THE AI TOOLCHAIN

NO. 023

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 11, 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 293 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 46 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: Pinecone makes document-shaped retrieval indexes explicit instead of convention-driven;

vim ? help

Braintrust uses AI to surface and debug agent failure patterns; Arcade gives every agent tool call a filterable audit trail instead of leaving teams to reconstruct runs from logs.

BUILD

Build retrieval indexes around document schemas instead of glue code conventions

SchemaBuilder and document-schema indexes make the shape of indexed documents part of the index contract, while metadata indexing controls let teams decide what is searchable up front. That reduces the hidden coupling between ingestion code, filters, and retrieval behavior that usually lives in notebooks or tribal knowledge.

Pinecone

EVALUATE

Find recurring agent failures without hand-reading every trace

AI-driven Patterns and Debugger group and diagnose failures across agent runs, turning trace review from a manual sampling exercise into a failure-discovery workflow. Teams can spend review time on the root causes Braintrust surfaces instead of grepping logs for examples.

Braintrust

OPERATE

Audit each agent tool call as a production event

Execution Tool Logs record per-run tool inputs and outputs with filtering, RBAC, and an API. When an agent takes a bad action or hits an unexpected integration result, operators can inspect the exact call path without rebuilding it from app logs and provider traces.

Arcade

DEPLOY

Run agents on durable session and job primitives instead of ad hoc alarm loops

Cloudflare Agents moved conversation state, prompt context, streams, and lifecycle jobs onto new durable primitives; OpenRath added a durable multi-agent execution framework with an Effect Ledger and Durable Interrupts. Both reduce the class of failures where long-running agent work disappears, duplicates, or resumes without a reliable execution record.

Cloudflare Agents · OpenRath

Does **GPT-5.5** read better? 👍 👎

BUILD

◆ AI Coding Agents

Warp

Feedback ⓘ 1 RELEASE · SEEN 2026-09-11 NOTES ↗ RANK

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

> Warp's latest updates extend team-scoping across the Oz CLI and environment pickers, add first-class Grok Build support, and improve fzf-based file and history search handoff.

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

01 Team-scoped Oz CLI and environment selection IMPROVED

85

Adds team selection to `oz model list` (for team-specific model catalogs), `oz run list`, runner listing and name-based updates, and the API key `list`, `create`, and `expire` commands in the Oz CLI, all of which now respect the selected team. Oz agent catalogs and environment lists/interactive pickers also follow active team selection, with environment pickers showing both personal environments and environments for the selected window team.

— Names every affected command but gives no runnable example

2026.09.09 (v0.2026.09.09.08.26)

02 fzf and Fish shell search handoff IMPROVED

80

Extends shell widget handoff to fzf's `ctrl-t` file search, alongside the existing `ctrl-r` history-search handoff; Fish's `ctrl-r` now hands off to PatrickF1/fzf.fish history search, not only junegunn fzf.

Fuzzy-search files in the current directory from the terminal without leaving Warp, using the new fzf ctrl-t handoff.

> 📍 Press ctrl-t in a Warp terminal session to invoke fzf file search and insert the selected path into the prompt.

— Names exact keybindings and includes a runnable example

2026.09.09 (v0.2026.09.09.08.26)

THINNER COVERAGE BELOW

03 **First-class Grok Build support** NEW

55

Warp adds first-class support for Grok Build (SpaceX AI Grok): it detects Grok Build sessions, adds branded chrome in the footer, and enables rich input mode.

– Describes behavior but requires no user action to trigger

2026.09.09 (v0.2026.09.09.08.26)

04 **Editable keybindings for Attach file and new window** IMPROVED

55

Attach file is now a Command Palette action with an editable, unbound-by-default keybinding, and Create New Window is now an editable keyboard shortcut that can be remapped or unbound.

– Names exact actions but not their default keybindings 2026.09.09 (v0.2026.09.09.08.26)

05 **Refreshed Entra credentials for Azure DevOps writes** IMPROVED

35

Oz agents can now use refreshed Entra credentials to write to Azure DevOps.

– Single sentence with no mechanism or scope detail product docs

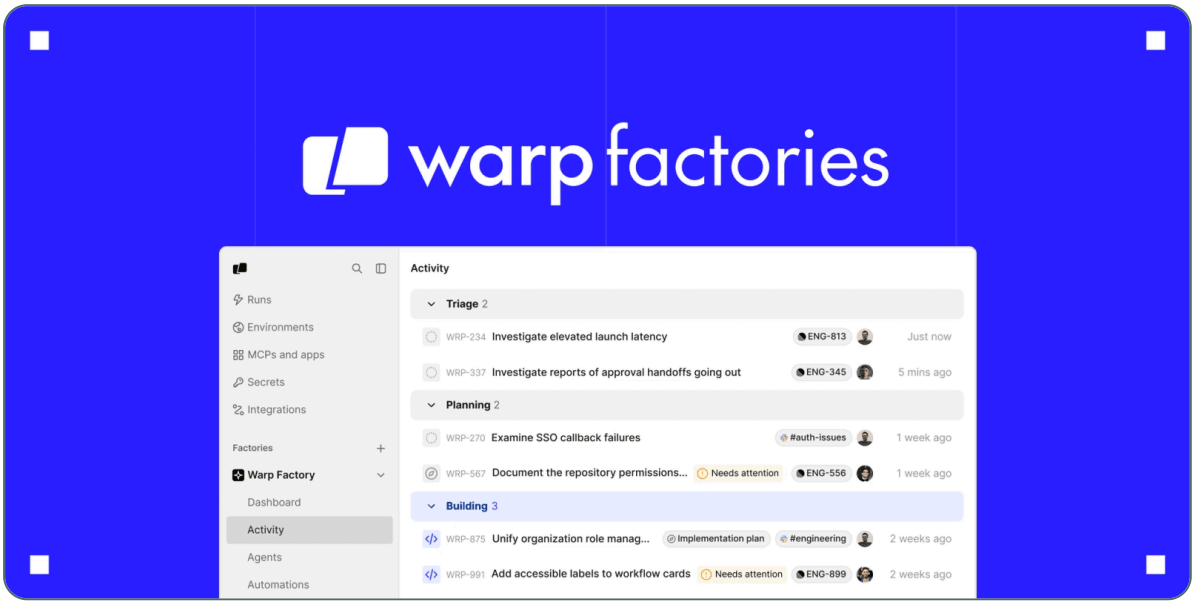
06 **Automatic Factory MCP access for Claude Code and Codex** NEW

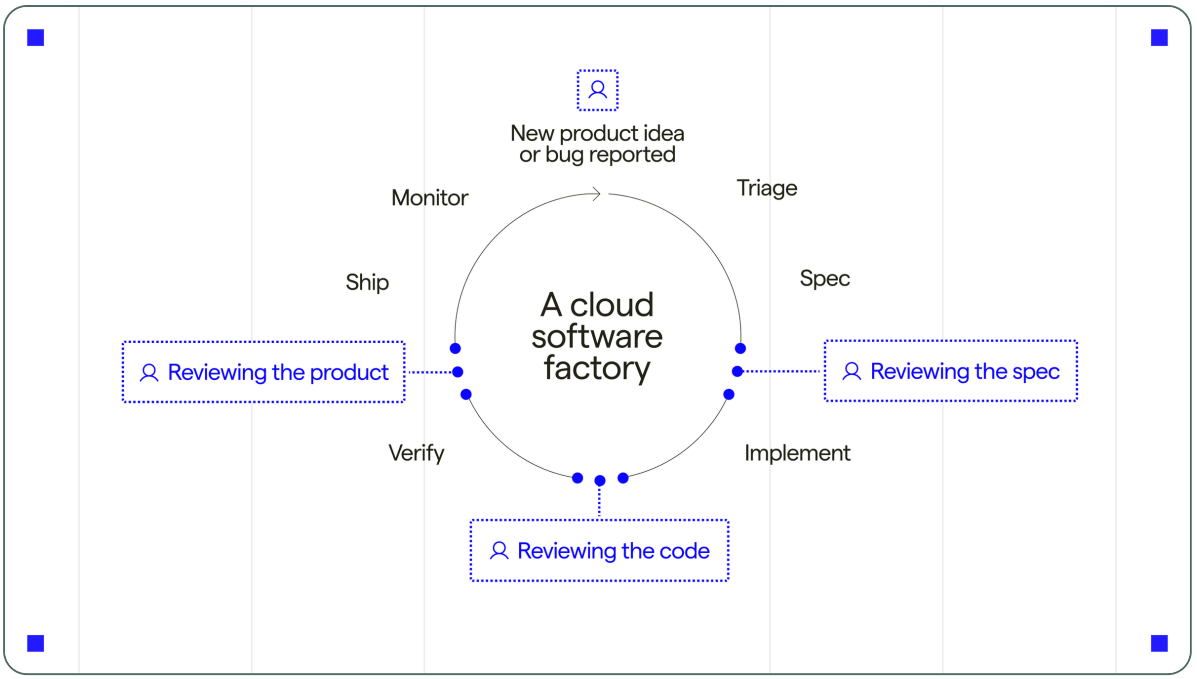
35

Claude Code and Codex runs can now access Factory MCP automatically.

– One-line mention with no mechanism or setup detail 2026.09.09 (v0.2026.09.09.08.26)

└──▷ ALSO FROM THESE RELEASES





[software factories as code]

```yaml
# factory.yaml
schemaVersion: v1alpha1
name: pr-review
repositories:
    owner: acme
    name: web
agentDefaults:
  model: claude-5-fable-high

# agents/foreman/agent.md
agentType: FOREMAN

# agents/review/agent.md
agentType: REVIEW
model: glm-5.2-fireworks

# automations/on-pr/automation.md
agent: review
triggers:
  - provider: github
    event: pull_request_ready

```

WAS THIS USEFUL?

StackBlitz bolt.new

Feedback ⓘ 1 RELEASE · SEEN 2026-09-11

NOTES ↗ RANK

Bolt.new is an AI-powered web development platform that builds websites and applications from prompts.

> Bolt.new launched Bolt Slides, a full slide-deck builder inside bolt.new with in-preview visual editing, rich speaker notes, a speaker view for presenting, slide reordering, and PDF/JSON export.

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

01 **Select tool for in-preview visual edits** NEW 65

Adds a Select tool in the chatbox for direct visual edits — text, colors, font size, formatting, and alignment — made right in the project preview without prompting. Edits are batched above the chatbox and only consume tokens when the user clicks Save changes, making preview edits free on all plans.

Iterate on a project's look and feel without spending tokens — use the Select tool to batch visual changes and only commit when satisfied.

> 📍 In the chatbox, activate the Select tool, click elements in the preview to edit text, colors, or font size, then click Save changes when done.

— Explains mechanism and token behaviour with exact UI steps snapshot-20260911

02 **Speaker view for presentations** NEW 60

Adds speaker view to Bolt Slides presentations: your tab shows speaker notes, the upcoming slide, and a timer, while the audience tab displays a fullscreen view of the slides.

Present to an audience while keeping notes and a timer visible only to you.

> 📍 In your Bolt Slides project, click Present, then choose Speaker view — your tab shows notes, the upcoming slide, and a timer, while the audience tab displays fullscreen slides.

— Describes exact tab layout with a concrete Present > Speaker view path

snapshot-20260911

THINNER COVERAGE BELOW

03 **PDF and JSON export for Slides decks** NEW 55

Exports Bolt Slides decks as PDF or JSON directly from the slide editor's Download menu, letting users share a file or import the JSON into another Bolt project to reuse its content.

Export a finished Slides deck as JSON to reuse its content in another Bolt project.

> 📍 In your Bolt Slides project, open the Download menu and select `JSON` to save the deck file locally.

Export a finished Bolt Slides deck as JSON to reuse slide content in another Bolt project.

> 📍 In your Slides project, open the deck, then select Download › JSON to save the deck file locally.

— Names both export formats and the exact menu path `snapshot-20260911`

---

04 **Bolt Slides deck builder** NEW 40

Adds Bolt Slides, enabling users to build, customize, share, and present slide decks directly within bolt.new.

When you need to generate and present a slide deck without leaving bolt.new.

> 📍 In bolt.new, open a project and select Slides from the available use cases to build, customize, and share your presentation.

— Names the product area and entry point, no deeper mechanism given `product docs`

---

05 **Rich-text speaker notes editor** NEW 35

Adds a speaker notes editor to Bolt Slides with rich text support including bold, italic, headings, and lists.

— Lists formatting options but gives no usage steps `snapshot-20260911`

---

06 **Slide reorder, duplicate and delete controls** NEW 35

Adds slide organization controls — reorder, duplicate, or delete slides — from the side panel or grid overview.

— Names the controls and UI location but no worked example `snapshot-20260911`

---

07 **Existing decks excluded from new Slides features** BREAKING 20

Existing Bolt Slides decks retain the controls they had at creation time and do not receive the new editing and export features added in this release.

— States the impact but offers no migration or workaround steps `snapshot-20260911`

└──▷ **BREAKING ON UPGRADE**

! Existing Bolt Slides decks retain the controls they had at creation time and do not receive the new editing and export features.

WAS THIS USEFUL?  

Anysphere Cursor

Feedback ⓘ

2 RELEASES · SEEN 2026-09-10 → 2026-09-11 NOTES ↗ RANK

Built to make you extraordinarily productive, agents turn ideas into code. Accelerate development by handing off tasks to Cursor.

> Cursor's headline release this window is Projects — a persistent coordinator agent that delegates work to thousands of parallel subagents, runs cloud-by-default, and can act autonomously via Slack or PR subscriptions — alongside new follow-up steering for running agents and private Origin Apps built on the Origin Public API.

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

01 Projects: coordinator agent with parallel subagents NEW

85

Projects, accessible from the left-hand nav, runs a persistent coordinator agent that plans and delegates large bodies of work (spanning months) across thousands of parallel subagents. Projects run cloud-by-default on their own persistent compute so work continues after closing your laptop, with local agents spun up on demand for machine-specific tasks. A shared context file set syncs across all cloud and local agents in a Project, accumulating codebase knowledge and working preferences so every new subagent starts already onboarded. Subscriptions let the coordinator watch a Slack channel, run on a schedule, or follow all PRs, acting autonomously on CI failures, opens, and merges without manual prompting.

Kick off a long-running security remediation task — e.g. eliminating all flaky auth tests — and let the coordinator run to completion without babysitting it.

```
$ /goal fix all flaky tests and make CI green
```

Route incoming bug reports from a Slack channel directly to the coordinator so it delegates fixes automatically, without waiting for a manual prompt.

📍 In Projects, open Subscriptions › Connect Slack, then point it at your bug-report channel so the coordinator delegates a fix each time a new report arrives.

Set up a recurring gardening Project that watches every new PR for design-system violations and auto-applies fixes without being prompted.

📍 In Cursor, open the left-hand nav, select Projects, create a new Project, and tell the coordinator: 'Follow all new PRs, extract components that belong in the design system, and add a lint rule whenever you see the same mistake twice.' Then configure a Subscription to trigger on PR open and merge events.

— Mechanism and scope well described, some UI-only steps `snapshot-20260911` `launch-20260910-72494b2b`

02 **Follow-up steering for running agents** NEW 60

Send a message to redirect an agent mid-run without interrupting it: follow-ups queue at the next tool call rather than cutting the agent off, or you can send immediately with Send now or by pressing ⏎ twice.

— Exact interaction mechanism given, no broader scope detail `snapshot-20260911`

THINNER COVERAGE BELOW

03 **Origin Apps private integrations** NEW 50

Origin Apps are private integrations built on the Origin Public API, created and installed from Repository Settings > Origin Apps by selecting Install.



Create a private Origin App to integrate a custom workflow with Cursor via the Origin Public API.

📍 In Cursor, go to Repository Settings › Origin Apps and select Install to create a new private Origin App.

— Names API and install path but little mechanism detail `product docs`

WAS THIS USEFUL?  

OpenAI Codex CLI

Feedback ⓘ 3 RELEASES · 2026-09-11 NOTES ↗

CODE ↗ RANK

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

> OpenAI Codex CLI shipped thread-scoped instructions, easier navigation back to the command center after canceled sessions, bundled Linux voice runtimes, a new managed-daemon shutdown flag, higher Guardian action review limits, and trigger propagation for delegated agents.

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

01 **Thread-scoped instructions via ThreadInstructionsProvider**

NEW 75

Adds `ThreadInstructionsProvider` via `StartThreadOptions` to supply per-thread instructions that compose after global instructions and before repository instructions, enforced with a 10,000 estimated-token cap independent of the repository instruction budget.

— Names the API and exact token cap, but no usage example. rust-v0.155.0-alpha.3.7

02 **Managed daemon shutdown flag for app-server** NEW 60

Adds `--managed-daemon` flag to the app-server to save loaded threads during managed daemon shutdown.

— Names an exact runnable flag, minimal mechanism explanation. rust-v0.155.0-alpha.3

THINNER COVERAGE BELOW

03 **Bundled Linux voice runtimes and audio buffer tuning** IMPROVED 55

Bundles GNU voice helpers and runtimes with primary Linux musl release archives and increases ALSA capture/playback buffer sizes to support larger PipeWire graph cycles, improving audio reliability.

— Names concrete audio subsystems but no configuration steps. rust-v0.155.0-alpha.3.7

04 **Turn trigger propagation to delegated agents** IMPROVED

55

Propagates `turn_trigger` through agent spawning and follow-up messages so delegated agents retain the trigger (composer input, scheduled automation) of the initiating turn.

— Names the mechanism and trigger types but no usage example. `rust-v0.154.0-alpha.6.2`

05 **Command center navigation after session cancellation** IMPROVED

50

Returns the user to the command center after session deletion, a canceled unarchive/fork, or pressing `Esc` in a conversation open in another app, instead of exiting the TUI.

— Clear behavior change but no exact navigation path given. `rust-v0.155.0-alpha.3.7`

WAS THIS USEFUL? 👍 👎

Cline

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ CODE ↗ RANK

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

> Cline desktop v0.0.26 brings live GitHub PR and CI status directly into the composer, streamlines provider selection with Cline Pass and a quick-add row, and reworks the Customize view's navigation and toggles.

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

01 Live GitHub PR and CI status in composer NEW

76

The composer now shows live GitHub PR status — PR number, merge status, changed-line totals, and CI checks with log inspection — refreshing every 30 seconds, on window focus, and on demand. Requires the `gh` CLI installed and signed in, with a `GitHub.com` `origin` remote.

