

# Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED SEPTEMBER 7, 2026 · EVERY DAY

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

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

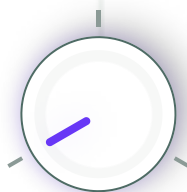
▸ // HOW THIS ISSUE IS MADE

SHOW

## VIEW

ISSUE VIEW

**scroll**



SCROLL

EXPAND  
IN  
PLACE

PREVIEW  
PANE

INFORMATION DENSITY

**everything**



BRIEFED

EVERYTHING

Every tool in the issue, in full.

all 193 changes

Do you prefer **scroll**?



▸ // CONTENTS SHOW

26 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 lets one document  
index serve keyword, dense, and sparse retrieval; Cursor can run cloud

vim ? help

agents on your own infrastructure; LocalAI brings continuous-batched vLLM-style serving into a Python-free local backend.

#### BUILD

### Build one retrieval surface instead of stitching keyword and vector paths together

The GA schema-based Documents API combines BM25 full-text, dense-vector, and sparse-vector search behind one document model. Teams can avoid maintaining separate search indexes and reconciliation code just to support lexical matches, semantic matches, and hybrid retrieval over the same corpus.

[Pinecone](#)

#### DEPLOY

### Run coding agents where your source and controls already live

Cursor opened Cloud Agents to self-hosted execution infrastructure, while Docker Sandboxes added scoped MCP OAuth authorization, Git-based skills, kit arguments, and machine-readable JSON for read-only commands. The practical change is that agent execution can move closer to private repos and internal automation without turning every setup step into a bespoke wrapper script.

[Cursor](#) · [Docker Sandboxes](#)

#### DEPLOY

### Serve local models with production-style batching and stricter output contracts

LocalAI added a Python-free vllm.cpp backend with continuous batching and paged KV cache, which are the serving mechanics that keep GPUs busier and reduce memory waste under concurrent requests. Jan added custom chat-format templates and GBNF grammar constraints, so local runs can enforce prompt shape and structured output instead of relying on best-effort text parsing.

[LocalAI](#) · [Jan](#)

#### EVALUATE

### Find bad agent behavior from traces instead of reading runs one by one

Braintrust added Patterns and Debugger for investigating agent behavior, while LangSmith added pre-save testing for thread evaluators. Together

they reduce the loop where teams manually inspect traces, edit an evaluator, and only later discover it scores the wrong thing.

[Braintrust](#)

Does **GPT-5.5** read better?



## BUILD

### ◆ AI Coding Agents

## Replit Agent

Feedback



1 RELEASE • SEEN 2026-09-07

NOTES ↗

RANK

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

Replit Agent added a machine-readable documentation index endpoint for programmatic discovery of its docs.

↳ WHAT SHIPPED • 1 FEATURE most completely described first ? what's the number?

### 01 Documentation index endpoint at `/llms.txt` NEW 55

Adds a `/llms.txt` endpoint to fetch the complete documentation index, enabling programmatic discovery of all available pages.

– Names exact endpoint but no further mechanism detail snapshot-20260907

WAS THIS USEFUL?



# Augment Code

Feedback

i

1 RELEASE • SEEN 2026-09-07

NOTES ↗

RANK

Your engineers have agents. Your organization doesn't.

Augment Code's Cosmos platform gained a set of Template Experts led by a Feedback Triager and a Ticket Manager, alongside a CLI flag for model reasoning effort and the removal of the `secret get --reveal` flag.

↳ WHAT SHIPPED • 5 FEATURES most completely described first ? what's the number?

01

Ticket Manager Expert for delegated tickets

NEW

56

A Ticket Manager Expert converts delegated Linear, Jira, or GitHub tickets into implementation-ready work and tracks them through closure.

– Names integrations but no invocation surface given. [product docs](#)

02

Feedback Triager Expert for Slack reports

NEW

54

A Feedback Triager Expert classifies Slack feedback, investigates technical reports, and creates tickets or launches fixes automatically.

– Explains behaviour but no command or config to invoke it. [product docs](#)

03

`'secret get --reveal'` flag removed

BREAKING

45

The `secret get --reveal` flag has been removed.

– Names exact removed flag but no migration guidance. [snapshot-20260907](#)

04

CLI flag for model reasoning effort

NEW

35

Adds a flag to control the model's reasoning effort from the CLI.

– Exact flag name not given, only its purpose. [snapshot-20260907](#)

05

Additional Template Experts for PR review and incident response

NEW

25

Augment Code also adds Template Experts for PR authoring and repair, code review, incident response, and project building, alongside the Feedback Triager and Ticket Manager experts.

– Only named in a list, no mechanism described. [product docs](#)

↳ ALSO FROM THESE RELEASES

Connectors

Team integrations

Atlassian

Azure DevOps App

GitHub App

GitLab (Service Account)

Linear App

Microsoft Teams

PagerDuty

Slack

Browse Connectors

All

Integrations

MCPs

AI & Automation

Hugging Face

Search models, datasets, and Spaces; run community tools

Sequential thinking

Think through complex problems step-by-step

Communication

Details

Diffst+1,651

Subscriptions

2 files changed +1651 • 12:1

augmentcode/augm

Show in right pane

FILES

Filter files...

clients

CHANGELOG

cosmos

audit

2 files

clients

1 # co

2

3 ## G

4 - \*\*

5 - \*\*Generated by\*\*: augment-app-stagin

6 - \*\*Request ID\*\*: unavailable (/requ

➡ **BREAKING ON UPGRADE**

! The `secret get --reveal` flag has been removed.

WAS THIS USEFUL? 

👍

👎

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

Bolt.new shipped a full visual editing mode for prompt-free UI tweaks, in-chat AI image generation, MCP server connectors for external tools, a new Claude Sonnet 4.6 model, team-scoped project templates, and direct file editing from folder view in Code view.

WHAT SHIPPED • 6 FEATURES most completely described first ? what's the number?

01 Visual editing mode in preview NEW 91

Adds a **Select** tool in the chatbox that lets you click any preview element and edit text, colors, font size, alignment, opacity, object fit, alt text, links, and list formatting directly in the UI without writing prompts. Edits are batched above the chatbox and only consume tokens when you click **Save changes** ; **CMD** +click (Mac) or **CTRL** +click (Windows) selects a child element inside a selected parent, with arrow-key navigation between nested elements. An optional batch note field lets you type cross-page instructions (e.g. apply a style change to matching elements on other pages) before saving, and a saved batch of visual edits can be rolled back via the project's version history when individual undo isn't available.

Restyle a heading across multiple pages without writing code — batch the change and use a note to propagate it site-wide.

- 1. In your Bolt project, click **Select** in the chatbox. 2. Click the section heading in the preview. 3. In the toolbar, change font size and type color, then click **Add changes** . 4. In the chatbox, type: Apply these changes to the section headings on the other pages too. 5. Click **Save changes** next to the batch.

Select a deeply nested child element (e.g. a button label inside a card) without accidentally editing the container.

- 1. Click **Select** in the chatbox. 2. Click the card element in the preview. 3. Use **CMD**+click (Mac) or **CTRL**+click (Windows) to target the button label inside the card. 4. Double-click the text and type your changes, then click **Add changes** .

— Names exact UI controls, click modifiers, and save/rollback mechanics product docs

02 MCP server connectors for external tools NEW 84

Adds MCP (Model Context Protocol) server connectors, letting Bolt integrate with external tools like Notion, Linear, and GitHub via built-in connectors or a custom MCP server URL, accessible via the plus icon in the chatbox under **Connectors > Manage connectors**.

Connect a GitHub MCP server so Bolt can read issues and pull requests without leaving the chat.

 In the Bolt chatbox, click the plus icon > Connectors > Manage connectors, then add GitHub from the built-in connectors list or paste a custom MCP server URL.

– Exact navigation path and named integrations given snapshot-20260907

03

**AI image generation in chatbox** NEW

79

Adds AI image generation directly in the Bolt chatbox, with transparent background support and automatic WebP conversion; requires enabling **Image Generation** under **Personal Settings > Add-on features**.

Enable AI image generation so you can describe and create ready-to-use WebP images with transparent backgrounds from the chatbox.

 Go to Personal Settings > Add-on features and turn on **Image Generation**, then return to the chatbox and describe the image you want to generate.

– Names exact settings toggle and output format snapshot-20260907

THINNER COVERAGE BELOW

04

**Team-scoped reusable project templates** NEW

49

Adds team-scoped reusable project templates (Teams plan only), allowing organizations to create standardized starter projects from existing projects without modifying the original.

– Describes scope and behavior but no UI path given snapshot-20260907

05

**Claude Sonnet 4.6 model option** NEW

45

Adds Claude Sonnet 4.6 as a selectable model in the agent selector, balancing speed, cost, and reasoning for planning, coding, debugging, and iterative development.

– Names the model and selector but no further detail snapshot-20260907

06

**Direct file editing in Code view folders** IMPROVED

30

Enables direct file editing from folder view in Code view.

– Single-line description with no mechanism or scope product docs

WAS THIS USEFUL?



# Vercel v0

[Feedback](#)[1 RELEASE · SEEN 2026-09-07](#)[NOTES ↗](#)[RANK](#)

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

v0 shipped a unified GitHub Publish flow with a full branch menu, zero-config AI Gateway authentication for generated apps, a new GPT-5.6 Sol model tier, and in-preview Vercel sign-in, alongside a long tail of editor, mobile, and billing improvements.

↳ [WHAT SHIPPED · 18 FEATURES](#) most completely described first [? what's the number?](#)

## 01 Publish flow with branch menu for GitHub projects NEW

85

A new branch menu gives access to the latest preview deployment, a code diff against the base branch, PR creation/open/merge, CI check and repository-rule status, pulling the base branch, and in-chat fix requests for preview, CI, or merge issues. The unified Publish action lets v0 create or reuse a pull request, wait for required checks and reviews, merge into the base branch, and deploy to production, pausing to link directly to the PR if a branch protection rule needs human action.

Publish a GitHub-backed project to production without leaving v0 — useful when you want branch protections and CI checks to run before merge but don't want to context-switch to GitHub.

📌 In v0, open your GitHub-backed project, make your changes, then select Publish. v0 creates or reuses the pull request, waits for required checks and reviews, merges into the base branch, and deploys to production. If a branch protection rule needs human action, v0 pauses and links you directly to the pull request.

— Full workflow described with concrete steps. [snapshot-20260907](#)

## 02 MCP server limit raised to 100 per scope IMPROVED

70

Raises the maximum number of MCP servers per scope from 10 to 100; in v0 settings, navigate to MCP Servers for your scope to add up to 100 servers.

Increase the MCP server limit for a scope that integrates more than 10 tools — the new ceiling of 100 lets large teams connect many MCP providers without hitting the old cap.

📌 In v0 settings, navigate to MCP Servers for your scope and add up to 100 servers. The previous limit of 10 no longer applies.

— Exact old/new limits and navigation path given. [snapshot-20260907](#)

### 03 Team or password visibility for published apps NEW

65

Adds a **Team or password** visibility option for published apps, requiring Vercel team sign-in while still allowing password-based access for others. Set via Share/Publish settings by choosing this visibility option and saving.

Set a published app to require Vercel team sign-in while still allowing external reviewers in via a shared password — useful for staging apps shared with contractors.

📌 In your project, open Share / Publish settings and set visibility to **Team or password**, then save. Team members sign in with Vercel; others use the password you supply.

– Exact settings path given for enabling. snapshot-20260907

### 04 Zero-config AI Gateway authentication for generated apps NEW

60

Generated apps get zero-config authentication to the AI Gateway covering text, structured output, embeddings, reranking, images, video, speech, and transcription — no provider or Gateway API keys required.

– Names capabilities covered but no setup steps given. snapshot-20260907

THINNER COVERAGE BELOW

### 05 GPT-5.6 Sol and Sol Fast model tier NEW

55

Adds GPT-5.6 Sol and a Sol Fast tier to the model picker, routed through OpenAI on Vercel AI Gateway, priced at 50% off through September 18.

– Names model, tier, and discount window; no usage steps. snapshot-20260907

### 06 Inline error bar for failed VM dev servers NEW

55

A compact error bar appears beneath the preview when a VM dev server fails to boot or install, and clicking it opens the console log to the exact error.

– Describes trigger and click-through behavior. snapshot-20260907

### 07 Sign-in with Vercel inside VM preview NEW

50

Sign-in with Vercel now works from inside a VM preview: the preview and app share an origin so the session persists, and sign-in opens in a new top-level window when the preview is embedded.

– Explains mechanism but no config steps. snapshot-20260907

### 08 Project-scoped chat starting source IMPROVED

50

New chats started inside a project now begin from that project's own source — GitHub branch, managed repository, or production deployment — so v0 has the code from the first message.

– Describes behavior change, no config surface. snapshot-20260907

## 09 Automatic VM tier upgrade on resource exhaustion IMPROVED

 45

Paid chats whose sandbox runs out of CPU or memory are automatically moved up a VM size tier and restarted in the background instead of timing out.

– Explains before/after behavior, no manual action needed. snapshot-20260907

## 10 Inline environment variable requests above composer IMPROVED

 45

Environment variable requests now appear above the composer like questions and approvals, with collapse and reopen that preserve entered values.

– Describes UI placement and state preservation. snapshot-20260907

## 11 Native mobile app messaging and history improvements IMPROVED

 40

The native mobile app adds message queuing, tappable links, shared-chat author avatars, and chat-history and image-caching improvements.

– Lists several additions with no mechanism depth. snapshot-20260907

## 12 Direct-to-chat Git repository imports IMPROVED

 40

Git repository imports go directly to chat, skipping the detection step and requirements modal.

