAI

Feedback ⓘ v4.11.0 NOTES ↗ CODE ↗ RANK

LocalAI runs language models locally through self-hosted, OpenAI-compatible APIs.

> LocalAI's v4.11.0 release adds model failover chains with new observability endpoints, a decision-model API (`/v1/systemone`), live realtime diarization and sound-detection events, Sigstore-signed OCI galleries, and a Kimodo text-to-animation backend, alongside a new Studio speaker-diarization and voice-memory workflow and finer-grained agent tool controls.

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

01 **Model failover chains and controls** NEW 95

Adds a `failover.targets` block to model config YAML to define an ordered failover chain of local and remote models under one model name. Adds `GET /api/failover` , `GET /api/failover/{chain}` , and `GET /api/failover/events` endpoints to observe chain health and target switches, plus `X-LocalAI-Served-Model` and `X-LocalAI-Failover` response headers identifying which target served each request. Adds a `localai-proxy` OCI backend to route REST capabilities to another LocalAI endpoint as a remote failover target, and `list_failover_chains` , `pin_failover_target` , and `unpin_failover_target` MCP tools to manage chain pins programmatically. Adds an Operate → Runtime → Failover page in Studio with a live health strip, pin controls, and an installed-models badge, and a `localai.model.failover` realtime session event for failover telemetry.

Keep a model available when your preferred local LLM is unavailable by defining an ordered failover chain that falls back to a remote LocalAI endpoint.

yaml

```yaml
name: assistant-llm
failover:
  targets:
    - model: preferred-local
    - model: remote-localai
      warm: true
```

Poll the live health and active target of all failover chains to confirm which model is currently serving requests.

```
$ curl http://localhost:8080/api/failover
```

Route all app requests to a local model with automatic fallback to a remote LocalAI endpoint when the local target is unavailable.

```yaml
name: assistant-llm
failover:
  targets:
    - model: preferred-local
    - model: remote-localai
  warm: true
```

— Config key, three endpoints, headers, MCP tools and a runnable curl `v4.11.0` `launch-20261002-7c60ba11`

02 **Decision models via `/v1/systemone` endpoint** NEW 95

Adds a `POST /v1/systemone` endpoint for structured decision requests on models declaring `known_usecases: [decisions]` in their config, enforcing a 64 KiB body limit and a maximum of 64 questions; decision models accept named questions about text and return structured answers for routing, moderation, and model selection. Models marked `known_usecases: [decisions]` are surfaced with a Decisions chip in the Studio React UI, and requests can be authenticated with the `LOCALAI_API_KEY` environment variable when auth is enabled.

Classify an incoming support ticket to choose the right queue and simultaneously check for a cancellation request — all in one API call.

```
$ curl --fail-with-body http://localhost:8080/v1/systemone \
  -H "Authorization: Bearer ${LOCALAI_API_KEY}" \
  -H 'Content-Type: application/json' \
  --data-binary '{
    "model": "laya-vllm-cpp",
    "state": "My subscription was charged twice this month and I
want to cancel.",
    "questions": {
      "queue": {
        "type": "choice",
        "instructions": "Which support queue should handle this
ticket?",
        "criteria": {
          "billing": "Issues with charges, invoices, or
payments",
          "cancellation": "Customer wants to end their
subscription",
          "technical": "Product bugs or outages"
        }
      },
      "cancel_intent": {
        "type": "choice",
        "instructions": "Does the customer explicitly ask to
cancel?",
        "criteria": {
          "yes": "Customer requests cancellation",
          "no": "No cancellation request present"
        }
      }
    }
  }'
```

Route an LLM request to the right specialist model by classifying the task type before forwarding — using `answers.route.choice` to drive the dispatch logic.

```
$ curl --fail-with-body http://localhost:8080/v1/systemone \
  -H "Authorization: Bearer ${LOCALAI_API_KEY}" \
  -H 'Content-Type: application/json' \
  --data-binary '{
    "model": "laya-vllm-cpp",
    "state": "Translate the attached release notes from English
into Italian.",
    "questions": {
      "route": {
        "type": "choice",
        "instructions": "What kind of assistance does this
request need?",
        "criteria": {
          "translation": "Translate text between languages",
          "code": "Write or debug software",
          "general": "Answer general questions"
        }
      }
    }
  }'
```

— Endpoint, limits, config field and runnable curl requests provided `v4.11.0` `launch-20261002-5085afdb`

03 **Tool allow/deny lists and pre-finish gate for agents** NEW

95

Adds `allowed_tools` and `excluded_tools` config keys to restrict which tools (including MCP tools) an agent can see — a tool listed in both is excluded. Adds `required_tool_before_finish` to force an agent to successfully call a specified tool (e.g. a validation or policy check) before delivering its final answer, with `required_tool_before_finish_prompt` to customize the reminder message and `required_tool_before_finish_attempts` (default: 3) to set how many reminders the model gets before the answer goes through anyway.

Restrict an agent to only approved tools and require a policy-check tool to run before it can return a final answer, with a custom reminder and 5 retry attempts.

```yaml
allowed_tools:
  - web_search
  - code_executor
excluded_tools:
  - file_delete
required_tool_before_finish: policy_check
required_tool_before_finish_prompt: 'You must run policy_check
before finishing.'
required_tool_before_finish_attempts: 5
```

— All config keys named with defaults and a runnable yaml example product docs

04

Live diarization and sound-detection events in Realtime API

NEW

90

Realtime sessions now stream live speaker-diarization and sound-classification events alongside transcript deltas with no client changes required. Adds `conversation.item.input_audio_transcription.segment` events when a `diarization_model` companion (e.g. `nemotron-3-diarization-q8_0.gguf`) is loaded with a `parakeet-cpp` transcription model, emitting speaker boundaries with `speaker_name`, `start`, and `end` per closed segment. Adds `conversation.item.sound_detection` events when a `sound_model` companion (e.g. `ced-tiny-q8_0.gguf`) is configured, tagging detected sounds with `label`, `score`, `start`, and `end`. Adds `pipeline.diarization` config key for the offline path (`parakeet-cpp-realtime-scene-tdt`), transcribing whole VAD-committed turns with Parakeet TDT 0.6B v3 (25 languages) and emitting labelled segments with `text` populated before the turn's completion event. Adds gallery models `parakeet-cpp-realtime-scene-tdt`, `parakeet-cpp-realtime-scene-base`, and `parakeet-cpp-realtime-scene-tdt-base` bundling Nemotron-3-Diarization and CED (Tiny or Base) with a single `parakeet-cpp` backend; supports any CED GGUF from `mudler/ced-gguf` (tiny, mini, small, base) as `sound_model`. Also adds `speaker_model`, `speaker_threshold`, and `speaker_margin` config keys to parakeet-cpp model configs for named speaker identification against the shared voice registry, and failover-chain support for realtime pipeline stages so a stage can switch targets without closing the session.

Enable live speaker-diarization and sound-detection on a realtime session using the pre-built live scene pipeline — no client changes needed, standard OpenAI Realtime clients ignore unrecognized events.

```yaml
name: parakeet-realtime-scene
backend: parakeet-cpp
parameters:
  model: realtime_eou_120m-v1-f16.gguf
diarization_model: nemotron-3-diarization-q8_0.gguf
sound_model: ced-tiny-q8_0.gguf
```

Use the offline TDT path to get diarized transcript segments and per-turn sound events with the `parakeet-cpp-realtime-scene-tdt` gallery model by enabling `pipeline.diarization` and pointing `sound_detection` at the same backend.

```yaml
name: gpt-realtime-scene
backend: parakeet-cpp-realtime-scene-tdt
sound_detection: parakeet-cpp-realtime-scene-tdt
pipeline:
  diarization: true
```

— Names events, fields, config keys and gives runnable yaml configs v4.11.0

05 **Speaker diarization and voice memory in Studio** NEW

88

Adds a Studio → Diarization page for speaker diarization — marking speaker turns in a recording without transcribing words — plus a transcript-with-speakers mode that combines word-level transcription with per-turn speaker attribution, and a named speaker-recognition mode that matches voices against saved profiles. The Prepare speakers to remember option captures a speaker's voice profile, and the Name and remember workflow in the Speakers panel lets you preview a clip, assign a name, and persist it for recognition across future recordings; compatible model bundles are installed via Models → Explore. Saved speaker voice profiles are lost when the LocalAI server restarts and are shared across all users of the same instance.