— Names exact requirements and refresh triggers, missing only a demo. desktop-v0.0.26

02 Customize view redesign and controls IMPROVED

65

Unifies the Customize view's Tools, Skills, and Rules tabs into a consistent list style; adds a search bar to the Tools tab that filters both sections, per-section Enable all/Disable all controls, and an in-place enable/disable toggle for Skills with Copy path / Uninstall menu options. Tabs are reordered to Tools, Plugins, Skills, Rules, MCP, Hooks, and the sidebar's New Task button is renamed to New Session.

— Names UI elements and controls precisely; navigation path given but no runnable command.

desktop-v0.0.26

THINNER COVERAGE BELOW

03 Provider picker additions: Cline Pass and quick add NEW

51

The composer's provider picker gains a Set up another provider row that opens Settings → Models to reach the full model catalog without leaving the composer, and a new Cline Pass entry alongside Cline, configured under the same Cline account credentials.

— Names the two picker additions and where they lead, but thin on mechanism.

desktop-v0.0.26

04 Recurring schedules default to local timezone IMPROVED

35

New recurring schedules now default to the local timezone instead of UTC.

— States the before/after but no further detail on configuration. desktop-v0.0.26

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ RANK

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

> Letta Code v0.32.2 adds host-provided Slack conversation binding and an experimental unified toolset, while removing AgentFile import/export support entirely.

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

01 **AgentFile import/export removed** BREAKING 80

The `--import` and `--from-af` CLI flags and the `/export` and `/download` slash commands are removed; AgentFile (`.af`) import and export are no longer supported, including imports from the agent registry.

— Names exact removed flags and commands with clear migration impact. v0.32.2

THINNER COVERAGE BELOW

02 **Slack conversation binding actions** NEW 38

Adds host-provided Slack conversation binding actions via the channels integration, enabling agents to bind and interact with Slack conversations.

— Names the integration but no command or config to invoke it. v0.32.2

03 **Experimental unified Letta toolset** NEW 23

Adds an experimental unified Letta toolset, consolidating agent tools into a single integrated set.

— Vague description with no naming of which tools are consolidated. v0.32.2

└──▷ **BREAKING ON UPGRADE**

! The `--import` and `--from-af` CLI flags and the `/export` and `/download` slash commands are removed; AgentFile (`.af`) import and export are no longer supported, including imports from the agent registry.

WAS THIS USEFUL? 👍 👎

Graphify Labs Graphify

Feedback ⓘ 1 RELEASE · 2026-09-10

NOTES ↗ RANK

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

> Graphify's docs and v0.9.58 release round out its CLI-first workflow: query/path/explain commands with edge provenance, a `graphify install` skill for 17 assistants, PR triage, a local MCP server, and broader SQL and Rust extraction.

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

01 **Graph query, path, and explain commands** NEW 91

`graphify query <symbol>` retrieves everything connected to a named symbol, and gained natural-language traversal with a `--dfs` flag to switch to depth-first search and now returns extracted edges with file and line references. `graphify path <symbol1> <symbol2>` finds the route between two symbols (e.g. `graphify path "UserService" "DatabasePool"`) and traces the shortest hop chain across the graph. `graphify explain <edge>` shows why a specific edge exists, including its provenance label — `parsed` (syntax tree), `inferred` (model), or unresolved (dynamic dispatch, reflection, runtime-built imports) — and can also explain a named node.

Trace every caller and dependency of an auth service before refactoring it, to understand blast radius.

```
$ graphify query AuthService
```

Find the call path between an authentication symbol and the database layer to audit privilege escalation routes.

```
$ graphify path AuthService Db
```

Investigate a suspicious or unexpected edge to see whether it was derived from source code or inferred by the model.

```
$ graphify explain <edge>
```

Trace how a request flows to the cache — useful for auditing lateral paths through your architecture.

```
$ graphify query "how does a request reach the cache?" --dfs
```

Find the explicit dependency path between two services to understand blast radius before a change.

```
$ graphify path "UserService" "DatabasePool"
```

Trace the dependency path between two services to understand blast radius during an incident or code review.

```
$ graphify path "UserService" "DatabasePool"
```

— Names commands, flags and provenance types with runnable examples. product docs

02 **MCP server for the graph** NEW 80

`python -m graphify.serve` exposes the graph over MCP (stdio transport) with 10 tools, writing no files and touching no code — letting an assistant query the repo graph directly.

Expose the repo graph to an AI assistant over MCP so it can answer architecture questions without touching your codebase.

```
$ python -m graphify.serve graphify-out/graph.json
```

— Runnable command, transport, tool count, and no-side-effect guarantee. product docs

03 **PR triage subcommand** NEW 65

`graphify prs` lists pull requests from the terminal, with `--triage` and `--conflicts` flags for filtered views; later described as surfacing the pull request queue from the terminal.

Surface PRs that introduce graph conflicts before merging — useful in CI or pre-review triage.

```
$ graphify prs --conflicts
```

— Names flags and a runnable example, but mechanism is thin. product docs

04 **Graph output files** NEW 65

Scanning produces `graphify-out/graph.html` (an interactive graph viewable directly in a browser), `graphify-out/GRAPH_REPORT.md` (a human-readable report covering god nodes, communities, and surprises), and `graphify-out/graph.json` (the full machine-readable graph for downstream tooling).

— Names three exact output files and their contents. product docs

05 **Assistant skill install across 17 tools** NEW 60

`graphify install` wires the `/graphify` skill into your assistant, auto-detecting and registering it across 17 supported assistants including Claude Code, Cursor, and GitHub Copilot.

— Names the command and assistants but no example run. product docs

06 **SQL index and Rust const/static extraction** NEW 60

SQL extraction now emits index nodes for `CREATE [UNIQUE] INDEX` statements, linked to the table they index. Rust module-level `static` and `const` declarations are now extracted as nodes in the knowledge graph.

— Names exact syntax covered but no example graph query shown. v0.9.58

THINNER COVERAGE BELOW

07 **Local parsing across 36 language grammars** NEW 55

Parses source files locally across 36 language grammars, turning functions, imports, and calls into edges without sending code to a model backend.

— Gives a concrete number but no command to try it. product docs

08 **Deep scan mode** NEW 45

`--mode deep` flag on the scan command performs a multi-pass read of the repository.

— Names the flag but not what the extra passes find. product docs

WAS THIS USEFUL? 👍 👎

Diagram Design

Feedback ⓘ changes since 2026-08-12 CODE ↗

Diagram Design is a Claude Code skill that generates editorial HTML and SVG diagrams and imports draw.io or Mermaid sources.

Adds treemap, dumbbell, slopegraph, ridgeline, bubble, beeswarm, and ten editorial diagram types, plus native Droid plugin packaging and named client profiles.

└──▷ **GET THIS VERSION**

```
$ git clone --branch commits-2026-08-12
https://github.com/cathrynlavery/diagram-design.git
# already have the repo? check out this version:
$ git checkout commits-2026-08-12
```

└──▷ **TRY IT**

Catch clipped SVG content and runtime paint errors in rendered diagram examples before shipping.

```
$ python3 scripts/lint-render.py
```

Verify that all beeswarm dots sit at their declared values and that no two dots overprint, as part of a geometry gate.

```
$ python3 scripts/verify-beeswarm.py
```

Run adversarial tests against the Sankey verifier in both pass and fail polarities to confirm the checker fires on real defects.

```
$ python3 scripts/test-verify-sankey.py
```

Adds `verify-beeswarm.py` to verify that beeswarm dot positions match their declared values and that no two dots overprint, enforcing nine geometric invariants including shared value scale and packing-only swarm offset.

Adds `lint-render.py`, a headless-Chromium rendering linter that catches clipped SVG content, collapsed SVGs, and page overflow by diffing screenshots with and without `overflow` released across staged ancestor levels.

Adds `test-verify-sankey.py` adversarial test suite for `verify-sankey.py`, covering ribbon narrowing, volume loss, stage carry, label mismatch, and dark/light variant drift

in both polarities.

Adds treemap diagram type for part-of-whole by area.

Adds dumbbell as a Bar chart variant.

+9 more

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ 1 RELEASE · 2026-09-10

NOTES ↗ CODE ↗ RANK

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

> Claude Code's v2.1.268 release expands gateway administration (pricing sync, internal-network allowlisting, access-control warnings), adds JSON output across plugin commands and a session-state cleanup flag for self-hosted runners, and restricts task-tracking tools to a specific model list.

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

01 Session-state cleanup flag for self-hosted runners NEW

85

Adds `--remove-session-state` flag to `claude self-hosted-runner` (default off) to delete each session's per-session directories under `<base-dir>/_sessions/` when the session ends.

Clean up session state automatically after each run in a CI self-hosted runner, keeping disk usage bounded across many short sessions.

```
$ claude self-hosted-runner --remove-session-state
```

— Exact flag, default state, and directory path given v2.1.268

02 Task-tracking tools restricted to specific models BREAKING

85

Changes `TaskCreate` / `Get` / `Update` / `List` and `TodoWrite` task-tracking tools to be offered only on Claude 3.x, Opus 4.0-4.7, Sonnet 4.0-4.6, Haiku 4.5; set `CLAUDE_CODE_ENABLE_TODO_TOOLS=1` to enable elsewhere.

— Exact model list and override env var given v2.1.268

03 JSON output across plugin commands IMPROVED

81

Adds `--json` output flag to `claude plugin install`, `uninstall`, `update`, `enable`, and `disable`, and adds `errorDetails` / `noteDetails` fields to each row of `claude plugin list --json`.

Parse plugin operation results in automation scripts by getting structured JSON back from install and list commands.

```
$ claude plugin install my-plugin --json && claude plugin list --json | jq '.[].errorDetails'
```

— Names every command and field, with runnable example `v2.1.268`

04 **gatewayInternalNetworks managed setting** NEW 75

Adds `gatewayInternalNetworks` managed setting, letting administrators allow `/login` to a Claude apps gateway on their organization's own public IPv4 block.

Allow org employees to authenticate via `/login` through a self-hosted Claude apps gateway on the company's own public IP block.

```yaml
gatewayInternalNetworks:
  - 203.0.113.0/24
```

— Named setting shown with concrete config example `v2.1.268`

05 **WebFetch deadline override environment variable** NEW 75

Adds `CLAUDE_CODE_WEBFETCH_DEADLINE_MS` environment variable to override the WebFetch 300-second deadline (set to 0 to disable the deadline entirely).

— Exact env var name, default, and disable value `v2.1.268`

06 **WebFetch rules no longer gate Artifact tool** BREAKING 70

Plain `WebFetch` deny and ask rules no longer apply to Artifact tool reads and updates; use an `Artifact` rule (or WebFetch(domain:claude.ai)) to block or gate them.

— Names exact rule migration path for affected users `v2.1.268`

07 **Built-in hooks-module plugins published as source** NEW 68

Publishes three built-in hooks-module plugins as source under `mods/`: `sec-default` (organization default outermost plugin), `diff` (`/diff`), and `telemetry` (`$.telemetry`).

— Names all three plugins and their paths/commands `v2.1.268`

08 **Gateway pricing sync in gateway.yaml** NEW 65

Adds `pricing:` support in `gateway.yaml` so signed-in Claude Code clients receive org-configured rates through managed settings, keeping `/cost` and telemetry aligned with the spend meter.

— Names config key and effect, lacks example syntax v2.1.268

THINNER COVERAGE BELOW

09 **Startup warnings for unrestricted gateway access** NEW

50

Adds startup warnings for gateways when `access_control.allow_cidrs` is empty, and a one-time warning the first time a request arrives from a public address.

— Describes trigger conditions, no remediation step given v2.1.268

10 **configDirectory field in auth status JSON** IMPROVED

40

Adds `configDirectory` field to the output of `claude auth status --json`.

— Single field addition, no example provided v2.1.268

11 **Browser-tab icons for published artifacts** NEW

30

Adds browser-tab icons for published artifacts, chosen by Claude to match each page.

— Thin description, no mechanism or setting named v2.1.268

└──▷ **BREAKING ON UPGRADE**

! `TaskCreate`, `Get`, `Update`, `List`, and `TodoWrite` task-tracking tools are no longer offered on models outside Claude 3.x, Opus 4.0-4.7, Sonnet 4.0-4.6, and Haiku 4.5; set `CLAUDE_CODE_ENABLE_TODO_TOOLS=1` to restore them on other models.

! Plain `WebFetch` deny and ask rules no longer apply to Artifact tool reads and updates; use an `Artifact` rule (or WebFetch(domain:claude.ai)) to block or gate them.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen Code

Feedback ⓘ 1 RELEASE · 2026-09-10 NOTES ↗

RANK

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

> Qwen Code's TypeScript SDK v0.1.12 renames all `disable*` settings keys to positive `enable*` naming (a breaking change) and adds a `memory.enableManagedAutoMemory` toggle for deterministic system prompts, alongside a batch of new SDK capabilities: multi-modal input, session resume, a batch runner, MCP progress reporting, Coding Plan auth, and a new `settings.env` config field.

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

01 Managed auto-memory control via `enableManagedAutoMemory`

NEW 85

The `memory.enableManagedAutoMemory` setting controls whether managed auto-memory injects remember/dream instructions into the system prompt. Setting it to `false`, as in `{ "memory": { "enableManagedAutoMemory": false } }`, gives fully deterministic system prompts on a given host.

Prevent managed auto-memory from injecting remember/dream instructions on a host where you want fully deterministic system prompts.

```json
{
  "memory": {
    "enableManagedAutoMemory": false
  }
}
```

— Named config key with a runnable example config. sdk-typescript-v0.1.12

02 Renamed negative settings keys to positive naming

BREAKING 65

Negative settings keys of the form `disable*` were renamed to positive `enable*` naming; any config or scripts using the old `disable*` form must be updated.

— Clear before/after but no exact key list given. sdk-typescript-v0.1.12

THINNER COVERAGE BELOW

03 `settings.env` config field NEW 50

A new `settings.env` field was added to SDK configuration for setting environment variables; the release notes give no further detail on its schema or usage.

— Named config field but no schema or example. sdk-typescript-v0.1.12

04 **MCP progress reporting** NEW 25

The SDK now surfaces progress information for MCP (Model Context Protocol) operations; no further detail on the mechanism is given.

— Names MCP but no interface or event detail. sdk-typescript-v0.1.12

05 **Coding Plan authentication** NEW 20

A new Coding Plan authentication method was added to the SDK; the release notes do not describe its setup or scope.

— Named but undescribed auth mechanism. sdk-typescript-v0.1.12

06 **Telemetry source tracking** NEW 15

Telemetry logs now track source information for events, though the release notes do not specify which sources or fields are recorded.

— Bare mention with no fields or usage named. sdk-typescript-v0.1.12

07 **Batch runner** NEW 15

A batch runner was added to the SDK; the release notes do not describe its interface or how it is invoked.

— Only named in summary, no usage shown. sdk-typescript-v0.1.12

08 **Session resume** NEW 15

SDK sessions can now be resumed, per the release summary; no mechanism or command for resuming is described.

— Only named in summary, no detail given. sdk-typescript-v0.1.12

09 **Multi-modal input support** NEW 15

The Qwen Code TypeScript SDK now accepts multi-modal input, though the release notes give no further detail on supported modalities or usage.

— Only named in summary, no detail given. sdk-typescript-v0.1.12

└──▷ **BREAKING ON UPGRADE**

! Negative settings keys (`disable*`) are renamed to positive naming (`enable*`); any config or scripts using the old `disable*` form must be updated.

WAS THIS USEFUL?  

Cotool

Feedback ⓘ 1 RELEASE · 2026-09-10 NOTES ↗ RANK

Cotool is an AI security operations platform for investigating alerts, enriching threat intelligence, and automating response workflows.

> Cotool v0.68.0 expands its agent integration surface with new Serval and Lumos lookups, adds Linear API sessions and Okta private key JWT auth, and rounds out with new LLM model options and detection-table UI improvements.

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

01 **Serval and Lumos integrations for agents** NEW 60

Agents can now look up users, groups, access requests, tickets, and workflow runs via the new Serval integration, and look up users, apps, groups, access requests, and activity logs via the new Lumos integration.

Enrich an investigation by having an agent query Serval for a user's open access requests and associated workflow runs.

> 📍 In an agent prompt, instruct: 'Use the Serval integration to look up access requests and workflow runs for user <username>.'

— Lists lookup objects for both integrations but no API mechanism v0.68.0

THINNER COVERAGE BELOW

02 **True/false positive rate columns in detections table** NEW 55

Adds true positive, benign, and false positive rate columns (combined, sortable) to the detections table.

— Names exact columns and sort behavior, no navigation path v0.68.0

03 **Default response agent for Slack mentions** NEW 50

Automatically sets up a default response agent to handle @mentions when you connect Slack.

— Describes trigger and setup but no config options v0.68.0

04 **Linear agent sessions via API** NEW 40

Supports starting a Linear agent session with any agent through the API.

— No endpoint name or example given v0.68.0

05 **New selectable LLM models added** NEW 40

Adds `GLM 5.3 Flash`, `GLM 5.2 Fast`, and `DeepSeek V4 Flash` as selectable models.

– Names models but no usage or selection detail v0.68.0

---

06 **Select read-only button in chat tool selector** NEW

40

Adds a Select read-only button to the chat tool selector.

– Names UI element but no further behavior described v0.68.0

---

07 **Markdown tables in Jira comments** IMPROVED

40

Agents can now write markdown tables in Jira comments and descriptions.

– States capability without example or mechanism v0.68.0

---

08 **Private key JWT auth for Okta** NEW

35

Supports connecting Okta with private key JWT authentication.

– No config steps or field names provided v0.68.0

---

09 **Tines record create, update, query** IMPROVED

35

Tines agents can now create, update, and query records.

– No record schema or endpoint specifics v0.68.0

WAS THIS USEFUL?  

Amazon Kiro Crew

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗

RANK

Kiro Crew is an open-source workspace for coordinating AI coding agents.

> Kiro Crew v0.6.0's biggest changes are selectable agent harnesses (Claude Code, Codex, KAS), remote crew routing for chat sessions, worker session dispatch, and much longer chat/sub-agent/auto-run timeouts, alongside new per-turn telemetry, sandbox security hardening, and an installer switch to a pinned CPython that drops Python 3.10/3.11 support.

└──▷ WHAT SHIPPED · 29 FEATURES most completely described first ? what's the number?

01 Extended timeout and duration limits for chat, sub-agents and auto-run plans

95

BREAKING

`agent.chat_turn_timeout_secs` in `config.json` is raised from 7200 to 14400 seconds, extending the maximum chat turn to four hours. `agent.subagent_timeout_secs` is raised from 1800 to 10800 seconds and `agent.subagent_max_turns` from 200 to 1000, giving sub-agents three hours and up to 1000 tool calls. `orchestrator.max_plan_duration_seconds` now enforces a two-hour default ceiling on unattended auto-run plans, which can be raised or set to 0 to remove the ceiling entirely.