– Names skipped steps but limited further detail. snapshot-20260907

## 13 Token billing default for paid Enterprise teams IMPROVED

 40

Paid Enterprise teams default to token billing, enabling teams with a shared credit pool but no seat credits to spend it.

– States billing change and who benefits. snapshot-20260907

## 14 Revision history for custom skills NEW

 35

Custom skills now show revision history, including file paths, authors, timestamps, and text diffs.

– Names fields shown but no usage steps. snapshot-20260907

## 15 Native Save As dialog for ZIP downloads NEW

 35

Adds a native Save As dialog to the Download ZIP action where the browser supports it.

– Names exact action, minimal further detail. snapshot-20260907

---

**16 Drag-to-Favorites for sidebar chats** NEW 30

Adds drag-to-Favorites for chats in the sidebar.

– Bare feature name, no further detail. snapshot-20260907

---

**17 Web-based VM code editor for all users** IMPROVED 30

Makes the web-based code editor in the VM panel available to all users.

– States rollout only, no mechanism. snapshot-20260907

---

**18 Illustrated tooltips for editor modes** IMPROVED 30

Editor modes — Preview, Design, Code, and Database — now have illustrated tooltips.

– Names the four modes, no other mechanism. snapshot-20260907

---

WAS THIS USEFUL?



# Anyosphere Cursor

Feedback



1 RELEASE • SEEN 2026-09-07

NOTES

RANK

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

Cursor's biggest shipment in this window opens Cloud Agents to self-hosted execution infrastructure and introduces Origin, a built-in code hosting service, alongside 3x-faster agent starts via Builds, persistent `/goal` objectives, subscriptions, custom modes, and new team controls over skill sharing.

WHAT SHIPPED • 10 FEATURES most completely described first ? what's the number?

## 01 Origin code hosting for Cloud Agents NEW 85

Introduces Origin, a built-in code hosting service with repos, pull requests, code browsing, and GitHub sync, available in early beta on all paid plans. Adds a Codebase tab as the home for Origin repos with a `+New` button to create repos whose names become part of every repo URL at `cursor.com/codebase/<name>`, and lets Cloud Agents start without a connected GitHub or third-party SCM provider via `Start from scratch` in the repo picker, saving work to a Cursor Origin repo. Adds pull request sync between Origin and GitHub so comments, reactions, and replies made in either surface appear in the other within seconds, app extensions (Vercel, Depot, Buildkite) to Origin repos via a per-repo Apps tab enabling preview deployments and CI on every PR, and per-repo Settings for managing GitHub sync status, access control, and connected apps.

.

– Names URL scheme, tabs, and specific app integrations snapshot-20260907

## 02 Self-hosted machines and execution infrastructure NEW 80

Introduces self-hosted machines so tool execution, codebase, build outputs, and secrets stay entirely within your own network. Adds dynamic pool scheduling via My Machines (single laptop/VM) and Team Pools (named queues of workers) that auto-scale and hibernate idle machines, plus support for running cloud agents on pre-existing sandbox infrastructure including AWS Lambda, Coder, Cloudflare, Daytona, Modal, Namespace, Vercel, and E2B. Also enables computer use (click, type, screenshot, browser control) on Linux and Mac for self-hosted workers, with live desktop viewing and takeover from Cursor.

.

– Many named integrations and mechanisms, no exact commands snapshot-20260907

## 03 Skill sync and team marketplace for Cloud Agents NEW

Adds a **Sync Skills for Cloud Agents** toggle under **Settings → Agents → Context and Tools** that lets individuals enable their personal skills for use in Cloud Agents, and a team admin control under **Team Settings → Security & Identity** to disable this org-wide for Teams and Enterprise plans. Also adds a team skill marketplace where members publish personal skills from `~/.cursor/skills/` to a shared Default marketplace via **Customize → Skills → Publish**, an **Allow Members to Publish** toggle in **Dashboard → Plugins** that restricts who can publish, and the ability for admins to mark a published plugin **Default On** or **Required** to control installation across the team.

Enable your local personal skills for use in Cloud Agents without sharing them with your team.

📌 Settings → Agents → Context and Tools → turn on **Sync Skills for Cloud Agents** → Confirm

As a Teams or Enterprise admin, prevent all members from syncing personal skills to Cloud Agents.

📌 Team Settings → Security & Identity → turn off **Sync Skills for Cloud Agents**

Share a personal skill with your team so teammates can discover and install it from the team marketplace.

📌 Go to **Customize → Skills**, open the personal skill you want to share from `~/.cursor/skills/`, choose **Publish**, and confirm. The skill becomes available in your team's Default marketplace.

Restrict skill publishing to admins only, so members cannot add unapproved skills to the team marketplace.

📌 Go to **Dashboard → Plugins**, open the team marketplace, and turn **Allow Members to Publish** off.

– Names exact UI paths and toggles, no API surface product docs

#### 04 Long-lived goals via `/goal` command NEW 70

Adds `/goal` command to give a cloud agent a long-lived objective that it pursues until fully complete — pair with a custom mode or `/loop` for recurring check-ins.

Give a cloud agent a persistent, long-running objective — such as eliminating all flaky tests — so it keeps working toward the goal across sessions without you re-prompting.

 In a new chat, type: `/goal` fix all flaky tests and make CI green

— Exact runnable command with pairing options named `snapshot-20260907`

#### 05 Custom Modes pin skills as always-on NEW 65

Adds Custom Modes, which pin any skill as an always-on mode in chat, activated via  (Mac) or `Alt+Enter` (Windows) from the skill picker.

Pin a skill as a Custom Mode so every new chat automatically follows a specific playbook — useful for always enforcing a code-review or security-audit workflow.

 In the chat panel, open the skill picker with `/`, select the skill you want to pin, then press `Alt+Enter` (Windows) or `Option+Return` (Mac) to activate it as a Custom Mode.

— Exact activation shortcut given `snapshot-20260907`

#### 06 Builds for faster Cloud Agent starts NEW 65

Introduces Builds for Cloud Agents: pre-prepared environment snapshots (repos cloned, dependencies installed) that make agents start 3x faster, with a Builds tab in the Cloud Agents dashboard for inspecting build status, logs, and commit SHAs.

— Concrete speedup and dashboard location, no command `snapshot-20260907`

#### 07 Non-interrupting steering messages IMPROVED 60

Adds steering improvements allowing mid-run follow-up messages that wait for the next tool call rather than interrupting the agent — send via `Send now` or by pressing  twice.

— Exact send mechanism named `snapshot-20260907`

THINNER COVERAGE BELOW

#### 08 Subscriptions for event-driven cloud agents NEW 50

Adds Subscriptions for cloud agents to monitor PRs, watch Slack threads, or run scheduled tasks — agents wake on events without manual intervention.

— Describes triggers but no setup steps `snapshot-20260907`

## 09 Isolated-VM subagents NEW

45

Enables subagents to run on their own isolated virtual machines, each with a clean copy of the project and an independent cloud environment.

Spin up a swarm of subagents to independently test a parent agent's changes in fresh, isolated environments — useful for parallel regression or security smoke-testing without environment collisions.

 In the chat panel, prompt: run a swarm of subagents to test my app for bugs, each in its own environment

– Describes mechanism but no config or limits snapshot-20260907

## 10 Live preview and publishing for Cloud Agents NEW

45

Adds in-browser live port-forwarding of a cloud agent's environment for real-time preview and design mode tooling, plus a Publish feature that connects a Vercel account and produces a live URL from a built project.

– Two thin additions with no exact steps snapshot-20260907

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.

Diagram Design 2.6 ships ten new diagram types, native Droid packaging, named client profiles, animated semantic patterns, and a live gallery.

---

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

Run the new rendered-lint pass against all examples to catch SVG clipping and runtime errors that source-level linting cannot see.

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

Verify beeswarm geometric invariants (value-axis accuracy, no overprint) after editing or generating a beeswarm example.

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

Run adversarial Sankey contract tests to confirm the checker fires on known mutations and stays quiet on clean files.

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

Adds `role="img"`, `aria-labelledby`, and first-child `<title>` / `<desc>` slots to `template.html`, `template-dark.html`, `template-full.html`, and `template-terminal.html`, and adds an `a11y` category to `scripts/lint-skin.py` that rejects a diagram SVG without a resolvable accessible name.

Adds `scripts/test-lint-a11y.py` covering nine rejection and seven acceptance cases for the accessible-SVG contract.

Adds `scripts/verify-beeswarm.py` to enforce nine geometric invariants — shared value scale, no overprint, and others — that `lint-skin.py` and `verify-geometry.py`

cannot catch.

Adds `scripts/lint-render.py` to lint diagram examples as rendered in headless Chromium, catching clipped SVG viewports, collapsed SVGs, page overflow, and runtime errors that source-level linting misses.

Adds `scripts/test-verify-sankey.py` with adversarial fixtures testing both passing and failing polarities of the Sankey geometric contract.

[+13 more](#)

---

WAS THIS USEFUL?



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

Kiro shipped its biggest platform update yet: Kiro Web reached general availability, account admins gained OpenTelemetry usage-metrics export, and the CLI got new session-management and configuration-inspection commands, alongside cloud-to-local config sync, git branch selection, and ISO/IEC 27001:2022 certification scope.

← WHAT SHIPPED • 9 FEATURES most completely described first ? what's the number?

## 01 OpenTelemetry usage metrics export NEW 95

Account administrators can export per-user Kiro usage metrics to any OpenTelemetry-compatible platform via OTLP/gRPC or HTTP/protobuf, configured as a single account-level export in the Kiro console using an AWS Secrets Manager secret and a public OTLP endpoint. Metrics are delivered server-side daily at 02:00 UTC, with support for Amazon CloudWatch, Datadog, Dynatrace, Elastic, Honeycomb, and OpenSearch via OTLP receiver.

Pipe Kiro adoption and credit-consumption metrics into your existing observability stack by setting up an account-level OTLP export in the Kiro console — no per-developer configuration needed.

📌 In the Kiro console, go to Account Settings > Usage Metrics Export. Enter your OTLP endpoint (OTLP/gRPC or HTTP/protobuf), select your AWS Secrets Manager secret for auth credentials, and save. Kiro will begin delivering metrics server-side daily at 02:00 UTC.

– Named protocols, integrations, schedule and exact console steps

changelog-20260907-2d81f3bc

## 02 CLI session browsing and management NEW 80

Adds `kiro-cli chat --sessions` (or `/sessions` in V3) to browse, search, resume, and clean up local and cloud past sessions directly from the terminal.

Find and resume a cloud session from the terminal when switching machines, without opening Kiro Web.

```
$ kiro-cli chat --sessions
```

– Runnable command named, but behaviour described briefly

changelog-20260907-2d81f3bc

03

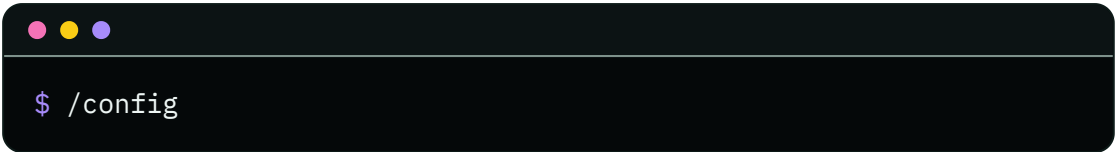
**/config command for session state overview**

NEW

80

Adds a `/config` command usable in any local or cloud V3 session to view all configured agents, MCP servers, Powers, Steering, Skills, and Hooks in one terminal panel, with local/cloud/both source labels for each.

Quickly audit everything Kiro has loaded in a running CLI session — agents, MCP servers, Powers, Steering, Skills, and Hooks — and confirm whether each comes from local config, cloud, or both.



– Exact command and full list of surfaces it inspects `changelog-20260907-2d81f3bc`

04

**Cloud configuration sync to local sessions**

NEW

70

New Configuration Sync option 'Apply your cloud configuration to local sessions' loads cloud Steering, custom agents, Skills, Powers, and Hooks into new local Kiro IDE and CLI sessions without writing files to the local `.kiro` directory.

– Names the exact option and mechanism, but not full setup path

`changelog-20260907-2d81f3bc`

THINNER COVERAGE BELOW

05

**Branch selection for repository sessions**

NEW

55

Users can pick a specific GitHub or GitLab branch when selecting a repository for a session, so the agent clones and builds from that branch instead of the default.

– Clear behaviour named, but no exact UI path given `changelog-20260907-2d81f3bc`

06

**Cloud-synced Powers in IDE**

NEW

50

Cloud-synced Powers now appear in the installed Powers list in the IDE, where users can select `Try power` to review bundled Skills and MCP configuration.