Transcribe a recorded meeting and name recurring speakers so their contributions are auto-attributed in future recordings.

📍 1. In the web UI, go to Models → Explore and install a model with diarization, transcription, and speaker recognition. 2. Go to Studio → Diarization, select that model, and upload your recording. 3. Enable Prepare speakers to remember, then select Diarize. 4. In the Speakers panel, click Preview next to each detected speaker, then choose Name and remember, enter their name, and select

Remember . 5. For future recordings, use the same recognition-capable model — named voices are matched automatically.

Name and remember a recurring meeting participant so LocalAI attributes their speech in future recordings automatically.

📍 1. In the web UI, go to Models → Explore and install the model with diarization, transcription, and speaker recognition. 2. Go to Studio → Diarization, select that model, and upload your recording. 3. Enable Prepare speakers to remember , then select Diarize . 4. In Speakers, click Preview for the target voice, choose Name and remember , enter their name, and select Remember . 5. On a later recording, use the same recognition-capable model — saved names will appear on matched speaker turns.

— Names UI paths, options and steps, plus a restart/sharing caveat v4.11.0

launch-20261002-7c60ba11 launch-20261001-7a2338a7

06 **Kimodo text-to-animation backend** NEW 80

Adds a `POST /3d/animate` endpoint (Kimodo backend) that generates 60–150 frame skeletal animation at 30 FPS from a UTF-8 text prompt, returning binary glTF (`.glb`); the `KIMODO_LIBRARY` environment variable configures the native library path for the backend.

— Endpoint, output format and env var named, no sample call v4.11.0

07 **Operate → This machine host inspection page** NEW 70

Adds an Operate → This machine page in Studio to inspect VRAM, RAM, CPU, disk usage, and per-model details — backend, resident memory, CPU share, uptime, and PID — for every model running on a single host, without distributed mode, with search, sort, log access, and stop controls.

.

— Clear UI path but no command or config to run v4.11.0 launch-20261002-7c60ba11

08 **Signed OCI gallery sources** NEW 68

Adds `oci://host/repository:tag` and digest-pinned OCI references as gallery sources, with Sigstore signature verification and digest-bound caches, and a `source_repository` Sigstore policy field in OCI gallery verification blocks to enforce an exact source constraint on signed gallery artifacts.

— Names the reference format and policy field but no worked example v4.11.0

09 **`POST /v1/audio/diarization` endpoint** NEW 65

Adds a `POST /v1/audio/diarization` endpoint supporting `include_text=true` and `include_speaker_profiles=true` parameters to combine ASR, diarization, and

versioned speaker profiles in a single API call.

— Endpoint and params named but no example call given v4.11.0

THINNER COVERAGE BELOW

10 **Gallery growth with new model families** NEW 58

Adds NeMo diarization and ASR models, four Italian Piper TTS voices, CED sound models, and decision models (kev, Nimble, CLM, Laya, GLiNER2.5-Decide, Qwen3-VL, Tev1) to the gallery, bringing total entries to 1,926.

— Lists named models and a total count but no usage path v4.11.0

11 **`hf_overrides` config key for Hugging Face models** NEW 50

Adds `hf_overrides` config key to merge required top-level model configuration without modifying the downloaded Hugging Face snapshot.

— Named config key but no example or scope detail v4.11.0

└──▷ BREAKING ON UPGRADE

! Saved speaker voice profiles are lost when the LocalAI server restarts and are shared across all users of the same instance — coordinate with other users before saving voices on a shared server.

v4.11.0 NOTES ↗ CODE ↗ LATEST STABLE

LocalAI v4.11.0 adds audio scenes, model failover chains, decision models via `/v1` /systemone, signed OCI galleries, and Kimodo text-to-animation.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v4.11.0 https://github.com/mudler/LocalAI.git
# already have the repo? check out this version:
$ git checkout v4.11.0
```

└──▷ USE IT

Keep a model available when your preferred local LLM is unavailable by defining an ordered failover chain that falls back to a remote LocalAI endpoint.

```yaml
name: assistant-llm
failover:
  targets:
    - model: preferred-local
    - model: remote-localai
      warm: true
```

Poll the live health and active target of all failover chains to confirm which model is currently serving requests.

```
$ curl http://localhost:8080/api/failover
```

Adds `failover.targets` block to model config YAML to define an ordered failover chain of local and remote models under one model name.

Adds `GET /api/failover`, `GET /api/failover/{chain}`, and `GET /api/failover/events` endpoints to observe chain health and target switches.

Adds `X-LocalAI-Served-Model` and `X-LocalAI-Failover` response headers to identify which failover target served each request.

Adds `POST /v1/systemone` endpoint for structured decision requests on models declaring `known_usecases: [decisions]`, enforcing a 64 KiB body limit and a maximum of 64 questions.

Adds `POST /3d/animate` endpoint (Kimodo backend) to generate 60–150 frame skeletal animation at 30 FPS from a UTF-8 text prompt, returning binary glTF (`.glb`).

+14 more

announced · Oct 2, 2026 BLOG ↗

LocalAI: Adds Operate → This machine page showing VRAM, RAM, CPU, model-disk usage, backend, resident memory, CPU share, uptime, and PID for each loaded model, with search, sort, log access, and stop controls.

└──▷ USE IT

Route all app requests to a local model with automatic fallback to a remote LocalAI endpoint when the local target is unavailable.

```yaml
name: assistant-llm
failover:
  targets:
    - model: preferred-local
    - model: remote-localai
  warm: true
```

Adds `Studio → Diarization` speaker-memory workflow: listen to per-speaker samples, name and save a voice profile, and have recognition-capable models surface saved names in later recordings.

announced · Oct 1, 2026 BLOG ↗

LocalAI adds speaker diarization, transcription with speaker turns, and named speaker recognition across recordings.

└──▷ HOW TO FIND IT

Name and remember a recurring meeting participant so LocalAI attributes their speech in future recordings automatically.

📍 1. In the web UI, go to Models → Explore and install the model with diarization, transcription, and speaker recognition. 2. Go to Studio → Diarization, select that model, and upload your recording. 3. Enable `Prepare speakers to remember`, then select `Diarize`. 4. In Speakers, click Preview for the target voice, choose `Name and remember`, enter their name, and select `Remember`. 5. On a later recording, use the same recognition-capable model — saved names will appear on matched speaker turns.

Adds `Models → Explore` installation path for diarization + transcription + speaker recognition model bundles, with setup guide covering model choices and installation requirements.

Supports three diarization modes: speaker turns only (no transcript), transcript with speaker attribution, and named speaker recognition matched against saved voice profiles.

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

AWS Context Ontology Accelerator

Feedback ⓘ

v0.3.4 CODE ↗ RANK

Context Ontology Accelerator provides an open-source semantic layer that helps AI agents make consistent, explainable decisions using formal ontologies and knowledge graphs.

> Context Ontology Accelerator's v0.3.4 release adds a multiplexed Databricks connector that lets one deployed instance serve every Databricks source in a deployment.

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

01 **Multiplexed Databricks connector** NEW 78

Adds `SsmConnectionConfigProvider`, which resolves each source's endpoint from one SSM parameter per Athena catalog name, so a single deployed connector serves every Databricks source in a COA deployment.

— Names exact class and SSM-per-catalog mechanism, no usage example given. v0.3.4

v0.3.4 CODE ↗ LATEST STABLE

Adds a multiplexed Databricks connector that serves every Databricks source in a deployment from a single deployed instance via per-catalog SSM parameters.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.3.4 https://github.com/aws/context-ontology-accelerator.git
# already have the repo? check out this version:
$ git checkout v0.3.4
```

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.133.0 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 latest window focuses on OCR tuning and document-conversion safety: a new RapidOCR checkpoint-size option for balancing accuracy against resource use, authenticated remote-image fetching scoped to trusted origins, and a sandboxed Tectonic engine for rendering TikZ diagrams.

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

01 **Configurable RapidOCR checkpoint size** NEW 95

Adds a `model_size` option (values `tiny`, `small`, `medium`, default `small`) to RapidOCR Python configuration via `options.model_size`, selecting among PP-OCRv6 detection/recognition checkpoints to trade off OCR accuracy against resource consumption. A matching `--rapidocr-model-size` CLI flag on `docling-tools models download rapidocr` prefetches a specific checkpoint for offline or `artifacts_path` use. Requesting an unsupported `model_size`/`language` combination (e.g. `tiny` with Japanese) immediately raises `RapidOcrModelSizeNotSupportedError` instead of failing later during download or inference, and specifying a non-default `model_size` for languages served by PP-OCRv5/PP-OCRv4 (which ignore the setting) logs a warning rather than silently no-oping.