Extend a long-running agentic plan beyond the new two-hour default ceiling so it can complete overnight without being cut.

json

```
{
  "orchestrator": {
    "max_plan_duration_seconds": 28800
  }
}
```

— Exact config keys, before/after numbers, and a runnable override example. v0.6.0

02 Per-turn telemetry for tokens, spend and latency NEW

86

`telemetry.enabled` in `config.json` (off by default) unlocks latency and fault-rate charts and per-turn token, spend and latency reporting across cron, heartbeat, subagents, workflows and every messaging channel, visible under Developer → Telemetry with Developer Mode on.

Enable per-turn token, spend and latency telemetry so you can track cost and performance across subagents and messaging channels.

```json
{
  "telemetry": {
    "enabled": true
  }
}
```

— Config key, UI location and a runnable enable example. v0.6.0

---

03 **Installer now provisions pinned CPython** BREAKING 83

The installer (`curl -fsSL https://download.crew.kiro.dev/cli.sh | sh`) now provisions a pinned CPython instead of using the system Python; pass `--system-python` to revert to the previous behaviour. Python 3.10 and 3.11 are no longer supported — a host must upgrade to Python 3.12 before installing or updating Kiro Crew.

— Names the installer command, flag and required Python version. v0.6.0

---

04 **kirocrew config defaults command** NEW 82

`kirocrew config defaults` reports which stored config values are holding a superseded default; `--adopt` takes the new defaults and `--keep` records existing values as intentional when a default has changed.

Check which stored config values are holding a superseded default and adopt all new defaults at once after upgrading.

```
$ kirocrew config defaults --adopt
```

— Exact command and flags with a runnable example. v0.6.0

---

05 **Sandbox fails closed on unsupported architectures** BREAKING 81

Hosts on unsupported architectures (armv7l, riscv64, ppc64le, s390x Linux) or a libc without `prctl` now fail closed unless `agent.sandbox` is set to `off` or `agent.sandbox_allow_unsandboxed_exec` is set to `true` in `config.json`.

— Lists exact architectures and the two override config keys. v0.6.0

---

06 **Remote crew routing for chat sessions** NEW 80

`instances.enabled` in `config.json`, combined with Settings → Developer → Feature Previews → Chat on a crew, lets a chat session be routed to a remote peer crew: the transcript stays local while every turn actually runs on that remote crew.

— Names config key and UI path but no example. v0.6.0

---

07 **Dev mode restricted outside app install root** BREAKING

77

Dev mode for a UI folder outside an app's install root is now refused by the dashboard and the API; `kirocrew app dev <name> --confirm-out-of-install-root` is the only way to grant it.

— Names the exact command and the enforced restriction. v0.6.0

---

08 **App manifest fields for jobs, commands and session controls** NEW

72

New app manifest fields `permissions.jobs`, `contributes.commands` and `contributes.sessionControls` let installed apps own background jobs, Command Bar rows and live composer controls.

— Names all three manifest fields and their purposes. v0.6.0

---

09 **AWS Control app** NEW

72

A new AWS Control app, disabled by default and enabled from Apps → Library, adds an Overview, Files, Library, Backup and Access panes, each with its own URL.

— Names every pane but not what each does. v0.6.0

---

10 **Command gate enforced via sandbox bind masks** BREAKING

72

The command gate no longer fences credential paths by reading command text; the sandbox bind masks are now the fence. Anyone relying on the text gate should verify that `agent.sandbox` in `config.json` is not set to `off`.

— Explains the mechanism shift and names the config key to check. v0.6.0

---

11 **Session search across Telegram and sidebar** IMPROVED

71

`/sessions <words>` in Telegram direct messages searches sessions by keyword, listing the ten most recent sessions when given no arguments; the dashboard's session sidebar search now also matches sessions by pull request, review or issue badge in addition to title.

— Names the Telegram command and sidebar behaviour, no config surface. v0.6.0

12 **send_message agent tool for cross-channel messaging** NEW

70

The new `send_message` agent tool lets agents message a channel by name across Slack, Discord, Telegram, WhatsApp, Webex, Teams, iMessage and Feishu.

— Names the tool and every supported channel. v0.6.0

13 **Selectable agent harnesses: Claude Code, Codex, KAS** NEW

70

Settings → Developer → Agent Backend (requires Developer Mode on) now lets a crew choose between Claude Code, Codex and KAS as its agent harness.

— Names the UI path and all three harness options. v0.6.0

14 **Apps → Library folder creation and Launchpad grid** NEW

70

Create Folders From Project under Apps → Library generates one nested sidebar folder per selected package from a project. The new Launchpad grid under Apps → Library presents installed apps as icon tiles with pin, open, details, update, disable and uninstall actions.

— Names both UI paths and the Launchpad's full action set. v0.6.0

15 **Crew avatar customization with state cues** NEW

70

Agent Capabilities → Agents → Avatar → Customize lets a crew avatar have per-state eyes, mouth and sound cues for working, done and failed states.

— Names the exact UI path and all three states. v0.6.0

16 **Artifact save slug flag and colour warning** IMPROVED

68

`kirocrew artifact save` gains a `--slug` flag for an exact handle, refusing a taken slug instead of renaming, and now warns when an artifact's colours are hardcoded, in both the agent tool result and on the CLI command.

— Names the command and flag; warning behaviour has no remediation detail. v0.6.0

17 **Full dashboard rebranding via theme packs** NEW

68

Installed theme packs can now rebrand the entire dashboard shell — including product name, logo and favicon — via Settings → Display → Install Theme.

— Names UI path and exact rebrand scope. v0.6.0

18 **Plain-text diffs and folded diff chips in chat** IMPROVED

66

Settings → Chat → Plain diffs (off by default) renders patches as plain monospace text instead of folded diff chips; by default, diffs still start folded as a chip showing filename and +N/`-M` counts.

— Names the setting path and default chip format. v0.6.0

19 **Worker session dispatch from crew threads** NEW 64

Crew members can now dispatch work into worker sessions from their own thread on Claude and KAS backends, with each worker inheriting its creator's trust posture.

— Explains mechanism but gives no command or config to invoke it. v0.6.0

20 **Built-in terminal completion and theming** IMPROVED 62

`dashboard.terminal.completion.enabled` in `config.json` can be set to false to silence inline completion menus in the built-in terminal, and the terminal now draws all sixteen ANSI colours from the active theme.

— Names the config key; ANSI colour change lacks further mechanism. v0.6.0

21 **Security policy files sealed read-only in sandbox** BREAKING 61

A sandboxed command can no longer rewrite the security policy, admission policy, profiles or denied-command list; these are now sealed read-only in every sandbox mode.

— Clear before/after but no config key to verify or override. v0.6.0

22 **MCP server saves no longer reset active session** IMPROVED 61

Saving an MCP server no longer resets the active session on Kiro CLI 2.10.0 or newer, and the chat session menu's MCP servers view now reports what the session actually mounted.

— Names the CLI version requirement and UI view but no config key. v0.6.0

23 **Global session-control kill switch** NEW 60

`agent.session_control` in `config.json` can be set to false to withdraw session control from every agent at once.

— Names the config key but no scenario or mechanism detail. v0.6.0

THINNER COVERAGE BELOW

24 **Denied Commands tagged by contributing edition** NEW 58

Settings → Security → Denied Commands now tags any deny rule contributed by the current edition with its id so it can be individually disabled.

— Names the UI path but no example of disabling a rule. v0.6.0

25 **Semantic memory and credential scan performance gains**

IMPROVED 58

Semantic memory uses approximately 1.2 GB less RAM, the session list loads over twice as fast on large stores, and credential scanning runs 2.6–2.9x faster.

— Concrete numbers but no way to invoke or verify them. v0.6.0

26 **Automatic meeting preparation via calendar integration** NEW

56

Meetings → Settings → Calendar integration enables automatic preparation of upcoming meetings via background polling.

— Names the UI path but not what preparation entails. v0.6.0

27 **Credentials section in kirocrew doctor** NEW 53

`kirocrew doctor` now outputs a Credentials section listing AWS profiles without exposing secrets.

— Names the command but no further detail on scope. v0.6.0

28 **Resumed sessions skip full memory re-injection** IMPROVED

42

A resumed session no longer re-injects its full memory, lessons and skills block, delaying compaction on idle sessions.

— Names the mechanism but no config or measurable effect. v0.6.0

29 **Blocked tool calls explain the alternative** IMPROVED

40

A blocked tool call now explains what the agent may do instead, stopping retry loops on the same blocked shape.

— Describes behaviour but names no surface or config. v0.6.0

└──▷ **BREAKING ON UPGRADE**

! Python 3.10 and 3.11 are no longer supported; a host must upgrade to Python 3.12 before installing or updating Kiro Crew.

! The installer now provisions a pinned CPython instead of using the system Python; pass `--system-python` to revert to the previous behaviour.

! The command gate no longer fences credential paths by reading command text; the sandbox bind masks are now the fence — verify that `agent.sandbox` in `config.json` is not `off` if

you relied on the text gate.

! An unattended auto-run plan now stops after two hours by default; raise the limit with `orchestrator.max_plan_duration_seconds` in `config.json` or set it to 0 to remove the ceiling.

! Dev mode for a UI folder outside an app's install root is refused by the dashboard and the API; only `kirocrew app dev <name> --confirm-out-of-install-root` grants it.

WAS THIS USEFUL? 👍 👎

AGENT

◆ AI Agent Frameworks

OpenClaw

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ CODE ↗ RANK

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

> OpenClaw v2026.9.4 focuses on cloud worker infrastructure — prepared sessions with a configurable reserve pool and full snapshot lifecycle management — alongside a unified Plugins workspace, a read-only deployment mode, and Deepgram Flux voice transcription.

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

01 Prepared cloud worker sessions and reserve pool NEW

92

Adds `cloudWorkers.profiles.<id>.readyWorkers` and `cloudWorkers.preparedPool.maxTotal` config keys to control how many prepared cloud-worker reserves are kept per project/profile and Gateway-wide (set either to zero to disable reserves). Prepared cloud sessions let eligible Linux sessions start from local Git projects or public GitHub repositories using these pre-built worker snapshots, avoiding repeated environment setup.

Disable all prepared cloud-worker reserves Gateway-wide to avoid incurring provider running-machine charges for warm workers.

yaml

```yaml
cloudWorkers:
  preparedPool:
    maxTotal: 0
```

Cap ready workers for a specific project profile to one instead of the default, controlling warm-worker spend at a per-project level.

yaml

```yaml
cloudWorkers:
  profiles:
    my-project-profile:
      readyWorkers: 1
```

— Names exact config keys with runnable example values and behavior. v2026.9.4

02 Cloud worker snapshot management NEW

75

Cloud worker snapshots can be built, rebuilt, inspected, pinned, deleted, and rolled back from Settings → Connections → Cloud workers → Snapshots in the Control UI, with support for cancelling an active build.

— Exact UI path and full set of lifecycle operations given. v2026.9.4

03 **/question command for pending prompts** NEW 70

The `/question` command reopens a pending terminal prompt in Gateway-connected and local TUI sessions, supporting keyboard-driven choices, free-text answers, and multi-question prompts.

— Named command with clear supported interaction modes. v2026.9.4

04 **Read-only deployment config mode** NEW 65

New `OPENCLAW_CONFIG_READONLY=1` environment variable prevents OpenClaw from rewriting deployment-managed configuration, while keeping read-only diagnostics and runtime state functional.

— Named env var is directly runnable, but scope of effect only briefly described. v2026.9.4

THINNER COVERAGE BELOW

05 **Deepgram Flux voice-note transcription** NEW 58

Voice notes can be transcribed using Deepgram Flux via the `flux-general-en` or `flux-general-multi` models; requires `ffmpeg` to be installed.

— Names models and dependency but no usage steps shown. v2026.9.4

06 **Unified Plugins workspace in Control UI** NEW 50

A single Plugins workspace in the Control UI lets users discover bundled and ClawHub plugins, install them, and manage setup, settings, and access all in one place.

— Describes scope well but only a UI area, no exact path. v2026.9.4

07 **Automatic rollback for failed updates** NEW 50

When an update fails but is compatible, OpenClaw automatically retains the previous package and restores it along with its prior configuration and service, provided schema and configuration checks confirm the rollback is safe.

— Mechanism explained but no config or trigger named. v2026.9.4

08 **Advanced cloud worker profile editing and Windows workers**

NEW

45

The Control UI adds advanced editing of cloud worker profiles and repository defaults, plus native Windows worker selection when the backend supports it.

— Names areas but no specifics on what 'advanced' editing entails. v2026.9.4

09 **GPT Image 2.5 Flare and Sunburst variants** NEW

45

Adds GPT Image 2.5 Flare and Sunburst variants for image generation and editing through OpenAI or fal, usable without changing the default model.

— Names the model variants and providers but no usage detail. v2026.9.4

WAS THIS USEFUL? 👍 👎

Daily Pipecat

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ RANK

Pipecat is a framework for realtime voice and multimodal agents, orchestrating speech-to-text, model inference, and text-to-speech as a streaming pipeline.

> Pipecat v1.9.0 ships a per-segment latency breakdown for pipeline turns, three new observability observers (service metrics, speech events, errors), a Meta streaming STT integration, and a broad set of new tuning settings across roughly a dozen STT/TTS providers.

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

01 Per-segment latency breakdown for pipeline turns NEW

90

Adds `LatencyBreakdown.contributions` to decompose user-to-bot latency into named segments (VAD silence, transcription, LLM inference, TTS, etc.), each with a stable `key`, `owner_kind`, `measured_from`, and `total_secs`. Print with LatencyBreakdown.turn_contribution_lines() and filter short contributions with UserBotLatencyObserver(min_contribution_secs=...).

— Names fields and methods; no full runnable example given. v1.9.0

02 Meta streaming STT service integration NEW

90

Adds `MetaSTTService`, a streaming STT service using Meta's `muse-voice-transcribe-1.0` model over the Muse Voice realtime API; install with `pipecat-ai[meta]`. Settings takes `language`, `language_bias`, `keywords`, and `mode` (model endpointing, client-delimited, or diarization).

Use MetaSTTService for multilingual streaming transcription with domain-specific keyword boosting, installed via the meta extra.

python

```python
# pip install 'pipecat-ai[meta]'
from pipecat.services.meta.stt import MetaSTTService

stt = MetaSTTService(
    settings=MetaSTTService.Settings(
        language_bias=['en', 'es'],
        keywords=['Pipecat', 'webhook', 'pipeline'],
        mode='endpointing',
    )
)
```

— Named install extra and settings, with a runnable example. v1.9.0

03 New observers for service metrics, speech events, and errors

NEW

85

Adds `ServiceMetricsObserver` with `on_service_latency` (time-to-first-byte, first audio, first answer token) and `on_service_usage` (STT audio seconds, TTS characters, LLM token counts), one record per metric per turn. Adds `SpeakingObserver` reporting `user_speech_started`, `user_speech_stopped`, `user_turn_started`, `user_turn_stopped`, `bot_speech_started`, `bot_speech_stopped`, and `interruption` events through `on_speech_event`. Adds `ErrorObserver` reporting every pipeline error through `on_error` as an `ErrorEvent` with `message`, `category`, `exception_type`, `processor`, and `processor_usable`.

Wire up an ErrorObserver to catch and categorise every pipeline error in one place — useful when running unattended agents where silent failures are hard to detect.

python

```python
from pipecat.observers.error_observer import ErrorObserver

error_obs = ErrorObserver()

@error_obs.event_handler('on_error')
async def on_error(observer, event):
    print(f'[{event.category}] {event.processor}: {event.message}
'
          f'(usable={event.processor_usable})')

runner.add_observer(error_obs)
```

— Full event/field names given; runnable example only for ErrorObserver. v1.9.0

04 Expanded pipeline startup timing report

IMPROVED

75

Adds `StartupTimingReport.warmup` with `warmup.duration_secs` and `warmup.blocking_duration_secs` to account for deferred-import warming cost, plus a new BaseObserver.on_startup_warmup() event. Also expands `StartupTimingReport` with `setup_phase_secs`, `start_phase_secs`, and per-processor `ProcessorStartupTiming.start_duration_secs` to break down the full pipeline startup span.

— Detailed named fields, no example usage shown. v1.9.0

05 Deepgram STT and TTS setting additions

NEW

73

Adds `profanity_filter` and `redact` to `DeepgramFluxSTTService.Settings` and `DeepgramFluxSageMakerSTTService.Settings` for profanity masking and number redaction. Adds a `version` field to `DeepgramSTTService.Settings` (and `DeepgramSageMakerSTTService.Settings`) to pin transcription to a specific

Deepgram model version instead of `latest`. Adds `speed` to `DeepgramTTSService` and `DeepgramHttpTTSService` for a Deepgram Aura speech-rate multiplier (0.7–1.5) passed to `/v1/speak`.

— Multiple named settings, classes and endpoint, no example. v1.9.0

---

06 **Azure STT and TTS tuning settings** NEW 71

Adds `voice_parameters` to `AzureTTSService` and `AzureHttpTTSService`, exposing SSML `<voice>` `parameters` (`temperature`, `top_p`, `top_k`, `cfg_scale`, `enhancePronunciation`) for HD voice tuning. Adds `segmentation_silence_timeout_ms` (100–5000 ms) to `AzureSTTService.Settings` to control how much in-phrase silence Azure tolerates before emitting a final transcript.

— Named settings and value ranges, no example code. v1.9.0

---

07 **AssemblyAI STT multilingual, timing, and new model support** NEW 68

Adds `language_codes` (e.g. `[Language.EN, Language.ES]`) and `timestamps` settings to `AssemblyAISyncSTTService` for multilingual/code-switching audio and per-word timing. Adds `universal-3-6-pro` as a supported model for `AssemblyAISTTService`, with the full U3-RT-Pro feature set (built-in turn detection, prompting, continuous partials, `interruption_delay`, context carryover, voice focus).

— Named settings and model, but no usage example. v1.9.0

---

08 **DeepSeek reasoning toggle** NEW 65

Adds `thinking` setting to `DeepSeekLLMService.Settings` to control DeepSeek's reasoning pass, e.g. `thinking={"type": "disabled"}` disables the V4 model's chain-of-thought.