– Names UI action but lacks deeper mechanism `changelog-20260907-2d81f3bc`

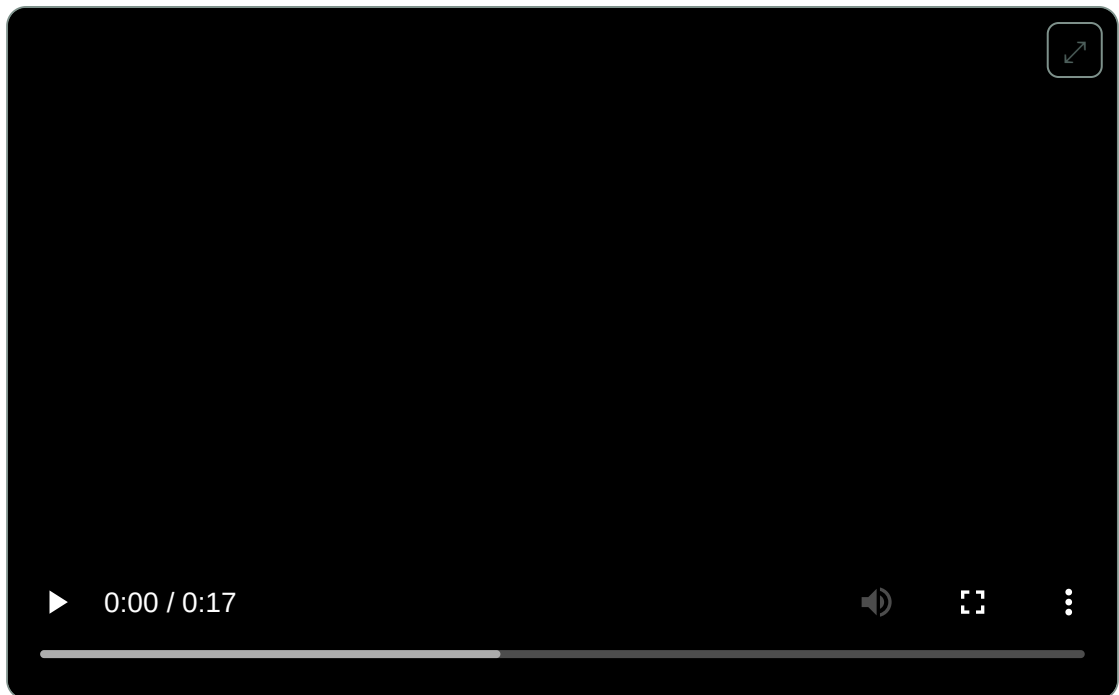
07

**Kiro Web general availability**

NEW

45

Kiro Web launches as generally available, enabling browser-based sessions with Specs, collaborative exploration, and full task delegation.



– Describes capability but no setup or navigation detail [changelog-20260907-2d81f3bc](#)

## 08 **Auto-generated session titles** NEW 30

New sessions now auto-generate a descriptive agent-created title based on session content; user-set titles are never overwritten by the agent.

– Simple behaviour change with no configuration surface [changelog-20260907-2d81f3bc](#)

## 09 **ISO/IEC 27001:2022 certification scope** NEW 30

Kiro is now included in the defined scope of AWS's ISO/IEC 27001:2022 certification, providing independently verified evidence for security, procurement, and vendor-risk teams.

– Compliance fact with no reader action available [changelog-20260907-2d81f3bc](#)

WAS THIS USEFUL?



---

**AGENT**

# Arcade

[Feedback](#)

1 RELEASE · SEEN 2026-09-07

[NOTES](#)[RANK](#)

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

Arcade's biggest release in this window adds ten new third-party toolkits (Power BI, Postman, Insightly, Fireflies.ai, OneDrive, Freshdesk, Ashby, Vercel, Datadog, and Nimble) alongside stateless MCP protocol support, new platform RBAC/org APIs, and a wide set of additions to existing Google Workspace, Linear, Telegram, X and Google Slides toolkits.

↳ WHAT SHIPPED · 12 FEATURES most completely described first ? what's the number?

## 01 Stateless MCP protocol support

NEW

83

Arcade's `/mcp` path now supports per-request protocol-version dispatch and a stateless `server/discover` endpoint, plus stateless `tools/list` and `tools/call` against the real catalog. The MCP `/list` page size is now configurable, and MRTR url-mode elicitation was added for tool OAuth challenges on the stateless MCP path.

Discover the stateless MCP server catalog without establishing a session, useful for lightweight integrations.

```
$ curl -X GET 'https://api.arcade.dev/mcp/server/discover' \
-H 'Authorization: Bearer <token>'
```

– Named endpoints with a runnable curl example against `server/discover`.

[snapshot-20260907](#)

## 02 Ten new toolkits for third-party services

NEW

68

Arcade adds the `microsoft_powerbi` toolkit for Power BI integrations, a Postman API toolkit with 30 tools and API-key authentication, an Insightly CRM toolkit with 29 tools for sales and account management, a Fireflies.ai meeting-intelligence toolkit with 17 functionalities, a Microsoft OneDrive toolkit with 16 tools for file management, a Freshdesk customer-support toolkit with API-key authentication, an Ashby recruiting/ATS toolkit for core recruiting operations (including a `pipeline exit` tool to archive applications), a Vercel toolkit for managing accounts via the REST API, a Datadog observability toolkit for log management, and Nimble as a Partner MCP Server for real-time web data in the search integrations catalog.

– Names ten toolkits with counts and auth types but no invocation examples.

[snapshot-20260907](#)

### 03 Google Workspace toolkit tool additions NEW 63

Google Workspace toolkits gain new tools: `GetSpreadsheetEditHistory` in the `google_sheets` toolkit, a secure Gmail unsubscribe tool with multiple flow outcomes and enhanced validation, new Gmail tools for label lifecycle management and conversation trash status handling, a `search_directory_people` tool for resolving coworkers' emails, a `search_drive` tool to search the user's entire Google Drive, and a `check_file_access` pre-flight probe plus a `file_types` picker filter for Drive access.

– Names every new tool across Sheets, Gmail and Drive without usage examples.

snapshot-20260907

THINNER COVERAGE BELOW

### 04 Builtin agent tooling and model selection NEW 57

Adds an `Arcade.ListApps` builtin that prefixes apps into the MCP prompt, MCP behavior annotations on Arcade builtin tools, an Agent Capability Gap self-report MCP builtin tool with a master switch and LLM instruction integration, a configurable chat-model selector for choosing from a curated catalog, and VS Code chat host support in agent mode for attachment capabilities.

– Named builtins and features, but no configuration details shown.

snapshot-20260907

### 05 RBAC and org management APIs NEW 55

New platform APIs add a project-scoped `tool_execution` list and detail endpoint for tool-execution history, surface `can_invite` on org endpoints for RBAC-aware UI, add a CLI command for operator authorization, enable default RBAC for new Cloud signups, and let project members leave projects voluntarily.

– Names endpoints and a field but no exact CLI syntax or paths.

snapshot-20260907

### 06 Dashboard UI for MCP gateways and Tools page IMPROVED 53

The dashboard gains dedicated create/edit pages for MCP gateways with new routing and tool selection, a catalog-driven server-side search with lazy loading on the Tools page, and an undo toast notification for destructive picker edits.

– Describes UI changes with clear navigation targets but no exact paths.

snapshot-20260907

### 07 Linear toolkit new tools NEW 45

The Linear toolkit adds a `resolve_comment` tool for managing comment resolution state and a read tool to retrieve project status updates with pagination.

– Names two tools but gives no invocation details.

snapshot-20260907

### 08 OIDC issuer validation and coordinator errors IMPROVED

Adds deep validation for OIDC user source issuers and improves coordinator error detail visibility.

– Names the auth area but no configuration or error format detail. snapshot-20260907

---

**09 Interactive messaging in Telegram toolkit** NEW 30

The Telegram worker toolkit gains interactive messaging with buttons and response handling.

– Brief description with no mechanism or usage detail. snapshot-20260907

---

**10 Complex schema support across surfaces** IMPROVED 23

Complex tool input/output schemas are now supported across all surfaces.

– Generic claim with no named surface or example. snapshot-20260907

---

**11 Google Slides integration overhaul** IMPROVED 15

Arcade performed a comprehensive overhaul of the Google Slides integration.

– One-line claim with no specifics of what changed. snapshot-20260907

---

**12 X toolkit rebuild** IMPROVED 15

Arcade did a ground-up rebuild of the X toolkit.

– No details on scope or changed behavior given. snapshot-20260907

---

WAS THIS USEFUL?



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

OpenBot v0.0.8 ships a cross-platform desktop shell and a production `serve` script, alongside a new `computer.action_stopped` audit event and a broad pass of startup and input validation hardening across config variables, the audit endpoint, and the component catalogue.

[WHAT SHIPPED](#) · [8 FEATURES](#) most completely described first [? what's the number?](#)

## 01 Startup configuration validation hardening

**BREAKING**

80

Validates `SERVER_INTERNAL_URL` at worker startup and requires an `http` or `https` URL, refusing to start and naming the variable when wrong. Validates `HostPort` with digit and range checks, refusing supervisor startup on an invalid port. Treats an empty `PORT` value on `agent-bot` and `agent-langgraph` as the shipped default rather than parsing it as `NaN` and binding an ephemeral port. Refuses malformed IPv6 entries in `AGENT_ENDPOINT_ALLOWED_HOSTS` at boot instead of silently accepting entries that never match. Reads `durationMs` units case-insensitively, accepting `30s` alongside `30s`.

— Names every affected variable and its exact validation rule `v0.0.8`

## 02 `DATABASE\_URL` password and parse validation

**IMPROVED**

75

Reports a malformed `DATABASE_URL` password containing a bare `%` with a structured error naming the variable and the required `%25` encoding, instead of crashing with `URIError: URI error`, and refuses any `DATABASE_URL` value that cannot be parsed without echoing the password into the log.

— Gives exact before/after error behaviour and the fix `v0.0.8`

## 03 `COMPUTER\_MEMORY\_BYTES` validation

**BREAKING**

75

Validates `COMPUTER_MEMORY_BYTES` at supervisor startup and refuses non-integer values, preventing a misconfigured cap such as `512m` — previously parsed via `parseInt` into `512` bytes — from ever reaching a container.

— Concrete before/after example of the bug it fixes `v0.0.8`

## 04 Strict audit endpoint parameter validation

**IMPROVED**

Validates `from`, `to`, and `limit` parameters on the audit endpoint strictly, returning HTTP 400 for an unparseable date or non-integer limit instead of coercing and querying.

Query the audit log for a specific time window when building a dashboard that needs to distinguish deliberate stops from real outages — filter by the new

`computer.action_stopped` type.

```
$ curl -G 'https://<openbot-host>/audit' \
  --data-urlencode 'from=2024-06-01T00:00:00Z' \
  --data-urlencode 'to=2024-06-30T23:59:59Z' \
  --data-urlencode 'limit=100' \
  -H 'Authorization: Bearer <token>'
```

– Names endpoint params, status code, and includes a runnable query v0.0.8

## 05 `'serve'` script for production deployment NEW

Adds a `serve` script that builds the app once and serves the bundle, replacing the dev-only `dev` script as the path to a production deployment.

– Names the exact script and what it replaces v0.0.8

## 06 `'computer.action_stopped'` audit event type BREAKING

Adds the `computer.action_stopped` audit event type to distinguish a user-initiated Stop from a genuine computer-unreachable failure, enabling accurate outage monitoring by type. Previously all stops were recorded under `computer.action_failed`, so monitoring that counts failures by that type will no longer count deliberate stops after upgrading.

– Names the event type and the migration impact clearly v0.0.8

## 07 Whitespace sanitization in component catalogue IMPROVED

Refuses titles, slugs, and IDs that are entirely whitespace in sandboxed components, skills, custom servers, and the component catalogue, and trims `name`, `title`, `kind`, and `description` fields before storing them in the catalogue, so a component with a stray space in its name is treated as the same component rather than creating a duplicate row.

– Names affected areas and fields but no direct command v0.0.8

THINNER COVERAGE BELOW

## 08 Cross-platform desktop shell NEW

Introduces a desktop shell for macOS, Windows, and Linux that fetches, pins, and raises the full OpenBot stack in a single window, no browser tab required.

– Describes mechanism but no install command or config path given v0.0.8

#### ! --> BREAKING ON UPGRADE

- ! The audit event type for a user-initiated Stop is now `computer.action_stopped` ; anything counting failures by type using `computer.action_failed` will no longer count deliberate stops.
- ! `SERVER_INTERNAL_URL` must be a valid `http` or `https` URL; the worker refuses to start if the value is any other string.
- ! `COMPUTER_MEMORY_BYTES` must be a whole number of bytes; values like `512m` that previously parsed silently via `parseInt` now cause the supervisor to refuse to start.
- ! IPv6 entries in `AGENT_ENDPOINT_ALLOWED_HOSTS` that are not parseable as a valid address are refused at boot; previously malformed entries were accepted and silently never matched.

---

WAS THIS USEFUL?



---

## DEPLOY

# Perplexity API

[Feedback](#)

1 RELEASE · SEEN 2026-09-07

[NOTES](#)

RANK

Perplexity API Platform - build with the Router, Agent, Search, and Embeddings APIs. Real-time, web-wide research and Q&A capabilities for your products.

Perplexity enriched its gateway chat completions responses with moderation results and deeper usage/audio metadata, and shipped a standalone Search API alongside an API key rotation mechanism.

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

## 01 Moderation results in chat completions

NEW

75

Adds a `moderation` object to `POST /router/v1/chat/completions` responses, with `input` and `output` sub-objects each carrying `type`, `model`, `flagged`, `categories`, `category_scores`, and `category_applied_input_types` fields for content moderation results.