Prefetch a tiny PP-OCRv6 English checkpoint for air-gapped or offline environments before running inference.

```
$ docling-tools models download rapidocr --rapidocr-backend-lang onnxruntime:en --rapidocr-model-size tiny
```

Select the medium PP-OCRv6 checkpoint in Python when you need higher-accuracy OCR and have measured it suits your workload.

```python
options.model_size = "medium"
```

— Names exact flag, config field, error class and default behavior v2.133.0

v2.133.0 NOTES ↗ LATEST STABLE

Docling v2.133.0 adds configurable RapidOCR model size for tuning OCR accuracy vs. resource use.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v2.133.0 https://github.com/docling-project/docling.git
# already have the repo? check out this version:
$ git checkout v2.133.0
```

Supports configuring RapidOCR model size, allowing practitioners to trade off OCR accuracy against resource consumption.

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ snapshot-20261003 NOTES ↗

RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith's biggest move this window is agent-based workspaces (beta), a new way to organize traces around named agents and environments instead of tracing projects, backed by environment-variable and code-level addressing. Alongside it, LangSmith shipped a thread/trace statistics endpoint, self-hosted licensing and ECR auth options, clearer failure logging for Slack-triggered deep agents, and a breaking tightening of dataset export permissions.

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

01 **Agent-based workspaces (beta) with agent/environment trace routing**

NEW 95

Introduces agent-based workspaces (beta) that organize a LangSmith workspace around named agents, each with four fixed environments — Production, Staging, Development, and Local — instead of tracing projects; every trace carries both an agent identifier and an environment, enabling alerts, evaluators, and prebuilt dashboards scoped per-environment. Agents have a stable, immutable `identifier` (used for trace addressing, visible in the browser address bar and the `ID` field) and a separate editable `display name`, so renaming an agent never breaks trace routing. Traces are routed via the `LANGSMITH_AGENT_ID` and `LANGSMITH_AGENT_ENVIRONMENT` environment variables, or per-call via an `address` argument added to `@traceable` and `tracing_context` in Python (`langsmith>=0.14.4`) and to `traceable` in TypeScript (`langsmith>=0.10.8`), with ls.address.agent(id, environment) (Python) and address.agent(id, environment) (TypeScript) helpers building the `lrn:agents/{id}/environments/{environment}` address string; sending a trace to a non-existent identifier auto-creates the agent with no advance setup. Workspace type (agent-based vs project-based) is identifiable via the top-left control: workspace name shown = agent-based (beta); LangSmith logo + Application sidebar = project-based.

Route all traces from a production checkout service to the correct agent and environment using environment variables — no code changes required.

```
$ export LANGSMITH_TRACING=true
export LANGSMITH_API_KEY=<your-api-key>
export LANGSMITH_AGENT_ID=checkout
export LANGSMITH_AGENT_ENVIRONMENT=production
```

Address a single decorated function to a specific agent and environment in Python, overriding any environment variables for that call.

```python
import langsmith as ls
from langsmith import traceable

@traceable(address=ls.address.agent("checkout", "production"))
def handle_order(order_id: str) -> dict:
    return {"order_id": order_id, "status": "charged"}
```

Address all traced calls within a block to a staging agent using `tracing_context`, useful when running integration tests against a non-production environment.

```python
from langsmith import traceable, tracing_context

@traceable
def handle_order(order_id: str) -> dict:
    return {"order_id": order_id, "status": "charged"}

with
tracing_context(address="lrn:agents/checkout/environments/staging"):
    handle_order("A-1")
```

Navigate to an agent's Overview to read its stable identifier — the value you need to route traces correctly without breaking existing trace history when you rename the agent.

📍 In LangSmith, open the agent switcher, select your agent, and go to its Overview page. Copy the value from the `ID` field — this is the immutable identifier also visible in the browser address bar.

Identify which workspace architecture you are on and navigate to the correct tracing container — agent-based or project-based.

📍 1. In LangSmith, look at the control at the top left. 2. If it shows your workspace name — you are in an agent-based workspace (beta). Traces are

organized by agent and environment (Production, Staging, Development, Local). 3. If it shows the LangSmith logo and the sidebar has an `Application` section with an application picker — you are in a project-based workspace. Traces are organized by tracing project.

— Full mechanism, env vars, code APIs, and UI navigation all given with runnable examples.

product docs

02 **Stricter permissions required for dataset exports** BREAKING

70

Dataset CSV and JSONL exports now require both dataset read and download permissions; API requests that previously succeeded with only dataset read permission now receive a forbidden response.

— Clear before/after behavior change, though no migration command given.

snapshot-20261003

THINNER COVERAGE BELOW

03 **Visible failure logging for Slack-triggered deep agents**

IMPROVED

45

Slack channel event failures occurring before reaching a Managed Deep Agent now surface in `mda` logs and the deployment logs UI instead of failing silently.

— Names the log surfaces but gives no exact command or UI path. snapshot-20261003

04 **IAM role support for ECR in self-hosted deployments** NEW

20

Self-hosted LangSmith deployments gain IAM role support for ECR, per the release notes; no further configuration detail was provided.

— Only named in passing with no mechanism or config shown. snapshot-20261003

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; API requests that previously succeeded with only dataset read permission now receive a forbidden response.

snapshot-20261003 NOTES ↗ LATEST STABLE

LangSmith adds `/v2` /threads/stats, ls_provider trace metadata, IAM role ECR support, and `LICENSE_EXPIRATION_WARNING_DAYS` for self-hosted operators.

└──▷ USE IT

Map a custom OpenAI-compatible model to provider-specific pricing and populate trace metadata through the LLM Gateway.

```json
{
  "model": "my-custom-gpt4",
  "base_url": "https://my-endpoint.example.com/v1",
  "ls_provider": "openai"
}
```

Configure a self-hosted LangSmith instance to warn operators 14 days before license expiration instead of the default 7.

```.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, including on self-hosted ClickHouse-backed deployments.

Slack channel event failures before reaching a Managed Deep Agent now surface in `mda` logs and the deployment logs UI instead of failing silently.

└──▷ BREAKING ON UPGRADE

! Dataset CSV and JSONL exports now require both dataset read and download permissions; API requests that previously succeeded with only dataset read permission now receive a forbidden response.

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261003 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

Adds Coding Agent Insights dashboards, SQL charts, Loop on dashboards, DPAPI-encrypted Windows credentials, and project group placement via `projectGroupName` / `project_group_name`.

Adds `%USERPROFILE%\.braintrust\auth\credentials.dpapi` DPAPI-encrypted credential storage on Windows so that `bt` profiles are shared between terminal sessions and packaged MSIX desktop apps.

Adds `projectGroupName` (TypeScript SDK v3.36.0+) and `project_group_name` (Python SDK v0.44.0+) to projects.create() so new projects are placed inside an existing project group; `bt functions push` also reads the setting when it auto-creates a missing project.

Adds `bt trace hook` exit-nonzero behaviour when it cannot capture an event — for example, when tracing settings are invalid or the background tracing process is unreachable.

Extends span customizers in the Python SDK v0.44.0 to run on all native SDK spans created with traced(), start_span(), and logger.log(), not only spans from integrations and wrappers.

Extends span customizers in the TypeScript SDK v3.36.0 to run on all native SDK spans created with traced(), startSpan(), and logger.log(), not only spans from instrumentation and wrappers.

+6 more

NEW INTEGRATION AgentScope TeamPipeline.reply_stream() tracing

UPDATED INTEGRATION Claude Agent SDK verbatim_prompts tracing

UPDATED INTEGRATION Claude Code subagent nesting in tracing

└─▷ BREAKING ON UPGRADE

! After downgrading from `bt` v0.22.2 to an earlier version on Windows, saved credentials in `credentials.dpapi` are not readable — you must log in again.

WAS THIS USEFUL? 👍 👎

agentacct

Feedback ⓘ v0.12.11 NOTES ↗ RANK

The agentacct dashboard records coding-agent work steps, tools, file changes, tests, time, and token costs locally.