— Named setting with example value, no full snippet. v1.9.0

09 **Additional STT/TTS provider tuning settings** NEW 65

Adds `no_verbatim` to `ElevenLabsSTTService` and `ElevenLabsRealtimeSTTService` to strip filler words, false starts, and non-speech sounds from transcripts. Adds `hotwords` to `WhisperSTTService.Settings` to bias Faster Whisper transcription toward specific words or phrases. Adds `reduce_silence` to `SonioxTTSService` to shorten inter-word pauses on supported models. Adds `include_results` to `SpeechmaticsSTTService` to request word-level results in transcript messages. Adds `temperature` to `HumeTTSService.Settings` to control Hume's synthesis sampling temperature. Adds `reconnect_backoff_min_wait` and `reconnect_backoff_max_wait` to `WebsocketService` to configure reconnection wait intervals (defaults: 4 s / 10 s).

— Six named settings across providers, no usage examples. v1.9.0

10 **Chirp 3 noise removal in Google STT** NEW 62

Adds `denoiser_config` to `GoogleSTTService`, exposing Chirp 3 background-noise removal via `denoise_audio` and `snr_threshold`.

— Named setting and sub-fields, no example. v1.9.0

11 **Server-side turn detection tuning for Cartesia** NEW 62

Adds `turn_start_threshold`, `turn_eager_end_threshold`, `turn_end_threshold`, and `turn_end_timeout_ms` to `CartesiaTurnsSTTSettings` for server-side turn detection tuning.

— Four named fields, no defaults or example given. v1.9.0

THINNER COVERAGE BELOW

12 **Progress and update handlers for eval framework** NEW 55

Adds `on_progress` and `on_update` event handlers to the eval framework; `EvalSession` and `EvalSuite` are now `BaseObject`s with `on_progress` (async task) and `on_update` (synchronous) handlers respectively.

— Names handlers and classes but no example. v1.9.0

13 **Cancellable function calls across LLMSwitcher** NEW 55

Adds `cancellable_by_llm` argument to LLMSwitcher.register_function(), forwarding it to every LLM the switcher fronts.

— Names flag and method, no example given. v1.9.0

WAS THIS USEFUL?  

NVIDIA NeMo Agent Toolkit

Feedback ⓘ

1 RELEASE · 2026-09-10 NOTES ↗ RANK

NeMo Agent Toolkit connects agent workflows to tools and model providers with profiling, evaluation, and observability.

> NeMo Agent Toolkit v1.9 adds new middleware for human-in-the-loop approval and tool fault tolerance, an MLflow OTLP telemetry exporter, and expanded authentication provider options, while removing the `local_sandbox` execution mode and changing how user identity is resolved.

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

01 **HITLMiddleware for human-in-the-loop tool calls** NEW

70

Adds HITLMiddleware for human-in-the-loop function interception, allowing agents to pause and await human approval before executing tool calls.

— Names the middleware and its mechanism but no usage example. v1.9.0

02 **fail_closed config for output verifier tools** IMPROVED

70

Adds a `fail_closed` boolean config key to `output_verifier_tools` to control failure behavior when output verification cannot complete.

— Exact config key and target field, ready to set. v1.9.0

03 **CircuitBreakerMiddleware for tool fault tolerance** NEW

60

Adds `CircuitBreakerMiddleware` for tool fault tolerance, automatically stopping calls to failing tools to prevent cascading failures.

— Explains mechanism and named class, lacks config detail. v1.9.0

THINNER COVERAGE BELOW

04 **MLflow OTLP telemetry exporter** NEW

55

Adds an MLflow OTLP telemetry exporter, enabling agent traces and metrics to be forwarded to an MLflow-compatible OTLP endpoint.

— Names the exporter and target but no config steps. v1.9.0

05 **Auth provider enhancements: redirects, preflight, WebSocket identity**

IMPROVED

55

Adds a redirect-based OAuth flow option for authentication providers, enables preflight authentication for applicable authentication providers, and adds WebSocket identity credential configuration via `auth` provider settings.

— Groups three thin auth additions, names `auth` settings. v1.9.0

06 **Plugin API exports for runtime context and HITL models** NEW 50

Exports runtime context and interactive HITL models from the plugin API, and exports interactive prompt content models for third-party plugin authors.

— Names exported model types for plugin authors. v1.9.0

07 **Changed user identity resolution and LLM exposure** BREAKING 50

The `user_id` parameter is no longer exposed to the LLM, and user identity resolution behavior has changed — existing setups relying on the previous resolution logic may require updates.

— Flags a behavior change requiring setup review, no exact migration steps. v1.9.0

08 **Opt-in provider hooks for IDs and timestamps** NEW 45

Adds opt-in provider hooks for generated IDs and timestamps, letting integrators supply custom values for interaction prompt IDs and timestamps via providers.

— Describes purpose but no hook names or config. v1.9.0

09 **LangChain Runnable callback tracking** NEW 40

Adds LangChain Runnable callback tracking for observability of LangChain-based workflows.

— Names integration but no mechanism or config detail. v1.9.0

10 **Configurable JobStore connection pooling** IMPROVED 35

Makes `JobStore` connection pooling configurable.

— Bare statement with no parameters or defaults given. v1.9.0

11 **LangChain integrations split into optional extras** IMPROVED 35

Splits LangChain provider integrations into optional extras for more granular dependency installation.

— States the change but no extra names or install commands. v1.9.0

12 **Removal of local_sandbox execution mode** BREAKING 35

The `local_sandbox` execution mode has been removed.

— Names removed mode but no migration guidance. v1.9.0

└──▷ BREAKING ON UPGRADE

! The `local_sandbox` execution mode has been removed.

! The `user_id` parameter is no longer exposed to the LLM.

! User identity resolution behavior has changed — existing setups relying on the previous resolution logic may require updates.

WAS THIS USEFUL? 👍 👎

LiveKit Agents

Feedback ⓘ 1 RELEASE · 2026-09-10 NOTES ↗

RANK

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

> LiveKit Agents 1.8.1 introduces full-duplex speech models via a new DuplexModel abstraction and its first implementation, GPTLiveModel, alongside new STT and inference integrations, expanded plugin configuration, and event-loop blocking telemetry.

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

01 **DuplexModel for simultaneous speech I/O** NEW 88

Adds a `DuplexModel` class enabling full-duplex speech models that can speak and listen simultaneously, with `GPTLiveModel` as the first implementation. `GPTLiveModel` accepts `voice` and `responses_options` (including `model` and `instructions`) so an agent can run a session using, for example, `voice="marin"` and `responses_options={"model": "gpt-5.6-luna", "instructions": ...}`.

Run a full-duplex voice session where the agent can speak and listen at the same time using OpenAI's GPT-Live model.

python

```python
session = AgentSession(
    llm=GPTLiveModel(
        voice="marin",
        responses_options={
            "model": "gpt-5.6-luna",
            "instructions": "Use tools when current information
is required.",
        },
    ),
)
```

— Named class, params, and a runnable code example `livekit-agents@1.8.1`

THINNER COVERAGE BELOW

02 **Phonic plugin watermarking and tool chaining options** NEW

50

Adds `enable_watermarking` realtime config option to the Phonic plugin and `allow_tool_chaining` as a per-tool configurable option in the same plugin.

— Names both flags but no mechanism or defaults given `livekit-agents@1.8.1`

03 **Speechmatics turn and VAD config exposure** IMPROVED

40

Exposes `end_of_turn_config` and `vad_config` options in the Speechmatics plugin.

— Names config options but no behavior explained `livekit-agents@1.8.1`

04 **Event-loop blocking detection telemetry** NEW

30

Adds telemetry detection that flags synchronous code blocking the event loop.

— Explains purpose but no config or threshold named `livekit-agents@1.8.1`

05 **Meta Muse Voice streaming STT plugin** NEW

25

Adds Meta Muse Voice streaming STT plugin.

— Bare name, no configuration or usage detail `livekit-agents@1.8.1`

06 **LiveKit inference realtime model in OpenAI plugin** NEW

25

Adds a LiveKit inference realtime model to the OpenAI plugin.

— Bare description with no parameters or setup `livekit-agents@1.8.1`

WAS THIS USEFUL? 👍 👎

Google ADK

Feedback ⓘ 1 RELEASE · 2026-09-10 NOTES ↗ RANK

The Agent Development Kit is Google's code-first Python toolkit for building, evaluating, and deploying agents, with built-in tools, sessions, and deployment to Vertex AI.

> ADK v2.9.0 adds automatic model failover, YAML-defined graph workflows, a LiveKit voice/telephony runner, and MCP SDK 2.x support, alongside several breaking changes to session, GCS and workflow-resume behavior.

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

01 **transfer_reason context on agent handoffs** NEW 75

Adds a `transfer_reason` parameter to `transfer_to_agent` to carry context across agent handoffs and prevent delegation loops.

Prevent agent-transfer loops by supplying a `transfer_reason` so the receiving agent knows why it was handed the conversation.

python

```python
from google.adk.tools import transfer_to_agent

await transfer_to_agent(
    agent_name="billing_agent",
    transfer_reason="User asked about invoice #4821, outside my scope.",
)
```

— Runnable example plus clear stated purpose. v2.9.0

02 **MCP SDK 2.x server support** BREAKING 70

Supports MCP SDK 2.x servers alongside existing 1.x servers, with 1.x remaining the default resolved version. Under MCP SDK 2.x, unknown fields a server sends on a `CallToolResult` or a tool declaration read via `raw_mcp_tool` are discarded during SDK validation before ADK sees them; only `_meta` (a declared field) still arrives, so servers relying on vendor extensions must move them under `_meta`.

— Names exact affected fields but gives no fix code example. v2.9.0

03 **Restricted local file access in GCS tools** BREAKING 70

GCS tools now restrict local file access to the directory named by `local_file_root` and refuse all local file access when `local_file_root` is absent.

— Names the exact config key and its before/after effect. v2.9.0

04 **YAML-defined graph workflows and resume behavior** BREAKING 60

Supports loading ADK 2.0 graph workflows from declarative YAML configuration. A workflow node that failed now re-executes when the workflow resumes, instead of replaying as though it had completed — node bodies with external side effects will repeat those side effects on every resume.

— Explains before/after resume behavior but no migration example. v2.9.0

05 **Strict session validation in InMemorySessionService** BREAKING 60

`InMemorySessionService` now raises `SessionNotFoundError` when an event is appended to a session it does not hold, instead of silently discarding the event.

— Names the exception and trigger but gives no handling example. v2.9.0

THINNER COVERAGE BELOW

06 **Pre-authorized artifact URLs in GcsArtifactService** NEW 50

Adds `get_authenticated_url` and `get_signed_url` methods to `GcsArtifactService` for pre-authorized artifact access.

— Names both methods but no usage example or scope detail. v2.9.0

07 **Telemetry tracking and OTLP export** NEW 50

Adds skill-load tracking and script execution instrumentation to the telemetry subsystem, and exports logs over OTLP to `telemetry.googleapis.com`.

— Names the OTLP endpoint but no configuration steps shown. v2.9.0

08 **Automatic model failover with FallbackModel** NEW 45

Adds a `FallbackModel` class for automatic failover to backup models when a primary model encounters errors.

— Names the class but no usage example or trigger conditions given. v2.9.0

09 **Evaluation set matching and reporting** IMPROVED 45

Adds an `ignore_args` option to tool trajectory evaluation to relax argument matching in eval sets, and agent evaluations now print detailed results when all tests pass.

— Names the flag but reporting change is only described in prose. v2.9.0

10 **FunctionNode auto-inference in NodeTool** NEW 30

Supports `FunctionNode` auto-inference in `NodeTool`.

— Single-line description with no usage shown. v2.9.0

11 **LiveKit voice and telephony runner** NEW 25

Adds a LiveKit runner integration for building voice and telephony agents.

— Bare description with no config or example. v2.9.0

12 **Hyphenated column names in BigQuery tools** IMPROVED 25

Supports hyphenated column names in BigQuery tools.

— Minimal description of the behavior change. v2.9.0

└──▷ **BREAKING ON UPGRADE**

! A workflow node that failed now re-executes when the workflow resumes, instead of replaying as though it had completed — node bodies with external side effects will repeat those side effects on every resume.

! GCS tools now restrict local file access to the directory named by `local_file_root` and refuse all local file access when `local_file_root` is absent.

! `InMemorySessionService` now raises `SessionNotFoundError` when an event is appended to a session it does not hold, instead of silently discarding the event.

! Under MCP SDK 2.x, unknown fields a server sends on a `CallToolResult` or a tool declaration read via `raw_mcp_tool` are discarded during SDK validation before ADK sees them; only `_meta` (a declared field) still arrives. Servers relying on vendor extensions must move them under `_meta`.

WAS THIS USEFUL? 👍 👎

Arcade

Feedback ⓘ 1 RELEASE · 2026-09-10 BLOG ↗ RANK

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

> Arcade's sole release this window introduces Execution Tool Logs, a per-run audit trail for every agent tool call, complete with input/output inspection, filtering, RBAC controls, and a programmatic API.

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

01 **Execution Tool Logs for agent tool calls** NEW 80

Arcade adds Execution Tool Logs, which record every tool and toolkit (including version) that ran in a project, the end user it ran for, timestamps (created, started, finished), retry attempts, and success or failure outcome. Executions can be filtered by tool name, toolkit, outcome, time range, user, or error text, and individual runs can be drilled into for a full timeline. Show inputs and Show outputs toggles on execution records let project admins inspect the exact arguments a tool received and what it returned, including full provider error messages. An API is provided to list, count, and retrieve individual executions programmatically, with admin permission required to access input and output payloads. RBAC restricts execution detail so project members see the list, status, timing, and errors, while inputs and outputs are limited to project admins in both the UI and API; records are stored encrypted at rest and in transit with a configurable retention window on Arcade Cloud.

Investigate a failing tool call by filtering the execution log to failures for a specific tool and user, then reviewing the exact inputs and provider error without digging through agent-side logs.

> 📍 In Arcade Cloud, go to your project › Tool Executions, set the outcome filter to failure, filter by tool name and affected user, open the run, then toggle Show inputs and Show outputs to see the arguments and full error the provider returned.

— Names UI toggles, RBAC scope, retention and API but no exact endpoint paths

launch-20260910-70f5950f

WAS THIS USEFUL? 👍 👎

Cloudflare Agents

Feedback ⓘ 3 RELEASES · 2026-09-11 NOTES ↗

RANK

The Cloudflare Agents SDK builds stateful agents on Workers and Durable Objects, with persistent state, scheduling, WebSockets, and MCP server support.

> Cloudflare Agents replaces its storage and scheduling core this window: a new `agents/sessions` engine (branching, compaction, full-text search) and `agents/context` module back conversation and prompt state, a durable Lifecycle job queue and alarm-memory-limit recovery system replace the old pull-based alarms, and both `AIChatAgent` and `Think` are migrated onto the new sessions/streams model with atomic stream settlement — while the experimental memory stack and legacy alarm APIs are removed outright.

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

01 **Durable job queue and alarm memory-limit recovery in Lifecycle**

NEW 96

Lifecycle gains a durable job queue driven as an alarm event loop: this.lifecycle.jobs.push({ id, fn, time }) schedules serializable callbacks, and capabilities handle them via onJob({ job }: LifecycleJobContext), returning nothing (complete), `{ rescheduleAt }` (suspend), or `'yield'` (wake immediately). Hosts get onJobError() for terminal job failures and onAlarmMemoryLimit() for alarm-memory-limit policy, with Lifecycle.install(host, { maxAlarmMemoryLimitStrikes }) now the canonical way to set the strike budget (replacing `setLifecycleAlarmMemoryLimitStrikes`) and capabilities exposing `onMemoryLimit`. Recovery-loop membership moves to the job row via `LifecycleJobPushOptions.recoveryLoop`, ResumableStream.progressMarker() derives forward-progress from durable stream-log segments (with creditProgress() for sub-agent forwarding), and `DEFAULT_CHAT_RECOVERY_MAX_WORK` rises from 1000 to 10000 segments. Existing `cf_agents_schedules` rows are migrated into the new `cf_agents_jobs` table on startup and the legacy table is dropped.

Declare a durable recurring sweep job from a LifecycleCapability so it survives isolate restarts and retries automatically.

```typescript
import { LifecycleCapability, LifecycleJobContext } from
'agents';

class Cleanup extends LifecycleCapability {
  async scheduleSweep(time: number) {
    await this.lifecycle.jobs.push({ id: 'sweep', fn: 'sweep',
time });
  }

  onJob({ job }: LifecycleJobContext) {
    // Return nothing to mark complete, { rescheduleAt } to
suspend,
    // or 'yield' to wake again immediately.
    console.log('Running sweep job', job);
  }
}
```

React to alarm memory-limit strikes in a capability without coupling to Agent internals, using the new `onMemoryLimit` hook via `Lifecycle.install`.

```typescript
import { Lifecycle } from 'agents';
import { AIChatAgent } from '@cloudflare/ai-chat';

export class MyChatAgent extends AIChatAgent<Env> {
  constructor(ctx: DurableObjectState, env: Env) {
    super(ctx, env);
    Lifecycle.install(this, { maxAlarmMemoryLimitStrikes: 3 });
  }

  onAlarmMemoryLimit(ctx: { job?: unknown }) {
    console.warn('Memory limit strike recorded', ctx);
  }
}
```

— Enumerates every method, hook and config, backed by runnable examples `agents@0.23.0` `@cloudflare/ai-chat@0.12.0` `@cloudflare/think@0.18.0`

02 **RoutedAgents capability for per-chat hub topology** NEW

85

Adds `RoutedAgents` (importable from `agents/routing`), a Lifecycle capability for the 'user hub with one Durable Object per chat' topology — create(), list(), setMetadata(),

get(), and delete() manage a durable catalog of independent top-level Agents routed by public ID.

Model a per-user hub that owns many independent chat Durable Objects, routing browser requests to each chat without waking the hub for WebSocket frames.

typescript

```typescript
import { Agent } from 'agents';
import { RoutedAgents } from 'agents/routing';
import { ChatAgent } from './chat-agent';

class UserAgent extends Agent<Env> {
  readonly chats = new RoutedAgents<ChatAgent, { title: string }>
({
    namespace: this.env.ChatAgent,
    route: 'chats',
  });

  constructor(ctx: DurableObjectState, env: Env) {
    super(ctx, env);
    this.lifecycle.use(this.chats);
  }

  async createChat(title: string) {
    return this.chats.create({ title });
  }

  async listChats() {
    return this.chats.list();
  }
}
```

— Named methods plus a full runnable code example `agents@0.23.0`

03

AIChatAgent messages migrated onto agents/sessions

BREAKING

85

`AIChatAgent` messages are now stored through `agents/sessions`, migrating existing `cf_ai_chat_agent_messages` rows into a linear Sessions chain on first wake; `this.messages` is empty until `onStart`. Wake-time hydration is capped via `hydrationByteBudget` (32 MiB default), while `maxPersistedMessages` now counts stored rows rather than the hydrated window, trimming the oldest stored messages even when the transcript exceeds the budget. The `get-messages` response is streamed instead of serializing the full transcript, and oversized messages are split across continuation rows and reassembled transparently on read.