Check whether the model flagged your prompt or its own output for policy violations, using the new moderation fields in the gateway response.

```
$ curl --request POST \
  --url https://api.perplexity.ai/router/v1/chat/completions \
  --header 'Authorization: Bearer <token>' \
  --header 'Content-Type: application/json' \
  --data '{ "model": "<model>", "messages": [{ "role": "user",
    "content": "<prompt>" } ] }' \
  | jq '{input_flagged: .moderation.input.results[0].flagged,
    output_flagged: .moderation.output.results[0].flagged,
    categories: .moderation.output.results[0].categories}'
```

– Names endpoint and every field, runnable curl example given

[product docs](#)

## 02 Expanded chat completion response metadata

IMPROVED

68

The `POST /router/v1/chat/completions` response gains `reasoning_content` on `message` for chain-of-thought exposure, an `annotations` array with `url_citation` entries ( `start_index`, `end_index`, `url`, `title` ) for inline source attribution, and an `audio` object ( `id`, `expires_at`, `data`, `transcript` ). `usage` gains `completion_tokens_details` ( `accepted_prediction_tokens`, `audio_tokens`, `reasoning_tokens`, `rejected_prediction_tokens` ) and

`prompt_tokens_details` ( `audio_tokens`, `cached_tokens`, `cache_write_tokens` ), plus top-level `service_tier` and `system_fingerprint` fields.

Inspect token-level cost breakdown — including cached, reasoning, and audio tokens — to optimise latency and billing for high-volume pipelines.

```
$ curl --request POST \
  --url https://api.perplexity.ai/router/v1/chat/completions \
  --header 'Authorization: Bearer <token>' \
  --header 'Content-Type: application/json' \
  --data '{"model": "<model>", "messages": [{"role": "user",
"content": "<prompt>"}]}' \
  | jq '.usage | {prompt_tokens, completion_tokens, total_tokens,
reasoning_tokens: .completion_tokens_details.reasoning_tokens,
cached_tokens: .prompt_tokens_details.cached_tokens}'
```

– Every added field named; example queries usage breakdown [product docs](#)

---

THINNER COVERAGE BELOW

---

### 03 Standalone Search API

NEW

55

A standalone Search API endpoint returns raw, ranked search results from Perplexity's index without any LLM processing, supporting all existing filters including domain, date range, and academic sources.

– Explains behaviour and filters but no endpoint path or example [snapshot-20260907](#)

### 04 API key rotation mechanism

NEW

35

Introduces an API key rotation mechanism allowing seamless replacement of API keys without service interruption, and supports automated rotation workflows.

– No named endpoint, flag, or example to act on [snapshot-20260907](#)

---

WAS THIS USEFUL?



# Eigen Labs Darkbloom

[Feedback](#)[1 RELEASE · 2026-09-07](#)[NOTES >](#)[CODE >](#)[RANK](#)

Private Inference Network on Idle Macs

Darkbloom's v0.9.0 release adds an operator-controlled idle-memory policy for provider nodes, per-model hardware eligibility checks in the model catalog, international payout support for providers, and a round of inference performance and UI improvements.

↳ [WHAT SHIPPED · 6 FEATURES](#) most completely described first [? what's the number?](#)

## 01 Idle-memory policy for provider nodes NEW 86

Operators choose how models are kept in memory at `darkbloom start`: `Always ready` keeps models loaded, `Free when idle` unloads them after 60 minutes (the default), or `Custom` lets the user define their own idle window.

Start a provider node interactively to be prompted for your idle-memory policy — choose between `Always ready`, `Free when idle`, or `Custom` timeout.

```
$ darkbloom start
```

— Names exact command, all three modes, and the default timeout. v0.9.0

## 02 Hardware eligibility verdicts in model catalog NEW 83

`darkbloom models catalog` now surfaces `required_provider_capabilities` per model and prints eligibility verdicts based on detected hardware, e.g. `requires: Apple M5, NAX runtime` or 'not eligible on this machine (missing: apple\_m5, mlx\_nax)'.

Check which catalog models your Apple Silicon Mac is eligible to serve before downloading, including any M5/NAX-gated models.

```
$ darkbloom models catalog
```

— Names the command, field, and sample verdict text verbatim. v0.9.0

THINNER COVERAGE BELOW

## 03 International bank withdrawals for providers NEW 41

Providers can now withdraw earnings via international bank transfer using Stripe Global Payouts.

– Names the payout mechanism but no UI path or limits given. v0.9.0

---

**04 Pinned model downloads with R2 fallback** IMPROVED  41

Model fetching now prefers pinned Hugging Face artifact downloads, falling back to a verified R2 mirror when needed.

– Names the mechanism but no command or config surface given. v0.9.0

---

**05 Inference performance improvements** IMPROVED  38

Adds paged attention and SSD caching for Qwen models, and accelerates GPT-OSS prefill and decode performance with measured controls.

– Names techniques and target models but no benchmarks or mechanism detail. v0.9.0

---

**06 Console, onboarding, and stats dashboard redesign** IMPROVED  27

Redesigns the console entry point, the provider onboarding flow, and the network stats dashboard.

– Names three UI areas but gives no detail on the changes. v0.9.0

---

WAS THIS USEFUL?



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

Together AI added a beta organization-wide billing usage API, progress tracking for file uploads in both the SDK and CLI, [Slurm](#) cluster kubeconfig downloads, legacy API key regeneration for admins, and refinements to the [OIDC](#) Kubeconfig verification flow.

WHAT SHIPPED • 5 FEATURES most completely described first ? what's the number?

## 01 Billing usage API NEW 80

Adds a new `GET /billing/usage` endpoint (beta) that returns organization-wide usage and cost line items, filterable by month via a `month` query parameter, e.g. `GET /billing/usage?month=2026-09`.

Pull your org's token spend for a specific month to feed into a cost-allocation report or budget alert.

```
$ curl -X GET 'https://api.together.ai/billing/usage?month=2026-09' \
-H 'Authorization: Bearer <your_api_key>'
```

– Named endpoint with method, path and runnable curl example. [changelog-20260907-23b77efe](#)

## 02 File upload progress tracking in SDK and CLI NEW 75

Adds a `progress_callback` parameter to `client.files.upload()` in the Python SDK so callers can receive upload progress events, and the `tg files upload` and `tg files check` CLI commands now show a progress bar in interactive terminals.

Track large [fine-tuning](#) dataset uploads in a CI pipeline or notebook by wiring a progress callback.

```
python

client.files.upload(
    file=open('dataset.jsonl', 'rb'),
    purpose='fine-tune',
    progress_callback=lambda bytes_sent, total:
    print(f'{bytes_sent}/{total} bytes')
)
```

– Names the SDK parameter and CLI commands with a working code example.

[changelog-20260907-23b77efe](#)

### 03 **OIDC Kubeconfig verify access improvements** IMPROVED

60

The **OIDC Kubeconfig** dialog's **Verify access** step now includes a copyable cache-clear command for refreshing OIDC tokens after RBAC or IdP group changes, and on Together-managed clusters it now suggests `kubectl get pods` instead of `kubectl get nodes` for members with namespace-scoped RBAC.

– Names both UI tweaks and the specific `kubectl` commands involved.

changelog-20260907-23b77efe

THINNER COVERAGE BELOW

### 04 **Slurm cluster kubeconfig download** NEW

50

Project members can now download the cluster kubeconfig for Slurm clusters from the cluster details page, matching the existing behavior already available for Kubernetes clusters.

– Describes scope and comparison but only a UI path, no command.

changelog-20260907-23b77efe

### 05 **Legacy API key regeneration for admins** IMPROVED

50

Legacy API keys can now be regenerated by any admin or editor of the default project; these keys appear in the API keys table marked with a **Deprecated** badge.

– States who can act and the UI marker, but no command or endpoint.

changelog-20260907-23b77efe

WAS THIS USEFUL?



## vMLX

Feedback



1 RELEASE · 2026-09-06

NOTES >

RANK

The vMLX server runs compressed MLX models on Apple Silicon with disk caching, paged memory, continuous batching, and hybrid SSM scheduling.

vMLX 1.6.55 sharpened tool-call validation and warning surfacing across API lanes, refined native MTP sampling-policy adoption, and added ordered replay of tool-step reasoning in chat history.

↳ WHAT SHIPPED · 3 FEATURES most completely described first ? what's the number?

### 01 Tool-call validation and warnings across API lanes

IMPROVED

55

Surfaces tool-call drop and schema warnings on every API lane (OpenAI, Anthropic, Ollama), and separates tool-call presence validation and schema validation into independent checks rather than one combined check.

– Names the three API lanes but no config or flag to act on.

v1.6.55

### 02 MTP sampling-policy adoption

IMPROVED

40

Native MTP adoption now keeps the live engine's sampling policy and depth, governed by one sidecar evidence rule.

– Mechanism named but no exposed setting or command.

v1.6.55

### 03 Ordered tool-step reasoning replay in chat history

NEW

35

Chat history now replays persisted tool-step reasoning ordered by step rank.

– Behavior described but no interface or command shown.

v1.6.55

WAS THIS USEFUL?



The vLLM inference engine runs large language models with high-throughput, memory-efficient GPU execution.

vLLM shipped a new out-of-tree Tenstorrent hardware plugin, adding platform registration for TT accelerators along with TT-specific model architectures, scheduling, sampling/fabric configuration, and Galaxy mesh optimisations.

←→ WHAT SHIPPED · 7 FEATURES most completely described first ? what's the number?

## 01 Custom model registration on Tenstorrent

NEW

90

Adds `EXTRA_MODELS_DIR` to register additional model architectures at startup from bundle folders each containing a `vllm_metadata.json` and an adapter class, without editing plugin source, and `TT_VLLM_BUILTIN_MODELS=0` to restrict the architecture registry to only models supplied via `EXTRA_MODELS_DIR`, excluding all built-in models.

Register a custom model bundle at startup without touching plugin source, then restrict the registry to only that model to avoid accidentally loading built-ins.

```
$ EXTRA_MODELS_DIR=/opt/tt-models TT_VLLM_BUILTIN_MODELS=0 vllm  
serve /opt/tt-models/my-custom-model
```

– Named env vars and file with runnable example command `launch-20260907-b130d8c8`

## 02 On-device sampling and fabric topology config

NEW

90

Adds `--additional-config.tt.sample_on_device_mode` (e.g. `all`) to enable on-device sampling so the host receives already-chosen tokens rather than raw logits, and `--additional-config.tt.fabric_config` to configure the Tenstorrent mesh fabric topology (e.g. `FABRIC_1D`, `FABRIC_2D`, `FABRIC_1D_RING`).

Serve Llama 3.3 70B on a Galaxy TG mesh with on-device sampling and ring fabric, so the host never processes raw logits and all parallelism is compiled into the mesh program.

```
$ vllm serve meta-llama/Llama-3.3-70B-Instruct --additional-  
config.tt.sample_on_device_mode all --additional-  
config.tt.fabric_config FABRIC_1D_RING
```

– Two flags with concrete values and a runnable serve command `launch-20260907-b130d8c8`

## 03 Tenstorrent hardware plugin and supported models

NEW

Introduces the `vllm-tt-plugin` out-of-tree platform plugin that registers Tenstorrent accelerators as a vLLM platform automatically when `ttnn` from TT-Metal is importable. Supports serving `TTLlamaForCausalLM`, `TTMllamaForConditionalGeneration`, `TTQwen2ForCausalLM`, `TTQwen3ForCausalLM`, `TTQwen3_5ForConditionalGeneration`, `TTQwen2_5_VLForConditionalGeneration`, `TTQwen3VLForConditionalGeneration`, `TTMistralForCausalLM`, `TTMistral3ForConditionalGeneration`, `TTGemma3ForConditionalGeneration`, `TTGemma4ForCausalLM`, `TTGemma4ForConditionalGeneration`, `TTGemma4UnifiedForConditionalGeneration`, `TTDeepseekV3ForCausalLM`, and `TTGpt0ssForCausalLM` model architectures on Tenstorrent hardware.

– Names plugin and 15 model classes but no install command given `launch-20260907-b130d8c8`

#### 04 Mesh device selection replacing tensor-parallel flags

75

##### BREAKING

Adds a `MESH_DEVICE` setting (e.g. `TG`) that replaces `--tensor-parallel-size` / `-tp` / `-pp` for Tenstorrent mesh targets; the plugin rejects those GPU-oriented flags outright.

– Names the env var and rejected flags but no runnable example `launch-20260907-b130d8c8`

#### 05 TTLaneCoordinator multi-lane scheduler NEW

75

Introduces `TTLaneCoordinator` as an alternative scheduler class, set via `scheduler_config.scheduler_cls`, that runs one independent `TTScheduler` per data-parallel lane within a single engine process for Galaxy-class single-execute models.

Serve Qwen3-32B on Galaxy using the galaxy-optimised single-execute variant, which activates the TTLaneCoordinator multi-lane scheduler inside a single engine process.

```
$ TT_QWEN3_TEXT_VER=qwen3_32b_galaxy vllm serve Qwen/Qwen3-32B --
  additional-config.tt.fabric_config FABRIC_2D
```

– Config key and mechanism named; example shows activation via variant env var

`launch-20260907-b130d8c8`

#### 06 Phase-constrained TTScheduler for traced execution NEW

75

Introduces a phase-constrained scheduler, `TTScheduler`, that issues exclusively prefill-only or decode-only steps, with chunked prefill and interleaved decode supported, to

match Tenstorrent traced-execution requirements; selected via

`scheduler_config.scheduler_cls`.

– Mechanism and config key named but no direct usage example

launch-20260907-b130d8c8

## 07 Galaxy-optimised model variant selection

NEW

70

Exposes `TT_LLAMA_TEXT_VER` (e.g. `llama3_70b_galaxy`) and `TT_QWEN3_TEXT_VER` (e.g. `qwen3_32b_galaxy`) environment variables to select Galaxy-optimised single-execute model variants.