> agentacct v0.12.11 tightens multi-account clarity by labeling every provider-limit reading with its account and source, and by making the terminal statusLine attribute itself correctly to the signed-in account.

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

01 **Account and source labels on provider-limit readings** IMPROVED 78

Every provider-limit reading in the macOS Usage page, the TUI, and `agentacct limits` is now labeled with its account and source — e.g. `CLI · you@example.com` or `desktop app · org 07c55a50` — so a second account's Weekly meter cannot be mistaken for the first account's.

— Names exact UI pages and CLI command with sample label format v0.12.11

02 **Account-aware statusLine attribution** IMPROVED 78

The terminal `statusLine` stream is now attributed to the account its config home is signed in as, reading the `oauthAccount` email and org UUID from that home's own `.claude.json` without opening credentials; a relocated spool claims no account rather than a guessed one.

— Explains precise mechanism and fallback behavior, no runnable command given v0.12.11

v0.12.11 NOTES ↗ LATEST STABLE

agentacct v0.12.11 labels every provider-limit reading with its account and source, eliminating cross-account meter confusion.

└──▷ **GET THIS VERSION** ▾

```
$ git clone --branch v0.12.11
https://github.com/mikehasa/agentacct.git
# already have the repo? check out this version:
$ git checkout v0.12.11
```

Labels each provider-limit reading in the macOS Usage page, the TUI, and `agentacct limits` with its account and source — e.g. `CLI · you@example.com` or `desktop app`

> `· org 07c55a50` — so a second account's Weekly meter cannot be mistaken for the first account's.
>
> Attributes the terminal `statusLine` stream to the account its config home is signed in as, reading `oauthAccount` email and org UUID from that home's own `.claude.json` without opening credentials; a relocated spool claims no account rather than a guessed one.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI/LLM Security

PipeLab Pipelock

Feedback ⓘ v3.6.0 NOTES ↗ RANK

Pipelock is an open-source AI agent firewall that scans MCP and HTTP egress for exfiltration, SSRF, and prompt injection and emits signed action receipts.

> Pipelock v3.6.0 is a dense release: it adds operator-facing containment controls (service publishing, display provisioning, VNC view/control, dry-run previews, loopback service declarations, workspace-statement verification) alongside per-process receipt chains, receipt-gated credential allows, GitLab token scanning, and redirect handling in the forward proxy, plus a long tail of breaking config-validation changes to blocklists, host patterns, TLS passthrough, DLP exemptions and temporary-exception expiry limits.

└──▷ WHAT SHIPPED · 26 FEATURES most completely described first ? what's the number?

01 Loopback service declarations for contained agents NEW

85

`containment.loopback_services` lets operators declare extra local destinations a contained agent may reach, each entry carrying `host`, `port`, `owner`, `reason` and `expiry`; the exemption is removed by a privileged timer at expiry and enforced via `contain verify`.

Declare a local service (e.g. a code-indexing daemon) that a contained agent is allowed to reach, with an expiry so the exemption is automatically removed.

yaml

```yaml
containment:
  loopback_services:
    - host: 127.0.0.1
      port: 7701
      owner: code-indexer
      reason: 'Agent requires local code-indexing service'
      expiry: '2026-12-31T00:00:00Z'
```

— Config fields and example shown, enforcement command named v3.6.0

02 VNC operator view and control of contained agents NEW

83

`pipelock contain view [--control]` lets an operator watch or take over a contained agent's screen with any VNC client through a caller-only local socket; view-only sessions cannot type or click, and at most one controller may hold the display at a time.

Watch a contained agent's screen live to audit its interactive behaviour, or pass `--control` to take over the session.

```
$ pipelock contain view --control
```

— Runnable command with concrete access-control behaviour v3.6.0

03 **Enforced max expiry for temporary exceptions** BREAKING

82

Temporary exception fields `request_body_scanning.sigv4_credential_routes[].expires`, `response_scanning.core_observe_exceptions[].expires`, `sandbox.best_effort_expiry` (30 days), `request_body_scanning.content_entropy_warn_routes[].expires`, `response_scanning.unscannable_passthrough[].expires` (90 days), `fetch_proxy.monitoring.path_entropy_exclusions[].expires` and `fetch_proxy.monitoring.query_entropy_param_exclusions[].expires` (180 days) now have enforced maximum expiry horizons, checked at load and reload; a config-sourced sandbox expiry must be an absolute RFC3339 time.

— Every field and its day limit named verbatim v3.6.0

04 **Dry-run preflight for contained sessions** NEW

81

`pipelock contain run --dry-run` runs preflight checks and prints the session contract — agent user, egress posture, capsule path, private `/tmp` state, and tools/workspace grants with expiry — without launching the agent.

Preview the full session contract for a contained agent before launch, catching expired grants in CI without starting the agent.

```
$ pipelock contain run --dry-run --config ./pipelock.yaml
```

— Runnable command printing a concrete contract v3.6.0

05 **Forward proxy returns redirects to the client** BREAKING

81

The absolute-URI forward proxy now returns allowed redirects (301, 302, 303, 307, 308) to the client instead of following them internally, so each hop is independently admitted, receipted and mediation-enveloped; a redirect whose `Location` carries a backslash, whitespace, control character or is duplicated is blocked with 403 and reason

`ambiguous redirect location`. The audit event message changes from `redirect followed` to `redirect observed`.

— Full before/after mechanism and status codes named v3.6.0

06 **Receipt-gated credential and issuer allows** NEW 76

With `flight_recorder.require_receipts`, a credential-audience, issuer-cookie or issuer-query allow now forwards only after its receipt is durably written; a failed write blocks the request with reason `receipt_emission_failed` and layer `credential_audience_receipt` or `issuer_allow_receipt`. More broadly, when MCP receipts are required, a request now forwards only after its receipt is durably written, with a failed or unsupported write blocking the request.

— Config key and exact reason/layer codes named v3.6.0

07 **Offline audit packets no longer auto-trusted** BREAKING 74

`audit-packet --offline` no longer reports a packet as trusted; the Go, Rust and TypeScript verifiers return `verdict: schema_checked_trust_unverified` with `trusted` and `valid` false and exit non-zero. `pipelock verify-receipt --clean-report` output now requires a `verification_mode` field (`pinned_provenance` or `unpinned_structural`).

— Exact verdict strings and required field named v3.6.0

08 **Deferred-resolution tracking for kill switch and gate** NEW 73

Activating the kill switch now blocks held MCP approvals that have not yet started sending, recording source `kill_switch` in the journal and layer `kill_switch` in the signed receipt; a call that previously forwarded after activation now resolves as blocked with error `-32004`. A held HTTP call refused by the upstream gate at release now records resolution source `upstream_contract` in deferred-resolution records.

— Exact field values named, no reproducible command v3.6.0

09 **Trusted hosts override for request-body scanning** BREAKING 73

`request_body_scanning.trusted_hosts` lets request-body injection and fully redacted critical credentials follow the configured action rather than a hard block for listed destinations; it is now the authoritative source for request-body trust, replacing reliance on `response_scanning.exempt_domains` to relax outbound request bodies.

— Config key and migration path from exempt_domains named v3.6.0

10 **Workspace change statement verification** NEW 69

`pipelock posture verify --workspace-statement` checks the signed workspace change statement produced after each contained session against its posture capsule, exiting 2 on an incomplete statement.

— Named command and exit code, no example given v3.6.0

11 **Default egress blocklist applied when unset** BREAKING

69

A config file that omits `fetch_proxy.monitoring.blocklist` now gets the shipped blocklist instead of an empty one; destinations on the shipped list that were reachable under 3.5.0 will now be blocked. Set `blocklist: []` to keep it off.

— Exact config key and opt-out value given v3.6.0

12 **Containment service exposure and private temp dirs** NEW

68

Containment gains `containment.published_services` to publish contained-agent services to the operator, `containment.display` (with `containment.display.geometry`) to provision a screen inside the namespace for a contained browser, and private `/tmp` and `/var/tmp` for contained agents through a transient systemd service that is verified before launch.

— Three config keys named but no worked example v3.6.0

13 **GitLab token scanning in DLP** NEW

68

GitLab `Private-Token` and `Job-Token` headers are now scanned for credentials by default, with `dlp.gitlab_hosts` available to declare self-managed GitLab hosts whose tokens are treated as going to their issuer.