Read conversation history only after the agent wakes, now that `this.messages` is populated in `onStart` rather than the constructor.

```typescript
import { AIChatAgent } from '@cloudflare/ai-chat';

export class MyChatAgent extends AIChatAgent<Env> {
  async onStart() {
    await super.onStart();
    // this.messages is now populated — safe to inspect history
here
    console.log(`Resuming with ${this.messages.length}
messages`);
  }
}
```

— Migration mechanism and config named, with a code example `@cloudflare/ai-chat@0.12.0`

04 **Rollover stream chunks with atomic settle** IMPROVED

82

Replaces the flat `cf_agents_stream_chunks` append log with mutable rollover blocks: each block row grows up to 256 KB before a new one opens, reducing a stream of thousands of chunks to a handful of rows. writer.close({ commit, discard }) and error(reason, { … }) settle a stream and delete its rows atomically in one SQLite transaction, guaranteeing a crash leaves either the live stream or the finished message, never neither; Think settles the same way plus Session.__DO_NOT_USE_WILL_BREAK__sync().upsert(). The backing streams capability is exposed as `readonly streams` on `AIChatAgent` and Think, letting any streams.read() consumer on the same Durable Object observe chat streams. `cleanupStreamBuffers` and `STREAM_CLEANUP_DELAY_SECONDS` are removed from `agents/chat`.

Observe in-flight chat streams from another consumer on the same Durable Object using the newly exposed `readonly streams` capability.

```typescript
import { AIChatAgent } from '@cloudflare/ai-chat';

export class MyChatAgent extends AIChatAgent<Env> {
  async onMessage(message: string) {
    // Tap the live stream for the current turn from within the
same DO
    const reader = await this.streams.read('<stream-id>');
    for await (const chunk of reader) {
      console.log('chunk', chunk);
    }
  }
}
```

— Names storage mechanism and removed APIs, one code example `@cloudflare/ai-chat@0.12.0` `@cloudflare/think@0.18.0`

---

05 **Think conversation storage migrated onto agents/sessions**

80

BREAKING

Rewrites Think conversation storage onto `agents/sessions`: `assistant_messages`, `assistant_compactions`, and `assistant_config` rows are lifted into `cf_agents_session_*` tables and the old tables dropped, with no rollback path once an object has woken on this version. `hydrationByteBudget` default rises from 24 MiB to 32 MiB and becomes a hard ceiling charging each row its full stored size including attachments, with no message-count floor. Direct SQL queries to `assistant_messages` are no longer supported — use this.session.history() or getHistory() instead — and getRecentHistory()'s `minRecentMessages` and compactAfter()'s `tokenCounter` options are now accepted but ignored, while `sessionAttachments` configuration is removed.

— Full migration and API-removal detail but no runnable example `@cloudflare/think@0.18.0`

---

06 **Durable conversation storage via agents/sessions** NEW

78

Adds `agents/sessions`, a Lifecycle capability providing durable conversation storage: a tree of messages with branches and compaction overlays, streamed and byte-budgeted reads via getRecentHistory(), and full-text search built on first call to search(). appendMessage(msg, { source: 'client' }) sanitizes reserved metadata, and messages exceeding the 1.5 MiB row budget are split across `cf_agents_session_message_chunks` continuation rows and reassembled transparently on read.

— Names methods and storage internals but no usage example `agents@0.23.0`

---

07 **Prompt-context management via agents/context** NEW 75

Adds the `agents/context` module, exposing `ContextBlocks`, `AgentContextProvider`, `AgentSearchProvider`, and skill providers for prompt-context management separate from conversation storage. Think rewrites prompt context onto this module, adding a configureContext() hook for declaring prompt blocks and exposing `this.context` (a `ContextBlocks`) as the primary way to read context.

Use configureContext() and `this.context` in a Think subclass instead of the deprecated withContext() / `this.session` pattern.

typescript

```
import { Think } from '@cloudflare/think';
import { ContextBlocks } from 'agents/context';

class MyAgent extends Think<Env> {
  configureContext() {
    return [
      { id: 'system', content: 'You are a helpful assistant.' }
    ];
  }

  async onStart() {
    await super.onStart();
    const block = this.context.getBlock('system');
    console.log('System prompt loaded:', block?.content);
  }
}
```

— Names the module's exports and shows a working subclass pattern `agents@0.23.0` `@cloudflare/think@0.18.0`

08 **Media eviction preserved to Workspace in Think** IMPROVED 71

Moves media eviction from a storage-layer concern to a context-window technique: evicted media is preserved at `/attachments/evicted/<messageId>-<n>.<ext>` in the Workspace with its real MIME type, and a marker like `[evicted image/png, 812004 bytes; preserved at /attachments/evicted/<messageId>-<n>.png]` is left in context. `WorkspaceLike.writeFileBytes` is now optional, so a custom workspace without it keeps working with eviction logging once and standing down, and `MediaEvictionConfig.externalizeToWorkspace` is deprecated and ignored since evicted bytes are always preserved.

— Concrete path and marker format but no command shown `@cloudflare/think@0.18.0`

09 **Removal of experimental memory stack and pull-based alarms**

70

BREAKING

The experimental memory stack is removed: `agents/experimental/memory/session` and `agents/experimental/memory/utils` subpaths no longer exist, taking Session.create(), `SessionManager`, `PostgresSessionProvider`, `PostgresContextProvider`, `PostgresSearchProvider`, `R2SkillProvider`, and the `SessionProvider` interface with them. The pull-based alarm-contribution model is also removed — capability getNextAlarm() / onAlarm(), host getNextAlarm(), `LifecycleServices.alarms` (`rearm` / `disabled`), and `AlarmContribution` are gone — along with Scheduler's `__DO_NOT_USE_WILL_BREAK__handleAlarmMemoryLimit` escape hatch, superseded by the new onAlarmMemoryLimit() host hook.

– Lists every removed symbol but no migration steps beyond the hook name `agents@0.23.0` `@cloudflare/think@0.18.0`

THINNER COVERAGE BELOW

10 **Dynamic sub-agent management via dynamicAgents** NEW

59

Adds `this.dynamicAgents.{get,abort,delete,has,list}` methods and the `DynamicAgentClass` / `DynamicAgentStub` types, extracting facet (sub-agent) machinery into `packages/agents/src/dynamic-agents/` as a Lifecycle capability (`capabilityId: 'dynamic-agents'`).

– Concrete API names but no example of use `agents@0.23.0`

11 **Fallback dispatch order in Lifecycle.use()** NEW

43

Adds a `{ fallback: true }` option to Lifecycle.use(), allowing a capability to be dispatched after all non-fallback capabilities regardless of installation order.

– Names the exact option but gives no example `agents@0.23.0`

└──▷ BREAKING ON UPGRADE

! The experimental memory stack is removed: `agents/experimental/memory/session` and `agents/experimental/memory/utils` subpaths no longer exist, taking Session.create(), `SessionManager`, `PostgresSessionProvider`, `PostgresContextProvider`, `PostgresSearchProvider`, `R2SkillProvider`, and the `SessionProvider` interface with them.

! The pull-based alarm-contribution model is removed: capability getNextAlarm() / onAlarm(), host getNextAlarm(), `LifecycleServices.alarms` (`rearm` / `disabled`), and `AlarmContribution` are gone.

! `Scheduler`'s `__DO_NOT_USE_WILL_BREAK__handleAlarmMemoryLimit` escape hatch is removed; alarm memory-limit policy now flows through the new onAlarmMemoryLimit() host hook.

- ! The `cf_agents_schedules` table is dropped on startup after its rows are migrated into `cf_agents_jobs`; any tooling or queries that read `cf_agents_schedules` directly will fail.
- ! `this.messages` is empty in the constructor and remains so until `onStart`; subclasses that read `this.messages` in their constructor now see `[]` — move such reads to `onStart` or later.
- ! Once an `AIChatAgent` object migrates its `cf_ai_chat_agent_messages` rows to the Sessions chain, rolling back to the previous release leaves it with an empty conversation (rolling forward again is safe).
- ! `cleanupStreamBuffers` and `STREAM_CLEANUP_DELAY_SECONDS` are removed from `agents/chat`.
- ! `Agent.onAlarmMemoryLimit`'s policy relay, the `_cf_recoveryAlarmCallbacks` template hook, `Scheduler.applyMemoryLimitPolicy`, and `setLifecycleAlarmMemoryLimitStrikes` are removed.
- ! `AIChatAgent` and `Think` now require `agents >= 0.23.0`.
- ! Storage migrates on first wake and cannot be rolled back: `assistant_messages`, `assistant_compactions`, and `assistant_config` rows are lifted into `cf_agents_session_*` tables and the old tables are dropped; an object that has woken on this version has an empty conversation if you roll back to the previous release.
- ! Existing `cf_agents_schedules` rows are migrated into `cf_agents_jobs` on startup and the legacy table is dropped.
- ! Existing `cf_agents_stream_chunks` rows are folded into rollover blocks lazily and the table is dropped once empty.
- ! `hydrationByteBudget` default changed from 24 MiB to 32 MiB and is now a hard ceiling with no message-count floor, so an unusually large recent window can hydrate fewer than four messages.
- ! The pull-based alarm-contribution model is removed: LifecycleCapability.getNextAlarm(), LifecycleCapability.onAlarm(), host getNextAlarm(), `LifecycleServices.alarms` (`rearm` / `disabled`), and `AlarmContribution` are gone.
- ! `Scheduler.__DO_NOT_USE_WILL_BREAK__handleAlarmMemoryLimit` is removed; alarm memory-limit handling now goes through the new onAlarmMemoryLimit() host hook.
- ! `sessionAttachments` configuration is removed; media layout in Sessions is automatic and lossless.
- ! getRecentHistory(budget, minRecentMessages): the `minRecentMessages` second argument is accepted and ignored.
- ! compactAfter(threshold, { tokenCounter }): the `tokenCounter` option is accepted and ignored.
- ! `MediaEvictionConfig.externalizeToWorkspace` is deprecated and ignored; evicted bytes are always preserved to the Workspace.
- ! Direct SQL queries to `assistant_messages` are no longer supported; use this.session.history() or getHistory() instead.
- ! Think requires `agents >=0.23.0`.

WAS THIS USEFUL?  

Mastra

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ RANK

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

> Mastra 1.66 focuses on observability and reliability: a new first-chunk streaming timeout with automatic model fallback, bulk deletion and richer filtering for traces/feedback/scores, typed span introspection helpers, and pagination improvements for message and experiment-result listing, plus new Observational Memory transform hooks.

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

01 First-chunk streaming timeout with model fallback NEW

85

Adds `modelSettings.timeout.firstChunkMs` to bound how long a streaming model call may take to produce its first content chunk; exceeding the limit raises a `MastraTimeoutError` with `timeoutType: 'firstChunk'` and advances to the next fallback model listed in `models`.

Fail a streaming agent call fast when the model stalls before producing its first real content, then fall back to an alternate model.

typescript

```typescript
const result = await agent.stream('Summarize this document', {
  modelSettings: {
    timeout: {
      firstChunkMs: 5000
    }
  },
  models: ['openai:gpt-4o', 'anthropic:claude-3-5-sonnet']
});
```

— Runnable example shows the config key and fallback behaviour `@mastra/core@1.66.0`

02 Batch deletion of feedback and scores NEW

75

Adds deleteFeedback() and deleteScores() to observability storage, accepting a batch of ids plus optional `organizationId` / `resourceId` tenant scope, implemented idempotently across in-memory, ClickHouse, DuckDB, and Postgres backends.

Bulk-delete stale feedback entries scoped to a specific organization to keep observability data clean.

```typescript
await observability.deleteFeedback({
  feedbackIds: ['feedback-1', 'feedback-2'],
  organizationId: 'org-acme'
});

await observability.deleteScores({
  scoreIds: ['score-9', 'score-10'],
  organizationId: 'org-acme'
});
```

— Named functions, params and backends, with a runnable example `@mastra/core@1.66.0`

03 **Typed span introspection helpers** NEW 75

Adds `isSpanRecordOfType` type guard and `SpanType` enum (importable from `@mastra/core/observability`) for narrowing typed `input`, `output`, and `attributes` on `AGENT_RUN`, `MODEL_GENERATION`, `MODEL_STEP`, and `MODEL_INFERENCE` spans without casting. Also adds `describeSpanInput`, `describeSpanOutput`, and `describeSpanError` helpers that return payloads tagged by content kind (`messages`, `agent-run-resume`, `interrupted`, `model-generation-result`, `json`, ...) so UIs can switch on `type` at render time.

Narrow a stored span to its typed input and attributes without a cast, using the new `isSpanRecordOfType` guard.

```typescript
import { SpanType, isSpanRecordOfType } from
'@mastra/core/observability';

if (isSpanRecordOfType(span, SpanType.MODEL_GENERATION)) {
  console.log(span.attributes?.usage);   // UsageStats |
undefined
  console.log(span.input?.messages);     // MessageListInput
}
```

— Many named symbols; example covers only one of the two helper sets `@mastra/core@1.66.0`

04 **Tags filter on experiment result listing** NEW 75

Adds a `tags` filter to dataset.listExperimentResults() and `GET /api/datasets/:datasetId/experiments/:experimentId/results`; pass repeated query params (`?tags=regression&tags=p0`) over HTTP.

— Named endpoint with method, path and exact query param usage `@mastra/core@1.66.0`

05 **Paginated message listing via Agent Controller** NEW

70

Adds storage-backed pagination, ordering, filtering, and message includes to Agent Controller message listing via session.listMessages(threadId, { perPage, page }).

— Function signature named; no worked example given `@mastra/core@1.66.0`

06 **Expanded trace query filtering in queryTraces()** NEW

65

Adds span-level filter predicates for name, model/provider, duration, outcome, identity, and version lineage via the `where.spans` field, portable top-level `metadata.*` string predicates with key/value validation inside recursive Boolean expressions, `where.feedback` predicates to filter traces by related feedback values, and richer `scores.some` / `scores.none` predicates covering scorer versions, sources, timestamps, span anchoring, and version lineage.

— All new predicate fields named but no worked example `@mastra/core@1.66.0`

07 **includeTotal flag for memory store listing** IMPROVED

60

Adds an optional `includeTotal` flag to memoryStore.listMessages() so callers can skip the count query when only `messages` and `hasMore` are needed; defaults to `true`.

— Named flag and default, but no example of use `@mastra/core@1.66.0`

08 **Observational Memory transform hooks** NEW

60

Adds `beforeObservation`, `afterObservation`, `beforeReflection`, and `afterReflection` async transform hooks to `@mastra/memory` Observational Memory, enabling filtering, redaction, or reshaping of messages and observations before model calls or persistence.

— Hook names and purpose given, no example implementation `@mastra/core@1.66.0`

THINNER COVERAGE BELOW

09 **Worker topology reporting** NEW

45

Adds Mastra.getWorkerConfig() to report the active worker topology and serializable instance-level settings for deployment topology comparison.

— Named method but no detail on output shape or usage `@mastra/core@1.66.0`

10 **Reduced Agent.stream() tracing overhead** IMPROVED

35

Reduces Agent.stream() tracing overhead by resolving the observability logger and metrics contexts once per model step rather than once per streamed chunk.

— Internal performance change with no measurable numbers `@mastra/core@1.66.0`

WAS THIS USEFUL? 👍 👎

DEPLOY

◆ Agentic Infrastructure

Together AI

Feedback ⓘ 2 RELEASES · SEEN 2026-09-10 → 2026-09-11

NOTES ↗ RANK

Run, train, and serve open-source AI models on Together AI.

> Together AI's biggest addition this window is preemptible compute for GPU clusters — discounted, sub-hourly-billed nodes controllable via CLI, API, and Kubernetes primitives — alongside support for uploading and serving custom LoRA adapters on dedicated endpoints.

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

01 **Preemptible compute for GPU clusters** NEW 97

Together GPU Clusters gain preemptible compute (public preview), a second compute type billed at a flat 50% of the on-demand rate with sub-hourly billing metered every one to two minutes and no spot-market price fluctuation. Preemptible nodes join the same Kubernetes cluster as standard nodes (no separate cluster type needed) and can be added at create time or later via the `--num-preemptible-gpus` flag on `tg beta clusters create` and `tg beta clusters update`, alongside the required standard `num_gpus` count. The Together API exposes `desired_preemptible_gpus` and `allocated_preemptible_gpus` fields to report requested vs. live capacity, and delivers preemption notices as Kubernetes events with `reason=TogetherPreempted` and via `node_lifecycle_events`, giving workloads up to `terminationGracePeriodSeconds` (max 300s) to checkpoint before removal. Preemptible nodes are labeled `together.ai/compute-class=preemptible` so workloads can target discounted capacity via `nodeSelector` while keeping coordinators and serving replicas on standard nodes.

Spin up a new cluster with both standard and preemptible GPUs to run batch fine-tuning jobs at half the cost.

```
$ tg beta clusters create \
  --name my-cluster --region us-central-2 --gpu-type RTX_6000_PCI
\
  --cluster-type KUBERNETES --num-gpus 8 --billing-type ON_DEMAND
\
  --num-preemptible-gpus 8 \
  --project-id <project-id>
```

Scale preemptible capacity up on a running cluster to absorb a batch inference burst, then scale it back down when done.

```
$ tg beta clusters update --num-preemptible-gpus 16
```

Pin interruptible worker pods to preemptible nodes while keeping coordinators on standard nodes, and give pods the full drain window to checkpoint on SIGTERM.

```yaml
nodeSelector:
  together.ai/compute-class: preemptible
terminationGracePeriodSeconds: 300
```

— Names flags, API fields, event reasons, labels, and grace period value.

launch-20260910-ad82b592 snapshot-20260911

THINNER COVERAGE BELOW

02 **Custom LoRA adapter uploads on dedicated endpoints** NEW

35

Dedicated endpoints now support uploading and serving custom LoRA adapters sourced from Hugging Face or S3.

— States the capability but gives no command, endpoint, or limit. snapshot-20260911

WAS THIS USEFUL? 👍 👎

ArcBox

Feedback ⓘ 1 RELEASE · 2026-09-11 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's fleet-agent now captures Docker and VM runner output per job and writes it to dedicated, locatable log files.

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

01 **Per-job log files in fleet-agent** NEW 55

The fleet agent now captures output from both Docker and VM runners and writes each job's output to its own dedicated log file, surfacing the log paths so practitioners can locate and inspect per-job logs.