Serve Qwen3-32B on Galaxy using the galaxy-optimised single-execute variant, which activates the TTTLaneCoordinator multi-lane scheduler inside a single engine process.

```
$ TT_QWEN3_TEXT_VER=qwen3_32b_galaxy vllm serve Qwen/Qwen3-32B --  
additional-config.tt.fabric_config FABRIC_2D
```

– Named env vars with example values and a runnable command

launch-20260907-b130d8c8

WAS THIS USEFUL?



LocalAI runs language models locally through self-hosted, OpenAI-compatible APIs.

LocalAI's headline addition is the vllm.cpp backend — a Python-free, C++20 port of vLLM with continuous batching and paged KV cache — alongside major expansions to text-to-speech (Qwen3-TTS, Higgs Audio v3, NeMo-Speech.cpp), MiniMax-H3 video generation with audio, conversation history compression, reversible PII masking, and a deny-by-default auth overhaul that closes several previously unauthenticated endpoints.

← WHAT SHIPPED · 15 FEATURES most completely described first ? what's the number?

## 01 Qwen3-TTS text-to-speech on llama-cpp NEW 95

LocalAI adds Qwen3-TTS support through the `llama-cpp` backend, enabling TTS on `CUDA`, `ROCm`, `SYCL`, `Vulkan`, `Metal`, and `L4T` accelerators alongside the existing `qwen3-tts-cpp` backend; gallery entries `qwen3-tts-llamacpp (Q8_0)` and `qwen3-tts-llamacpp-q4 (Q4_K_M)` install via `local-ai models install qwen3-tts-llamacpp-q4`. A `known_usecases: [tts]` config declaration distinguishes the `GGUF` from chat/vision models (without it the model is treated as chat and Voice Library profiles are refused), and an `mmproj` config key points to the multimodal projector `GGUF` (speaker encoder and code predictor). Requests support `max_frames` to cap generation length (engine default 512 frames = 40.96s at 12.5 frames/second), ISO 639-1 `language` codes, and `top_k` / `top_p` sampling parameters, with streaming TTS supported on this backend.

Install and configure Qwen3-TTS via llama-cpp for zero-shot voice cloning with a hard audio length cap.

```
$ local-ai models install qwen3-tts-llamacpp-q4
```

Wire up the Qwen3-TTS backbone and its multimodal projector in a model config file so LocalAI routes it as a TTS model rather than a chat model.

```
yaml
name: qwen3-tts-llamacpp
backend: llama-cpp
mmproj: qwen3-tts-llamacpp/mmproj-Qwen3-TTS-12Hz-1.7B-Base-Q8_0.gguf
model: qwen3-tts-llamacpp/Qwen3-TTS-12Hz-1.7B-Base-Q8_0.gguf
known_usecases: [tts]
```

Synthesise a short sentence with Qwen3-TTS, pinning language and capping generation to ~8 seconds to avoid runaway output on short inputs.

```
$ curl http://localhost:8080/v1/audio/speech \
  -H 'Content-Type: application/json' \
  -d '{"model": "qwen3-tts-llamacpp", "input": "Hello, world.",
    "voice": "voices/my-reference.wav", "language": "en", "params":
    {"max_frames": "100"}}' \
  --output out.wav
```

– Config keys, install command, and a runnable curl example `launch-20260820-d873b3b1`

## 02 **vllm.cpp backend for local inference** NEW

90

LocalAI ships `vllm.cpp`, a C++20 port of vLLM with continuous-batching, paged KV cache, and automatic prefix caching, served from a HuggingFace `safetensors` directory or single file with no Python at inference time. It installs via `local-ai backends install vllm-cpp`, with images for CPU, CUDA 13, Vulkan, Metal, and Jetson L4T, plus gallery-ready configs ( `qwen3-0.6b-vllm-cpp`, `qwen3-4b-vllm-cpp`, `qwen3-coder-30b-a3b-vllm-cpp`, `qwen3.6-27b-nvfp4-vllm-cpp`, `qwen3.6-27b-nvfp4-mtp-vllm-cpp`, `qwen3.6-27b-nvfp4-dflash-vllm-cpp`, `qwen3.6-35b-a3b-nvfp4-vllm-cpp`, `qwen3.6-35b-a3b-nvfp4-mtp-vllm-cpp` ) with tool calling, reasoning split, and speculative decoding pre-configured. CUDA architecture coverage was expanded from `120a;121a` (amd64) / `121a` (arm64) to `80;86;89;90a;100a;103a;120a;121a` on amd64 and `87;90a;100a;110;121a` on arm64, adding A100, A10, 3090, L4, 4090, RTX 6000 Ada, H100, H200, B200, B300, Jetson Orin, and Jetson Thor. New image variants include `docker.io/localai/localai:latest-gpu-nvidia-cuda-13` (Blackwell sm\_120a/sm\_121a GPUs), `docker.io/localai/localai:latest-gpu-vulkan` (fallback for unmatched GPU vendors), and `docker.io/localai/localai:latest-nvidia-l4t-arm64-cuda-13` (Jetson DGX Spark).

Install and run a small vllm.cpp model on CPU or any supported GPU without hand-editing YAML.

```
$ local-ai backends install vllm-cpp
local-ai models install qwen3-0.6b-vllm-cpp
```

– Names images, archs and gallery models, with an install command

`launch-20260820-d873b3b1`

## 03 **External source management for agent collections** NEW

90

LocalAI adds `GET /api/agents/collections/:name/sources` to list external sources attached to an agent collection, `POST /api/agents/collections/:name/sources` to attach one (with an `update_interval` field in minutes, default 60, controlling sync frequency), and `DELETE /api/agents/collections/:name/sources` to remove one.

Verify which external sources are currently registered on a collection before modifying them.

```
$ curl http://localhost:8080/api/agents/collections/my-collection/sources
```

– Three full endpoints with a runnable curl example [product docs](#)

## 04 Alias-based scheduling for distributed deployments

NEW

85

`POST /api/nodes/scheduling` attaches a node-selector and replica rule to a model alias, enabling stable distributed-mode deployment slots whose underlying model can be swapped without changing the scheduling policy.

Pin a distributed scheduling rule to a model alias so swapping the underlying model preserves GPU placement and replica requirements.

```
$ curl -X POST http://localhost:8080/api/nodes/scheduling \
-H 'Content-Type: application/json' \
-d '{"model_name": "production", "node_selector": {"tier": "gpu"}, "min_replicas": 2}'
```

– Named endpoint with a runnable curl example [product docs](#)

## 05 Reversible PII masking in responses

NEW

80

The `pii.reverse_in_response` config option restores PII-masked values in responses, replacing identical `[REDACTED:EMAIL]` -style markers with unique deterministic pseudonyms like `EMAIL_001` and `EMAIL_002`, including handling across split SSE tokens.

Enable reversible PII masking so that addresses and emails in model responses are restored to their originals rather than left as generic redaction markers.

```
yaml

pii:
  reverse_in_response: true
```

Enable PII reversal so masked values in requests are restored to their originals in the model response, with unique pseudonyms per entity.

```
yaml

pii:
  reverse_in_response: true
```

– Named config key with a working YAML snippet `launch-20260820-d873b3b1`

## 06 Deny-by-default auth middleware

BREAKING

80

Auth middleware is now method-aware and denies by default: routes must appear in an explicit registry to be public, closing unauthenticated access to

`/mcp/chat/completions`, `/moderations`, `/models`, `/backends`, and `/import-model`. Narrow exceptions remain configurable via

`ApplicationConfig.PathWithoutAuth` for embedded deployments.

– Names every affected route and the exception mechanism `launch-20260820-d873b3b1`

## 07 Unified model & backend management in WebUI

IMPROVED

80

The WebUI gains a `Write YAML` tab on the model import page for direct YAML config editing with syntax highlighting and a copy button, plus a `Source` field accepting `huggingface://`, `hf://`, `oci://`, `ocifile://`, `ollama://`, full Hugging Face URLs, direct `https://` URLs, and absolute host paths. An `Import options` expansion lets you override backend, name, description, quantizations, MMProj quantizations, model type, `embeddings` support, diffusers-specific fields, and arbitrary custom key-value preferences before import. `/app/models` now unifies Explore and Installed views (filterable by running, idle, disabled, pinned, or distributed state) and `/app/backends` unifies Catalog and Installed views, with search, filter, and selection state persisted in the URL for shareable links.

– Names every field and page but only UI navigation, not commands

`launch-20260820-d873b3b1`

## 08 Higgs Audio v3 and NeMo-Speech.cpp TTS backends

NEW

75

LocalAI adds the `audio-cpp-higgs-audio-v3` gallery model (install via `local-ai models install audio-cpp-higgs-audio-v3`) for expressive multilingual TTS and zero-shot voice cloning via the `voice` field (path to a server-local WAV file), and a `magpie-tts-cpp` backend wrapping NeMo-Speech.cpp for Nemotron Speech synthesis via a NanoCodec decoder, requiring `codec_model` and `tokenizer_dir` config keys. Streaming TTS is supported on the `llama-cpp`, `magpie-tts-cpp`, and `sherpa-onnx` backends, with others falling back to non-streaming mode.

## 09 Video generation with audio via vllm-cpp NEW 65

LocalAI adds video generation with synchronised AAC audio through the `vllm-cpp` backend when a model config declares the MiniMax-H3 checkpoint set, served through the existing `/video` endpoint, producing MP4 output with lip-sync when speech is requested. Gallery entries `minimax-h3-fl2va-q4` and `minimax-h3-ref2va-q4` install the two checkpoint sets.

– Names endpoint and gallery models but no runnable example launch-20260820-d873b3b1

## 10 Conversation history compression NEW 60

A `compression` block in model config files enables automatic summarization of older conversation turns before inference, preserving system prompts, `tool-call` pairs, and the newest messages; older complete turns are summarised by a chosen LocalAI model.

– Names the config block but no worked example launch-20260820-d873b3b1

### THINNER COVERAGE BELOW

## 11 Authentication required for /version and asset URLs BREAKING 55

`/version` now requires credentials when database authentication or legacy API keys are configured, and URLs LocalAI returns for generated audio, images, video, and 3D assets likewise now require authentication — clients that previously fetched them anonymously must send the key.

– Clear migration note but no mechanism detail launch-20260820-d873b3b1

## 12 In-flight backend operations UI and process-wide admission control IMPROVED 55

Running backend operations are now surfaced in the UI while still in flight, with direct links to their logs. HTTP admission control is now process-wide rather than per-backend, also bounding trace-list growth. Cold model loads run as durable jobs, so the advisory lock is taken and released around the dedup check only, rather than held across the full multi-gigabyte transfer.

– Describes mechanism but no config surface to act on launch-20260820-d873b3b1

## 13 Unauthenticated API discovery endpoints NEW 55

`GET /api/instructions/{name}` returns a markdown API guide for a named instruction area and remains available even when database authentication or legacy API keys are active. Full interactive API documentation is also available without authentication at `/swagger`, listing available endpoints and capabilities.

#### 14 **User-Agent header on model downloads** NEW 50

LocalAI now emits a `LocalAI/<version> (<os>; <arch>)` User-Agent header (e.g. `LocalAI/v3.2.1 (linux; amd64)`) on every model download from Ollama, OCI registries, the model gallery, and plain HTTP(S) URLs, so registry operators can attribute usage; source builds without a stamped version send `LocalAI (<os>; <arch>)`.

– Exact header format given but nothing for a reader to configure [product docs](#)

#### 15 **Parallel Hugging Face shard downloads** IMPROVED 45

Hugging Face snapshot materialization now downloads shard files in parallel up to a configured limit, instead of transferring many-shard repositories one file at a time.

– States the change but not the configurable limit's name or default

`launch-20260820-d873b3b1`

#### ↳ **BREAKING ON UPGRADE**

- ! `/version` now requires credentials when database authentication or legacy API keys are configured — clients that fetched it anonymously must send the key.
- ! URLs LocalAI returns for generated audio, images, video, and 3D assets now require authentication when database authentication or legacy API keys are configured — clients that fetched those URLs anonymously must send the key.
- ! `/version` now requires credentials when database authentication or legacy API keys are configured.
- ! URLs LocalAI returns for generated audio, images, video, and 3D assets now require authentication credentials when database authentication or legacy API keys are configured; clients that previously fetched those URLs anonymously must now send the key.

WAS THIS USEFUL?



## DATA

## Pinecone

Feedback



1 RELEASE · 2026-09-07

CHANGELOG >

RANK

Pinecone is a managed vector database that stores and queries embeddings for AI applications.

Pinecone shipped a GA schema-based Documents API combining BM25 full-text, dense-vector, and sparse-vector search, introduced egress metering on read operations across all plans, and moved BYOC cluster metadata storage in-cluster via FoundationDB.

↳ WHAT SHIPPED · 3 FEATURES most completely described first ? what's the number?

## 01 Schema-based Documents API with BM25 and metadata operations

95

### BREAKING

On API version `2026-07`, `POST /indexes` now creates schema-based document indexes: a `schema` object combines BM25 full-text, dense-vector, and sparse-vector ranking fields in a single index, replacing the previous top-level `dimension`, `metric`, `vector_type`, and `spec` fields (which will break existing index-creation code on `2026-07`). The Documents API supports BM25 token matching, Lucene query syntax, and vector similarity queries with metadata and text-match filters, and adds fetch, update, and delete of documents by metadata filter in addition to by ID.