— Config key and header names given, no example v3.6.0

14 **Named agent policy binds to identity only** BREAKING

65

Named agent policy (Enterprise) now follows bound identity only; a self-declared `X-Pipelock-Agent` header is recorded for attribution but receives the fallback policy, and session state folds to the client IP. `contain install` now refuses to swap in a core build that cannot enforce a named-agent config.

— Behaviour change explained, no migration command given v3.6.0

15 **Per-process receipt chains** BREAKING 65

Each Pipelock process now writes its own receipt chain; `verify-receipt --chain` and `evidence doctor` follow restart links between chains. Tools that previously read a single session file must now read the run chains instead.

— Commands named, migration note given v3.6.0

16 **Interception CA added to contained Chromium** NEW 60

`contain install` now adds the interception CA to the contained agent's Chromium certificate database, reported as probe 20 in verification output, with rollback removing only what install added.

— Mechanism and probe number named, no runnable example v3.6.0

THINNER COVERAGE BELOW

17 **Scan API can return a warn decision** BREAKING 59

The Scan API can now return `decision: "warn"` for a `tool_call` matching a warn-configured tool-policy rule, aligning with live MCP proxy behaviour; consumers that expected only `allow` or `deny` must now handle the new value.

— API field named, no endpoint path given v3.6.0

18 **Wildcard restriction on TLS passthrough domains** BREAKING 59

`tls_interception.passthrough_domains` now refuses a wildcard over a public suffix, including private-section suffixes shared by many tenants; operators must use exact hosts or intercept with a trusted local CA.

— Config key named with concrete restriction and workaround v3.6.0

19 **Host pattern validation across allow/deny/trust/bypass lists** 56

BREAKING

Host patterns in allow, deny, trust and bypass lists are now validated and refused if they carry a URL, `host:port`, fragment, interior wildcard, malformed label, or a trailing extra dot.

— Validation rules named but no affected config keys listed v3.6.0

20 **SSH public key reads exempted from credential rule** IMPROVED 56

Reading an SSH public key (`id_*.pub` under `.ssh`) no longer trips the built-in Credential File Access rule when the path is the call's only string value, across the MCP

proxy, `pipelock explain`, the Scan API and hook events.

— Exact path pattern and affected surfaces named v3.6.0

21 **Session-profiling volume spike ratio removed** DEPRECATED 52

`session_profiling.volume_spike_ratio` is removed; a config that still sets it, top-level or per agent, is refused at load, and the canonical policy hash changes as a result.

— Field removal named, no replacement guidance given v3.6.0

22 **Custom DLP patterns can't reuse core pattern exemptions** 52

BREAKING

A custom DLP pattern that reuses a core pattern name can no longer carry `exempt_domains`; config validation refuses it at startup and reload.

— Rule named but no example of affected config v3.6.0

23 **file_sentry block action refused on server listener** BREAKING 52

`pipelock run` now refuses `file_sentry.action: block` on the server listener; a reload requesting it is rejected atomically.

— Exact config value and command named v3.6.0

24 **GitHub release-download JWT egress allow** IMPROVED 52

The default canonical policy hash changes because a signed GitHub release-download JWT may now reach `release-assets.githubusercontent.com` in an `https` URL query; receipt consumers that pin the default hash will see a new value.

— Destination named but underlying rule change is thin v3.6.0

25 **Reload now waits for systemd verdict** IMPROVED 50

`systemctl reload pipelock` now waits for the reload verdict on systemd 253 and newer, using `Type=notify-reload`.

— Mechanism named but effect on operator workflow is thin v3.6.0

26 **Tool-policy rules default to patch_targets arg source** IMPROVED 43

Shipped presets now set `arg_source: patch_targets` on tool-policy rules; binaries older than 3.6.0 refuse to load them.

— Field value named but effect barely explained v3.6.0

╰──▷ BREAKING ON UPGRADE

! A config file that omits `fetch_proxy.monitoring.blocklist` now gets the shipped blocklist instead of an empty one; destinations on the shipped list that were reachable under 3.5.0 will now be blocked. Set `blocklist: []` to keep it off.

! The `request_body_scanning.trusted_hosts` list is now the authoritative source for request-body trust; operators who relied on `response_scanning.exempt_domains` to relax outbound request bodies must add those hosts to `request_body_scanning.trusted_hosts`.

! Host patterns in allow, deny, trust and bypass lists are now validated and refused if they carry a URL, `host:port`, fragment, interior wildcard, malformed label, or a trailing extra dot.

! `tls_interception.passthrough_domains` now refuses a wildcard over a public suffix (including private-section suffixes shared by many tenants); use exact hosts or intercept with a trusted local CA.

! Temporary exception expiry limits are now enforced at load and reload: `request_body_scanning.sigv4_credential_routes[].expires` and `response_scanning.core_observe_exceptions[].expires` (30 days), `sandbox.best_effort_expiry` (30 days), `request_body_scanning.content_entropy_warn_routes[].expires` and `response_scanning.unscannable_passthrough[].expires` (90 days), `fetch_proxy.monitoring.path_entropy_exclusions[].expires` and `fetch_proxy.monitoring.query_entropy_param_exclusions[].expires` (180 days). A config-sourced sandbox expiry must be an absolute RFC3339 time.

! `session_profiling.volume_spike_ratio` is removed; a config that still sets it (top-level or per agent) is refused at load. The canonical policy hash changes as a result.

! A custom DLP pattern that reuses a core pattern name can no longer carry `exempt_domains`; config validation refuses it at startup and reload.

! `pipelock run` now refuses `file_sentry.action: block` on the server listener; a reload requesting it is rejected atomically.

! Named agent policy (Enterprise) now follows bound identity only; a self-declared `X-Pipelock-Agent` header is recorded for attribution but receives the fallback policy, and session state folds to the client IP.

! `contain install` now refuses to swap in a core build that cannot enforce a named-agent config.

! Each Pipelock process now writes its own receipt chain; tools that read a single session file must read the run chains instead. `verify-receipt --chain` and `evidence doctor` follow the links.

! Activating the kill switch now blocks held MCP approvals that have not yet started sending; a held call that previously forwarded after activation now resolves as blocked with error `-32004`.

! The default canonical policy hash changes because a signed GitHub release-download JWT may now reach `release-assets.githubusercontent.com` in an `https` URL query; receipt consumers that pin the default hash will see a new value.

! The Scan API can now return `decision: "warn"`; consumers that expected only `allow` or `deny` must handle the new value.

- ! Shipped presets now set `arg_source: patch_targets` on tool-policy rules; binaries older than 3.6.0 refuse to load them.
- ! Contained hosts must run `pipelock contain install` with the new binary to adopt the private network namespace, private temporary directories, `0600` config mode and refreshed CA export.
- ! The absolute-URI forward proxy now hands allowed redirects back to the client instead of following them; clients that do not follow redirects themselves will no longer have hops followed on their behalf.
- ! The `redirect` audit event message changes from `redirect followed` to `redirect observed`; anything matching on the old message text must be updated.
- ! With `flight_recorder.require_receipts` enabled, a credential-audience, issuer-cookie or issuer-query allow is now blocked if its receipt write fails, with reason `receipt_emission_failed`.
- ! When MCP receipts are required, a request now forwards only after its receipt is durably written; a failed or unsupported receipt write blocks the request.
- ! `audit-packet --offline` no longer reports a packet as trusted; the Go, Rust and TypeScript verifiers return `verdict: schema_checked_trust_unverified` with `trusted` and `valid` false and exit non-zero. `pipelock verify-receipt --clean-report` output now requires a `verification_mode` field (`pinned_provenance` or `unpinned_structural`).
- ! GitLab `Private-Token` and `Job-Token` headers are now scanned for credentials by default; self-managed GitLab operators must add their host to `dlp.gitlab_hosts` so tokens are treated as going to their issuer.
- ! Reading an SSH public key (`id_*.pub` under `.ssh`) no longer trips the built-in Credential File Access rule when the path is the call's only string value on the MCP proxy, `pipelock explain`, the Scan API and hook events.

v3.6.0 NOTES ↗ LATEST STABLE

Pipelock v3.6.0 adds contained-agent network namespaces, VNC operator control, loopback service declarations, and session workspace statements.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v3.6.0
https://github.com/luckyPipewrench/pipelock.git
# already have the repo? check out this version:
$ git checkout v3.6.0
```

└──▷ TRY IT

Preview the full session contract for a contained agent before launch, catching expired grants in CI without starting the agent.