— Describes mechanism and behaviour but no exact path or flag given `fleet-agent-v0.1.5`

WAS THIS USEFUL? 👍 👎

Eigen Labs Darkbloom

Feedback ⓘ 1 RELEASE · 2026-09-11

NOTES ↗ CODE ↗ RANK

Private Inference Network on Idle Macs

> Darkbloom v0.9.2 adds Nemotron Lightning model serving with native reasoning and tool support, turns on Gemma QAT prefix caching by default (invalidating existing checkpoints), and introduces adaptive Gemma MTP alongside a zero-downtime assistant activation mechanism.

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

01 **Gemma QAT SSD prefix caching by default** BREAKING 80

Enables Gemma QAT SSD prefix caching by default for exact `gemma-4-26b-qat-4bit`, with authenticated complete paged checkpoints preserving tenant, model, prompt, binary, metallib, and numerical-state identity checks; other Gemma artifacts and GPT-OSS caching remain opt-in. Note: existing 0.9.1 Gemma QAT prefix checkpoints are not reused after upgrading to v0.9.2 — a new binary starts a new checkpoint identity, so the prefix cache is cold on first run.

— Names exact model ID and includes concrete upgrade migration note v0.9.2

02 **Zero-downtime assistant activation** NEW 80

Downloads and verifies the optional assistant while the current engine serves, advertises `reloading` status, drains accepted work before swapping, and returns transient 503 `slot_state` refusals to racing requests during the swap.

— Names concrete status field and HTTP response code for integrators v0.9.2

03 **Nemotron Lightning model serving** NEW 70

Adds serving for Nemotron Lightning, admitting the three explicitly qualified registry/Hugging Face IDs on existing network and standalone paths, with native paged KV, complete encrypted prefix reuse, native activation/KV precision, FP32 persistent Mamba state, and adaptive MTP depth up to seven.

— Detailed mechanism but no exact model IDs or commands given v0.9.2

04 **Adaptive Gemma MTP decode selection** NEW 60

Adds adaptive Gemma MTP that automatically resolves the catalog assistant for the QAT target and selects ordinary decode or one draft token from measured committed output and elapsed time, with support for temperature/top-p/top-k/min-p target-prefix sampling.

— Mechanism described but nothing a reader can directly invoke v0.9.2

THINNER COVERAGE BELOW

05 **Native reasoning and tool support for Nemotron** NEW

50

Adds native reasoning and tool support for Nemotron, separating reasoning before tool parsing and validating required/named calls before publishing through the existing encrypted response stream.

— Explains behavior but no configuration surface named v0.9.2

06 **Immutable assistant revision support with R2 fallback** NEW

45

Adds catalog-declared immutable Hugging Face assistant revision support with checksum-verified R2 fallback and jittered fetch retries for assistant downloads.

— Names R2 fallback and checksum verification but little further detail v0.9.2

└──▷ **BREAKING ON UPGRADE**

! Existing 0.9.1 Gemma QAT prefix checkpoints are not reused after upgrading to v0.9.2 — a new binary starts a new checkpoint identity, so the prefix cache is cold on first run.

WAS THIS USEFUL? 👍 👎

Google Gen AI JavaScript SDK

Feedback ⓘ

1 RELEASE · 2026-09-10 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 support for initializing a client directly from environment configuration.

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

01 **Environment-based client initialization** NEW 45

Adds `from_environment` support for copying environment configuration into a `GoogleGenAI` client instance, enabling client initialization directly from ambient environment variables.

– Names the feature and mechanism but no usage example v2.22.0

WAS THIS USEFUL? 👍 👎

Google Gen AI Python SDK

Feedback ⓘ

1 RELEASE · 2026-09-10 NOTES ↗ RANK

The Google Gen AI Python SDK provides Python interfaces to Gemini model generation, streaming, tools, and embeddings.

> Google Gen AI Python SDK v2.23.0 adds dynamic turn completion for live sessions and a client construction helper that copies environment configuration.

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

01 **Dynamic turn completion in live sessions** NEW 45

Adds `interaction_status` field support in `live.py` receive() to enable dynamic turn completion in live sessions.

— Names exact field and file but no usage example v2.23.0

02 **Environment-copying client construction** NEW 40

Adds `from_environment` support for copying client environment configuration.

— Names the method but no further mechanism detail v2.23.0

WAS THIS USEFUL? 👍 👎

HeyGen HyperFrames

Feedback ⓘ 1 RELEASE · 2026-09-10

NOTES ↗ RANK

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

> HyperFrames v0.8.34 expands its Registry with new carousel blocks while significantly cutting CPU overhead in the Studio editor, both at idle and during composition scrubbing.

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

01 **Idle CPU reduction in paused Studio editor** IMPROVED

55

Reduces idle CPU in a paused Studio editor from ~10% of a core to ~3%, dropping scheduled frames from 301 per second to 6.

— Concrete before/after numbers but no user action v0.8.34

02 **Faster scrubbing via cached ancestor measurement** IMPROVED

50

Cuts main-thread work for scrubbing a large composition by roughly half by measuring each ancestor once per redraw instead of once per element.

— Explains mechanism and scope but no reader action v0.8.34

03 **Image carousel blocks in Registry** NEW

35

Adds 25 image carousel blocks to the Registry.

— Names a count and area but no further mechanism v0.8.34

WAS THIS USEFUL? 👍 👎

Anthropic TypeScript SDK

Feedback ⓘ

1 RELEASE · 2026-09-10 NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> The Anthropic TypeScript SDK added a new user profiles beta field, a new error code for the web_fetch tool, and support for mounting public GitHub repos in Managed Agents sessions without authentication.

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

01 **Public GitHub repo mounting in Managed Agents** NEW

60

Managed Agents sessions can now mount public GitHub repositories without providing an `authorization_token`.

– Names the field removed and the feature area clearly sdk-v0.125.0

THINNER COVERAGE BELOW

02 **User profiles beta field addition** NEW

52

Adds the `user-profiles-2026-09-04` beta value and an `external_user_details` field to the user profiles API.

– Names the beta value and field but no usage detail sdk-v0.125.0

03 **New error code for web_fetch tool** NEW

43

Adds a `content_too_large` error code to the `web_fetch` tool.

– Bare addition, no handling guidance given sdk-v0.125.0

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ 1 RELEASE · 2026-09-10

NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> Anthropic Python SDK v1.5.0 adds a direct way to convert Claude responses back into request parameters, accepts tool objects natively in core methods, and extends Managed Agents and error handling.

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

01 **Message-to-param conversion methods** NEW 75

Message.to_param() and BetaMessage.to_param() convert message objects directly into API-compatible parameter dicts, letting a completed assistant message be reused as an entry in a `messages` list for the next turn without manually rebuilding the dict.

Turn a completed assistant message back into a request parameter for multi-turn conversations without manually rebuilding dicts.

python

```python
import anthropic

client = anthropic.Anthropic()

# First turn
message = client.messages.create(
    model="claude-opus-5",
    max_tokens=1024,
    messages=[{"role": "user", "content": "What is 2+2?"}],
)

# Convert response to param and continue the conversation
conversation = [
    {"role": "user", "content": "What is 2+2?"},
    message.to_param(),
    {"role": "user", "content": "Now multiply that by 10."},
]

follow_up = client.messages.create(
    model="claude-opus-5",
    max_tokens=1024,
    messages=conversation,
)
print(follow_up.content)
```

– Includes a runnable multi-turn example using the new method. v1.5.0

02 **Tool objects accepted in core API methods** IMPROVED

60

`messages.create`, `parse`, `stream`, and `count_tokens` now accept tool objects directly instead of requiring manual dict construction.

– Names all four affected methods but no usage example given. v1.5.0

THINNER COVERAGE BELOW

03 **Managed Agents enhancements** NEW

55

Managed Agents gain `auto` mode tool permissions and support for mounting public GitHub repositories in a session without an `authorization_token`.

– Names both additions verbatim but no mechanism detail. v1.5.0

04 **New credentials error classes** NEW

45

Adds `CredentialsError` and `IdentityTokenFileError` exception classes for more granular credentials error handling.

– Names the two new classes but no usage detail. v1.5.0

05 **User profiles beta with external_user_details field** NEW

40

Adds the `user-profiles-2026-09-04` beta value and an `external_user_details` field to user profiles.

– Names beta value and field but no usage shown. v1.5.0

06 **content_too_large error code for web_fetch** NEW

35

Adds a `content_too_large` error code to the `web_fetch` tool.

– Bare naming of a single error code with no context. v1.5.0

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ 1 RELEASE · 2026-09-10

NOTES ↗ RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> OpenAI TypeScript SDK v7.14.0 adds support for the Live API.

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

01 **Live API support** NEW 25

The SDK adds the Live API to the OpenAI TypeScript/JavaScript client.

— Only names the feature, no mechanism or usage shown v7.14.0

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ 1 RELEASE · 2026-09-10 NOTES ↗

RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK v3.12.0 adds support for the Live API.

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

01 **Live API support** NEW 23

The SDK adds support for the Live API, extending the typed Python client to this new API surface.

— No mechanism, endpoints, or usage details given beyond the name. v3.12.0

WAS THIS USEFUL? 👍 👎

Microsoft ONNX Runtime

Feedback ⓘ 1 RELEASE · 2026-09-10

NOTES ↗ RANK

ONNX Runtime runs machine-learning models across hardware and operating systems with an open-source inference engine.

> ONNX Runtime 1.30.0 adds Go bindings for the C API, a batch of CUDA operators for generative-AI serving (paged attention, DeepSeek Engram, GatedDeltaNet, speculative XQA), FP4/FP8 QMoE kernel support, LoRA adapter loading, new CPU SIMD kernels, and several build-default changes affecting CUDA quantized-GEMM and CPU FP16 execution.

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

01 **FP4/FP8 QMoE kernel support** NEW 90

Adds an opt-in FP8 DeepGEMM MoE decode path for fixed-shape QMoE workloads on Hopper GPUs, enabled via `ORT_QMOE_FP4_DEEPGEMM=1` (default off, unavailable on Windows); FP4 QMoE kernels are now enabled by default in CUDA builds, with source builds able to opt out via `-Donnxruntime_USE_FP4_QMOE=OFF`.

Enable the FP8 DeepGEMM MoE decode path on a Hopper GPU to accelerate fixed-shape QMoE workloads without rebuilding from source.

```
$ ORT_QMOE_FP4_DEEPGEMM=1 python run_inference.py
```

— Exact env var, build flag, GPU target, and runnable command v1.30.0

02 **CUDA fpA-intB GEMM default kernel set change** BREAKING 85

CUDA fpA-intB builds now default to a compact kernel set (FP16 activations, INT4/INT8 weights, scale-only quantization, `block_size=32`); set `-Donnxruntime_USE_FPA_INTB_GEMM_FULL=ON` at source build time to retain BF16, zero-point, bias, larger-block-size, and native Hopper variants.

— Exact build flag and default behavior change stated v1.30.0

03 **CUDA generative-AI attention operators** NEW 70

Adds the `VarlenCausalConvWithState` CUDA operator for continuous batching with compact variable-length causal-convolution state updates; adds an `is_causal` attribute to `PagedAttention` plus INT4 paged KV caches with per-channel scales on CUDA; adds a compact `GatedDeltaNet` operator with BFloat16 support on CUDA; and

extends paged XQA with speculative decoding, query-to-KV head group size 6, head size 256 with FP16 caches, and native block tables for 128-token pages.

— Names several CUDA ops and parameters, no runnable example v1.30.0

THINNER COVERAGE BELOW

04 **Custom kernel and memory-import APIs** NEW 55

Adds `KernelContext::GetPreallocatedOutput` to access preallocated outputs from custom kernels, and extends memory importing with host-pointer support.

— Names exact API but no usage detail v1.30.0

05 **CPU SIMD kernel acceleration** IMPROVED 55

Adds fused CPU LinearAttention kernels for AVX-512, Arm64 NEON, and SVE, plus AVX2 LayerNorm/RMSNorm acceleration.

— Names instruction sets and kernels but no usage path v1.30.0

06 **CPU FP16 Gemm/MatMul fallback to FP32** IMPROVED 50

CPU-assigned FP16 `Gemm` and `MatMul` nodes without hardware acceleration now automatically fall back to FP32 execution.

— States exact node types and behavior change, no config needed v1.30.0

07 **WebGPU plugin EP packaging for Linux AArch64** IMPROVED 45

Adds WebGPU plugin EP packaging support for Linux AArch64; plugin versions advanced to WebGPU 0.4.0 and CUDA 0.2.

— Names platform and version bumps, no usage steps v1.30.0

08 **Go bindings for the C API** NEW 40

Adds Go bindings for the ONNX Runtime C API, enabling Go applications to run ONNX model inference directly.

— Names the new language binding but no usage example v1.30.0

09 **DeepSeek Engram contrib operators** NEW 30

Adds DeepSeek Engram contrib operators `EngramGate` and `NGramHashMapping`.

— Bare operator names with no behavioral detail v1.30.0

10 **LoRA adapter support with plugin EPs** NEW 25

Adds LoRA adapter support with plugin EP allocators.

— One-line mention with no mechanism detail v1.30.0

WAS THIS USEFUL?

Baseten

Feedback ⓘ 2 RELEASES · 2026-09-10 NOTES ↗ RANK

Baseten deploys and serves machine learning models as autoscaling inference endpoints, with a model API catalog and dedicated GPU deployments.

> Baseten added DeepSeek V4.1 Flash to its Model APIs catalog and introduced regional deployments for pinning replicas to a specific region.

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

01 **Regional deployments for replica placement** NEW 67

Deployments can now be restricted to run every replica in a specific Baseten region, selectable via the dashboard, the Baseten CLI, or the Management API, to meet compliance or data residency requirements or reduce latency.

— Names three access surfaces but no exact API/CLI syntax Regional deployments

THINNER COVERAGE BELOW

02 **DeepSeek V4.1 Flash on Model APIs** NEW 41

DeepSeek V4.1 Flash, a 552B-parameter multimodal model purpose-built for agentic and coding workflows, is now available through Baseten Model APIs.

— Names model and size but no usage detail DeepSeek V4.1 Flash available on Baseten

WAS THIS USEFUL?

Thinking Machines Lab Tinker

Feedback ⓘ

5 RELEASES · 2026-08-22 → 2026-09-04 NOTES ↗ RANK

Tinker provides a managed API for training and sampling language models while users control their training algorithms and data.

> Tinker's SDK and Cookbook releases this window added authentication and audit-log infrastructure (browser-based `tinker auth` login/logout, API key verification, expanded audit log with an `event_details` field), a weight-copying API, provenance tracking for training data, and new renderer support for GLM-5.3 and Qwen3.8, alongside a forecasting RL recipe and the removal of subprocess-isolated sampling.

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

01 **tinker auth CLI: login, API-key flag, logout** NEW 95

Adds `tinker auth login` to authenticate through the browser via WorkOS device authorization, mint an API key, and store it as the default credential; `tinker auth login --api-key` to store a manually pasted API key, which now verifies the key with the server before storing it and records it under its real key id with the server-held name and note; and `tinker auth logout` to remove the default credential, deleting a browser-minted key server-side or removing a manually entered one locally.

Authenticate to Tinker from a new workstation without managing API keys manually — the browser flow mints and stores the credential automatically.

```
$ tinker auth login
```

Authenticate in a headless CI environment by supplying a pre-generated API key directly.

```
$ tinker auth login --api-key
```

Authenticate with an API key and have it validated server-side before storage.

```
$ tinker auth login --api-key <your-api-key>
```

— Three commands named with exact CLI syntax and examples. SDK 0.26.0 SDK 0.26.2

02 **Weight copying via copy_weights()** NEW 90

Adds ServiceClient.copy_weights() to copy weights into the client's own project, returning the tinker path of the copy; accepts optional `ttl_seconds` and `weights_access_token` parameters.

Copy weights from another account's source into your project with a 24-hour expiry, for use in a short-lived experiment.

```python
copy_path = client.copy_weights(
    source_weights_path,
    ttl_seconds=86400,
    weights_access_token='<token>'
)
print(copy_path)
```

— Full signature, params, and runnable example given. SDK 0.26.2

---

03 **Audit log expansion and field deprecation** BREAKING

90

Expands RestClient.get_audit_log() with an `event_type` filter accepting `'projects'`, `'teams'`, or `'organizations'` to cover administrative activity alongside checkpoints, and adds an `event_details` field to `AuditLogEntry` carrying the full record — actor, resource, and action with its own parameters — for every event type. The `model_id`, `tinker_path`, and `purpose` fields on `AuditLogEntry` are now deprecated: they remain set for checkpoint events but are unset for the new `projects`, `teams`, and `organizations` event types, so code reading them unconditionally will see missing values.

Audit all team-level administrative actions in your org to investigate a permission change.

```python
entries = rest_client.get_audit_log(event_type='teams')
for entry in entries:
    print(entry.event_details)
```

— Names filter values, new field, deprecated fields, and impact. SDK 0.26.2

04 **Subprocess-isolated sampling removed** BREAKING 75

Subprocess-isolated sampling has been removed: `TINKER_SUBPROCESS_SAMPLING=1` no longer has any effect, and passing `subprocess_sampling=` to a `SamplingClient` is now an error — sample() and compute_logprobs() always run in the calling process.

— Clear before/after and named surfaces, no migration example. SDK 0.27.1

05 **Datum provenance tracking** NEW 75

Adds datum.with_provenance(model_input=..., loss_fn_inputs=...) to attach `PromptProvenanceSpan` / `SampledProvenanceSpan` runs that tile input tokens and loss rows, with partition validation at the call site.

— Named signature and mechanism, but no runnable example. SDK 0.26.0

06 **get_server_capabilities() trainable/sampleable fields** NEW 75

Adds `trainable` and `sampleable` fields to `SupportedModel`, exposed via get_server_capabilities() to identify which models can host training runs and which have sampling capacity.

Before launching a training run, check which models support training vs. sampling to pick the right target.

```python
capabilities = get_server_capabilities()
trainable_models = [m for m in capabilities if m.trainable]
sampleable_models = [m for m in capabilities if m.sampleable]
```

— Named fields and method with a working code example. SDK 0.26.0

07 **Qwen3.8 renderer and reasoning-turn enforcement** BREAKING 70

Adds a `qwen3_8` renderer family with tool calling, tool-declaration, and parsing support. Additionally, a renderer with reasoning disabled now refuses a turn that carries reasoning instead of silently dropping it — workflows that passed reasoning-bearing turns to a reasoning-disabled renderer will now fail explicitly.