Create a document index combining BM25 full-text and dense-vector fields in a single index using the new `2026-07` API version.

```
$ curl -X POST https://api.pinecone.io/indexes \
  -H 'Api-Key: <your-api-key>' \
  -H 'X-Pinecone-API-Version: 2026-07' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "my-doc-index",
    "schema": {
      "fields": [
        {"name": "body", "type": "text", "ranking": "bm25"},
        {"name": "embedding", "type": "dense_vector",
"dimension": 1536, "metric": "cosine"}
      ]
    }
  }'
```

— Names endpoint, API version, schema fields, and includes a runnable example

changelog-20260907-be541c1b

## 02 Egress metering on read operations NEW

70

Reads via `query`, `fetch`, `list`, and `search` are now metered for egress, with monthly allowances of 1 GB (Starter), 10 GB (Builder), and 100 GB (Standard and Enterprise). Standard and Enterprise plans continue serving reads beyond the allowance at an overage rate, while Starter and Builder reads are blocked until the next billing period.

— Concrete limits and plan behavior given, but no config or command to act on

changelog-20260907-be541c1b

THINNER COVERAGE BELOW

## 03 In-cluster metadata storage for BYOC IMPROVED

45

BYOC deployments now run FoundationDB inside the customer's Kubernetes cluster to store cluster metadata, eliminating the need for a separately managed database.

– Describes the architectural change but gives no configuration or migration steps

changelog-20260907-be541c1b

#### ↳ **BREAKING ON UPGRADE**

! On API version `2026-07`, `POST /indexes` is schema-only: the top-level `dimension`, `metric`, `vector_type`, and `spec` fields are replaced by a `schema` field. Code that passes those top-level fields when creating indexes on `2026-07` will break.

---

WAS THIS USEFUL?



# Weaviate

Feedback

2 RELEASES · 2026-09-07

NOTES

RANK

Weaviate is an open-source vector database that stores both objects and vectors, allowing for the combination of vector search with structured filtering with the fault tolerance and scalability of a cloud-native database.

Weaviate shipped always-on MCP support for Cloud clusters with a read-only toggle, structured reference selection in the REST API, a new Meta AI generative module, and backend improvements including concurrent collection backups and a debug endpoint for vector reassignment.

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

01 Always-on MCP server for Weaviate Cloud clusters NEW 70

MCP server is now always enabled on Weaviate Cloud clusters, making the write tool available by default. An **Enable MCP Read-Only** switch restricts the MCP server to its read tools only, configurable on the cluster creation form or post-creation from cluster settings, with no restart required.

Restrict an existing Weaviate Cloud cluster's MCP server to read-only tools after creation — no restart required.

In the Weaviate Cloud console, open your cluster, go to Settings, and toggle **Enable MCP Read-Only** on.

— Names the exact toggle and UI path, no restart mechanism noted [product docs](#)

02 Shard-count Prometheus metric NEW 60

Adds **weaviate\_shards** Prometheus metric reporting shard count broken down by state and eager/lazy loading mode.

— Names exact metric and its breakdown dimensions [v1.39.3](#)

THINNER COVERAGE BELOW

03 Structured reference selection via returnReferences NEW 55

The REST API adds **returnReferences** structured reference selection, enabling fine-grained control over which references are returned in object responses.

— Named REST field but no request example given [v1.39.3](#) [v1.38.14](#)

04 Concurrent collection backups via shared worker pool IMPROVED 45

Backs up collections concurrently through one shared worker pool, enabling faster parallel collection backups.

– Describes mechanism but no config or command shown v1.38.14

---

**05 Meta AI generative module** NEW 40

Adds the Meta AI generative module ( `generative-meta` ) as a new generative integration.

– Bare name of new module, no usage detail v1.39.3

---

**06 Debug endpoint to reassign vectors in place** NEW 40

Adds a debug endpoint to reassign all vectors in place, intended for hfresh index maintenance.

– No exact endpoint path or usage given v1.38.14

---

WAS THIS USEFUL?



EverOS provides a local-first memory layer that stores and evolves context for AI agents.

EverOS v1.3.0 introduces an optional Milvus/Zilliz Cloud remote index backend with strict startup schema validation, and aligns vector search across backends by switching LanceDB to cosine distance.

←> WHAT SHIPPED · 3 FEATURES most completely described first ? what's the number?

## 01 Milvus/Zilliz Cloud remote index backend NEW 92

Adds Milvus and Zilliz Cloud as an optional remote derived-index backend, installed via `pip install everos[milvus]` and selected by setting `backend = "milvus"` under `[index]` in config; every field also binds to an environment variable such as `EVEROS_INDEX_BACKEND` and `EVEROS_MILVUS_URI`. A new `[milvus]` config section adds keys `uri`, `token`, `db_name`, `consistency_level` (accepting `Strong`, `Bounded`, `Session`, or `Eventually`), and `collection_prefix` to control the remote connection.

Switch a production EverOS instance from the default LanceDB backend to a self-hosted remote Milvus Server.

```
toml

[index]
backend = "milvus"

[milvus]
uri = "http://127.0.0.1:19530"
token = ""
db_name = ""
consistency_level = "Session"
collection_prefix = "everos"
```

— Names every config key, env var and install command with a runnable example v1.3.0

## 02 Startup schema validation for Milvus collections IMPROVED

58

Startup now validates each physical Milvus collection field-by-field — datatype, primary key, nullability, vector dimension, and index metric — failing immediately on schema mismatch rather than at first write or search.

– Explains mechanism but gives no command or config to invoke it v1.3.0

## 03 Cosine distance for LanceDB vector search

**BREAKING**

54

Vector search now uses cosine distance on LanceDB instead of L2, making the metric consistent across LanceDB, Milvus, and `agent_skill` recall paths; ranking order can shift for existing deployments on upgrade with no rebuild required.

– Describes behavior change and impact but no migration step given v1.3.0

### ↳ **BREAKING ON UPGRADE**

! Vector search on LanceDB now uses cosine distance instead of L2; ranking order can shift for existing deployments on upgrade with no rebuild required.

WAS THIS USEFUL?



# ai-memory

[Feedback](#)**1** RELEASE · 2026-09-06[NOTES](#)[RANK](#)

Solution for long term memory for agent coding CLIs and to facilitate handoff between different agent vendors

ai-memory 2.1 adds ordered LLM provider fallback chains with health status reporting, a resumable bootstrap, and first-party OpenCode 2.0 and Codex SessionEnd handoff support.

→ WHAT SHIPPED · 5 FEATURES most completely described first ? what's the number?

## 01 LLM fallback chains with health status

**NEW**

95

Adds `[[llm_fallbacks]]` table to `config.toml` with per-entry `model`, `base_url`, and `api_key_env` fields to define ordered provider fallback chains that advance on 429, 5xx, timeout, or connection errors while stopping immediately on deterministic failures. `ai-memory status` now exposes passive per-candidate health for fallback providers, showing labels and status only, never keys or response bodies.

Route LLM calls through a primary provider and fall back to a secondary when rate-limited or the upstream is down.

```
toml

[[llm_fallbacks]]
model = "gpt-4o"
api_key_env = "OPENAI_API_KEY"

[[llm_fallbacks]]
model = "gemini-2.0-flash"
base_url =
"https://generativelanguage.googleapis.com/v1beta/openai/"
api_key_env = "GEMINI_API_KEY"
```

– Full mechanism, named config keys, and runnable config example given `v2.1.0`

## 02 OpenCode 2.0 beta integration

**NEW**

85

Adds `install-mcp --client opencode2`, `install-hooks --agent opencode2`, and `ai-memory run opencode2` for first-party OpenCode 2.0 beta support with full lifecycle capture, handoff injection, and managed launch/resume/import.

Wire OpenCode 2.0 beta into ai-memory for full lifecycle capture and cross-agent handoff support.

```
$ ai-memory install-mcp --client opencode2 --apply
ai-memory install-hooks --agent opencode2 --apply
```

– Named subcommands with a runnable setup example v2.1.0

### 03 Resumable bootstrap runs

NEW

75

Adds `--resume` flag to `bootstrap` to restart an interrupted bootstrap from durable per-chunk progress rather than re-running all LLM calls from scratch.

Resume a large bootstrap that was interrupted mid-way without re-paying for already-completed LLM calls.

```
$ ai-memory --data-dir ~/.local/share/ai-memory bootstrap --
resume
```

– Exact flag, mechanism, and runnable command provided v2.1.0

### 04 Codex SessionEnd handoff wiring

NEW

70

Adds native `SessionEnd` event wiring for `install-hooks --agent codex` (requires Codex CLI 0.145.0+), giving Codex sessions the same automatic end-of-session summary and cross-agent handoff as Claude Code.

– Names flag and version requirement but lacks a usage example v2.1.0

THINNER COVERAGE BELOW

### 05 Operator headers on LLM requests

NEW

50

Adds `AI_MEMORY_LLM_HEADERS` environment variable and `llm_headers` config key to attach caller-identifying HTTP headers to every LLM request.

– Named config surfaces but no mechanism or example shown v2.1.0

WAS THIS USEFUL?



EVALUATE

# Helicone

Feedback



1 RELEASE · SEEN 2026-09-07

NOTES &gt;

RANK

Routing and monitoring for reliable AI apps - the LLMOps platform behind the fastest-growing AI companies.

Helicone's biggest move this window is launching an open-source Rust-based AI Gateway proxy alongside Prompt Management V2, a Go SDK, one-command self-hosting, and expanded cost tracking and context-window support across model providers.

↳ WHAT SHIPPED · 11 FEATURES most completely described first ? what's the number?

## 01 Open-source AI Gateway proxy NEW 90

Launches Helicone AI Gateway as an open-source Rust-based proxy available via `docker run -p 8080:8080 helicone/ai-gateway` at [github.com/helicone/ai-gateway](https://github.com/helicone/ai-gateway), supporting smart load balancing, semantic caching, automatic failover, and rate limiting across OpenAI, Anthropic, Azure, and more.

Spin up the self-hosted Helicone AI Gateway locally to proxy LLM traffic without sending data to third-party servers.

```
$ docker run -p 8080:8080 helicone/ai-gateway
```

— Runnable docker command with named repo, providers, and mechanisms snapshot-20260907

## 02 Prompt Management V2 and AI Gateway prompt referencing NEW 80

Adds `prompt_id` and `inputs` fields to AI Gateway API calls, enabling prompt references by ID with runtime variable substitution (e.g., `{{hc:customer_name:string}}`) so prompts can be updated in the dashboard without code changes. Prompt Management V2 introduces version control, typed variables in system prompts, messages, and tool schemas, and instant deployment through the AI Gateway.

— Named fields and syntax but no full endpoint example snapshot-20260907

## 03 Go SDK for Helicone Helpers NEW 75

Introduces a Go SDK for the Helicone Helpers Package, installable via `go get github.com/helicone/go-helicone-helpers`, enabling custom LLM integration with the Go Manual Logger.

Install the Go Helicone Helpers Package to instrument a custom LLM with the Manual Logger in a Go service.

```
$ go get github.com/helicone/go-helicone-helpers
```

– Runnable install command for a new SDK snapshot-20260907

#### 04 Self-hosting via Docker Compose NEW 75

Adds self-hosting support via a single `docker compose up -d` command from the `github.com/Helicone/helicone` repository, keeping all LLM interactions and analytics within your own infrastructure.

– Runnable command and repo named for self-hosting snapshot-20260907

#### 05 Country-based request filtering NEW 65

Adds `country_code:<ISO>` filter (e.g., `country_code:US`) to the requests dashboard advanced filters section for geographic analytics on LLM usage.

– Named filter syntax and UI location, no deeper mechanism snapshot-20260907

#### 06 Cost tracking for new models across providers NEW 60

Adds cost tracking for GPT-5, GPT-5-Mini, GPT-5-Nano, and GPT-5-Chat-Latest models across multiple providers with Playground support; for GPT-OSS models (`accounts/fireworks/models/gpt-oss-20b`, `accounts/fireworks/models/gpt-oss-120b`, `openai/gpt-oss-20b`, `openai/gpt-oss-120b`) across Fireworks, Groq, and OpenRouter; for Claude Opus 4.1 (`claude-opus-4-1-20250805`, `anthropic/claude-opus-4.1` on OpenRouter) and `grok-4` / `x-ai/grok-4`; and for `gpt-4o-search` with automatic pricing detection.

– Lists exact model ids but tracking is passive, no user action snapshot-20260907

#### 07 1M token context window for Claude Sonnet on AI Gateway IMPROVED 60

Extends Claude Sonnet 4 and Sonnet 4.5 on the AI Gateway to support a 1M token context window by default across Anthropic API, AWS Bedrock, and Google Vertex AI.

– Concrete limit and scope but no config example snapshot-20260907

THINNER COVERAGE BELOW

- 08

**Reasoning effort control in Playground** NEW

55

Adds reasoning effort control in the Playground with **minimal**, **low**, **medium**, and **high** reasoning levels for supported thinking models.

– Named levels but only a UI starting point

snapshot-20260907
- 09

**Redesigned Sessions view with metrics** IMPROVED

45

Introduces redesigned Sessions view with session-level metrics including average latency, total cost, and improved time filtering.

– Names metrics shown but no navigation detail

snapshot-20260907
- 10

**OpenAI Responses API monitoring** NEW

40

Adds OpenAI Responses API support to Helicone for monitoring and analyzing text and image input/output responses.

– Describes capability but no config or endpoint detail

snapshot-20260907
- 11

**Automatic timezone detection for timestamps** IMPROVED

30

Adds automatic local timezone detection and clear timezone indicators for all timestamp displays in the dashboard.

– Minor UI change with no named surface

snapshot-20260907

WAS THIS USEFUL?



LangSmith provides tracing, evaluation, and deployment tools for LLM applications.

LangSmith's latest release changes bulk export compression defaults, adds pre-save testing for thread evaluators, reworks trace sharing to operate at the trace root, and ships several breaking API and logging changes.