```
$ pipelock contain run --dry-run --config ./pipelock.yaml
```

Declare a local service (e.g. a code-indexing daemon) that a contained agent is allowed to reach, with an expiry so the exemption is automatically removed.

pipelock.yaml

```
containment:
  loopback_services:
    - host: 127.0.0.1
      port: 7701
      owner: code-indexer
      reason: 'Agent requires local code-indexing service'
      expiry: '2026-12-31T00:00:00Z'
```

Watch a contained agent's screen live to audit its interactive behaviour, or pass `--control` to take over the session.

```
$ pipelock contain view --control
```

Adds `containment.published_services` to publish contained-agent services to the operator and `containment.display` (with `containment.display.geometry`) to provision a screen inside the namespace for a contained browser.

Adds `containment.loopback_services` to declare extra local destinations for contained agents, each entry carrying host, port, owner, reason and expiry, with privileged timer removal on expiry and `contain verify` enforcement.

Adds `pipelock contain view [--control]` to let an operator watch or take over a contained agent's screen with any VNC client through a caller-only local socket; view-only sessions cannot type or click, and at most one controller may hold the display at a time.

Adds `pipelock contain run --dry-run` to run preflight checks and print the session contract (agent user, egress posture, capsule path, private `/tmp` state, tools and workspace grants with expiry) without launching the agent.

Adds `pipelock posture verify --workspace-statement` to check the signed workspace change statement produced after each contained session against its posture capsule, exiting 2 on an incomplete statement.

+13 more

└──▷ **BREAKING ON UPGRADE**

- ! A config file that omits `fetch_proxy.monitoring.blocklist` now gets the shipped blocklist instead of an empty one; destinations on the shipped list that were reachable under 3.5.0 will now be blocked. Set `blocklist: []` to keep it off.
- ! The `request_body_scanning.trusted_hosts` list is now the authoritative source for request-body trust; operators who relied on `response_scanning.exempt_domains` to relax outbound request bodies must add those hosts to `request_body_scanning.trusted_hosts`.
- ! Host patterns in allow, deny, trust and bypass lists are now validated and refused if they carry a URL, `host:port`, fragment, interior wildcard, malformed label, or a trailing extra dot.
- ! `tls_interception.passthrough_domains` now refuses a wildcard over a public suffix (including private-section suffixes shared by many tenants); use exact hosts or intercept with a trusted local CA.
- ! Temporary exception expiry limits are now enforced at load and reload: `request_body_scanning.sigv4_credential_routes[].expires` and `response_scanning.core_observe_exceptions[].expires` (30 days), `sandbox.best_effort_expiry` (30 days), `request_body_scanning.content_entropy_warn_routes[].expires` and `response_scanning.unscannable_passthrough[].expires` (90 days), `fetch_proxy.monitoring.path_entropy_exclusions[].expires` and `fetch_proxy.monitoring.query_entropy_param_exclusions[].expires` (180 days). A config-sourced sandbox expiry must be an absolute RFC3339 time.
- ! `session_profiling.volume_spike_ratio` is removed; a config that still sets it (top-level or per agent) is refused at load. The canonical policy hash changes as a result.
- ! A custom DLP pattern that reuses a core pattern name can no longer carry `exempt_domains`; config validation refuses it at startup and reload.
- ! `pipelock run` now refuses `file_sentry.action: block` on the server listener; a reload requesting it is rejected atomically.
- ! Named agent policy (Enterprise) now follows bound identity only; a self-declared `X-Pipelock-Agent` header is recorded for attribution but receives the fallback policy, and session state folds to the client IP.
- ! `contain install` now refuses to swap in a core build that cannot enforce a named-agent config.
- ! Each Pipelock process now writes its own receipt chain; tools that read a single session file must read the run chains instead. `verify-receipt --chain` and `evidence doctor` follow the links.
- ! Activating the kill switch now blocks held MCP approvals that have not yet started sending; a held call that previously forwarded after activation now resolves as blocked with error `-32004`.

- ! The default canonical policy hash changes because a signed GitHub release-download JWT may now reach `release-assets.githubusercontent.com` in an `https` URL query; receipt consumers that pin the default hash will see a new value.
- ! The Scan API can now return `decision: "warn"`; consumers that expected only `allow` or `deny` must handle the new value.
- ! Shipped presets now set `arg_source: patch_targets` on tool-policy rules; binaries older than 3.6.0 refuse to load them.
- ! Contained hosts must run `pipelock contain install` with the new binary to adopt the private network namespace, private temporary directories, `0600` config mode and refreshed CA export.
- ! The absolute-URI forward proxy now hands allowed redirects back to the client instead of following them; clients that do not follow redirects themselves will no longer have hops followed on their behalf.
- ! The `redirect` audit event message changes from `redirect followed` to `redirect observed`; anything matching on the old message text must be updated.
- ! With `flight_recorder.require_receipts` enabled, a credential-audience, issuer-cookie or issuer-query allow is now blocked if its receipt write fails, with reason `receipt_emission_failed`.
- ! When MCP receipts are required, a request now forwards only after its receipt is durably written; a failed or unsupported receipt write blocks the request.
- ! `audit-packet --offline` no longer reports a packet as trusted; the Go, Rust and TypeScript verifiers return `verdict: schema_checked_trust_unverified` with `trusted` and `valid` false and exit non-zero. `pipelock verify-receipt --clean-report` output now requires a `verification_mode` field (`pinned_provenance` or `unpinned_structural`).
- ! GitLab `Private-Token` and `Job-Token` headers are now scanned for credentials by default; self-managed GitLab operators must add their host to `dlp.gitlab_hosts` so tokens are treated as going to their issuer.
- ! Reading an SSH public key (`id_*.pub` under `.ssh`) no longer trips the built-in Credential File Access rule when the path is the call's only string value on the MCP proxy, `pipelock explain`, the Scan API and hook events.

WAS THIS USEFUL? 👍 👎

ai-safe2-framework

Feedback ⓘ 2026-10-02_AISM_Scoring_Update

NOTES ↗ RANK

Ai-Safe2-Framework provides governance, risk, and compliance controls for securing agentic AI systems and non-human identities.

> AI SAFE² (now branded AI Sovereign Maturity, AISM) adds two new scoring layers — Technology Contribution Profile and Evidence Assurance — plus a reusable Technology Card template, validation/incident documentation, and example reviews, while clarifying how these relate to the existing Sovereignty Score.

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

01 **Validation and incident evidence guidance** NEW 75

Adds validation and incident guidance covering clean/attacked pair planning, side-effect and bypass tests, policy-integrity checks, reconstruction, and notification ownership across `TECHNOLOGY-VALIDATION.md` and `INCIDENT-EVIDENCE.md`.

– Names both source files and the concrete test types covered.

2026-10-02_AISM_Scoring_Update

02 **Reusable Technology Card template** NEW 60

Adds a reusable Technology Card template to capture version, scope, controls, mechanisms, evidence, failure behavior, retained authority, and adoption conditions for any reviewed artifact.

– Names the template and its exact fields. 2026-10-02_AISM_Scoring_Update

THINNER COVERAGE BELOW

03 **Technology Contribution Profile v1.0** NEW 55

Introduces `AISM Technology Contribution Profile v1.0` to structure reviews of what an artifact addresses, implements, enforces, validates, evidences, or challenges within a stated boundary.

– Names the framework but no worked scoring procedure given.

2026-10-02_AISM_Scoring_Update

04 **Technology Contribution review examples** NEW 55

Provides nine source-attributed Technology Contribution review candidate examples in `research/technology-contribution-examples.md` to frame scoped acceptance work without adopting unsupported scores.

– Gives exact file path and count, no content detail. 2026-10-02_AISM_Scoring_Update

05 **Evidence Assurance v1.0 claim scoring** NEW 45

Introduces `AI SAFE² Evidence Assurance v1.0` to attribute evidence strength, origin, independence, and limitations to each individual claim.

— Describes purpose but not how attribution is scored. 2026-10-02_AISM_Scoring_Update

06 **Sovereignty Score separation clarified** IMPROVED 40

Clarifies the existing AISM Sovereignty Score to keep organizational maturity across five pillars and six dimensions explicitly separate from technology contribution and claim-level evidence assurance.

— States the distinction but not what changed mechanically.

2026-10-02_AISM_Scoring_Update

07 **AISM rebrand and repository navigation** NEW 40

Adopts the name AI Sovereign Maturity (AISM) Model and adds dated AISM change summaries with repository-wide navigation connecting methods to pillars, implementation guidance, research, Challenge Lab, and agent discovery.

— Names linked areas but no specific navigation path given.

2026-10-02_AISM_Scoring_Update

2026-10-02_AISM_Scoring_Update NOTES ↗ LATEST STABLE

AI SAFE² adds Technology Contribution Profile v1.0, Evidence Assurance v1.0, and a reusable Technology Card to separate org maturity from scoped AI tool claims.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 2026-10-02_AISM_Scoring_Update
https://github.com/CyberStrategyInstitute/ai-safe2-framework.git
# already have the repo? check out this version:
$ git checkout 2026-10-02_AISM_Scoring_Update
```

Introduces AISM Technology Contribution Profile v1.0 to structure reviews of what an artifact addresses, implements, enforces, validates, evidences, or challenges within a stated boundary.