— Names renderer and behavior change, no example. Cookbook 0.5.5

08 **get_console_url() on client and checkpoint responses** NEW 65

Adds get_console_url() on `ServiceClient` to return the current session's Console page URL, and on the responses from save_state() and save_weights_for_sampler() to

return the saved checkpoint's Console page URL.

— Named methods and behavior, but no usage example. SDK 0.27.1

09 **GLM-5.3 renderer support** NEW 60

Adds `glm5_3` renderer family with tool calling, reasoning, and `<|observation|>` / `<|user|>` turn terminator support for GLM-5.3 models, plus model metadata and hyperparameter lookup entries for GLM-5.3.

— Named renderer features but no usage example or command. Cookbook 0.5.6

THINNER COVERAGE BELOW

10 **Forecasting RL recipe on Prophet Arena** NEW 55

Adds a forecasting recipe using RL on binary prediction markets from the Prophet Arena subset, with a chronological train/test split and a Brier-score reward.

— Describes mechanism but no runnable recipe reference. Cookbook 0.5.6

11 **loss_fn_config accepts string values** IMPROVED 45

Allows `loss_fn_config` to accept string values alongside floats on forward(), forward_backward(), and their async variants.

— Names methods and change but no example or rationale. SDK 0.26.2

12 **Inkling-Small hyperparameter lookup entry** NEW 25

Adds Inkling-Small to the hyperparameter lookup tables.

— Bare one-line addition with no further detail. Cookbook 0.5.6

└──▷ **BREAKING ON UPGRADE**

! Subprocess-isolated sampling has been removed: `TINKER_SUBPROCESS_SAMPLING=1` no longer has any effect, and passing `subprocess_sampling=` to a `SamplingClient` is now an error — sample() and compute_logprobs() always run in the calling process.

! The `model_id`, `tinker_path`, and `purpose` fields on `AuditLogEntry` are deprecated: they remain set for checkpoint events but are unset for new `projects`, `teams`, and `organizations` event types — code that reads these fields unconditionally will see missing values for those event types.

! A renderer with reasoning disabled now refuses a turn that carries reasoning instead of silently dropping it — workflows that passed reasoning-bearing turns to a reasoning-disabled renderer will now fail explicitly.

WAS THIS USEFUL?  

OpenRouter

Feedback ⓘ 1 RELEASE · 2026-08-17 NOTES ↗ RANK

OpenRouter is a routing gateway that exposes hundreds of models from many providers behind one OpenAI-compatible API with failover and unified billing.

> OpenRouter's analytics tooling gets a significant upgrade, adding a queryable Analytics API alongside a more capable dashboard with team-level spend breakdowns, saveable charts, and log drill-down.

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

01 **Analytics API for spend and usage data** NEW 45

OpenRouter adds an Analytics API so teams can query spend and usage data from the terminal rather than only through the dashboard.

— Named API but no endpoint path or method given snapshot-20260911

02 **Analytics dashboard: team spend, saved charts, log drill-down**

IMPROVED 45

The analytics dashboard gains per-model spend breakdowns scoped to team-level usage, the ability to save charts so practitioners can preserve views they rebuild repeatedly, and click-through from any chart bar directly into the underlying request logs.

— Describes three dashboard additions but no exact UI paths snapshot-20260911

WAS THIS USEFUL? 👍 👎

scikit-learn

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ RANK

scikit-learn: machine learning in Python

> scikit-learn 1.9.1 adds support for Python 3.15 alongside bug fixes.

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

01 **Python 3.15 support** NEW 30

scikit-learn 1.9.1 adds support for running on Python 3.15.

— Names the version but no mechanism or details 1.9.1

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

ggml whisper.cpp

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗

RANK

whisper.cpp is a C/C++ port of OpenAI's Whisper speech recognition model, running transcription on CPU and GPU with no Python runtime.

> whisper.cpp v1.9.4 adds language detection in the server API, NPU encoder offload via a new VitisAI plugin, and Windows on ARM release binaries, alongside callback and determinism fixes.

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

01 **Detected language in server /detect response** IMPROVED

55

The server's `/detect` endpoint response now includes the detected language in its JSON payload.

— Names exact endpoint and field change but no schema example v1.9.4

02 **VitisAI NPU encoder offload plugin** NEW

55

Adds a VitisAI plugin (`vitisai`) enabling encoder offload to AMD Ryzen AI NPUs.

— Names plugin and hardware target, no setup steps given v1.9.4

03 **Earlier encoder-begin callback during language auto-detection**

IMPROVED

55

`encoder_begin_callback` is now called before language auto-detection runs, enabling callback-based control earlier in the pipeline than before.

— Names the callback and the behavioural fix precisely v1.9.4

04 **Windows on ARM release binaries** NEW

35

Adds Windows on ARM (WoA) support to the release CI job, producing WoA binaries.

— States what was added but no download or build detail v1.9.4

05 **Deterministic decoder re-seeding across calls** IMPROVED

35

Decoder 0 is now re-seeded between calls for more deterministic repeated inference.

— Explains mechanism briefly, no numbers or reproduction steps v1.9.4

WAS THIS USEFUL?

vMLX

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ RANK

The vMLX server runs compressed MLX models on Apple Silicon with disk caching, paged memory, continuous batching, and hybrid SSM scheduling.

> vMLX 1.6.57 adds Spark-X2.5 model integration and improves handling of mflux image generation exports.

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

01 **mflux image generation export handling** IMPROVED 55

Improves local-folder discovery and launch validation for mflux image generation/editing exports, including quantization metadata, progress, cancellation, and output history.

— Lists concrete behaviors but no exact commands or config keys v1.6.57

02 **Spark-X2.5 model integration** NEW 25

vMLX adds integration with the Spark-X2.5 model.

— Names the model but no mechanism or usage detail v1.6.57

WAS THIS USEFUL?

OpenWhispr

Feedback ⓘ 1 RELEASE · 2026-09-11 NOTES ↗ RANK

OpenWhispr is a desktop dictation and meeting-transcription app that supports local speech models and user-configured cloud providers.

> OpenWhispr 1.10.0 adds a terminal transcription command, three new bring-your-own-key streaming providers, managed enterprise speech-to-text on Azure, and a much faster local model on Apple silicon, alongside a new offline model, AI action and assistant-tool additions, and UI redesign work.

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

01 **openwhispr transcribe CLI command** NEW 85

Adds `openwhispr transcribe <file>` CLI command to run local transcription from the terminal using whichever local model the app is configured to use, with no audio leaving the machine.

Transcribe a local audio file from the terminal without sending audio to any cloud service, using whatever local model the app is already configured to use.

```
$ openwhispr transcribe meeting-recording.mp3
```

— Named CLI command with a runnable example v1.10.0

02 **Managed enterprise speech-to-text on Azure** NEW 70

Adds managed enterprise speech-to-text for workspaces on Azure OpenAI or AI Foundry, distributing a transcription deployment to every member with no API keys on user machines; requires version 1.10.0 or later for members.

— Describes mechanism and version requirement but no setup steps v1.10.0

03 **arm64-native ONNX Runtime speedup** IMPROVED 70

Ships arm64-native ONNX Runtime in sherpa-onnx, making local models up to three times faster on Apple silicon — an 11-second clip drops from 605 ms to 173 ms on Orukeet.

— Concrete before/after benchmark numbers given v1.10.0

04 **Cohere Transcribe offline model** NEW 65

Adds Cohere Transcribe as an offline model with 5.42% average WER across 14 languages, running on the same sherpa-onnx server as Parakeet.

— Concrete WER and language-count figures given v1.10.0

05 **Automatic background updates** NEW 60

Adds automatic background updates that install on next quit, opt-in for existing installs under Settings → System.

Opt existing installs into automatic background updates so the app self-updates without manual intervention.

> 📍 In the app, go to Settings → System and enable automatic updates.

— Gives exact UI path and opt-in mechanism v1.10.0

06 **New AI assistant tools and CLI bridge routes** NEW 60

Adds three new assistant tools enabling the AI to fetch a snippet by trigger, add or remove dictionary words, and create, replace, or delete snippets; the CLI bridge gains matching routes.

— Names each tool capability and CLI bridge tie-in v1.10.0

THINNER COVERAGE BELOW

07 **Orukeet local model as new default** NEW 55

Adds Orukeet as the new recommended local model — a multilingual Parakeet fine-tune running on sherpa-onnx — and the default selection for new installs choosing the local route.

— Names model and runtime, no setup instructions v1.10.0

08 **Default transcription model switched to gpt-transcribe** IMPROVED 55

Switches the OpenAI default transcription model from whisper-1 to gpt-transcribe (gpt-4o transcribe models), with old models remaining selectable.

— Clear before/after model names, no exact selection path v1.10.0

09 **Note editor redesign** IMPROVED 55

Redesigns the note editor with a combined date-and-attendees capsule, unified toolbar, and a split Share dialog containing Export.

— Names specific UI elements changed, no interaction detail v1.10.0

10 **New built-in AI actions for notes** NEW 50

Adds Detailed Notes and Follow-up email as two new built-in AI actions alongside Generate Notes, with every built-in prompt remaining user-editable and never overwritten by updates.

— Names the actions and editability behaviour, no exact UI path v1.10.0

11 **Streaming transcription for Deepgram, AssemblyAI and Gemini Live**

NEW

45

Adds streaming transcription support for Deepgram, AssemblyAI, and Gemini Live as bring-your-own-key providers for dictation.

— Names providers but no setup or config detail given v1.10.0

12 **Local Insights usage view** NEW

45

Adds a local Insights usage view backfilled from existing history, with an optional opt-in leaderboard for colleagues.

— Describes scope and opt-in behaviour, no navigation path v1.10.0

13 **Cmd+K search in main window** NEW

45

Adds ⌘K search across notes and transcripts in the redesigned main window top bar.

— Names the shortcut and location, but little else v1.10.0

14 **Arabic UI language support** NEW

35

Adds Arabic as the eleventh UI language with full right-to-left layout.

— Simple language addition, minimal detail beyond naming v1.10.0

15 **New reasoning models Astra and Fable 5.1** NEW

30

Adds Astra and Fable 5.1 as available reasoning models.

— Bare model names with no further context v1.10.0

WAS THIS USEFUL?  

DATA

◆ Vector DBs & RAG

Pinecone

Feedback ⓘ 1 RELEASE · SEEN 2026-09-11 NOTES ↗ RANK

Pinecone is a managed vector database that stores and queries embeddings for AI applications.

> Pinecone shipped a major 2026-07 API version that restructures index responses and introduces document-schema indexes via SchemaBuilder, alongside new metadata indexing controls, a Nexus knowledge-curation product, and a broad set of admin/lifecycle management endpoints for projects, organizations, API keys, and service accounts.

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

01 **Document-schema indexes via SchemaBuilder** NEW 95

Pinecone adds document-schema index creation via `SchemaBuilder` in the Python and JavaScript SDKs and REST, using a `schema.fields` structure with typed fields such as `dense_vector`, `sparse_vector`, and `string` with `full_text_search`, alongside a `deployment` object specifying `deployment_type: managed`. Available under API version `X-Pinecone-Api-Version: 2026-07`.

Create a serverless document-schema index with a dense-vector embedding field using the 2026-07 API version.

```
$ curl -s -X POST 'https://api.pinecone.io/indexes' \
  -H 'X-Pinecone-Api-Version: 2026-07' \
  -H 'Api-Key: <your-api-key>' \
  -H 'Content-Type: application/json' \
  -d '{
    "deployment": {
      "deployment_type": "managed"
    },
    "schema": {
      "fields": {
        "embedding": {
          "type": "dense_vector",
          "metric": "cosine"
        }
      }
    }
  }'
```

Build a document schema with a dense-vector field in Python and create a managed serverless index.

```python
from pinecone import Pinecone, SchemaBuilder

pc = Pinecone(api_key='<your-api-key>')

schema = (
    SchemaBuilder()
    .add_dense_vector_field('embedding')
    .build()
)

pc.indexes.create(
    deployment={'deployment_type': 'managed'},
    schema=schema,
)
```

— Runnable curl and Python examples with named schema fields. product docs

02

Project, organization, API key and service account management endpoints

NEW

95

Pinecone adds full lifecycle management endpoints for admin resources: projects via List projects, Create a new project, Get project details, Delete a project, and Update a project (including setting maximum Pod count and customer-managed encryption key (CMEK)); organizations via List organizations, Get organization details, Delete an organization, and Update an organization; API keys via List API keys and Delete an API key; and service accounts via List service accounts, Create a service account, Get service account details, Delete a service account, Update a service account, and Rotate a service account's OAuth client secret. A role-binding endpoint (`POST /organizations/<org-id>/role-bindings`) is also available for assigning scoped roles such as `editor` to principals like service accounts.

Grant a service account a scoped role at the project level immediately after creating it, using the new role-binding endpoint.

```
$ curl -X POST 'https://api.pinecone.io/organizations/<org-id>/role-bindings' \
  -H 'Api-Key: <api-key>' \
  -H 'Content-Type: application/json' \
  -d '{"principal_id": "<service-account-id>", "role": "editor", "scope": "project", "project_id": "<project-id>"}'
```

Rotate a compromised service account secret to immediately revoke old tokens and retrieve the new secret — returned only once.

```
$ curl -X POST 'https://api.pinecone.io/organizations/<org-
id>/service-accounts/<service-account-id>/rotate-secret' \
  -H 'Api-Key: <api-key>'
```

— All endpoint names given plus runnable curl examples. snapshot-20260911

03 **List documents endpoint with pagination** NEW 88

Pinecone adds a List documents endpoint that returns up to 100 documents per page in a namespace, with an optional `prefix` parameter to filter by ID prefix and a `pagination` token for iterating through results.

Paginate through all documents in a namespace whose IDs share a common prefix — useful for auditing or bulk-processing a logical subset of records.

```
$ curl -X GET
'https://api.pinecone.io/namespaces/<namespace>/documents?
prefix=user-&pagination=<token>' \
  -H 'Api-Key: <api-key>'
```

— Named endpoint, parameters, and runnable curl example. snapshot-20260911

04 **2026-07 API version restructures index responses** BREAKING 78

Under `X-Pinecone-Api-Version: 2026-07`, REST index responses are restructured: the top-level `deployment_type` and `read_capacity` fields move into a top-level `deployment` object, and the dense-vector metric now appears on the schema's dense-vector field rather than at the top level. SDKs return both the old and new shapes for backward compatibility, but direct REST callers parsing the old top-level fields will break.

— Clear before/after and named fields, but no dedicated migration example. product docs

05 **Metadata indexing controls at index and namespace level** NEW 67

Pinecone supports limiting metadata indexing at index creation or later via a per-field `filterable` setting that controls which metadata fields are indexed for filtering. Namespace-level metadata indexing rules can override index-level rules, letting each namespace configure its own indexed fields, while index-level rules apply as defaults to namespaces without their own rules. A document-schema index cannot accept metadata fields in its schema at creation, so its metadata indexing must instead be configured at the namespace level.

— Mechanism and scope explained but no example or exact call shown. product docs

THINNER COVERAGE BELOW

06 Nexus knowledge-curation and retrieval product NEW

45

Pinecone launches Nexus, a new knowledge-curation and retrieval product, with manifest-driven context design that lets practitioners define custom artifact and edge types in the console to control how sources become queryable knowledge.

— Product described in prose only, no concrete config or endpoint. product docs

07 2026-07 API version for Inference and Admin APIs NEW

30

Pinecone adds `inference_2026-07.oas` and `admin_2026-07.oas` API specifications, introducing a new `2026-07` versioned release of the Inference and Admin APIs.

— Only spec file names given, no detail on what changed. product docs

└──▷ **BREAKING ON UPGRADE**

! With `X-Pinecone-Api-Version: 2026-07`, REST responses remove the top-level `deployment_type` and `read_capacity` fields, replacing them with a top-level `deployment` object; callers parsing those top-level fields directly will break. (SDKs return both old and new shapes for backward compatibility.)

WAS THIS USEFUL? 👍 👎

timgordontg engrim

Feedback ⓘ 1 RELEASE · 2026-09-10

NOTES ↗ RANK

The Universal Cross-Model Episodic Memory Standard. Local-first, project-scoped SQLite memory engine for Google Antigravity, Claude Code, Cursor, Windsurf, and Codex.

> engrim v1.4.0 adds a live, WAL-safe SQLite backup command, a session/CI teardown command for stale resume pointers, scoped per-project record-count introspection, and a reference multi-agent CI architecture using GitHub Agentic Workflows.

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

01 **Live SQLite backup command** NEW 88

`engrim backup COPY.db [--force] [--json]` creates consistent online snapshots of active SQLite stores via Connection.backup(), guaranteeing zero WAL tearing even while agents are actively writing to the store.

Take a safe, consistent snapshot of the live memory store before a destructive operation, even while agents are mid-write.

```
$ engrim backup ~/.engrim/memory.backup.db --json
```

— Exact flags and underlying mechanism (Connection.backup) both given. v1.4.0

02 **Resume-pointer teardown command** NEW 88

`engrim retire [-p P | --all] [--dry-run] [--json]` monotonically closes active `resume-pointer` records when a session or CI job concludes, preventing stale `[▶ RESUME HERE]` pointers from accumulating across runs.

At the end of a CI job, close any stale resume-pointers for the current project without committing changes — useful for validating teardown in a dry run first.

```
$ engrim retire --dry-run --json
```

▶ View command output v1.4.0

— Full flag set, exact marker string, and dry-run behavior specified. v1.4.0

03 **Scoped project record-count introspection** NEW 76

`engrim project [-p P | --global | --all] [--json]` returns single-project record counts, complementing the existing `engrim projects` whole-store listing.

Inspect memory record counts scoped to a specific project by tag, in machine-readable form for dashboards or health checks.

```
$ engrim project -p my-app --json
```

— Runnable command with flags but little mechanism detail. v1.4.0

04 **GitHub Agentic Workflows reference implementation** NEW

62

A full multi-agent CI reference implementation at `examples/gh-aw` demonstrates engrim inside GitHub Agentic Workflows, covering ephemeral runner memory sync, safe SQLite backups, and multi-agent issue triage.

— Names the example path and scope but gives no runnable command. v1.4.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

LangChain LangSmith

Feedback ⓘ 1 RELEASE · 2026-08-20

NOTES ↗ RANK

LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

> LangSmith added monthly trace limits scoped to projects and users, improved code evaluator typing and image rendering in dataset tables, and began deprecating legacy feedback-formula and dataset-comparison endpoints in favor of newer APIs.