WHAT SHIPPED • 5 FEATURES most completely described first ? what's the number?

## 01 Thread evaluator pre-save testing and turn-count filter NEW

90

The thread evaluator config now shows a locked `Trace count >= 2` filter for managed thread evaluators, making explicit that they only run on threads with multiple turns. Evaluators can be validated against a real conversation before saving via `POST /runs/rules/validate` with `test_thread_id`, `session_id`, and `rule` parameters, catching misconfiguration early.

Test a multi-turn thread evaluator against a real conversation before saving it to catch misconfiguration early.

```
$ curl -X POST
'https://api.smith.langchain.com/runs/rules/validate' \
-H 'X-API-Key: <your-api-key>' \
-H 'Content-Type: application/json' \
-d '{"test_thread_id": "<thread-uuid>", "session_id": "
<session-uuid>", "rule": { ... }{'
```

— Names endpoint, method, and parameters with a runnable curl example. snapshot-20260907

## 02 Dataset comparison helpers removed from public API

BREAKING

60

Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; existing HTTP routes continue to work only for LangSmith UI clients, so integrations relying on the SDK methods must migrate.

— Names the removed surface and migration constraint clearly. snapshot-20260907

THINNER COVERAGE BELOW

## 03 Trace-root scoped sharing

BREAKING

35

Sharing, viewing, or unsharing any run in a trace now operates on the trace root, meaning every run in a shared trace becomes publicly viewable rather than just the individual run that was shared.

– Explains behavior change but no config or opt-out named. snapshot-20260907

---

#### 04 **Dataset split chips in Examples table** NEW 35

Each example's dataset splits now render as chips in the dataset Examples table, with a clickable +N overflow menu for examples belonging to many splits.

– UI change described but no exact navigation path given. snapshot-20260907

---

#### 05 **ABAC-aware access denial messaging** IMPROVED 30

Thread pages now show an explicit access-control message when trace loading is denied by ABAC, instead of a generic retrieval error.

– Describes the change but no mechanism or navigation detail. snapshot-20260907

---

#### **BREAKING ON UPGRADE**

- ! Legacy dataset comparison helpers are removed from the public OpenAPI spec and generated SDKs; existing HTTP routes continue to work only for LangSmith UI clients.
- ! The batched-run ingestion log now emits `run_verbs` as a list of `run_id` and `verbs` objects instead of a map keyed by run UUID, which will break any structured-log aggregators that parse the old map format.

---

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 push this cycle is around agentic debugging and improvement: two new investigative surfaces (Patterns and Debugger) plus a major upgrade to Loop that lets it act autonomously across a project with built-in models, alongside SDK tracing additions for OpenAI Batch jobs and Agno evals.

WHAT SHIPPED · 7 FEATURES most completely described first [? what's the number?](#)

## 01 OpenAI Batch API tracing in TypeScript SDK **NEW** 90

The TypeScript SDK (v3.30.0) adds `openaiFilesCreateTraced`, `openaiBatchesRetrieveTraced`, and `completeOpenAIBatchTrace` to trace asynchronous OpenAI Batch API requests end-to-end so batch inference jobs appear as first-class spans in Braintrust experiments.

Trace async OpenAI Batch requests end-to-end so batch inference jobs appear as first-class spans in your Braintrust experiments.

```
typescript

import { openaiFilesCreateTraced, openaiBatchesRetrieveTraced,
  completeOpenAIBatchTrace } from 'braintrust';

const file = await openaiFilesCreateTraced(openai, {
  file: fs.createReadStream('requests.jsonl'),
  purpose: 'batch'
});
const batch = await openai.batches.create({
  input_file_id: file.id,
  endpoint: '/v1/chat/completions',
  completion_window: '24h'
});
await openaiBatchesRetrieveTraced(openai, batch.id);
await completeOpenAIBatchTrace(openai, batch.id);
```

— Named functions with a full runnable code example [snapshot-20260907](#)

## 02 Auto-instrumentation for Agno eval types **NEW** 80

Calling `auto_instrument()` automatically instruments all Agno eval types — `AccuracyEval`, `AgentAsJudgeEval`, `ReliabilityEval`, and `PerformanceEval` — so their runs appear as Braintrust experiments without manual tracing setup.

Automatically instrument all Agno eval types so their runs appear as Braintrust experiments without manual tracing setup.

```
python

from braintrust import auto_instrument
from agno.evals import AccuracyEval, AgentAsJudgeEval,
ReliabilityEval, PerformanceEval

auto_instrument()

eval_suite = [
    AccuracyEval(agent=my_agent, questions=questions),
    AgentAsJudgeEval(agent=my_agent, rubric=rubric),
    ReliabilityEval(agent=my_agent, scenarios=scenarios),
    PerformanceEval(agent=my_agent, tasks=tasks),
]
for e in eval_suite:
    e.run()
```

– Named function and eval classes with a runnable code example

snapshot-20260907

### 03 Loop runs autonomously with built-in models

IMPROVED

75

Loop now runs in a Braintrust-managed runtime, persists threads, and operates across logs, experiments, and datasets project-wide. It can create and edit prompts, scorers, datasets, facets, custom views, dashboards, and automations, with a per-write approval step unless auto-accept is enabled. Loop also runs built-in on the GPT-5.6 Sol, GPT-5.6 Terra, and GPT-5.6 Luna models with no AI provider configuration required.

– Detailed scope and named models, but no direct invocation shown

snapshot-20260907

### 04 Patterns: scheduled trace investigation

NEW

70

Patterns is a scheduled automated investigation of traces that surfaces recurring failure modes, cost trends, and cohort-level problems. Each finding includes supporting traces and a suggested fix, and can be used to create a scorer, continue in Loop, or copy as a prompt.

– Mechanism and outputs described, but no config key or command shown

snapshot-20260907

### 05 Debugger for single-trace failure analysis

NEW

65

Debugger reports likely failure modes for a single trace, citing spans, tool calls, and model outputs, and groups a run into **Work** sections via the **Analyze trace** action to explain what the agent did.

– Named UI action given but no runnable command

snapshot-20260907

## 06 GitHub eval action score and metric filtering IMPROVED

 65

The GitHub eval action (v2.1.0) adds `report_scores` and `report_metrics` inputs, each accepting a comma- or newline-separated list of names to filter what appears in the PR comment.

– Named config inputs but no example workflow shown snapshot-20260907

---

THINNER COVERAGE BELOW

---

## 07 Invocation parameters in Pydantic AI span metadata IMPROVED

 50

The Python SDK (v0.37.0) adds invocation parameters — `temperature`, `max_tokens`, and similar model settings — to Pydantic AI LLM span metadata when model settings are provided.

– Named fields but no example or config path given snapshot-20260907

---

WAS THIS USEFUL?



---

## GOVERN

# Promptfoo

Feedback



1 RELEASE • SEEN 2026-09-07

NOTES >

RANK

Promptfoo tests, evaluates, and red-teams LLM applications to detect vulnerabilities and compare model outputs.

Promptfoo expanded its provider ecosystem with new integrations for Docker Model Runner, MCP servers, and several models and vision providers, while also adding enterprise-grade mutual TLS support for the HTTP provider.

WHAT SHIPPED • 5 FEATURES most completely described first ? what's the number?

01 MCP provider and red-team agent example NEW 65

Adds an MCP (Model Context Protocol) provider to connect Promptfoo evals to MCP servers, plus an MCP Agent example for red team testing that demonstrates how to test AI systems using tool call results via MCP.

- Names provider and example but no exact config shown snapshot-20260907

THINNER COVERAGE BELOW

02 JKS and PFX client certificates for HTTP provider NEW 55

The HTTP provider now supports JKS and PFX client certificate formats, enabling mutual TLS authentication against enterprise endpoints.

- Names cert formats but no config key or flag shown snapshot-20260907

03 New model and provider integrations NEW 55

Adds the Google Imagen provider for generating images in multimodal testing scenarios, the AIMLAPI provider for unified access to various AI models, support for OpenAI's o3-deep-research and o4-mini-deep-research models, Grok-4 model support via the xAI provider, LiteLLM embeddings support for similarity testing and semantic search evaluations, and a Google Vision provider for image understanding in multimodal evaluations.

- Six providers named but each described in one line only snapshot-20260907

04 Docker Model Runner provider NEW 50

Adds a Docker Model Runner provider to run models in isolated Docker containers for improved security and reproducibility.

- Names the provider but no setup command or config given snapshot-20260907

05 System prompt support in Azure provider IMPROVED 25

The Azure provider now supports system prompts for better control over model behavior.

– Thin description with no mechanism or config detail

snapshot-20260907

#### **L--> ALSO FROM THESE RELEASES**

.

---

WAS THIS USEFUL?



# Docker Sandboxes

[Feedback](#)[1 RELEASE · 2026-09-07](#)[NOTES ↗](#)[RANK](#)

Docker Sandboxes runs AI agents in isolated Docker environments with controlled filesystem, network, and tool access.

Docker Sandboxes v0.42.0 adds machine-readable JSON output across read-only commands, scoped [MCP OAuth](#) authorization, Git-based skills, a full argument system for kits and environment files, and a Windows MSI installer — alongside several breaking changes to port defaults, built-in agents, and environment-file semantics.

[←](#) [WHAT SHIPPED · 23 FEATURES](#) [most completely described first](#) [? what's the number?](#)

## 01 Kit arguments (`--kit-arg`, `args:` block) NEW 95

Kits can now declare arguments in an `args:` block and receive values with `--kit-arg name=value` or `--kit-arg kit.name=value` to target a specific kit; a `kits:` entry in `sbxenv.yaml` can carry per-kit arguments under `kits[].args`.

Pass parameterized values into a kit at launch time — useful when a kit exposes configurable targets or modes.

```
$ sbx run docker.io/myorg/pentest-kit:latest --kit-arg
target=192.168.1.0/24 --kit-arg mode=aggressive
```

Pin per-kit arguments in an environment file so the same values are applied every time the environment is started.

```
yaml

sandbox:
  name: pentest-env
kits:
  - ref: docker.io/myorg/pentest-kit:latest
    args:
      target: 192.168.1.0/24
      mode: aggressive
```

– Full flag and config key with runnable examples v0.42.0

## 02 JSON output for read-only commands NEW 85

Adds a `--json` flag to `sbx secret ls`, `sbx version`, `sbx mcp ls`, `sbx skills ls`, `sbx policy inspect`, and the `sbx kit` verification commands for machine-

readable output; `sbx version --json` also reports a `server.state` field ( `running` or `unavailable` ) so scripts can check daemon reachability without parsing error text.

Check daemon reachability in CI scripts without parsing human-readable error text.

```
$ sbx version --json | jq '.server.state'
```

– Names every affected command and field, with a runnable jq example v0.42.0

### 03 Default TCP4 port publishing

**BREAKING**

80

`sbx ports --publish` and kit-declared ports now default to `tcp4` instead of dual-stack `tcp`, so published ports no longer listen on `::1`; pass the protocol explicitly (e.g. `--publish 8080:3000/tcp`) to restore dual-stack behavior.

– Exact flag and migration example provided v0.42.0

### 04 Scoped MCP OAuth authorization

**IMPROVED**

75

`sbx mcp auth` now requests only the chosen scopes (or those the resource requires) instead of a broad default set, accepts a new `--no-scope` flag to suppress server-required scopes, explains refused scopes with a narrower retry command, and reports granted/requested/supported scope sets in `sbx mcp auth status`.

– Mechanism and flag named but no runnable example given v0.42.0

### 05 Optional workspace bind mounts

**BREAKING**

75

`sbx create` can now omit the workspace path to create a sandbox with no workspace bind mount; correspondingly, an environment file that declares no `workspace:` now creates a sandbox with no bind mount instead of mounting the directory holding the file — add `workspace: .` to restore the previous behavior.

– Behavior change and explicit migration note included v0.42.0

### 06 Environment-file arguments (``args:`` block)

**BREAKING**

75

Environment files ( `sbxenv.yaml` ) can now declare their own arguments in an `args:` block, referenced as `${{ env.args.NAME }}` and supplied with `sbx env --env-arg`; this replaces the previous `${VAR}` interpolation syntax, which is no longer expanded.

– Config syntax and breaking migration named, no example v0.42.0

### 07 Built-in agents removed in favor of community kits

**BREAKING**

The built-in `droid`, `copilot`, and `kiro` agents are removed and must now be run as community kits, e.g. `sbx run docker.io/sbx/droid-kit:latest`, `sbx run docker.io/sbx/copilot-kit:latest`, or `sbx run docker.io/sbx/kiro-kit:latest`.

– Exact migration commands given for each removed agent v0.42.0

## 08 **sbx env plan-and-confirm workflow**

NEW

70

`sbx env` now shows a plan of all changes an environment file will make to the host (lifecycle commands, credentials, bindings, MCP servers, workspaces, kits, ports) and prompts for approval unless `env.rememberHostCommands` is set.

– Names the config key and scope of the plan, no example v0.42.0

## 09 **Sandbox default disk size reduced**

BREAKING

70

Sandbox default Docker volume size is reduced to 10 GB (from 50 GB); configurable via the `sandbox.disk.dockerVolume` setting or the `DOCKER_SANDBOXES_DOCKER_SIZE` environment variable.

– Exact size numbers and config keys named v0.42.0

## 10 **sbx mcp add flag rename and address flexibility**

IMPROVED

65

`sbx mcp add` gains a `--skip-auth` flag (replacing `--skip_auth`) and now allows `--url` values pointing at private, loopback, or cloud-metadata addresses, printing a warning instead of refusing.

– Flag rename and behavior change named, no example command v0.42.0

## 11 **Kit reference as positional run argument**

IMPROVED

60

Adds an `sbx run <sandbox-kit-ref>` positional form so kit references can be passed directly as the agent argument; the `--kit` flag is now reserved for mixins.

– Names the new syntax and repurposed flag, no example v0.42.0

THINNER COVERAGE BELOW

## 12 **Environment file resolution and defaults**

IMPROVED

55

`sbx env` now reads a non-hidden `sbxenv.yaml` in preference to `.sbxenv.yaml` when both exist in a directory, and merges a `sbxenv.yaml` from the home directory as project-wide defaults.

– File precedence and merge behavior named clearly v0.42.0

## 13 **CPU and memory limits via kit spec**

NEW

55

Kits can now set a sandbox's `CPU` and memory limits via the `sandbox.resources` block in their spec.

– Config block named, no example of values `v0.42.0`

#### 14 Registry mirror applied inside sandboxes IMPROVED 55

A registry mirror configured with `platform.images.registryMirror` is now also applied to Docker running inside a sandbox when the value is a bare host.

– Config key and condition named, no example `v0.42.0`

#### 15 Audit records for declined MCP approval prompts NEW 55

Declined `@requireApproval` prompts on a local `MCP` server now produce their own audit record with policy attribution and a context digest.

– Mechanism named but no command to inspect the record `v0.42.0`

#### 16 Cloud sandbox port publishing NEW 55

`sbx --cloud run` and `sbx --cloud create` now publish a kit's declared `TCP` ports and print their public URLs.

– Commands and behavior named, no worked example `v0.42.0`

#### 17 Git-based sandbox skills management NEW 45

Adds an `sbx skills` subcommand to add, update, list, and remove sandbox skills directly from Git repositories.

– Names the subcommand but no flags or example shown `v0.42.0`

#### 18 Governance resolution visibility IMPROVED 45

Governance resolution issues are now shown in `sbx policy` output and the dashboard.

– Names the command surface but not the mechanism `v0.42.0`

#### 19 Terminal display fixes IMPROVED 45

`sbx terminal` output now adapts colours for light terminal backgrounds and uses a distinct pink spinner glyph; CJK and combining-mark column widths in table output are now measured correctly.

– Cosmetic fixes described, no config surface involved `v0.42.0`

#### 20 Relative kit path resolution changed BREAKING 45

Relative kit paths in an environment file now resolve against the file's directory instead of the directory `sbx` was run from.

– Behavior change named, no example or migration command v0.42.0

## 21 **sbx env create display, GPU, and USB options** NEW 40

`sbx env create` now accepts `display`, `GPU`, and `USB` options in `sbxenv.yaml` and shares imported skills by default.

– Options named generically, no exact keys given v0.42.0

## 22 **Automatic diagnostics bundle upload** NEW 35

Docker Sandboxes can now upload a diagnostics bundle automatically when the daemon hits an error, after opt-in.

– No detail on the opt-in mechanism or command v0.42.0

## 23 **Windows MSI installer** NEW 30

Provides a machine-wide Windows MSI installer for administrator-managed installations.

– Bare description with no options or install steps v0.42.0

### ↳ **BREAKING ON UPGRADE**

! `sbx ports --publish` and kit-declared ports now default to `tcp4` instead of dual-stack `tcp`, so published ports no longer listen on `:::1`; pass the protocol explicitly (e.g. `--publish 8080:3000/tcp`) to restore dual-stack behavior.

! The built-in `droid`, `copilot`, and `kiro` agents are removed; existing workflows that invoke them directly must be updated to `sbx run docker.io/sbx/droid-kit:latest`, `sbx run docker.io/sbx/copilot-kit:latest`, or `sbx run docker.io/sbx/kiro-kit:latest`.

! `${VAR}` interpolation in `.sbxenv.yaml` is no longer expanded; use the new `${env.args.NAME}` syntax with an `args:` block instead.

! An environment file that declares no `workspace:` now creates a sandbox with no workspace bind mount instead of mounting the directory holding the file; add `workspace: .` to restore the previous behavior.

! Relative kit paths in an environment file now resolve against the file's directory instead of the directory `sbx` was run from.

WAS THIS USEFUL?



# NVIDIA SkillSpector

Feedback



1 RELEASE • 2026-09-07

NOTES >

RANK

SkillSpector scans AI agent skills for vulnerabilities and malicious patterns before installation.

SkillSpector v2.11.1 raises the default scan workflow deadline and makes it configurable via a new environment variable.

WHAT SHIPPED • 1 FEATURE most completely described first ? what's the number?

## 01 Configurable aggregate workflow deadline IMPROVED 86

Adds the `SKILLSPECTOR_MAX_WORKFLOW_SECONDS` environment variable to set a custom aggregate workflow deadline (must be a positive, finite number of seconds); invalid, zero, negative, infinite, or NaN values fall back to the default. The default aggregate end-to-end workflow deadline itself is raised from 60 seconds to 600 seconds, applying across direct CLI, recursive, transitive, and multi-skill analysis paths.

Override the aggregate scan deadline for an environment that routinely scans large transitive skill graphs and hits the default ceiling.

```
$ SKILLSPECTOR_MAX_WORKFLOW_SECONDS=1800 skillspector scan https://github.com/user/my-skill
```

– Names exact env var, old/new defaults, fallback behaviour, and a runnable example.

v2.11.1

WAS THIS USEFUL?



## AI MODELS

# Anthropic

[Feedback](#)

1 RELEASE • SEEN 2026-09-07

[NOTES](#)[RANK](#)

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

Anthropic shipped Claude Fable 5.1 and Claude Mythos 5.1 with 1M-token context and always-on adaptive thinking, alongside an `ant apply` subcommand for infrastructure-as-code agent deployments, a full Admin API in the CLI and SDKs, expanded Compliance API coverage, and several breaking changes to `tool_choice`, thinking-block replay, and beta SDK clients.

↳ WHAT SHIPPED • 15 FEATURES most completely described first ? what's the number?

## 01 Admin API in CLI and SDKs NEW 95

The Admin API is now available in the `ant` CLI and Python, TypeScript, C#, Go, Java, PHP, and Ruby SDKs under `client.beta.organization`, covering organization info, members, invites, workspaces, API keys, rate limits, service accounts, workload identity federation issuers and rules, and customer-managed encryption keys; it reads credentials from `ANTHROPIC_API_KEY` or `ANTHROPIC_AUTH_TOKEN`.

– Names the client path, env vars, and every covered resource. snapshot-20260907

## 02 Mid-conversation effort control beta NEW 90

With the `mid-conversation-output-config-2026-07-01` beta header, a `role: "system"` message inside `messages` can include `output_config.effort` to change effort mid-conversation on Claude Fable 5.1, Claude Mythos 5.1, and Claude Opus 5 while preserving the prompt cache.

– Exact header and JSON field make this runnable as described. snapshot-20260907

## 03 Turn-scoped system messages beta NEW 90

The `mid-conversation-system-clear-at-2026-08-21` beta header lets a mid-conversation `role: "system"` message set `clear_at: "next_user_message"` so it applies only to the current turn and remains in history at no token cost.

– Exact header and field value give a runnable configuration. snapshot-20260907

#### 04 **`ant apply` subcommand for agent infrastructure-as-code** NEW

85

The `ant` CLI (v1.30.0) adds `ant apply` to create and update agents, environments, skills, memory stores, and deployments directly from repository files, writing a `claude-lock.json` lockfile to keep CI runs reproducible.

– Names exact subcommand, CLI version, and lockfile file. `snapshot-20260907`

#### 05 **Thinking block binding controls beta header** NEW

85

The `thinking-binding-controls-2026-08-01` beta header reports dropped thinking blocks in an `input_transformations` response field and adds `thinking.block_binding.prefix_mismatch_behavior` to choose between rejecting or dropping blocks whose conversation history changed.

– Named header, response field, and config key are directly usable. `snapshot-20260907`

#### 06 **Compliance API session coverage expands and exits beta for Cowork/Code**

IMPROVED

85

The Compliance API's local session endpoints (beta, Claude Enterprise) now return transcripts for Claude Science sessions ( `product_surface` value `claude_science` ) and Claude for Microsoft 365 sessions in Excel, PowerPoint, Word, and Outlook ( `product_surface` values beginning with `office_agents` ), using the existing Compliance Access Key and `read:compliance_user_data` scope; the same session endpoints for Cowork and Claude Code sessions are promoted out of beta.

– Names `product_surface` values, scope, and beta status change. `snapshot-20260907`

#### 07 **Beta files/skills SDK client changes** BREAKING

85

In Python SDK 1.2.0, TypeScript SDK 0.122.0, Go SDK 1.68.0, Java SDK 2.59.0, Ruby SDK 1.67.0, and C# SDK 12.44.0, `client.beta.files` and `client.beta.skills` no longer send the `files-api-2025-04-14` and `skills-2025-10-02` beta headers and return the same response shapes as `client.files` and `client.skills`; `client.beta.skills.delete()` now deletes a Skill together with all of its versions, and the beta Messages type `BetaSkill` is renamed `BetaContainerSkill`.

– Names exact SDK versions, headers, methods, and type rename. `snapshot-20260907`

#### 08 **Claude Fable 5.1 and Claude Mythos 5.1 launch with 1M context**

NEW

80

`claude-fable-5-1` launches with a 1M token context window, 128k max output tokens, always-on adaptive thinking, priced at \$10/\$50 per MTok input/output with prompt cache reads at \$0.25 per MTok (0.025x base input price); `claude-mythos-5-1` ships the same context, output, thinking, and pricing for Project Glasswing participants.

– Concrete specs and pricing but only a model id to call. snapshot-20260907

## 09 **`thinking.display` "updates" mode beta** NEW 80

With the `thinking-display-updates-2026-08-18` beta header, `thinking.display` accepts the value `"updates"`, returning short progress updates Claude writes between tool calls as text, with at most one thinking block before each tool call.

– Named header and field value with a stated behavioural limit. snapshot-20260907

## 10 **Computer and browser use toolsets on Google Cloud** IMPROVED 70

The `computer_toolset_20260801` and `browser_toolset_20260801` toolsets are now available on Google Cloud for Claude Fable 5, Claude Mythos 5, Claude Opus 5, Claude Sonnet 5, and Claude Opus 4.8.

– Names exact toolsets and model availability, no mechanism detail. snapshot-20260907

## 11 **`tool\_choice` restrictions on Fable/Mythos 5.1** BREAKING 65

On Claude Fable 5.1 and Claude Mythos 5.1, `tool_choice` types `any` and `tool` are not supported and return a 400 error; only `auto` and `none` are accepted.

– Clear before/after but no migration guidance. snapshot-20260907

## 12 **Thinking block replay restrictions** BREAKING 65

For accounts created on or after August 31, 2026, replaying a thinking block on Claude Fable 5.1 after the system prompt, tools, or an earlier message has changed now returns a 400 error, and thinking blocks produced by Claude Fable 5.1 and Claude Mythos 5.1 are dropped by the API if replayed to an earlier model.

– Names exact conditions but no remediation steps. snapshot-20260907

THINNER COVERAGE BELOW

## 13 **`anthropic-version` header now required for Enterprise APIs** BREAKING 55

The Claude Enterprise Admin API endpoints, Claude Enterprise Analytics API, and Compliance API now require the `anthropic-version` header on every request.

– Exact header and affected APIs named, minimal further detail. snapshot-20260907

## 14 **Automatic content watermarking** NEW 50

Text output from Claude Fable 5.1 and Claude Mythos 5.1 automatically carries Anthropic's text watermark; image, video, and audio files produced via the code execution tool carry C2PA Content Credentials when retrieved through the Files API, with no request or response changes required.

– Automatic behaviour with nothing for a reader to configure.

snapshot-20260907

## 15 Personal and service account API keys in Console

NEW

40

The Console adds personal keys and service account keys that can be scoped to a specific workspace or made active across any workspace or admin endpoints the linked account can access.

– Describes scope options but only a UI starting point.

snapshot-20260907

### ↳ BREAKING ON UPGRADE

- ! On Claude Fable 5.1 and Claude Mythos 5.1, `tool_choice` types `any` and `tool` are not supported and return a 400 error; only `auto` and `none` are accepted.
- ! For accounts created on or after August 31, 2026, replaying a thinking block on Claude Fable 5.1 after the system prompt, tools, or an earlier message has changed returns a 400 error.
- ! Thinking blocks produced by Claude Fable 5.1 and Claude Mythos 5.1 are dropped by the API if replayed to an earlier model.
- ! In Python SDK 1.2.0, TypeScript SDK 0.122.0, Go SDK 1.68.0, Java SDK 2.59.0, Ruby SDK 1.67.0, and C# SDK 12.44.0, `client.beta.files` and `client.beta.skills` no longer send the `files-api-2025-04-14` and `skills-2025-10-02` beta headers and return the same response shapes as `client.files` and `client.skills`.
- ! In the SDK versions above, `client.beta.skills.delete()` now deletes a Skill together with all of its versions, and the beta Messages type `BetaSkill` (the container Skill reference) is renamed `BetaContainerSkill`.
- ! The Claude Enterprise Admin API endpoints, Claude Enterprise Analytics API, and Compliance API now require the `anthropic-version` header on every request.

WAS THIS USEFUL?



// END OF TRANSMISSION



The AI Toolchain · curated daily · September 7, 2026

Releases & features from the open-source and commercial AI tools that practitioners actually run.

▶ [Subscribe](#)

[Feedback on this issue](#)