Introduces AI SAFE² Evidence Assurance v1.0 to attribute evidence strength, origin, independence, and limitations to each individual claim.

Adds a reusable Technology Card template to capture version, scope, controls, mechanisms, evidence, failure behavior, retained authority, and adoption conditions for any reviewed artifact.

Adds validation and incident guidance covering clean/attacked pair planning, side-effect and bypass tests, policy-integrity checks, reconstruction, and notification ownership

> across `TECHNOLOGY-VALIDATION.md` and `INCIDENT-EVIDENCE.md`.
>
> Provides nine source-attributed Technology Contribution review candidate examples in `research/technology-contribution-examples.md` to frame scoped acceptance work without adopting unsupported scores.
>
> +2 more

WAS THIS USEFUL? 👍 👎

Docker Sandboxes

Feedback ⓘ announced · Oct 1, 2026 BLOG ↗

RANK

Docker Sandboxes runs AI agents in isolated Docker environments with controlled filesystem, network, and tool access.

> Docker launched Cloud Sandboxes, letting AI agent tasks run in isolated microVMs on Docker-managed cloud compute via the `sbx` CLI, and introduced the open Sandbox Kit specification along with `sbxenv` config files and MCP tool support inside sandboxes.

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

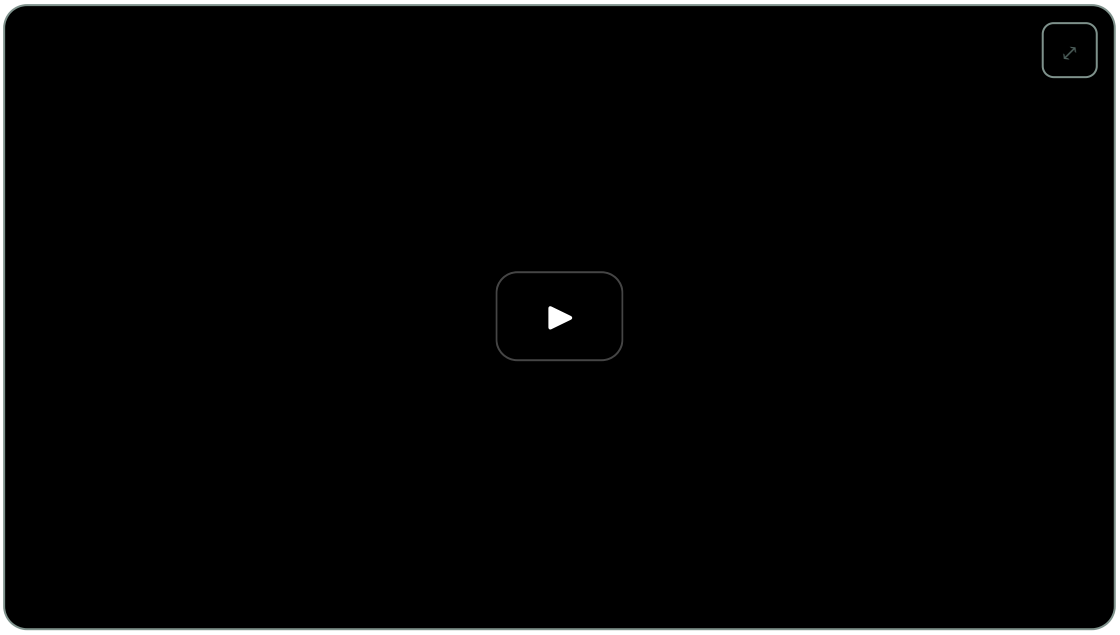
01 **Docker Sandbox Kit specification** NEW 80

The Docker Sandbox Kit is an OCI image format that packages an agent, its tools, and access declarations, letting teams build, push, pull, scan, and pin agent environments by digest. The spec is published under Apache 2.0 with a commitment to bring it to the CNCF for neutral governance, and Docker Sandboxes is the first runtime to implement it.

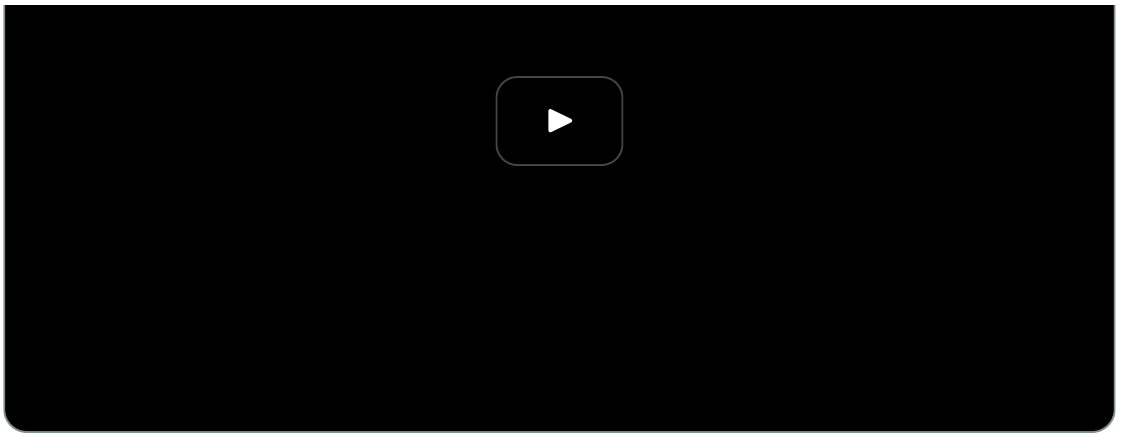
— Concrete format, verbs and licensing named, no sample command launch-20261001-8151baca

02 **Cloud sandbox migration via sbx CLI** NEW 60

Using the `sbx` CLI, a sandbox's filesystem can move from a local machine to Docker-managed cloud compute mid-task, letting longer agent runs continue after a laptop is closed.









— Names the `sbx` CLI and mechanism but no exact commands launch-20261001-8151baca

THINNER COVERAGE BELOW

03 **sbxenv configuration files** NEW 55

`sbxenv` files describe a set of sandbox environments as configuration, enabling reproducible AI-ready developer environments across machines.

— Names config file but no schema or example launch-20261001-8151baca

04 **MCP tool support in sandboxes** NEW 40

Sandboxes support tools connected through the Model Context Protocol (MCP), including network rules and Kit packaging for agent workflows.

— Mentions MCP and network rules only in brief prose launch-20261001-8151baca

announced · Oct 1, 2026 BLOG ↗

Docker Cloud Sandboxes launch: run agent tasks in isolated microVMs on Docker-managed cloud compute using the `sbx` CLI

> Supports moving a sandbox's filesystem from a local machine to the cloud mid-task, allowing longer agent runs to continue after closing a laptop.
>
> Introduces the Docker Sandbox Kit specification — an OCI image format containing the agent, its tools, and access declarations — enabling teams to build, push, pull, scan, and pin agent environments by digest.
>
> Introduces `sbxenv` files to describe a set of sandbox environments as configuration, enabling reproducible AI-ready developer environments across machines.
>
> Supports tools connected through Model Context Protocol (MCP) inside sandboxes, including network rules and Kit packaging for agent workflows.
>
> Publishes the Sandbox Kit specification under Apache 2.0 with a commitment to bring it to the CNCF for neutral governance; Docker Sandboxes is the first runtime to implement it.

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Open-Weight Models

DeepSeek V4 Flash 0731 Feedback ⓘ price cut · Oct 2, 2026

MODEL ↗

DeepSeek V4 Flash 0731 is a chat model from DeepSeek.

Price cut on OpenRouter (deepseek/deepseek-v4-flash): $0.047 → $0.028 in, $0.0941 → $0.056 out per million tokens (-40%), first seen 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 4.5 Feedback ⓘ retirement scheduled · Oct 2, 2026

MODEL ↗

GLM 4.5 is a chat model from Z.ai.

OpenRouter lists a retirement date of 31 Dec 2026 for z-ai/glm-4.5, first seen 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 4.7 Feedback ⓘ retirement scheduled · Oct 2, 2026

MODEL ↗

GLM 4.7 is a chat model from Z.ai.

OpenRouter lists a retirement date of 31 Dec 2026 for z-ai/glm-4.7, first seen 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.2 Feedback ⓘ price increase · Oct 2, 2026 MODEL ↗

GLM 5.2 is a chat model from Z.ai.

Price increase on OpenRouter (z-ai/glm-5.2): $0.6496 → $0.41 in, $2.0416 → $3.99 out per million tokens (+95%), first seen 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.3

Feedback ⓘ context change · Oct 2, 2026 MODEL ↗

GLM 5.3 is a reasoning model from Z.ai.

Context window shrank from 1311K to 1049K tokens on OpenRouter (z-ai/glm-5.3), first seen 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

Moonshot AI Kimi K3

Feedback ⓘ price cut · Oct 2, 2026

MODEL ↗

Kimi K3 is a multimodal model from Moonshot AI.

Price cut on OpenRouter (moonshotai/kimi-k3): $3.00 → $2.70 in, $15.00 → $13.50 out per million tokens (-10%), first seen 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

NVIDIA Nemotron 3 Ultra

Feedback ⓘ

price cut · Oct 2, 2026 MODEL ↗

Nemotron 3 Ultra is a chat model from NVIDIA.

Price cut on OpenRouter (nvidia/nemotron-3-ultra-550b-a55b): $0.60 → $0.50 in, $2.40 → $2.20 out per million tokens (-17%), first seen 2 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Models

Feedback ⓘ qwen-image-2.1-pro NOTES ↗

RANK

Qwen is Alibaba's language and multimodal model family, including coding, vision, image, speech, and reasoning models.

> Alibaba released Qwen-Image-2.1-Pro, a unified 7B-parameter model that merges text-to-image generation and image editing with native transparent image support.

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

01 **Qwen-Image-2.1-Pro unified generation and editing model** NEW

65

Qwen-Image-2.1-Pro is a single 7B-parameter model that unifies text-to-image generation and image editing, eliminating the need for separate models per task. It supports native transparent image generation and editing, including transparent layer editing and subject extraction from photos, and accepts up to 10 reference images for versatile editing tasks including local edits, people and product preservation, while keeping a lightweight architecture that balances image quality and computational cost.

— Names model, parameter count, and capabilities but no API or usage path

qwen-image-2.1-pro

qwen-image-2.1-pro NOTES ↗ LATEST STABLE

Qwen-Image-2.1-Pro unifies text-to-image generation and image editing in a single 7B-parameter model with native transparent image support.

Supports native transparent image generation and editing, including transparent layer editing and subject extraction from photos.

Unifies text-to-image generation and image editing in a single model, eliminating the need for separate models per task.

Accepts up to 10 reference images for versatile editing tasks including local edits, people and product preservation.

Delivers a lightweight 7B-parameter visual generation architecture balancing image quality and computational cost.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-35B-A3B

Feedback

price cut · Oct 2, 2026 MODEL ↗

Qwen3.5-35B-A3B is a multimodal model from Alibaba.

Price cut on OpenRouter (qwen/qwen3.5-35b-a3b): $0.1625 → $0.15 in, $1.30 → $1.00 out per million tokens (-23%), first seen 2 Oct 2026.

WAS THIS USEFUL?

Ai2 Models

Feedback ⓘ announced · Oct 2, 2026 BLOG ↗ RANK

Ai2 develops the Olmo, Molmo, Tulu, and OlmoEarth families of open language and multimodal models.

> Ai2 open-sourced AstaBrief 8B, a cited-report-generation model that now powers a new Fast mode in Asta, alongside its training data and a local-report-generation workflow.

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

01 **AstaBrief 8B cited report generation model** NEW 60

Ai2 released AstaBrief 8B, an open-weights model that generates cited scientific reports from a research question and retrieved literature excerpts in a single pass; it now powers Asta's Generate a report feature as Fast mode. Ai2 also published the AstaBrief training data alongside the model weights for researchers to study, reproduce, and build on, and released an example workflow for generating reports from researchers' own PDFs, enabling local report generation on institutional infrastructure.

— Names model, size, feature mode, but no benchmarks or exact steps.

launch-20261002-c8bbc3dc

> announced · Oct 2, 2026 BLOG ↗
>
> Ai2 open-sources AstaBrief 8B, a fast cited-report-generation model now powering Asta's Fast mode
>
> Releases AstaBrief 8B, an open-weights model that generates cited scientific reports from a research question and retrieved literature excerpts in a single pass, now available in Asta's Generate a report feature as Fast mode.
>
> Publishes the AstaBrief training data alongside the model weights so researchers can study, reproduce, and build on the report-generation approach.
>
> Releases an example workflow for generating reports from researchers' own PDFs, enabling local report generation on institutional infrastructure.

WAS THIS USEFUL? 👍 👎

OTHER / UNCATEGORIZED

◆ UNCATEGORIZED

Foremerge

Feedback ⓘ v0.5.1 NOTES ↗ RANK

Catch intent conflicts before code conflicts. The open-source coordination protocol for coding agents, built above Git.

> Foremerge v0.5.1 adds npm as an install channel for the coordination protocol and makes `foremerge mcp` handle handshakes gracefully outside a Git repository.

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

01 **npm install channel and launcher detection** NEW 80

Foremerge can now be installed via `npm install -g foremerge`, which selects one of five platform-specific binary packages with no downloads, compile step, or install scripts. `foremerge doctor` recognises the npm launcher (`foremerge.cmd` on Windows) via `npm_launcher: true` and no longer flags it as a duplicate installation, also detecting the shim in a global prefix, `node_modules\.bin`, and an `npx` run.

Install Foremerge globally via npm on any supported platform — no Rust toolchain or shell installer required.

```
$ npm install -g foremerge
```

— Names exact command, flag output, and detection paths v0.5.1

02 **MCP handshake outside a Git repository** IMPROVED 78

`foremerge mcp` started outside a Git repository (without `--database`) now completes the MCP handshake instead of closing the connection, answering `initialize` and `tools/list` as an unavailable server with `isError: true` and codes `NOT_A_REPOSITORY`, `GIT_UNAVAILABLE`, or `REPOSITORY_UNREADABLE`, along with a remedy naming `--cwd`.

— Names flags, error codes and behaviour change precisely v0.5.1

v0.5.1 NOTES ↗ LATEST STABLE

Foremerge v0.5.1 adds npm as an install channel and makes `foremerge mcp` answer MCP handshakes outside a Git repository.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v0.5.1 https://github.com/naw103/foremerge.git
# already have the repo? check out this version:
$ git checkout v0.5.1
```

└──▷ **TRY IT**

Install Foremerge globally via npm on any supported platform — no Rust toolchain or shell installer required.

```
$ npm install -g foremerge
```

Installs via npm (`npm install -g foremerge`), selecting one of five platform-specific binary packages — no downloads, no compile step, no install scripts.

`foremerge mcp` started outside a Git repository (without `--database`) now completes the MCP handshake instead of closing the connection: it answers `initialize` and `tools/list` as an unavailable server, returning `isError: true` with codes `NOT_A_REPOSITORY` , `GIT_UNAVAILABLE` , or `REPOSITORY_UNREADABLE` and a remedy naming `--cwd` .

`foremerge doctor` recognises the npm launcher (`foremerge.cmd` on Windows) with `npm_launcher: true` and does not warn about it as a duplicate installation; it also recognises the shim in a global prefix, `node_modules\.bin` , and an `npx` run.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 3, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe | Feedback on this issue