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

01 Batch add runs and threads to annotation queues NEW

90

A single request can now add a mixed batch of runs and conversation threads to an annotation queue for human review, via `POST https://<langsmith-host>/annotation-queues/<queue-id>/items` with an `items` array of objects carrying `item_type` (`RUN` or `THREAD`), `run_id`, `thread_id`, and `session_id`.

Add a batch of runs and conversation threads to an annotation queue in one request for mixed human review.

```
$ curl -X POST https://<langsmith-host>/annotation-queues/<queue-id>/items \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: <API_KEY>' \
  -d '{"items": [{"item_type": "RUN", "run_id": "<run-id>"}, {"item_type": "THREAD", "thread_id": "<thread-id>", "session_id": "<session-id>"}]}'
```

— Runnable curl example with full endpoint and payload shape. snapshot-20260911

02 Legacy dataset comparison SDK helpers removed BREAKING

70

Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; code using the generated SDK methods for dataset comparison must be migrated to `POST /v2/datasets/<id>/experiment-runs`.

— Named replacement endpoint provided for migration. snapshot-20260911

THINNER COVERAGE BELOW

03 PEP 604 union return types in code evaluators IMPROVED

50

Code evaluators annotated with PEP 604 union return types (e.g., `-> dict | None`) are now accepted on upload.

— Concrete syntax example given but no CLI or endpoint. snapshot-20260911

04 Base64 image thumbnails in dataset tables IMPROVED

40

Dataset and experiment tables now detect and render base64 images as small thumbnails with a delayed larger preview instead of showing raw encoded strings.

— UI-only change, no named endpoint or flag. snapshot-20260911

05 Monthly trace limits per project and user NEW

30

Enforces user-defined monthly trace limits scoped to individual projects and users, rejecting new traces once the configured limit is exceeded.

— No config key or API surface named for setting the limit. snapshot-20260911

└──▷ BREAKING ON UPGRADE

! The legacy feedback formula endpoints (`POST/GET /feedback/formulas` and `GET/PUT/DELETE /feedback/formulas/{feedback_formula_id}`) are deprecated and scheduled for removal on 2026-08-20; migrate existing feedback formulas to composite evaluators before that date.

! Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; code using the generated SDK methods for dataset comparison must be migrated to `POST /v2/datasets/<id>/experiment-runs`.

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ 2 RELEASES · 2026-09-11 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse's v4.34 and v4.35 releases bootstrap a new Rust-based AI gateway service with an organization control plane, add user-triggered session revocation and a redesigned session timeline, and extend score filtering into both the Scores view and Experiments search bar.

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

01 **Legacy ingestion API deprecation signalling** DEPRECATED

70

Adds a `_deprecation` field attached to successful legacy ingestion API write responses, and documents that legacy endpoints delay data by approximately 10 minutes.

— Names exact field and quantifies the delay behavior v4.34.0

THINNER COVERAGE BELOW

02 **OTEL trace IDs on tRPC error responses** IMPROVED 55

Attaches OTEL trace IDs to tRPC error responses to aid distributed tracing of server errors.

— Names OTEL and tRPC mechanism, no example given v4.34.0

03 **User-triggered session revocation** NEW 45

Adds user-triggered session revocation so users can invalidate their own active sessions.

— Clear self-service mechanism but no endpoint or UI path named v4.34.0

04 **Score filtering in Scores view and Experiments** NEW

45

Adds score filters and a search bar to the Scores view, and adds score filter display in the Experiments search bar.

— Names both UI surfaces but not filter mechanics v4.34.0

05 **Experiments UI polish for mismatches and comparisons** IMPROVED

45

Names the expected variables when a dataset does not match in experiments, surfacing mismatches explicitly; colour-codes compared runs; and aligns score cell icons in the experiments results view.

— Describes three UI changes but no config or command v4.35.0

06 **AI gateway Rust service with control plane** NEW 40

Bootstraps a new AI gateway Rust service with an organization control plane.

— Names the new service and control plane but no mechanism detail v4.34.0

07 **AI gateway trusted execution and provider attribution** NEW 40

Adds the ability to resolve trusted executions through the web interface in the AI gateway, and adds attribution of provider connections in the AI gateway.

— Names two gateway capabilities without mechanism v4.35.0

08 **Unified filter UI across sidebar and Users/Traces** IMPROVED 40

Unifies sidebar search and saved views in filters for a consistent navigation experience, and unifies the Users filter experience to match the Traces filter UI.

— Names affected views but not specific filter fields v4.35.0 v4.34.0

09 **Session timeline redesign and observation actions** IMPROVED 35

Redesigns the session timeline UI in the Sessions view and adds observation timeline actions to the session timeline.

— UI path given, mechanism of actions unspecified v4.34.0 v4.35.0

WAS THIS USEFUL? 👍 👎

Arize Phoenix

Feedback ⓘ 2 RELEASES · 2026-09-10 NOTES ↗ RANK

Arize Phoenix is an open-source platform that monitors LLM application traces and evaluates their outputs.

> Arize Phoenix added a new completeness evaluator to its evals library and let prompt versions carry arbitrary metadata via the JS client.

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

01 **Metadata parameter on promptVersion()** IMPROVED 55

Phoenix client's promptVersion() now accepts a `metadata` parameter so prompt versions can carry arbitrary JSON metadata.

— Names exact function and parameter, no usage example `@arizeai/phoenix-client@7.11.0`

02 **Completeness evaluator in evals library** NEW 40

Adds a `completeness` evaluator to the evals library for assessing whether a response fully covers the expected content.

— Names the evaluator but no scoring mechanism detail `arize-phoenix-evals-v3.7.0`

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ 1 RELEASE · 2026-09-01 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust's biggest additions this window are AI-driven Patterns and Debugger for surfacing and diagnosing agent failures, plus a managed-runtime upgrade to Loop automations, alongside broad SDK expansions across Python, TypeScript, Go and Java and infrastructure work like columnar metric storage and asynchronous Brainstore log search.

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

01 Java SDK eval concurrency, lifecycle APIs, and thread-safety requirement

90

BREAKING

Adds Eval.start() to start an eval in the background and return an `EvalResult` `/status` immediately, with lifecycle APIs getStatus(), isDone(), awaitCompletion(), getStartedAt(), getDuration(), getCasesExecuted(), and getAbortCause(). Concurrency is controlled via .maxConcurrency(int) on `Eval.Builder` or `Devserver.Builder` (default 3 concurrent cases) or a custom `Executor` via .executor(Executor), plus global defaults via environment variables `BRAINTRUST_DEFAULT_MAX_CONCURRENCY`, `BRAINTRUST_OTEL_MAX_QUEUE_SIZE`, `BRAINTRUST_OTEL_MAX_EXPORT_BATCH_SIZE`, and `BRAINTRUST_OTEL_EXPORT_INTERVAL_MILLIS` (Java SDK v0.3.23). Breaking: `Task` and `Classifier` implementations in the Java SDK must now be thread-safe.

Cap Java eval concurrency and tune OTel export batching via environment variables for high-throughput eval pipelines.

```
$ export BRAINTRUST_DEFAULT_MAX_CONCURRENCY=10
export BRAINTRUST_OTEL_MAX_QUEUE_SIZE=2048
export BRAINTRUST_OTEL_MAX_EXPORT_BATCH_SIZE=256
export BRAINTRUST_OTEL_EXPORT_INTERVAL_MILLIS=2000
```

— Exact methods, env vars, defaults, and a stated migration requirement snapshot-20260911

02 Custom iframe panels in trace spans NEW

90

A Span iframe project setting renders custom iframe URLs as fields inside trace spans, using mustache `{{parameters}}` to reference span data paths (e.g. `https://example.com/{{input.id}}?q={{metadata.foo.bar}}`). A Post span data toggle POSTs the full span payload to the iframe on load, useful when data is too large for URL parameters or the iframe writes data back to the span.

Embed a custom review tool inside every trace span by linking span fields into the iframe URL — no separate lookup needed.

> 📍 In the project settings, go to Span iframe › Create span iframe. Set URL to your tool's endpoint using mustache parameters, e.g. `https://example.com/{{input.id}}?q={{metadata.foo.bar}}`, give it a unique Field name, then save.

— Exact setting names, template syntax, and step-by-step setup `product docs`

---

03 **Metric columnstore for faster log queries** NEW 75

A Metric columnstore toggle in project settings writes trace-level metrics in a columnar format during compaction, alongside log search optimization, speeding up loading, sorting, and querying traces on the Logs page for high-volume projects.

Speed up Logs page queries on a high-volume project by enabling columnar storage for trace-level metrics.

> 📍 In the project settings, go to Metric columnstore and enable the toggle to write trace-level metrics in a columnar format.

— Named toggle and exact UI navigation, mechanism explained `snapshot-20260911`

---

04 **Trace-level SpanFilters for scorers** NEW 75

trace.get_spans() in the Python SDK (v0.39.0) gained `has_error`, `metadata`, and `duration` fields on `SpanFilters`, letting trace-level scorers filter spans by error state, metadata values, and execution time bounds.

Filter spans in a trace-level scorer to only those that errored and took longer than 2 seconds, narrowing scorer focus to slow failures.

```
$ trace.get_spans(SpanFilters(has_error=True, duration={"gt": 2.0}))
```

Filter spans in a trace-level scorer to only those that errored and ran longer than 5 seconds, using the new `has_error` and `duration` SpanFilters fields.

```python
spans = trace.get_spans(SpanFilters(has_error=True, duration={"gt": 5.0}))
```

— Named fields, function, version, and a runnable code example `snapshot-20260911`

---

05 **TypeScript SDK instrumentation and prompt template additions** NEW 75

The TypeScript SDK gained several tracing surfaces across versions: OpenAI Batch API instrumentation via `openaiFilesCreateTraced`, `openaiBatchesRetrieveTraced`, and `completeOpenAIBatchTrace` (v3.30.0); `wrapGoogleGenerativeAI` instrumentation for the legacy `@google/generative-ai` SDK v0.24.x (v3.32.0); `toolCallId` added to Vercel AI SDK v6 tool call span metadata (v3.31.0); and `input_audio` content type support in Mustache and Nunjucks prompt template rendering (v3.32.0).

— Names every function, SDK version, and format added `snapshot-20260911`

06 **Agno eval instrumentation** NEW 75

Adds Agno eval instrumentation to the Python SDK (v0.36.0): `AccuracyEval`, `AgentAsJudgeEval`, `ReliabilityEval`, `PerformanceEval`, and eval suites are traced automatically via auto_instrument(), and eval suite runs automatically open a Braintrust experiment.

Instrument Agno evals for automatic tracing and experiment creation in Braintrust, using the new auto_instrument() integration.

```python
import braintrust
from agno.eval.accuracy import AccuracyEval

braintrust.auto_instrument()

eval_run = AccuracyEval(agent=my_agent, dataset=my_dataset)
eval_run.run()
```

— Names every eval class, the function, version, and a runnable example `snapshot-20260911`

07 **Loop automations and managed runtime upgrade** NEW 70

Loop automations are scheduled Loop runs with their own instruction, model, and write permissions that produce a read-only thread you can Continue, with results deliverable to a Slack channel or webhook. Loop itself was upgraded to run in a Braintrust-managed server-side runtime (not the browser), persisting threads and enabling work across logs, experiments, and datasets, including creating and editing prompts, scorers, datasets, facets, custom views, dashboards, and automations, powered by built-in `GPT-5.6 Sol`, `GPT-5.6 Terra`, and `GPT-5.6 Luna` models.

— Names models and delivery channels, actioned via Continue button `snapshot-20260911`

08 **ElevenLabs tracing integration** NEW 65

Adds ElevenLabs integration to trace ElevenLabs SDK calls for debugging audio generation, evaluating models, and monitoring production usage, implemented in the TypeScript SDK (v3.32.0) via `wrapElevenLabs` for text-to-speech and speech-to-text calls, including streaming and timestamped audio.

Instrument all ElevenLabs TTS/STT calls in a TypeScript agent for automatic tracing, including streaming and timestamped audio.

typescript

```
import { wrapElevenLabs } from 'braintrust';
import ElevenLabs from 'elevenlabs';

const client = wrapElevenLabs(new ElevenLabs({ apiKey:
process.env.ELEVENLABS_API_KEY }));
```

— Names the wrap function and version with a runnable code example `snapshot-20260911`

09 **Harbor plugin verifier output upload** NEW 60

The Harbor plugin in the Python SDK (v0.36.0) now uploads standard verifier outputs `test-stdout.txt`, `test-stderr.txt`, and `ctrf.json`, configurable via `attachments` and `redact_patterns` settings.

— Names files and config keys but no full example shown `snapshot-20260911`

10 **Blind human reviews** NEW 60

Blind human reviews hide peer scores, comments, and aggregates until a reviewer submits their own; enabling the Blind human reviews project setting extends that hiding for the full review session.

— Names the exact project setting to enable `snapshot-20260911`

THINNER COVERAGE BELOW

11 **Patterns: scheduled AI analysis of recurring issues** NEW 55

Braintrust runs Loop against a project on a schedule to surface recurring problems and trends not covered by any scorer, saving each finding with supporting traces and a suggested fix; findings can create a scorer or classifier, continue investigation in Loop, or be copied as a prompt.

— Explains mechanism and outputs but no exact config or UI path `snapshot-20260911`

12 **Debugger for single-trace failure analysis** NEW 55

Debugger reports likely failure modes for a single trace, citing spans, tool calls, and model outputs; Analyze trace groups a run into Work sections explaining what the agent did.

– Names the Analyze trace action but no deeper config snapshot-20260911

13 **Asynchronous Brainstore log search** IMPROVED 50

Brainstore's log search gained an async execution engine that streams results as segments finish scanning, for full-text, phrase, and multi-clause trace searches including ANY_SPAN() (with negation) — no configuration required.

– Explains search types affected but nothing for reader to configure snapshot-20260911

14 **Go SDK v0.13.0 with prompt support** NEW 50

Ships Braintrust Go SDK v0.13.0 with a new Go API reference, adding prompt support: load a Braintrust prompt, render its variables locally, call any LLM client with the result, drive eval parameters, and link rendered calls back to their prompt in Braintrust traces.

– Describes the workflow but no concrete function names or example snapshot-20260911

15 **glm-5.3-flash built-in model** NEW 50

Adds `glm-5.3-flash` as a built-in Braintrust-served model requiring no AI provider setup, selectable under the Braintrust provider in playgrounds, prompts, and scorers, or requestable via `glm-5.3-flash` through the Braintrust Gateway.

– Names the model and gateway path but no usage example snapshot-20260911

16 **Dashboard sections** NEW 50

Dashboard charts can now be grouped into named, collapsible, reorderable, and duplicable sections, with per-browser collapse-state memory, available on Pro and Enterprise plans.

– Describes behavior and plan gating but no exact UI steps snapshot-20260911

17 **Pydantic AI invocation parameters in spans** NEW 40

Adds invocation parameters (`temperature`, `max_tokens`, and similar model settings) to Pydantic AI LLM span metadata in the Python SDK (v0.37.0).

– Names fields and version but is a thin one-line addition snapshot-20260911

18 **Scorer functions can omit name field** IMPROVED 40

Scorer functions in the Python SDK (v0.38.0) and TypeScript SDK (v3.31.0) can now return `{"score": 0.5}` without a `name` field, using the function's own name as the score key.

– Names the return shape and versions but no worked example snapshot-20260911

└──▷ **BREAKING ON UPGRADE**

! Task and Classifier implementations in the Java SDK must now be thread-safe.

WAS THIS USEFUL?

AI MODELS

◆ Frontier Models

Mistral AI

Feedback ⓘ 1 RELEASE · SEEN 2026-09-11 NOTES ↗ RANK

Learn how to deploy and use Mistral AI's Large Language Models with our comprehensive documentation, guides, and tutorials.

> Mistral shipped OCR 4.1 to general availability and added block-level granularity to OCR confidence scoring.

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

01 Block-level granularity for OCR confidence scores NEW

78

The OCR API's `confidence_scores_granularity` parameter now accepts `"block"` in addition to `"page"` and `"word"`. `"page"` returns page-level scores only, `"block"` returns page-level and block-level scores, and `"word"` returns page-level and word-level scores.

— Names exact parameter and all three accepted values. snapshot-20260911

THINNER COVERAGE BELOW

02 OCR 4.1 general availability NEW

50

`mistral-ocr-4-1` is now Generally Available, and both `mistral-ocr-latest` and `mistral-ocr-4` now point to it.

— Names model ids and aliasing but no capability detail. snapshot-20260911

WAS THIS USEFUL?

Anthropic

Feedback ⓘ 2 RELEASES · SEEN 2026-09-11 CHANGELOG ↗

RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic added infrastructure-as-code style agent management with `ant apply`, an `auto` permission-evaluation policy for Claude Managed Agents tool calls, and a live terminal/web session viewer via `ant beta:sessions connect`.

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

01 `ant beta:sessions connect` live session control NEW

88

Adds `ant beta:sessions connect <session-id>` CLI command to attach a terminal to a Claude Managed Agents session for live monitoring and interactive allow/deny of pending tool calls; the `--web` flag serves the Claude Console session viewer locally and opens the session in a browser instead of the terminal.

Attach to a running Managed Agents session from your terminal to monitor activity and interactively approve or deny tool calls waiting for human review.

```
$ ant beta:sessions connect <session-id>
```

Open the Claude Console session viewer locally instead of using the terminal interface — useful when you want a richer UI to review tool calls.

```
$ ant beta:sessions connect <session-id> --web
```

— Exact command and flag with runnable usage examples changelog-20260911-a850a3bd snapshot-20260911

02 `ant apply` CLI subcommand for agent resources NEW

80

Adds `ant apply` CLI subcommand (v1.30.0) to create and update agents, environments, skills, memory stores, and deployments from repository files; it writes a `claude-lock.json` lockfile so subsequent runs — locally or in CI — update existing resources rather than creating duplicates.

Declare all your agent resources as files in a repo and apply them in CI so each run updates existing resources tracked in the lockfile rather than duplicating them.

```
$ ant apply && git add claude-lock.json && git commit -m 'chore: update agent lockfile'
```

— Command, lockfile name and resource types given, no rollback notes

changelog-20260911-a850a3bd

03 `auto` permission policy for Managed Agents tool calls NEW

73

Claude Managed Agents permission policies gain an `auto` policy that evaluates each `agent.tool_use` and `agent.mcp_tool_use` call and lets the server automatically run it, deny it, or pause for human approval; the events report the outcome via new `evaluation` and `evaluated_permission` fields. Policy support extends to Claude Fable 5.1, Claude Mythos 5.1, and Claude Opus 5.

— Names events, fields and models but no config surface to set it

changelog-20260911-a850a3bd snapshot-20260911

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · September 11, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue